



中文

目录

1 控制器技术数据及安装接线	01	2.2.8 查找	45
1.1 CPU模块	01	2.2.9 仿真	46
1.2 技术数据	01	2.2.10 上传	50
1.2.1 端子介绍	04	2.2.11 固件升级	51
1.2.2 CPU模块接线示意图	04	2.3 编程语言	54
1.3 扩展模块	05	2.3.1 编程语言类型	54
1.3.1 扩展模块通用数据	05	2.3.2 梯形图 (LD)	55
1.3.2 数字量输入模块	05	2.3.3 结构化文本编程 (ST)	61
1.3.3 数字量输出模块	06	2.3.4 功能块图 (FBD)	63
1.3.4 模拟量输入模块	06	2.3.5 顺序功能图 (SFC)	63
1.3.5 模拟量输出模块	07	2.3.6 指令表 (IL)	63
1.3.6 通讯接口模块	07	2.4 扩展模块	63
1.3.7 接线示意图	08	2.4.1 数字量输入DI	66
1.4 安装尺寸	08	2.4.2 数字量输出DQ	69
2 控制器软件安装及编程调试	09	2.4.3 模拟量输入AI	70
2.1 快速入门	09	2.4.4 模拟量输出AQ	70
2.1.1 BonengLogicLab软件安装	09	2.4.5 通讯模块CI	71
2.1.2 软件编程	10	2.5 串口通信	71
2.1.3 通讯设置	11	2.5.1 概述	71
2.1.4 用户编程的典型步骤	12	2.5.2 Modbus-RTU主站配置	71
2.1.5 BonengLogiclab程序运行基本框架	12	2.5.3 Modbus-RTU从站配置	75
2.1.6 编程与调试	12	2.5.4 Modbus 故障码说明	76
2.1.7 切换PLC工作模式	21	2.6 以太网通信	78
2.1.8 在线监控与调试	21	2.6.1 概述	78
2.1.9 示波器应用	24	2.6.2 Modbus-TCP主机配置	78
2.1.10 程序扫描周期设置	27	2.6.3 Modbus-TCP从机配置	79
2.1.11 EtherCAT任务周期设置	28	2.6.4 Modbus故障码说明	79
2.1.12 导入与导出对象	28	2.7 EtherCAT通信	79
2.2 编程基础	29	2.7.1 概述	79
2.2.1 概述	29	2.7.2 EtherCAT主站配置	80
2.2.2 变量	29	2.7.3 EtherCAT从站配置	84
2.2.3 资源-参数 (掉电保持)	39	2.8 运动控制	92
2.2.4 资源-状态变量 (掉电不保持)	41	2.8.1 运动控制轴简介	92
2.2.5 库管理向导	41	2.8.2 创建轴对象一般步骤	94
2.2.6 交叉参考	44	2.8.3 轴页面配置	97
2.2.7 重构	45	2.9 电子齿轮	110

目录

2.9.1 齿轮动作	110	3.5.5 Vars (EtherCAT库中的全局变量)	300
2.9.2 电子齿轮使用	111	3.6 ModbusRTU	301
2.9.3 轴配置/状态总览	112	3.6.1 Definitions	301
2.9.4 故障类别	113	3.6.2 ModbusRTU_ReadCoils2	309
2.10 电子凸轮	113	3.6.3 ModbusRTU_ReadDeviceIdent	311
2.10.1 电子凸轮简介	113	3.6.4 ModbusRTU_ReadDiscrInputs2	311
2.10.2 软件配置	114	3.6.5 ModbusRTU_ReadHoldingRegs2	313
2.10.3 电子凸轮使用	119	3.6.6 ModbusRTU_ReadInputRegs2	314
2.11 轴组	121	3.6.7 ModbusRTU_WriteMultCoils2	315
2.11.1 轴组简介	121	3.6.8 ModbusRTU_WriteMultRegs2	316
2.11.2 软件配置	122	3.6.9 ModbusRTU_WriteSingleCoil2	317
3 指令说明	128	3.6.10 ModbusRTU_WriteSingleReg2	317
3.1 BonengControlLib	128	3.6.11 sysMbMRtu_ChangePortSettings	318
3.1.1 BN_EtherCAT	128	3.6.12 sysMbMRtu_DisableWriteForSlave	318
3.1.2 BN_Basic	151	3.6.13 sysMbMRtu_GetMasterStatus	319
3.1.3 BN_Modbus	153	3.6.14 sysMbMRtu_GetSlaveHandle	319
3.2 BonengEmbedLogicLib	155	3.6.15 sysMbMRtu_GetSlaveStatus	320
3.2.1 ArithmeticOperationClassExtLib	155	3.6.16 sysMbMRtu_SetMaxRetriesForSlave	321
3.2.2 ClockClassExtLib	156	3.6.17 sysMbMRtu_SlaveActivation	321
3.2.3 DataConversionClassExtLib	156	3.6.18 sysMbMRtu_SlaveChangeAddress	322
3.2.4 ConfigClassExtLib	157	3.6.19 ModbusVars	322
3.3 BonengMachiningProcessLib	161	3.6.20 sysMbMRtuNetList	323
3.3.1 BN_TensionControl	161	3.7 ModbusTCP	324
3.4 MotionControl轴控	168	3.7.1 Enumerations	324
3.4.1 MotionControl库介绍	168	3.7.2 Structures	325
3.4.2 Basic基础指令	178	3.7.3 sysMbMTcp_ReadReg_RO	326
3.4.3 Cam and Gear 凸轮齿轮指令	221	3.7.4 sysMbMTcp_ReadReg_RW	326
3.4.4 Encoder编码器轴	235	3.7.5 sysMbMTcp_WriteReg_M	327
3.4.5 MotionGroupControl轴组控制	237	3.7.6 sysMbMTcp_WriteReg_S	327
3.5 EtherCAT	278	3.7.7 Diagnostics	328
3.5.1 Definitions (定义)	278	4 其他说明及注意事项	328
3.5.2 Master control functions (主站控制功能)	291	4.1 故障说明	328
3.5.3 Master diagnostic functions (主站诊断功能)	292	4.2 注意事项	329
3.5.4 SDO function blocks	296	售后服务	331

1 控制器技术数据及安装接线

1.1 CPU模块

1.2 技术数据

博能X3000系列控制器是中型PLC且支持运动控制，具体订货号如下

订货号	描述
X3050-EC-00-S	通用PLC并支持Motion运动控制

X3050的主要功能如下

	X3050
PLC基本功能	✓
EtherCAT	✓
modbus	✓
Ethernet	✓
最大EtherCAT轴数	32
最大本地扩展IO数量	16
速度控制	✓
扭矩控制	✓
定位控制	✓
PID控制	✓
张力控制	✓
MC标准功能块	✓
编码器轴	✓
虚拟轴	✓
电子齿轮同步	✓
电子凸轮同步	✓
轴组(点到点运动)	✓
轴组(空间直线插补)	✓
轴组(空间圆弧插补)	✓
轴组(龙门主从同步)	✓
轴组(姿态控制)	✓

下表是控制器功能在具体行业的应用

X3050	
调速	同步/凸轮/轴组
行车	凹版印刷机
门机	复合机
塔机	涂布机
拉丝机	复卷机
升降机	分切机
输送机	压花机
辊压机	印花机
卷板机	拉幅定型机
搅拌机	浆纱机
	纺纱机
	倍捻机
	着色机
	多线切割机
	灌装机
	旋盖机
	贴标机
	装箱机
	折弯机/弯管机
	剪板机
	成型机
	轮切/飞锯
	追剪
	点胶机
	桁架机器人
	龙门机床
	智能行车

主要参数如下表

订货号	X3050-EC-00-S	标准控制性
CPU模块	X3050	ARM-A8 800MHz
存储	用户程序容量	1MB
	用户数据空间	8MB
	掉电保存空间	4kB
工作电源	外部端子供电	电压范围： DC 19.2 ... 28.8V 最大电流：2.6A
通讯接口	EtherCAT	1个RJ45端口
	Ethernet	1个RJ45端口
	Modbus	无内置端接电阻
指示灯	3个单色指示灯	PWR指示灯：绿色 RUN指示灯：绿色 ERR指示灯：红色
日时钟	断电维持一周	
编程语言	LD	✓
	ST	✓
	IL	✓
	FBD	✓
	SFC	✓
扩展数量	最大15个，最多允许一个通讯接口扩展模块	
工作温度	-20℃ ... +50℃	
存储温度	-40℃ ... +70℃	
相对空气湿度	<95%	不允许有凝露
污染	符合IEC 61800-5-1	适用于污染等级 2 的环境
电磁兼容性	符合IEC 61800-3	
防护等级	IP20	
重量	230g	

1.2.1 端子介绍

Ethernet接口

类型	带金属屏蔽的RJ45母座
功能	连接Ethernet协议设备
	升级模块固件
	烧录用户程序



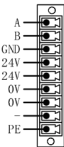
EtherCAT接口

类型	带金属屏蔽的RJ45母座
功能	连接EtherCAT协议设备



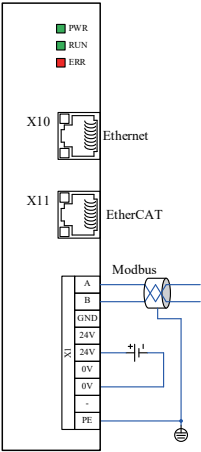
X1端子

电源供电	24V DC (± 20%)，max. 2.6A
通讯	Modbus-RTU通讯



1.2.2 CPU模块接线示意图

当绿色的PWR指示灯长亮的时候，表示供电已经接通，当绿色的RUN指示灯闪烁时表示该CPU模块运行正常，当RUN不在闪烁，表示CPU模块运行异常。红色的ERR指示灯亮起时表示CPU模块出现错误，需要排除错误后重新断电上电。查看绿色RUN指示灯是否闪烁运行正常。如果还不正常，需要联系相关技术部门进行检修。



1.3 扩展模块

1.3.1 扩展模块通用数据

通用数据	单色指示灯	RDY指示灯：绿色
	工作温度	-20℃ ... +50℃
	存储温度	-40℃ ... +70℃
	相对空气湿度	<95%，不允许有凝露
	污染	符合IEC 61800-5-1，适用于污染等级 2 的环境
	电磁兼容性	符合IEC 61800-3
	防护等级	IP20

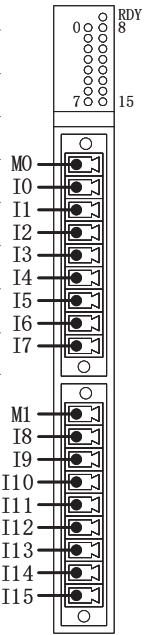
1.3.2 数字量输入模块

技术数据

X3000-DI-16-1	不需要外部电源供电
	电气隔离
	16个点
	每8个公共端，M0、M1
	电压：DC24V
	24V典型电流：4mA
	支持source和sink模式
	"1"信号电压：>11V
	"0"信号电压：<5V

指示灯状态

- RDY 绿灯指示模块连接状态
- 快闪表示等待配置中
- 慢闪表示正常运行中
- 0~15 绿灯指示输入状态
- 常亮表示输入1



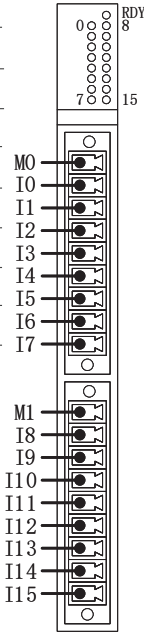
1.3.3 数字量输出模块

技术数据

X3000-DQ-16-3	不需要外部电源供电
	继电器类型
	16个点
	每8个公共端，M0、M1
	电压：DC24V、AC36V、AC48V
	连续电流：max 2A
	切换电流：max 2A
	触点类型：1常开

指示灯状态

- RDY 绿灯指示模块连接状态
- 快闪表示等待配置中
- 慢闪表示正常运行中
- 0~15 绿灯指示输入状态
- 常亮表示输入1



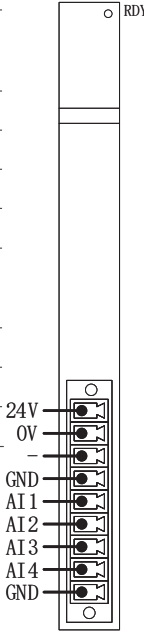
1.3.4 模拟量输入模块

技术数据

X3000-AI-04-3	外部提供DC24V ± 20%/max. 150mA
	4个通道
	单端输入，共GND
	电压/电流模式可编程
	电压型支持0V...10V
	电流型支持0mA...20mA、4mA...20mA
	电流型内阻250 Ω
	电压型：大于50k Ω
	精度：± 1%

指示灯状态

- RDY 绿灯指示模块连接状态
- 快闪表示等待配置中
- 慢闪表示正常运行中



1.3.5 模拟量输出模块

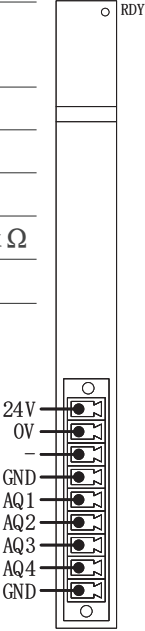
端子连接

X3000-AQ-04-1	外部提供DC24V ± 20%/ max. 150mA
	4个通道
	单端输入，共GND
	电压型支持0V ... 10V
	电压型负载电阻需大于5k Ω
	精度：± 1%

指示灯状态

RDY 绿灯指示模块连接状态
快闪表示等待配置中
慢闪表示正常运行中

注：模拟量的电压和电流输入检测切换是由，博能自定义库实现的。具体调用查看3.2.4小节



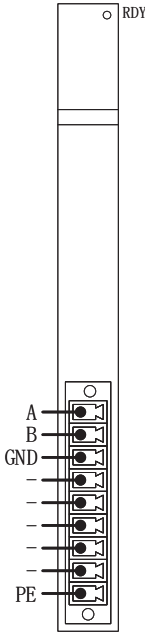
1.3.6 通讯接口模块

技术数据

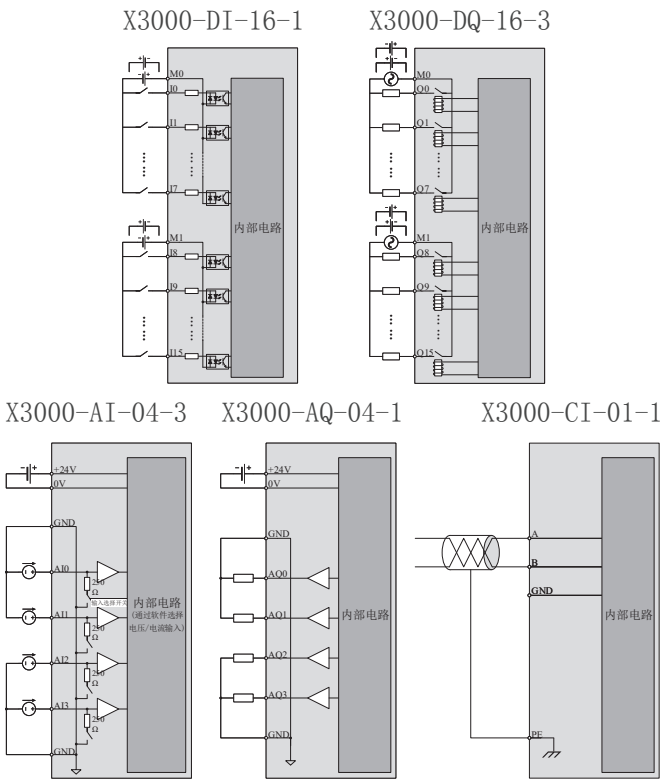
X3000-CI-01-1	不需要外部电源供电
	1个通道
	支持Modbus
	波特率：最大115200bps
	无板载终端电阻

指示灯状态

RDY 绿灯指示模块连接状态
常亮表示正常运行中

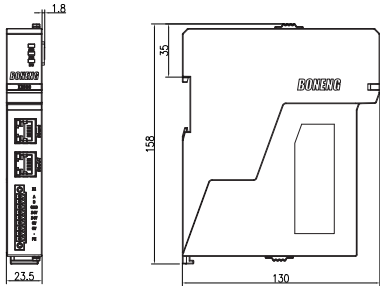


1.3.7 接线示意图

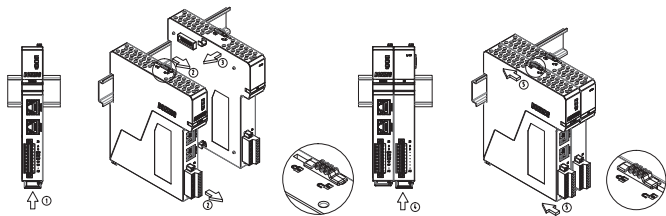


1.4 安装尺寸

模块外形及安装尺寸如下图所示。实际安装时，模块上方和下方需保证有60mm的空间，利于散热。



使用标准的35mm宽DIN导轨安装，将CUP安装至导轨上正确位置，底部的卡接滑块向上推使其固定。扩展模块对接完成后，上推卡接滑块使其固定，并将左侧模块本体上方和下方的固定锁扣拨至锁定状态。



2 控制器软件安装及编程调试

2.1 快速入门

2.1.1 BonengLogicLab软件安装

安装包请在博能传动（苏州）有限公司官网

www.boneng.com 下载最新版本。具体安装步骤如下：

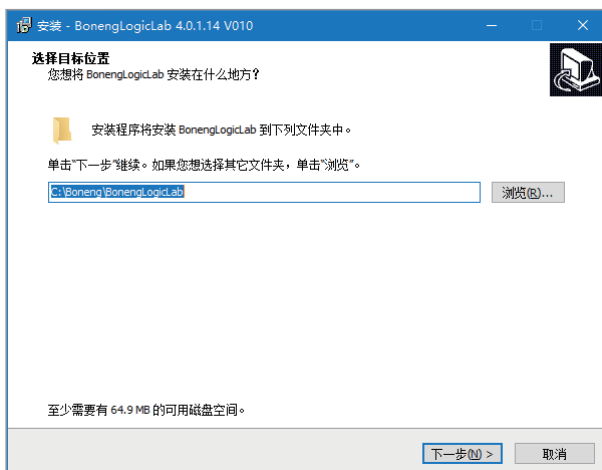
1) 运行安装包



2) 选择语言：简体中文、English两种语言可选；

3) 同意协议；

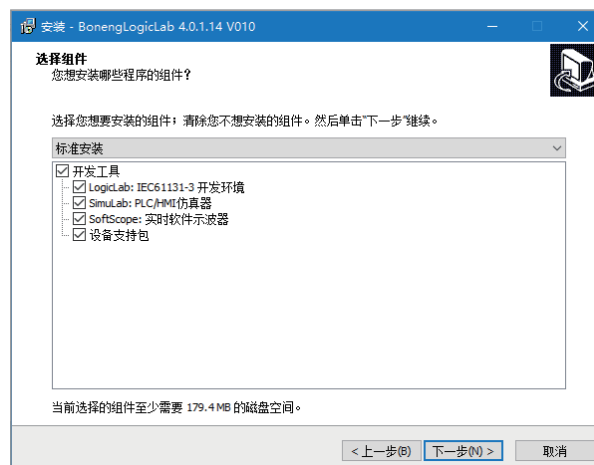
4) 选择软件的安装路径，点击浏览可以更改安装路径位置，安装路径下不能有中文；



如果点击下一步遇到文件夹已经存在，这时候需要去这个检查是否卸载了以前的编程软件，如果卸载了，点击是即可继续安装。



5) 选择标准安装；



6) 根据系统提示完成软件安装，安装好后桌面图标如图；



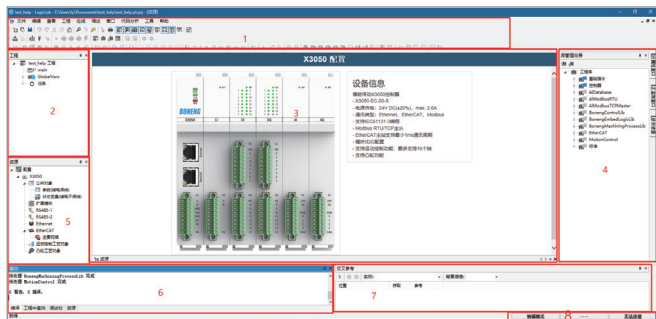
2.1.2 软件编程

2.1.2.1 软件简介

中小型的X3000系列的编程软件为BonengLogiclab，BonengLogiclab编程软件的用户界面友好，方便用户根据应用需求对PLC进行配置、编程、调试、下载、监控等操作。

编程界面

编程软件的主界面如下图所示：



- 1、主菜单和快捷工具:编程软件操作菜单,包含编程,调试,通信等相关设置,文件管理与编程工具的快捷方式。
- 2、程序工程区域,创建任务程序、全局变量列表、结构体、功能块、枚举。
- 3、主要显示区域,主要显示资源配置,和编程区域
- 4、其他辅助显示区域,主要包含库管理向导、属性窗口、监控窗口、示波器。
- 5、资源配置区域,主要是对外设设备的配置
- 6、输出显示区域,主要是显示编译结果,编译信息,版本信息等显示
- 7、交叉参考区域,主要是方便与快速定位全局查找变量,功能块的使用情况等信息;
- 8、状态显示区域,包含通讯连接状态等。

2.1.3 通讯设置

2.1.3.1 通讯连接方式

仅支持以太网与BonengLogicLab软件建立通信连接,实现程序的上传,下载,监控,调试。

2.1.3.2 以太网连接

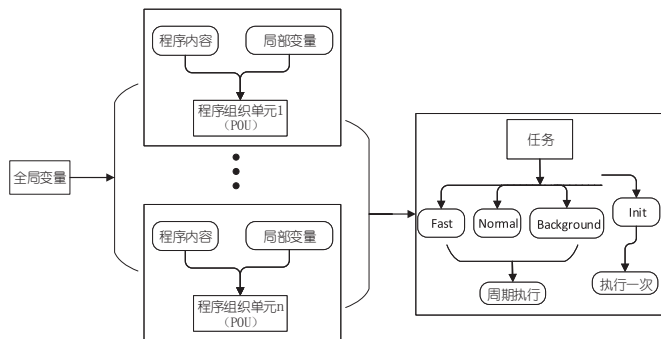
BonengLogicLab的连接非常简单,不需要手动去设置IP地址,只需要点击工具栏中的执行网络扫描,选择用户的设备然后点击确定即可,软件就会自动将该工程自动IP切换。便于后面的连接。

2.1.4 用户编程的典型步骤

初次使用博能PLC的用户需要注意,编写调试一个程序需要完成以下几个步骤,以X3000系列的PLC为例,进行说明。

- 1、新建X3000系列的工程
- 2、根据需要使用的外设进行硬件系统的配置
- 3、根据自己的任务要求建立任务程序,和变量
- 4、扫描网络选择控制,连接上PLC后,编译,下载,调试。

2.1.5 BonengLogiclab程序运行基本框架



全局变量与局部变量的区别：

全局变量：可以在该工程中任意程序组织单元中使用

局部变量：只可以在该程序组织单元中使用。

全局变量包括自定义的全局变量组, GlobalVars (原工程自带), I_0_mappings (配置扩展时产生), Global shared (配置参数, 状态变量时产生), Modbus_Vars (配置Modbus通讯时产生), 库管理向导中的变量等。

2.1.6 编程与调试

2.1.6.1 示例工程要求

在BonengLogiclab环境下完成一个灯闪烁程序的编程及其调试;使用一个16点数字输出模块, PLC编程实现

每间隔1秒导通和断开。由于数字输出模块是继电器类型的，在点亮的同时能听到滴嗒，继电器中电磁铁吸合的声音，并且指示灯也会亮起表示吸合。

2.1.6.2 新建工程

在BonengLogicLab编程软件中新建X3000系列的工程的操作如下：

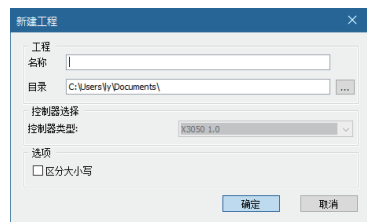
创建一个工程的方法有两种，第一种是在有控制器的条件下创建工程，另外一种是在没有控制器的条件下创建工程。具体的使用那种方法根据用户自己现场的情况而定。下面将介绍这两种方法：

第1种：

1. 打开BonengLogicLab编程软件，来到欢迎界面。点击扫描本地网络... 会弹出扫描到的实际连接好的设备，这里值得注意的是电脑的IP地址需要和控制器的IP地址处于同一网段。否则可能会扫描到设备后建立工程后无法直接连上设备。



2. 点击⊕会弹出如下对话框，输入用户的工程名称，选择用户要存放的工程的路径，点击确定即可。



第2种

1. 在欢迎页面中点击选择用户所需要的控制设备，然后点击创建基础工程然后，输入工程名称，工程存放路径点击确定，就会根据用户所选的设备创建一个基础工程。

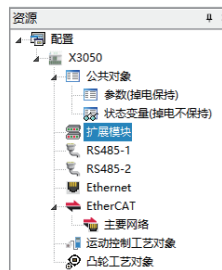


值得关注的是，在欢迎页面下方还有最新使用的工程的一个列表，会把用户最近打开的工程信息都记录并保存这个列表当中，方便直接双击打开该工程。在控制选择的时候，点击控制器下面的最近使用的工程列表会自动帮用户过滤掉其他控制器类型的工程，以便于快速锁定工程。

2.1.6.3 配置硬件系统

硬件配置的操作步骤如下：

创建一个新的工程后，在资源列表里面可以看到扩展模块：如下图所示：



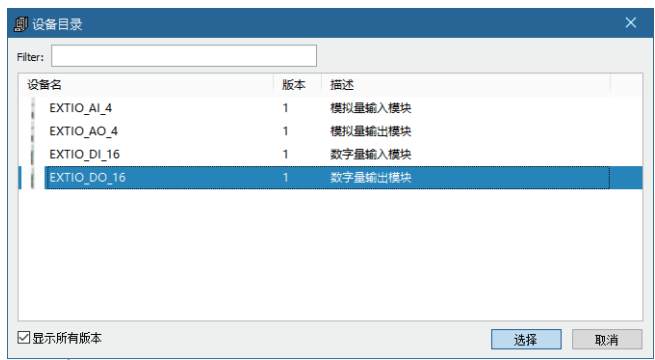
双击打开扩展模块，此时可以看到如下操作页面：



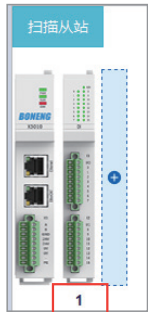
此时可以根据实际扩展模块的配置，手动或者自动添加5种扩展模块。具体使用那种，需要按照具体情况使用。

方式一、手动添加：点击上图加号，会弹出如下页面。

选择一种扩展模块，该模块的选择要对应实际安装的槽号，否则需要修改槽号的位置，修改扩展槽号，下面讲解。

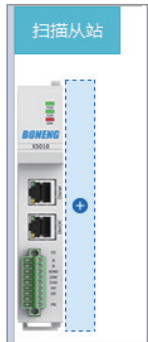


点击选择即可添加成功，添加成功后如图下所示：

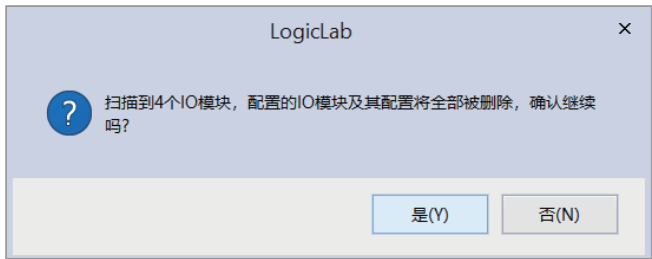


可以看到DQ模块下有一个标号1，可知手动添加的DQ模块在一号槽内。

方式二、自动添加，在下图中点击扫描从站，



点击后，会弹出没有发现扩展RS485接口，点击确定后出现扫描到的实际配置的扩展模块数量。如下图所示：



点击是（Y），软件内从站显示界面会按照实际背板槽号的摆放顺序进行排列，与之对应。在这里添加了1个扩展模块。

按照上述要求配置好扩展模块后，将每个输出点，分配其对应的PLC变量(可直接输入任意变量名称，保存后会直接生成对应的变量)。

点击添加好的模块，点击输出，在变量栏中输入与该点位对应的PLC变量名称与之绑定，如下图所示：



点击保存后，在工程栏中会出现一个新的全局变量组 I_0_mappings 在这个组内存放有上述输入的变量名称。方便后续在程序的编写当中使用。

2.1.6.4 IP地址修改

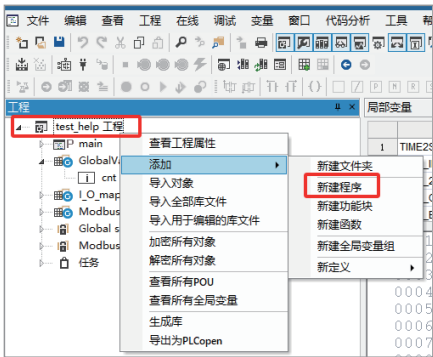
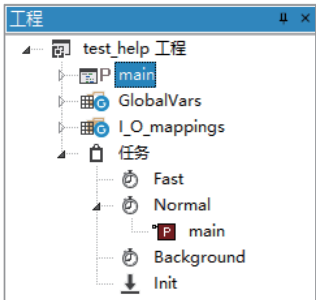
本控制器使用以太网与HMI或者是其他上位机建立通信时，此时，需要配置好PLC的IP地址。



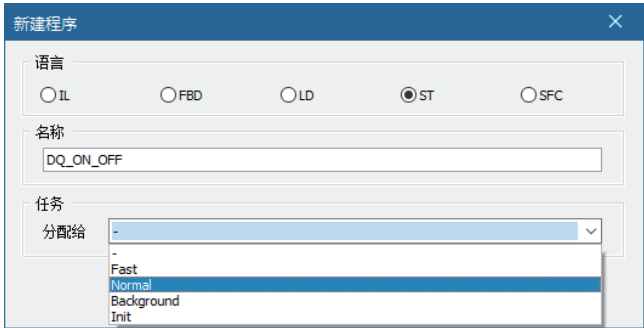
在欢迎界面中，先点击扫描本地网络，在扫描到的结果当中，点击后，会弹出对话框，修改该控制器的IP地址。控制器的端口号默认502，暂不支持修改。

2.1.6.5 编程与编译

创建好新的工程后会自带一个程序组织单元名称是main，该程序默认是使用ST 语言，在这里为了方便让大家了解如何创建程序组织单元，从而新建一个程序组织单元。

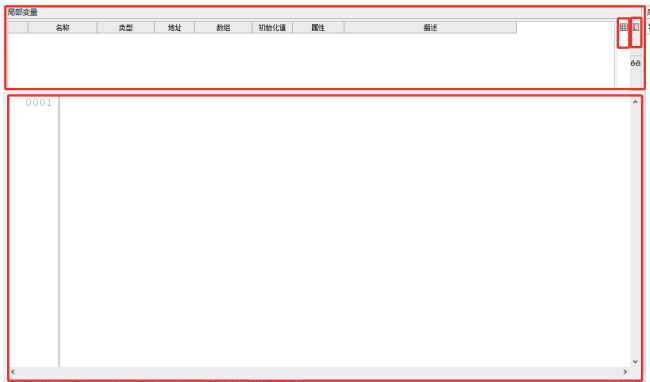


点击新建程序如下图所示：

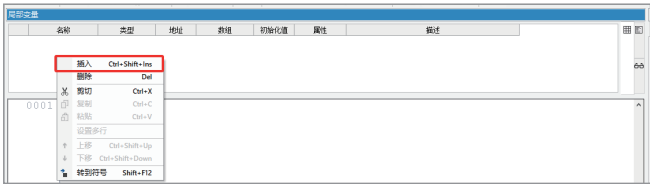


这里可以选择五中编程语言，这里选择ST，（其余的编程语言的操作方法和ST，基本是一样的），然后输入程序组织单元的名称这里取名叫DQ_ON_OFF，在这里可以将程序分配给自己想要放入的任务当中，这里选择Normal任务即可。然后点击确定。

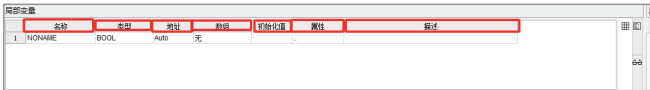
在工程中点击上面新建的POU, 如下图所示：



在局部变量在可以新建局部变量，局部变量的创建有分为两种模式，第一种是表格形式，另一种是文本形式。在这里推荐使用表格形式，在局部变量的空白区域右击鼠标插入，这样就可以添加一个变量。



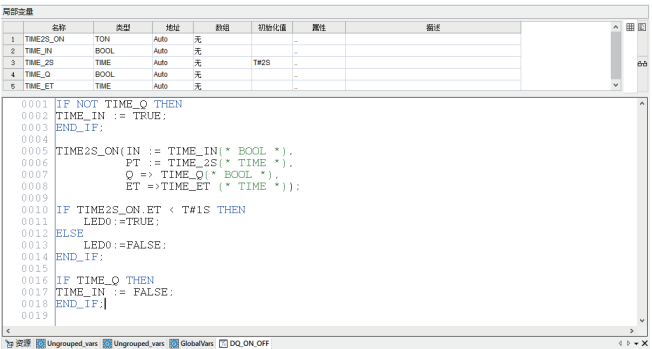
在添加好一个默认的变量后，用户可以去修改这个变量的相关配置。一个变量该有的设置，包含变量名称，变量类型，变量地址（不设置默认为Auto），变量是不是数组，变量的初始化值，变量是常量还是掉电保持变量（在变量的属性当中去修改），以及该变量的描述。值得注意的是，掉电保持变量只能是全局变量。如下图所示：



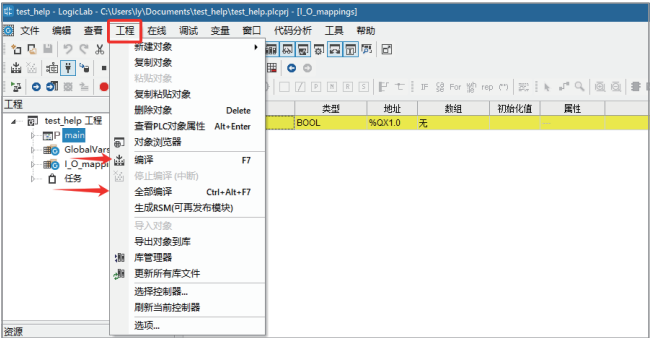
在修改变量的类型时，可以手动输入变量类型。也可以点击下图所示位置，调出对象浏览器选择所需要的变量类型。



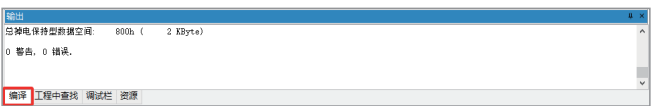
现在利用上面创建PLC变量的方法，相继创建如下图所示变量，将上述DQ_ON_OFF中的程序进行编辑，编辑后的程序如下：



点击快捷菜单栏中的 按钮，或者点击菜单栏中的工程，点击编译，或者全部编译。如下图所示：



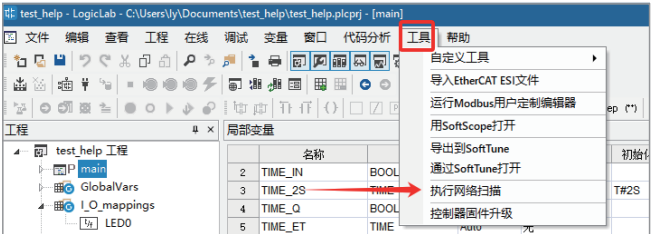
点击后在输出框中的编译栏中会输出许多信息，最后没有错误之后可以进行下载操作。



2.1.6.6 通信设置

本控制器使用以太网与PLC进行连接。

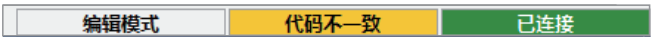
在菜单栏中点击 按钮进行连接，如果提示连接失败，则点击工具，点击执行网络扫描。



如果网线连接正常将会弹出扫描到的设备选择用户所需要连接的设备，点击确定即可。如下图所示：



这时点击连接即可成功连接，在状态栏中即可明显看到连接状态，如果还是没有连接成功，检查网线是否插好。



2.1.6.7 程序下载

BonengLogicLab 编程软件中的程序下载操作步骤如下：在快捷菜单栏中，点击如下图所示按钮即可



然后会弹出一些配置发生了变化，需要重新启动控制器，点击是即可。

下载完成后状态栏会显示工程源代码匹配，如下图所示。



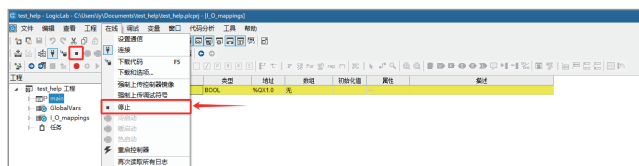
2.1.6.8 显示结果运行

当下载成功后PLC会自动运行起来，当听到每间隔1s响一次时，程序就成功运行了。

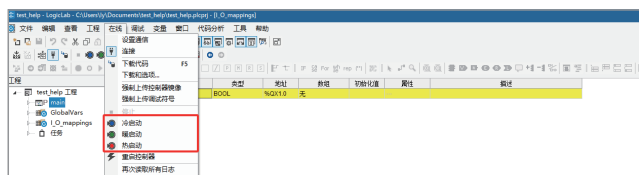
2.1.7 切换PLC工作模式

PLC有运行和停止两种工作模式，

停止PLC的运行只能通过BonengLogicLab 编程软件点击快捷菜单栏中的按钮或者点击在线点击停止如下图所示：



运行PLC程序有三种运行模式，点击快捷菜单中的按钮  ，按照颜色顺序依次为，冷启动、暖启动、热启动，或者在菜单栏中点击“在线”然后选择用户要启动的模式。如下图所示



三种启动方式之间的差异。

冷启动：PLC应用程序将被重新启动，可保持型变量和非保持型变量都将被复位；

暖启动：PLC应用程序将被重新启动，并且非掉电保持型变量将被复位；

热启动：PLC应用程序将被重新启动，不会复位任何类型的变量；

2.1.8 在线监控与调试

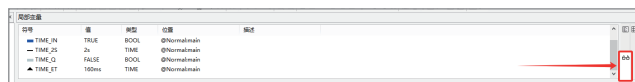
2.1.8.1 在线监控

BonengLogicLab 编程软件支持在线修改变量的实时

值，这样方便用户在使用中更快的进行调试。在线调试又分为变量列表监视（方便直接快速监控这个变量列表中的变量值），和单独的监视窗口（适合将各个地方的变量放到一块进行整合监控）。和直接在程序中监控变量值。

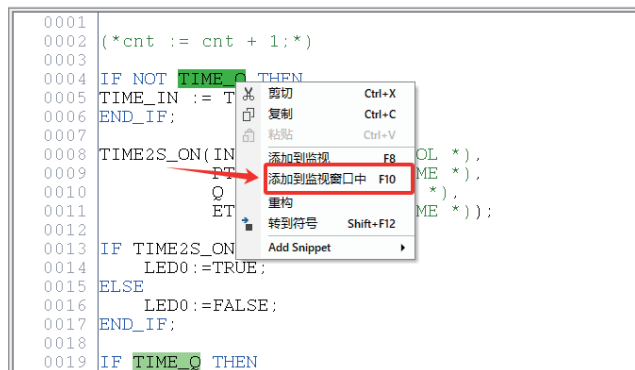
- 在变量列表中直接监控变量的值

选择一个变量列表，点击即可查看，如下图所示

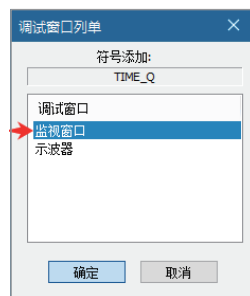


- 使用单独的监视窗口

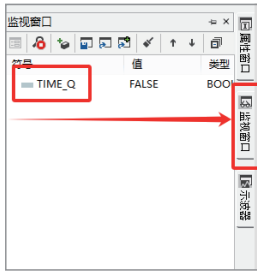
选择中TIME_Q这个变量右击，点击添加到监视窗口中。



然后点击选择监视窗口即可。

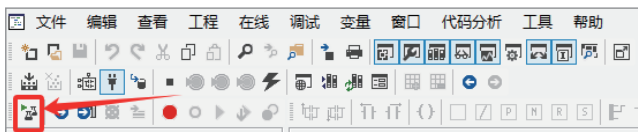


将鼠标放在右侧监控窗口上，监控串口弹出即可监控刚才添加的变量。

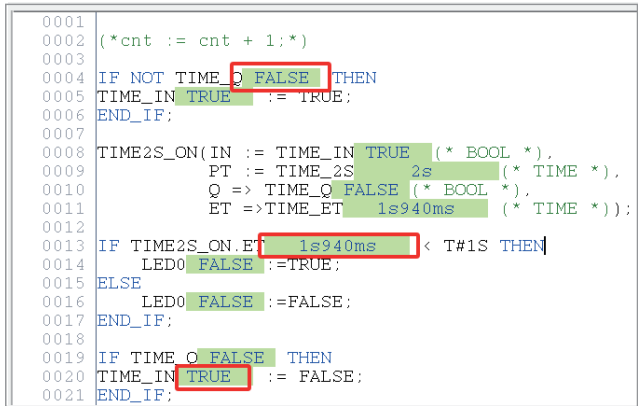


• 直接在程序中监控（也是在线调试所需步骤）

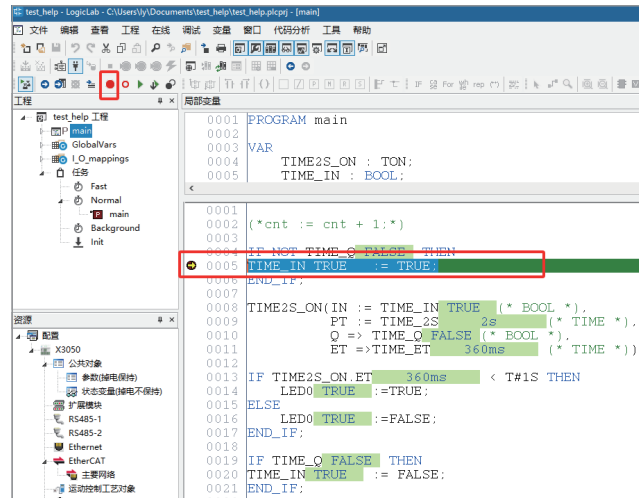
在快捷菜单栏中点击如下图所示按钮（在线调试模式）即可进入在线调试页面。



点击后程序编辑页面将不再运行被编辑，可直接查看该程序页面中所含变量的值，如下图所示：



行，断点列表。



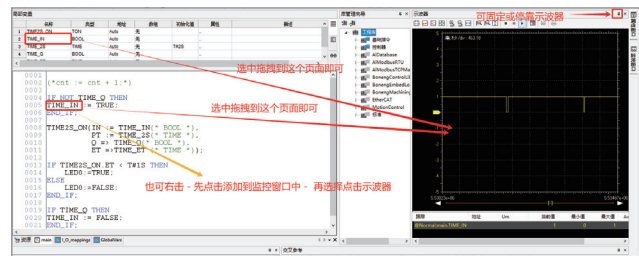
2.1.9 示波器应用

2.1.9.1 示波器功能简介

示波器分析该变量的最大值。最小值，信号之间的差值监控等。往往在数据复杂的时候就需要借助示波器来分析，多个数据之间的关系。以图形化的方式，查看变量的值的变化情况，以帮助用户快速分析程序的运行情况。

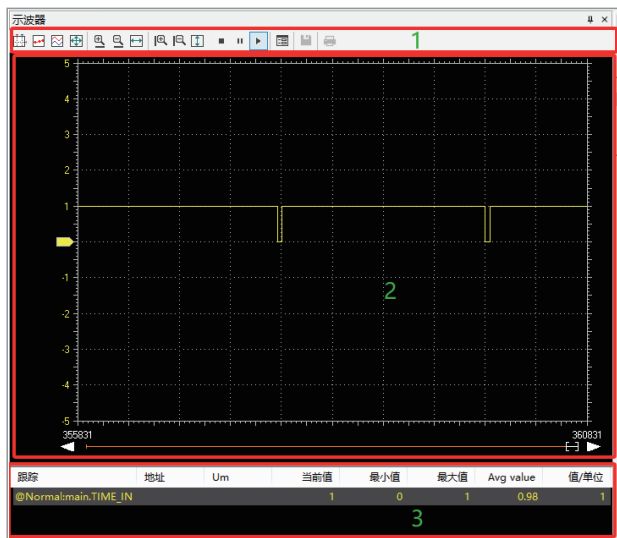
2.1.9.2 添加变量到示波器

添加变量到示波器分为两种，第一种是直接拖拽的形式，将在变量窗口区域的变量，或者是程序串口中的变量选中后，拖拽到示波器的窗口当中。第二种是右击点击的形式，右击添加到监控窗口中，在选择示波器即可。



2.1.9.3 示波器的功能介绍

示波器页面主要分为三个区域1、示波器控制及调试区域。2、主要波形显示区域 3，监控的变量列表存放区域。如下图所示：



一、示波器控制及调试区域，包含以下几个常用调试按钮，如下图表格所示：

图标	图标名称	功能
	显示测量栏	开启后会出现两条水平光标（分别为一条蓝色光标，和一条红色光标），和两条垂直光标（分别为一条蓝色光标，和一条红色光标）。分别用于测量水平以及垂直方向的值。开启后，鼠标可以在示波器的显示区域，调整光标位置来获取当前水平或者是垂直高度的值。
	显示样本	将示波器的采样的样本点给显示出来。
	垂直分割	用于当页面有多个变量同时采样时，可以一键将其分割开，便于观察
	显示所有值	将采样到的所有的水平或者竖直方向的值，全部显示到当前页面中

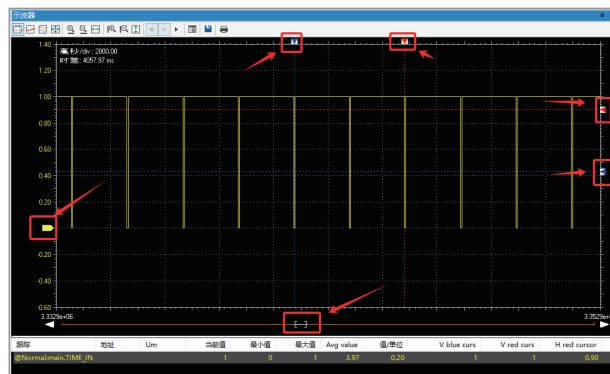
图标	图标名称	功能
	水平放大	选中某个变量后可以将其水平放大
	水平缩小	选中某个变量后可以将其水平缩小
	水平显示所有值	选中某个变量后可以显示其水平方向的所有值
	垂直放大	选中某个变量后可以将其垂直放大
	垂直缩小	选中某个变量后可以将其垂直放大
	垂直显示所有值	选中某个变量后可以显示其垂直方向的所有值
	停止采样	停止采样，停止后重新开始会丢失缓冲区数据
	暂停采样	暂停采样，点击开始不会丢失缓冲区数据
	开始采样	点击后示波器开始采样运行

二、主要示波器显示区域

示波器显示区域主要有以下几点可操作

- ◆ 两条水平光标（分别为蓝色，和红色），和两条垂直光标的拖动（分别为蓝色，和红色）
- ◆ 调整波形的零点起始位置（也就是可以上下查看其波形）
- ◆ 调整波形的左右位置（也就是可左右查看缓冲区其余波形）

以上三种可操作的操作方式均是使用鼠标左键点击拖拽完成，如下图所示：



三、监控的变量列表存放区域。

该区域的主要功能是，查看添加进来的变量名称，已经该变量的最大值，最小值，平均值（Avg value），测量水平的蓝色光标值（V blue cursor），测量水平的红色光标值（V red cursor），测量垂直的蓝色光标值（H blue cursor），测量垂直的红色光标值（H red cursor），以及描述等。如下图所示：



如果没有完全显示，鼠标拖拽标题栏分隔线位置（上图红色箭头所指部分），以展开完全显示。

2.1.10 程序扫描周期设置

BonengLogiclab 的任务分为4个，分别为Fast、Normal、Background、Init，将程序组织单元放到里面过后，该程序才能正常运行。在工程窗口中，任意选择一个任务（这里右击Fast），右击，或者双击任务，如下图所示：



点击任务配置后如下图所示：

ID	名称	类型	设置周期	周期(ms)	描述
0	Fast	Cyclic	No	2	Fast task
1	Normal	Cyclic	No	20	Normal task
2	Background	Cyclic	No	100	Background task
3	init	Single	No	..	init task

修改程序的周期，需要先将上图中设置周期修改为Yes, 然后点击后面的数字进行选择，或者输入周期值，对于Fast任务的周期值时只允许，进行选择。而Normal任务和Background 任务可以直接键盘输入。但是，在输入周期值时，只能是Fast任务的周期的整数倍。

2.1.11 EtherCAT任务周期设置

EtherCAT的任务程序是在后台运行的，与Fast任务进行绑定，当Fast任务的周期值发生改变时，EtherCAT的周期也会随之发生改变。

2.1.12 导入与导出对象

在BonengLogiclab中，可以将工程窗口中的POU导出，成为一个单独的文件，在其他的工程中使用到的时候，可以直接导入这个POU。操作方法下：

2.1.12.1 导出对象生成库

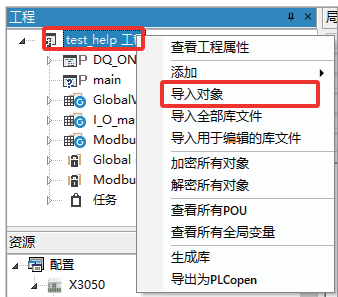
选择一个POU然后右击，选择导出到库，输入文件名称，选择是否加密这个库，然后点击确定即可，在该工程目录下，会自动生成该名称的文件。



名称	修改日期	类型	大小
Build	2024/3/4 10:09	文件夹	
Download	2024/3/4 16:38	文件夹	
Previous Versions	2024/3/5 10:17	文件夹	
test_EXT	2024/3/5 10:18	文件	1 KB
test_help.exp	2024/3/5 10:17	EXP 文件	1 KB
test_help.imgx	2024/3/4 16:40	IMGX 文件	27 KB
test_help	2024/3/5 10:17	PLCPRJ 文件	778 KB
test_help.PLK	2024/3/5 10:17	PLK 文件	1 KB
test_help.wkpx	2024/3/4 17:08	WKPX 文件	1 KB

2.1.12.2 导入对象

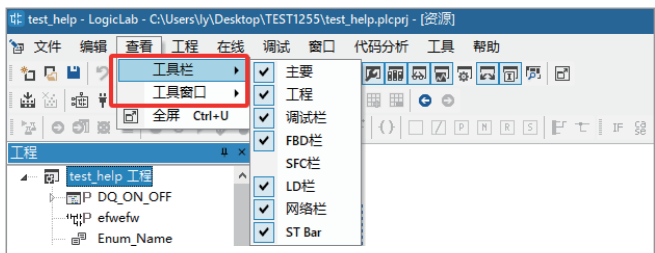
右击工程的名称，点击导入对象，找到用户需要导入的文件，点击即可，选择性导入用户需要的对象。方便快速移植程序。



2.2 编程基础

2.2.1 概述

BonengLogiclab的变量分配是比较灵活的,可以将变量分为全局变量区域和局部变量两大类。在快速入门中已经介绍了,相关变量的参数设置。这里就不在重复说明。弄清楚变量的类型,即可快速上手该软件。简单来说,在左侧,工程窗口中的变量,都是全局变量;当然还有少部分全局变量在库中。在使用过程中,没有找到某些窗口,可通过,点击菜单栏中的查看,将工具栏中,工具窗口,勾选上自己想要打开的窗口。如下图所示:

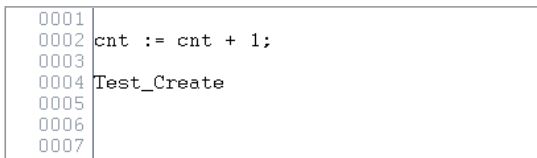


2.2.2 变量

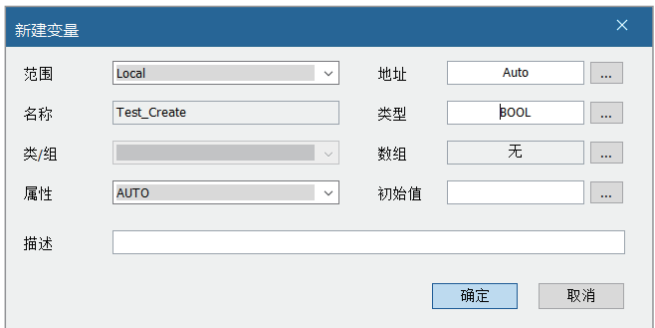
2.2.2.1 创建简单变量

在BonengLogiclab中创建一个变量的方法有很多,在快速入门当中提到,可以在表格编辑格式中鼠标右击

添加一个变量,也可以切换到文本格式中,进行文本格式的编辑。下面将介绍其余的快速创建变量的方法。在ST程序当中,可以直接在文本编辑区中,输出用户想要创建的新的变量名称,如下图所示:



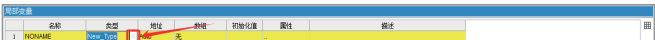
然后回车 (Enter), 结束输入,会弹出如下对话框:



在这个页面当中可以设置变量的范围, 变量的地址, 没有特殊的用法情况下, 变量的地址都是默认Auto即可, 还可重新更改变量的名称, 变量的类型。在ST的编辑模式下, 这里可以选择功能块、函数类型, 而在LD当中出现这个对话框时, 只能选择基础变量类型。如果是在这个地方选择全局变量的话, 需要给变量分配一个组别。将该全局变量放到该组别当中。

2.2.2.2 对象浏览器

在创建一个变量的时候, 通常需要去修改其变量类型, 此时变量类型分为, IEC基本类型, 和用户类型。或者是分配一个功能类型, 函数类型。此时庞大的类型种类名称无法做到每次直接输入正确, 此时就需要一个搜索引擎帮助查找自己想要的变量类型。在变量的表格中, 点击类型中的小白块, 如下图所示:

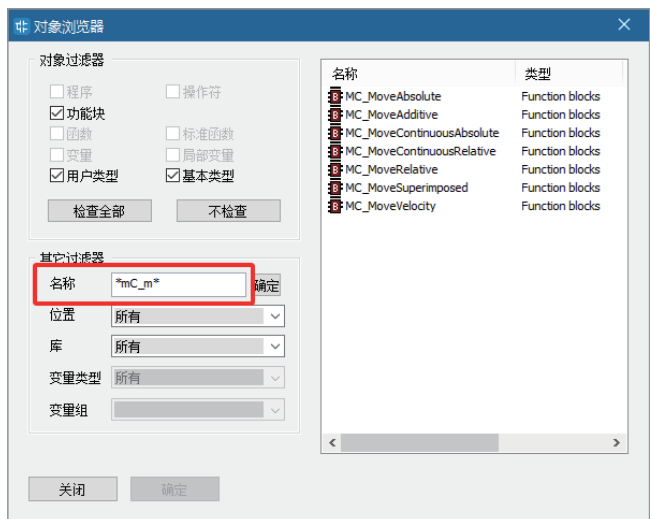


点击后会出现, 对象浏览器, 如下图所示:



在这个对象浏览器中，可以分为3部分，其中包含两个过滤器（左侧），一个过滤结果框（右侧），通常使用上面的过滤器即可，其他过滤器的适用于当工程已经非常庞大时，通过上面的对象过滤器后依然会有很多结果时，使用。且其他过滤器时，会受到上面的过滤器的影响，当上面的对象过滤器不勾选时，下面的其他过滤器是无法筛选出结果的。

在其他过滤器中，名称一栏的输入规则是，使用*为通配符，且不区分大小写。如下图所示：

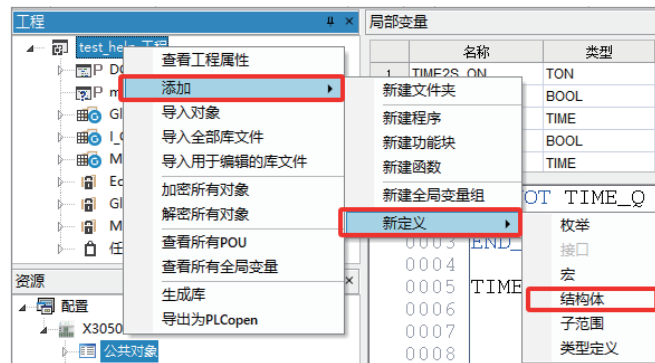


2.2.2.3 创建复杂变量类型

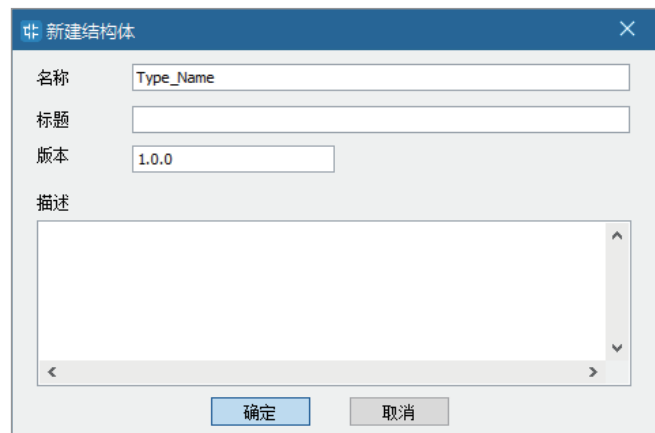
在BonengLogiclab中可以在工程窗口中创建复杂的变量类型，方便自己使用。比如，结构体，枚举等。

2.2.2.3.1 结构体类型的创建

在工程窗口中，右击工程名称，添加，点击新定义，点击结构体。如下图所示：



点击结构体后，会弹出如下图所示窗口

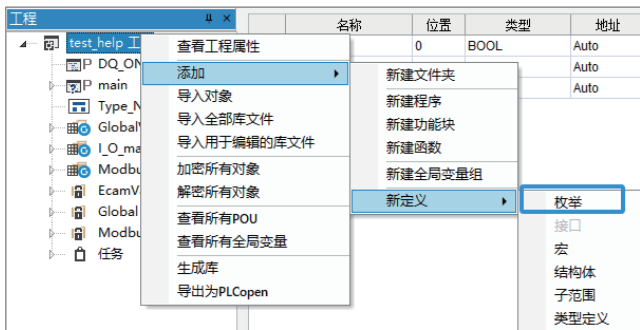


输入结构体类型的名称，点击确定即可。其中标题和描述都是用以给自定义的结构体类型做解释说明的作用。留空即可，而版本号则是为了在导入，导出时，更好的区分该版本。创建好该结构体类型后，即可下一步创建该结构体类型的成员。在工程窗口中，点击刚才创建好的结构体类型。在右侧主窗口中，鼠标右击即可添加该结构体类型的成员。

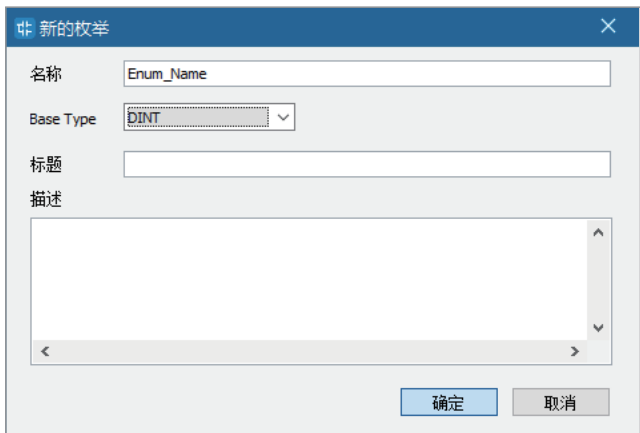


2.2.2.3.2 枚举类型的创建

在工程窗口中，右击工程名称，添加，点击新定义，点击枚举。如下图所示：



点击后如下图所示：在下图可选择用户需要的枚举的基本类型。标题和描述依然可以留空。

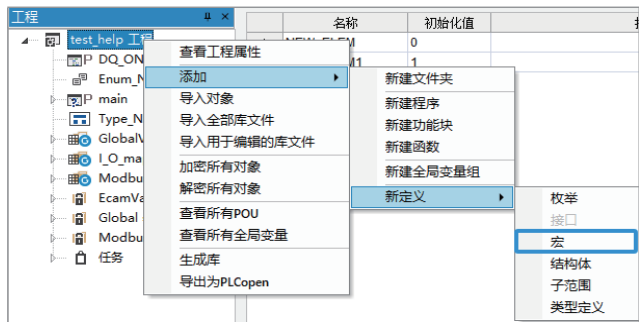


点击添加好的枚举类型，在右侧主窗口中可以鼠标右击，插入枚举的成员。枚举的成员可以更改其名称，以及初始的值。如下图所示：

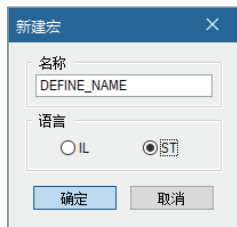


2.2.2.3.3 宏创建

在工程窗口中，鼠标右击工程名称，添加，点击新定义，点击宏。如下图所示：



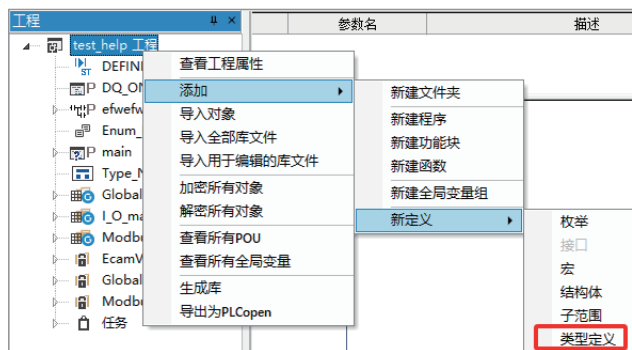
点击宏后会弹出如下对话框，在名称中，输出宏的名称，选择语言，在使用宏的时候只支持在ST和IL中使用宏。这里选择常用的ST。



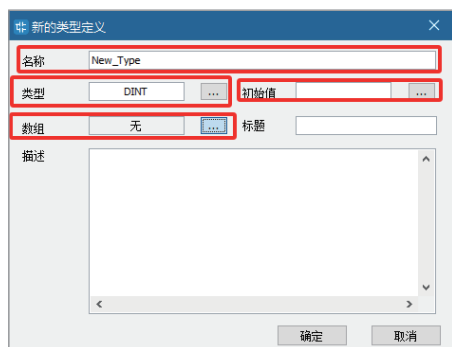
点击确定后，会弹出一个编辑窗口，在该编辑窗口中，可以编辑程序，然后再ST中使用这个宏，即可代替大段程序。

2.2.2.3.4 类型定义创建

在工程窗口中，右击工程名称，添加，点击新定义，点击类型定义。如下图所示：



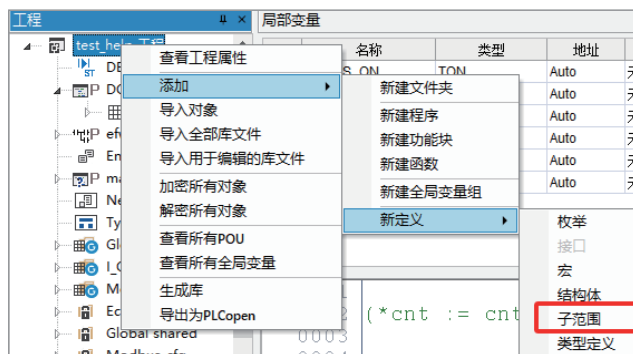
点击后如下图所示：



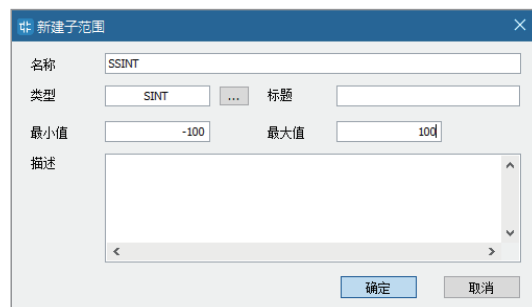
在新的类型定义中，需要设置新类型的名称，类型，初始值，数组，等。总而言之，新类型定义，实际就是把软件中自带的变量类型，包含结构体，功能块，函数等类型的名称，重新命名，然后再去用户再去调用这个自己习惯的变量类型名称进行创建变量的命名。

2.2.2.3.5 子范围类型

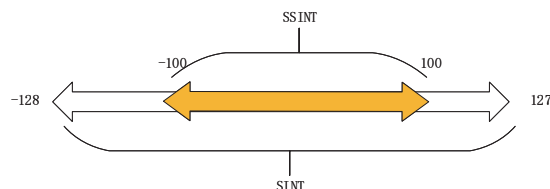
在工程窗口中，鼠标右击工程的名称，添加，点击新定义，然后点击子范围。如下图所示：



点击后出现如下对话框，这里需要填写子范围的名称、选择或输入类型、输入最大值、最小值。如下图所示：

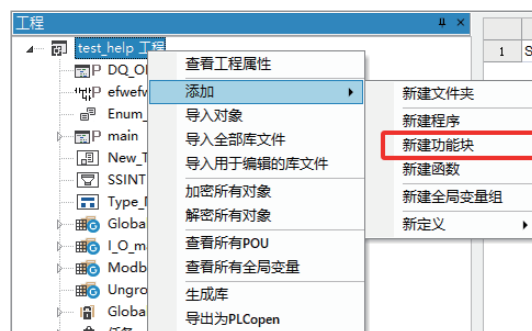


根据上述设置后子范围示意图如下图所示：

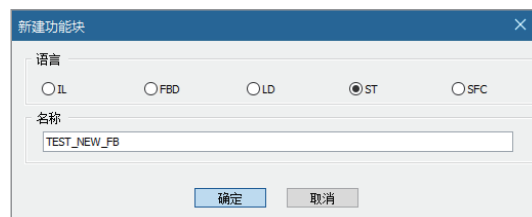


2.2.2.4 创建功能块

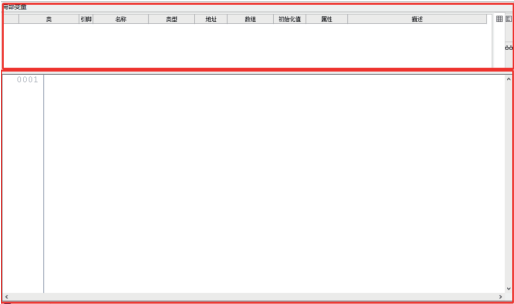
鼠标右击工程名称，点击添加，点击新建功能块，如下图所示：



点击后会弹出如下对话框，需要选择创建该功能块所使用的语言，以及该功能块的名称。



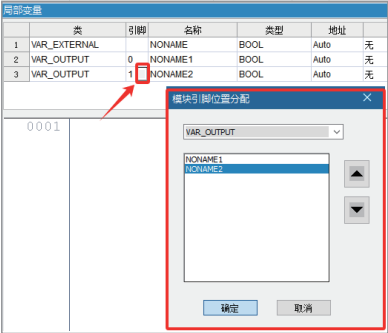
点击上面创建好的功能块，会弹出编辑该功能块的窗口，



该窗口和程序编辑窗口类似，但是功能块中的局部变量编辑，需要将该变量分类，包含普通内部变量，输入变量，输出变量，输入输出变量，外部变量。如下图所示，

类	引脚	名称	类型	地址	数组	初始化值	属性	描述
VAR_INPUT	0	NONAME	BOOL	Auto	无			
VAR_OUTPUT	1	NONAME1	BOOL	Auto	无			
VAR_EXTERNAL	2	NONAME2	BOOL	Auto	无			

还可以根据自己的习惯，可以将引脚的顺序进行重新排序，排序的作用是为了改变引脚在功能块中的顺序，点击引脚的白色小方块，可弹出模块引脚位置分配对话框，如下图所示：



在该对话框中，可选择需要排序的引脚类型，以及鼠标点击上下按钮即可排序。

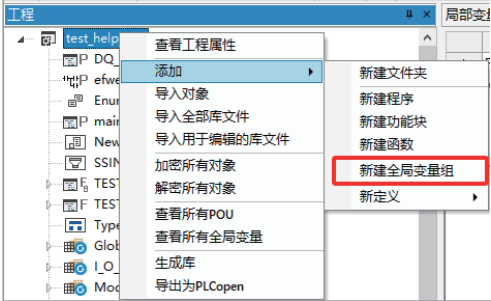
2.2.2.5 创建函数

创建函数与创建功能块类似，但是由于函数本身的特点，只允许有一个输出管脚，所以在添加的时候，局部变量中，已经规定好输出引脚为RESULT。且在类的选择中，无法选择输出引脚。

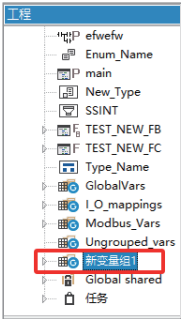
类	引脚	名称	类型	数组	初始化值	属性	描述
RESULT		TEST_NEW_FC	BOOL	无			运行结果
VAR		NONAME	BOOL	无			

2.2.2.6 创建全局变量组

创建全局变量组的作用是为了更好的管理自己的全局变量，鼠标右击工程，点击添加，点击新建全局变量组，如下图所示：



添加后可实现自己的变量分组管理，如下图所示：



2.2.2.7 指针变量类型

指针类型变量使用起来比较简单，但是同时也是比较危险的操作。所以不建议使用指针类型的变量。指针类型的变量需要在类型的前面插入@符号以表示该变量是一个指针变量。

名称	类型	地址	数组	初始化值	属性	描述
cnt	@DINT	Auto	无	---		
cnt1	DINT	Auto	无	---		

取地址操作和指针变量，一般是成对出现的。在ST中取地址可以使用“?”来取一个变量的地址例如：cnt := ?cnt1;在变量的前面加上?表示取该变量的地址，当然也可以使用库中的指令“ADR ()”实现取地址操作。

2.2.3 资源-参数（掉电保持）

在资源窗口中，展开公共对象，可以看到参数（掉电保持）的节点，该参数的页面主要是用于设计Modbus通讯（TCP/RTU）时，控制器作为“从站”时的寄存器地址。

2.2.3.1 添加一个寄存器地址

如下图所示：点击添加即可添加一个寄存器地址



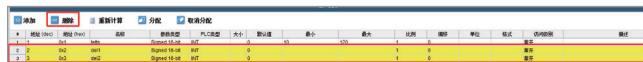
添加完成后需要给该寄存器地址输入一个名称，该名称自动变动该地址



地址(dec)	十进制的方式显示寄存器地址
地址(hex)	16进制的方式显示寄存器地址
名称	变量名称，可自行输入，可也以分配，之后自动绑定该地址。
参数类型	通讯的参数类型，可选择
PLC类型	PLC中显示的参数类型，可选择
默认值	默认的初始值
最大	能接收的最大值，超出范围通讯失败
最小	能接收的最小值，超出范围通讯失败
比例	接收完成后的数据放大或缩小的比例（仅对于PLC端而言，对于主站端不变），无特殊要求默认为1。
偏移	接收完成后的数据偏移的大小（仅对于PLC端而言，对于主站端不变），无特殊要求默认为0。
描述	对这个变量添加描述

2.2.3.2 删除一个寄存器地址

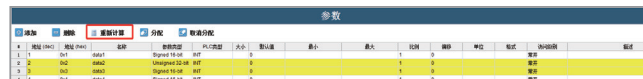
选择需要删除的行,点击删除按钮即可,如下图所示:



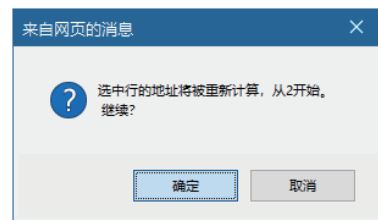
按住Shift键，或者Ctrl键可以多选，选中的对象变为黄色。

2.2.3.3 重新计算

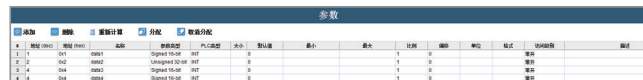
重新计算的功能是当数据类型出现32位时，因为通讯的数据都是按照16位进行传输的，而如果此时的地址还只是加1的话就会出现错误，那么就需要重新计算PLC地址。重新计算的操作方式如下图所示：



选择中32位数据以及下方一行的数据，点击重新计算。点击后如下图所示：



点击确定



可以看到data3的地址已经不是从3开始了，而是从4开始的。但是发现data4的地址已经是4了，所以，重新计算时，通常需要将所有的地址全部选择起来，然后点击重新计算，这样的话可以保证每个变量的地址的唯一性，以及合理性。全选之后重新计算如下图所示：

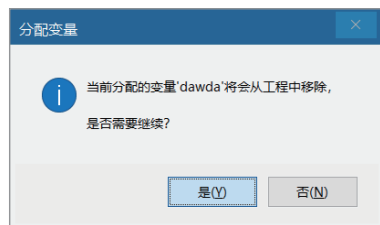


另外一种需要重新计算的情况是，由于复制粘贴时，多个变量使用一个地址(或者地址顺序错乱)，这样是不允许的，所以可以重新计算以便于重新分配地址。

2.2.3.4 分配

选中个地址（某行），点击分配，软件会提示，将

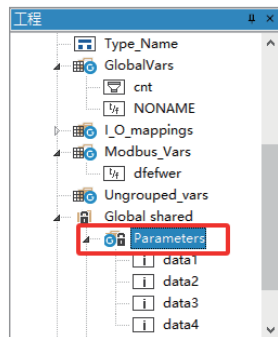
会把原来的变量给移除，并且会将该变量从工程删除，是否继续？如下图所示：



点击是，选择好变量（只有全局变量）后，即可。

2.2.3.5 取消分配

选中某个地址（某行），点击取消分配，会把该变量从工程中参数组中移除



2.2.4 资源-状态变量（掉电不保持）

状态变量和参数基本相同，也是Modbus通讯使用的。但是状态变量的寄存器地址是从8192开始的，且掉电不保持，值得注意的点是，如果没有做特殊修改，在添加一个地址后。该地址是只读的，无法修改。如下图所示：



如果需要写入数据的话，那么需要将，只读改为false即可。

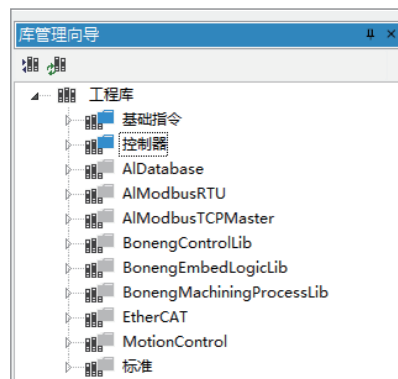
2.2.5 库管理向导

这里只介绍库管理向导的功能，包含导入库，导出库，等操作，不会去说明，其中每个功能块，以及

函数的功能。详细功能块说明请看第三章指令说明。在BonengLogiclab中，库文件对于整个工程，具有非常重要的作用。管理好自己的库文件，以及学会调用库文件去编写自己的程序。库文件是以“.plclib”结尾的文件。

2.2.5.1 库文件简介

BonengLogiclab提供了许多库，当新建工程后，就可以看到，以X3050工程为例，如下图所示：



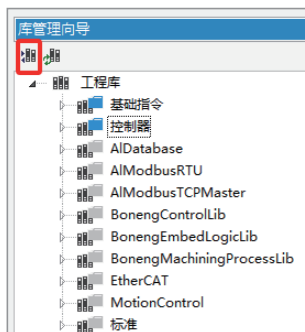
根据上图，下面的表格将对上图做出简单说明：

库名称	说明
基础指令	包含一些最基本的指令，例如转换，算数，逻辑，比较等
控制器	包含一些系统变量，系统时钟等。
AlDatabase	包含对数据区域的存储
AlModbus-RTU	包含一些和ModbusRTU的功能块，函数，一般使用页面配置来完成，而不使用这个库
AlModbusT-CPMaster	包含一些和ModbusTCP的功能块，函数，一般使用页面配置来完成，而不使用这个库
BonengCon-troLib	包含了对伺服，变频器的简单控制的库
BonengEm-bedLogiclib	包含了一些需要特殊用法的嵌入式库

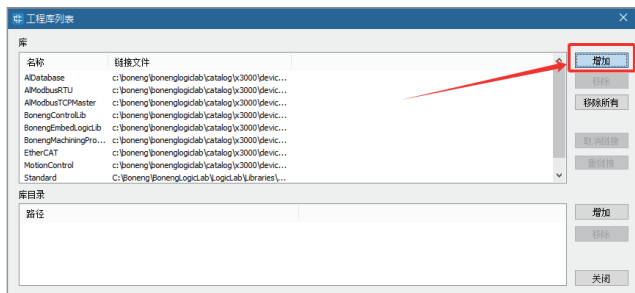
库名称	说明
BonengMachining-ProcessLib	包含了收放卷辊相关功能的库
EtherCAT	包含了对ECAT通讯相关的功能库，一般不使用
MotionControl	包含了复杂的运动控制相关的库
标准	包含了定时器，置位，复位，边沿触发，等功能的库

2.2.5.2 导入库文件

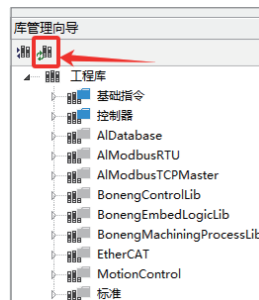
点击库管理向导窗口页面点击库管理器按钮，如下图所示：



点击后会弹出如下图所示窗口：

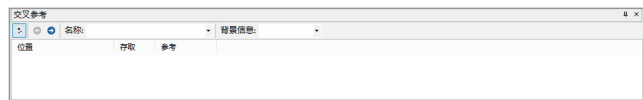


点击增加，找到以“.plc.lib”文件结尾的，点击确定即可。如果库中仍然没有显示增加的文件，可尝试点击刷新库文件如下图所示：



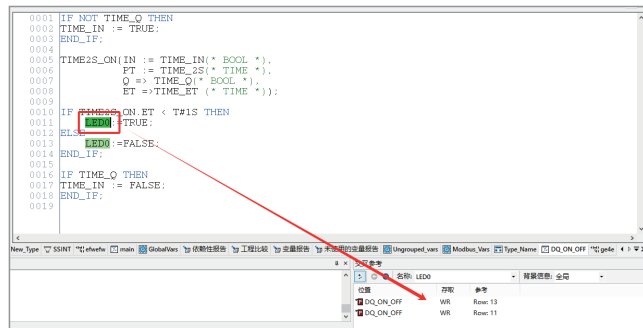
2.2.6 交叉参考

交叉参考的作用是，快速定位某个全局变量，在整个工程中使用的位置及其使用情况，交叉参考，只能检索到加入了任务的程序。如果该程序没有加入任务，是无法检索到的。交叉参考窗口如下图所示：



2.2.6.1 交叉参考的一般步骤

在交叉参考中，需要点击这个按钮，以保证能自动显示对象。点击后，只需要在程序中，使用鼠标点击该变量，即可在交叉参考的窗口中，出现该变量的使用情况，点击交叉参考中检索出来的结果，会自动跳转到该变量所在的行。LD语言也同样适用。



2.2.6.2 直接选择或输入变量名称

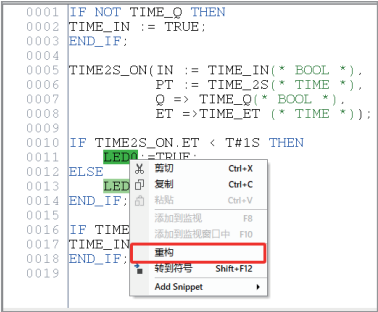
在交叉参考中，还可以在名称一栏中，直接输入或者选择变量的名字，输入完毕后，下面也会直接显示检索结果。还可以检索某种功能块的使用情况，

例如检索TON定时器 在整个工程中的使用情况等如下图所示：

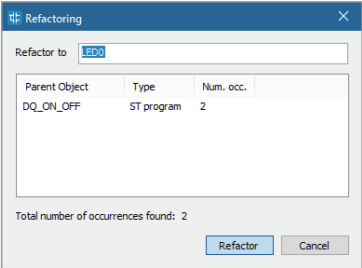
交叉参考			
名称: TON		背景信息: 全局	
位置	存取	参考	
ge4e	DECL		
DQ_ON_OFF	DECL		

2.2.7 重构

重构的作用是，将某个变量的名称重新命名，且会修改在整个工程当中，使用到这个变量的名称也会被更改。具体的操作方法如下：在程序中，选择一个变量，鼠标右击，选择重构：如下图所示：



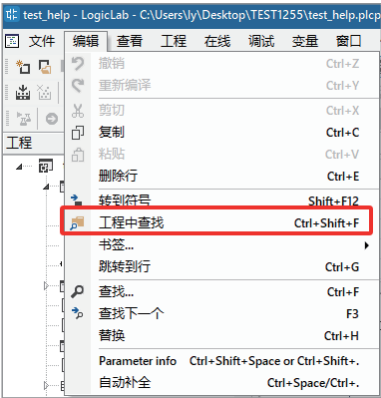
点击重构后会弹出如下窗口：



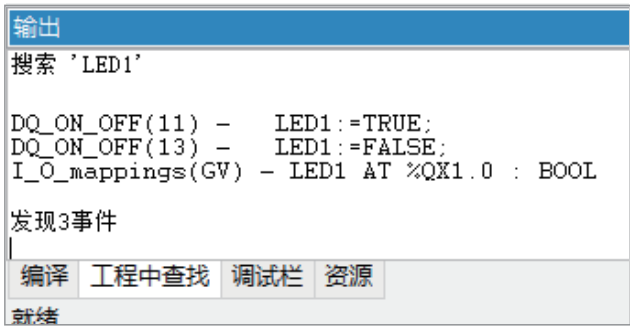
在该窗口中，会列出那些地方使用到这个变量。在这里还需要输入重构后的名称，点击重构，即可完成。

2.2.8 查找

在BonengLogiclab 中提供了查找变量的工具，和交叉参考类似，分为局部查找，和在工程中查找（全局查找）。在菜单栏中点击编辑（需要在程序编辑页面），点击工程中查找如下图所示：



点击后会弹出过滤器窗口，如下所示：



点击查找到的结果，可直接跳转所在位置。

2.2.9 仿真

对于BonengLogiclab 的仿真，只能做一些简单的逻辑的仿真，如果在程序中，配置了其他的资源。例如Modbus, EtherCAT, 以及使用了一些博能相关的库等资源的话，会导致编译无法通过。无法使用仿真。

2.2.9.1 仿真结构

在BonengLogiclab软件快捷菜单栏中，点击  按钮，之后，软件会弹出如下显示窗口。

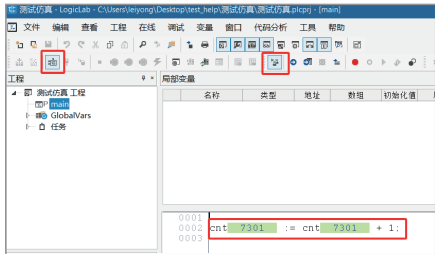


该窗口中含4个按钮，按钮的具体功能如下表所示：

图标	功能
	仿真模式下停止PLC程序运行
	仿真模式下启动PLC程序运行
	使能所有的I/O控制面板
	展开/收缩I/O仿真窗口

点击仿真按钮后，需要编译并且下载PLC程序，当右下角显示工程源代码匹配时，可开始正常使用 BonengLogiclab

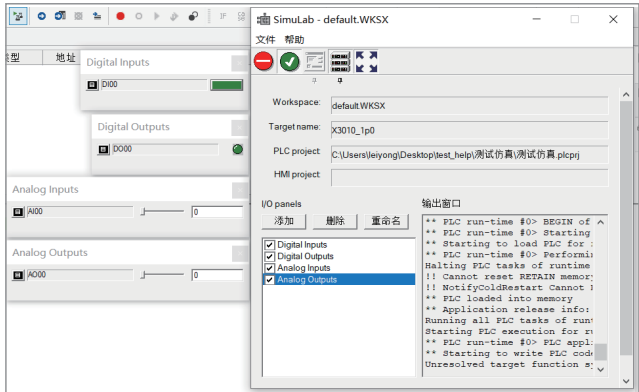
软件进行在线监控值。如下图所示：



如若遇到PLC程序编译失败，报错，需要到官网更新最新版本的软件方可使用。

2.2.9.2 扩展I/O仿真

点击展开I/O仿真窗口，并且使能I/O控制面板如下图所示：

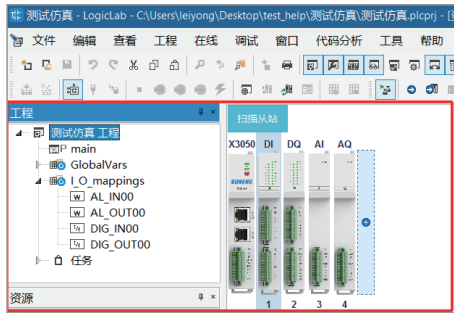


在该仿真页面中，已经预设好了4个I/O控制面板，默认四个I/O控制面板都是显示的，如果不想显示某个

控制面板，可以取消勾选。而其中的DI00, D000, AI00, A000, 只是添加的一个显示例子，在用户的工程中没有实际的意义，可以将其直接删除。在该窗口中也可以创建自己的控制面板。

• 设计测试程序

在PLC工程中需要添加四种扩展模块，并且在扩展模块中创建变量。如下图所示：



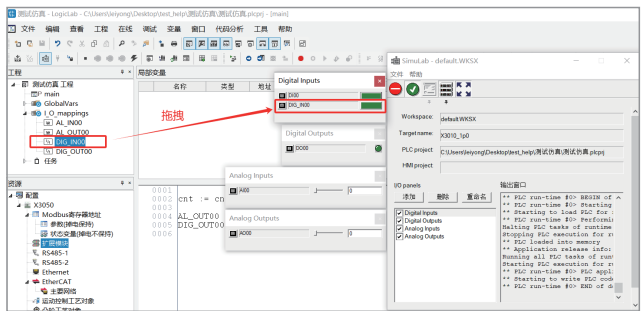
利用上述创建的扩展I/O变量在main()任务在添加如下程序：

```
0001 cnt := cnt + 1;
0002
0003
0004 AL_OUT00 := AL_IN00;
0005 DIG_OUT00 := DIG_IN00;
```

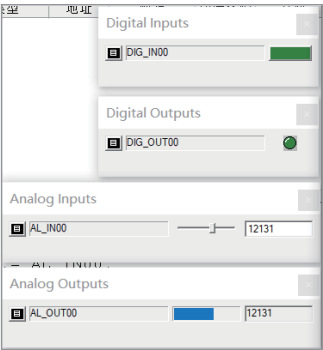
该测试程序方便后续查看实验结果。

• 添加扩展I/O变量到I/O控制面板中

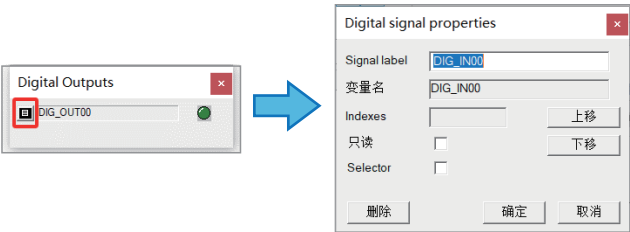
编译保存下载后，将工程中的扩展I/O变量拖拽到I/O仿真控制面板中



并且删除原来初始的自带的点位控制DI00 ,D000, AI00,A000。具体删除操作往下看，如下图所示：



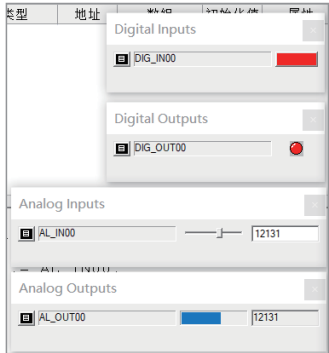
• 调整扩展IO面板设置



在该控制面板中可以设置该控制面板的该点位只读，或者删除该面板中的该点位，或者是在该面板中上下移动该点位。变量名称和标签名称保持一致即可，一般不做修改。这里需要将DIG_OUT00和AL_OUT00在点位配置属性中设值为只读。方便观察演示结果。

• 最终操作演示

当点击绿色方形状按钮时，同时输出点位的灯也会同时变为红色，当滑动模拟量输入的滑动条时，模拟量输出的大小也会随之发生改变。

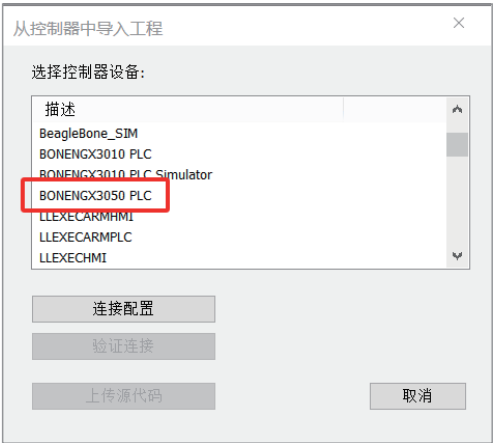


2.2.10 上传

BonengLogiclab 支持将程序上传到电脑，点击菜单栏中的文件，点击从设备中导入工程，操作如下图



点击后，找到BONENGX3050 PLC，点击选中。



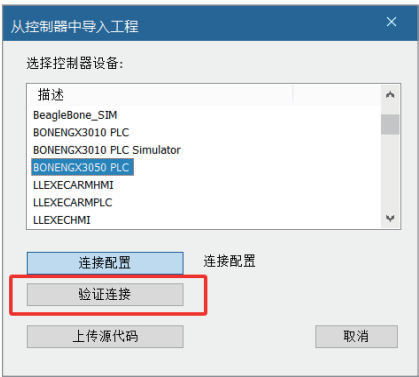
点击连接配置,弹出如下窗口。



选择GDB，点击激活，然后点击属性。



选择TCP/IP协议，输入控制器的IP地址，端口为5000，超时1000，然后依次点击确定关闭前面打开的窗口。回到下图所示的窗口。点击验证连接，则会弹出提示，验证有效。



验证有效后，点击上传源代码，弹出窗口，需要选择一个文件夹作为存放该源代码。选择好文件夹后点击确定即可，且该工程会自动关闭，并打开后面的工程。

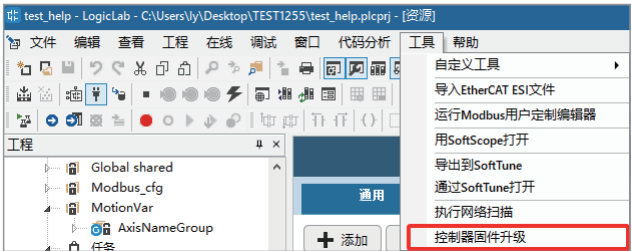
2.2.11 固件升级

该固件升级方式只有在BonengLogiclab Version: V0.1.4中才有，可以在BonengLogiclab帮助页面中

点击关于，查看当前软件版本。

2.2.11.1 打开升级工具

在使用PLC时可能会遇到功能不满足需要固件升级的情况。以X3050 固件升级为例操作步骤如下所示：



点击菜单栏中的工具，点击控制器固件升级。

2.2.11.2 进入升级页面

进入升级页面后，该页面会自动扫描出，电脑网线实际连接的控制器设备情况，如下图所示：



显示出实际连接的控制类型，以及IP地址信息。如果实际通过交换机连接有，多台控制器的情况，也会全部展示在表格中。如果中途拔插网线，需要重新关闭该页面，再次进入，方可显示最新的控制器设备信息

2.2.11.3 选择升级文件

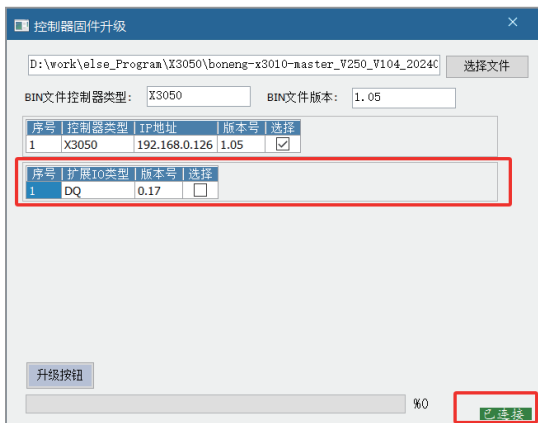
点击选择文件，会弹出文件对话框，选择由Boneng发布的控制器烧录文件，选择好烧录文件后如下图所示：



选择好后需要升级的固件后，会自动解析该BIN文件，将该bin文件的控制器类型，以及BIN文件的版本号显示到相应的对话框中。

2.2.11.4 连接控制器

选择一个控制器，勾选上后，会自动读取控制器中的版本号,如果该控制器还配有扩展IO,还会将扩展IO的组态信息也一并给显示在下方对话框中。且勾选上控制器后右下角会显示已连接。表示已连接上该控制器。



2.2.11.5 控制器本体升级

选择好控制器后，不用选择扩展IO，选择扩展IO，表示升级的对象是该控制器下的某个扩展IO。点击升级按钮即进行控制器升级。升级过程中请不要断电！否则会导致控制器永久死机。



当进度条达到100%时候表示升级完成，且会自动断开与控制器的连接。

2.2.11.6 扩展模块升级

扩展模块升级的方式，与升级控制本体类似，先选择好需要烧录的扩展模块文件，然后勾选上控制器，在弹出的扩展IO列表中选择用户需要升级的扩展IO，点击升级即可。如下图所示：



2.3 编程语言

2.3.1 编程语言类型

BonengLogicLab支持的编程语言：

- 梯形图（LD）
- 结构化文本编程（ST）

- 功能模块图 (FBD)
- 顺序功能图 (SFC)
- 指令表 (IL)

2.3.2 梯形图 (LD)

前面快速入门已经介绍过，关于如何添加（创建）一个程序，这里就不在赘述。在添加好梯形图程序的前提下，即可进行梯形图的网络中编辑。如下图所示：



其中 ? 表示该触点或者是线圈还未分配变量。

2.3.2.1 网络

在梯形图中，需要新建网络，才能在该网络中编辑 LD 程序，右击该网络序号，可以在本行网络的前，后，顶部，底部，新建网络。如下图所示：



如上图所示，点击属性可编辑该网络的标签（程序中跳转，而不是注释），以及是否禁用该网络。

网络属性

标签:

禁用 ☐

如果未定义则禁用:

确定 取消

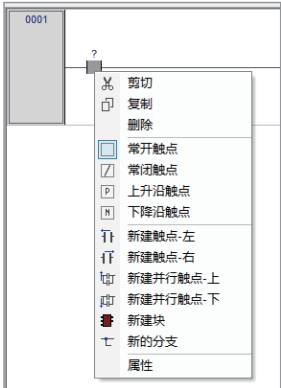
在快捷菜单栏按钮中也具有新建网络的功能，。新建完成网络后，如下图所示：



2.3.2.2 触点

2.3.2.2.1 添加触点

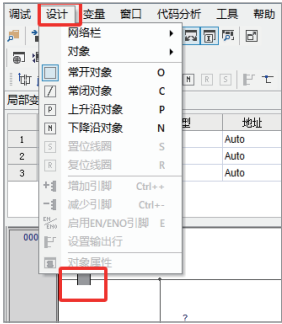
鼠标右击某个触点可以弹出右击菜单栏,如下图所示:



在这个菜单中可以看到，相关对这个触点的操作从上到下依次，包括剪切，复制，删除该触点，设置为常开触点，常闭触点，设置该触点为上升沿或者下降沿，在该触点左侧新建触点，在该触点右侧新建触点，在该触点上方新建并行触点，在该触点下方新建并行触点。在该触点右侧新建一个块，在触点右侧添加一个新的分支。以及设置该触点的属性。

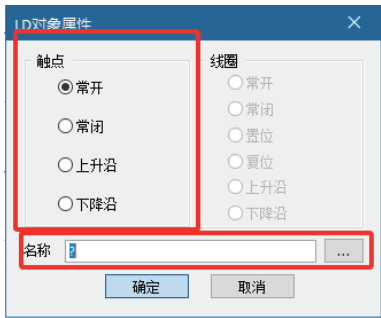
在选中某个触点后，上方和LD相关的快捷菜单按钮会高亮，表示可进行点击的操作。如下图所示：

，同样的，也可以在选择某个触点后，在上方菜单栏中，点击设计，对该触点进行操作。如下图所示：



2.3.2.2.2 触点的属性设置

双击这个触点会弹出该触点的属性设置，（或者按照上面的方法，鼠标右击选择属性），操作后如下图所示：



对于触点的类型选择有，常开，常闭，上升沿，下降沿，在这里需要给该触点输入一个名称。输入名称后，如果输入的名称在全局变量或者在该局部变量中没有找到，那么将会弹出新建变量的对话框如下图所示：



由于这是给触点添加变量，所以该类型只能是BOOL类型。

也可以在打开对象属性后，点击名称右侧的按钮进入对象浏览器，选择变量。如下图所示：



2.3.2.3 线圈

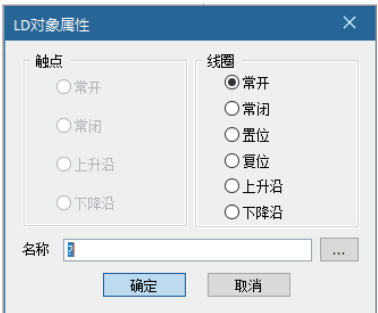
2.3.2.3.1 添加线圈

在一个网络中，鼠标右击线圈，可以剪切，复制，删除，再添加一个线圈（双线圈输出），以及属性设置，如下图所示：



2.3.2.3.2 线圈属性设置

双击线圈，可以进入线圈的属性设置，可以设置该线圈为常开，常闭，置位，复位，上升沿，下降沿。等

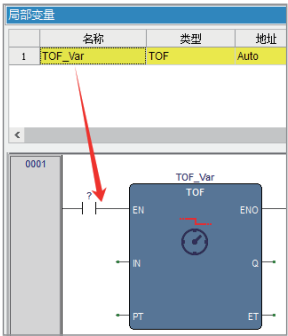


在名称设置上，与触点的名称设置方法一致。

2.3.2.4 功能块以及函数

2.3.2.4.1 添加功能块以及函数

BonengLogicLab在程序中添加功能块以及函数，非常简单。具体添加方法2种。第一种是在变量设计列表中，设计好所需要的变量，然后使用鼠标拖拽到，触点，即可在该触点的右边添加该功能块，或者函数。如下图所示



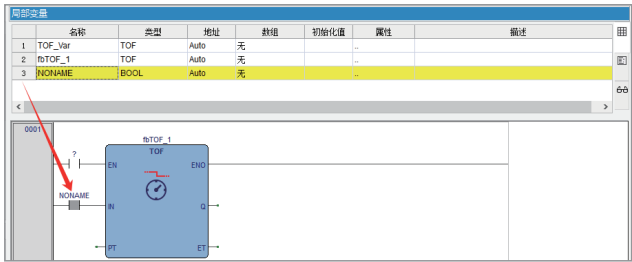
第二种方法是直接从库里面拖拽，拖拽后，软件会生成对话框，也就是功能块的实例。



如果不更改名称，直接点击确定即可。

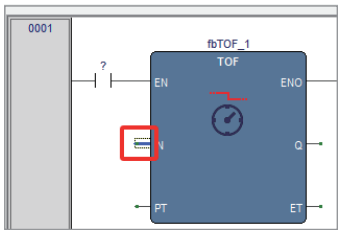
2.3.2.4.2 设置功能块管脚

在使用功能块时，需要对该功能块的管脚进行设置，也就是需要给该功能块的管脚分配变量。给管脚分配变量只需要，将变量表中的变量拖拽到该管脚上即可。如下图所示：

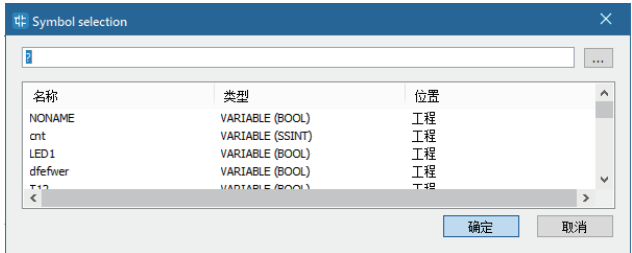


对于拖拽的变量，会自动连接到左母线。所以在这里介绍另一种管脚添加变量的方法。

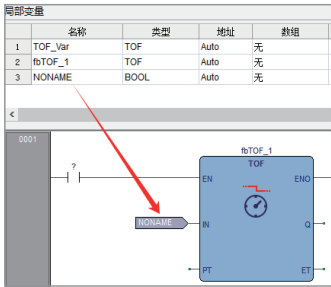
鼠标点击功能块的管脚如下图所示：



点击后会弹出对象选择窗口。



在这个窗输入名称进行筛选，筛选后双击选择需要的变量点击确定即可。

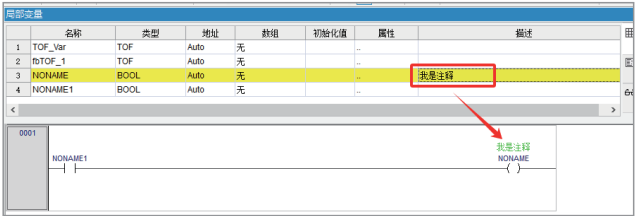


也可以在对象选择对话框中，点击右侧  按钮，进入对象浏览器直接选择变量名称。

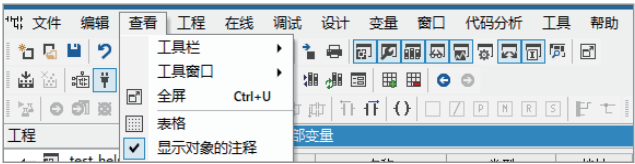
2.3.2.5 添加注释

2.3.2.5.1 变量注释

在LD当中只可以在触点，或者线圈的上方显示注释，如下图所示：

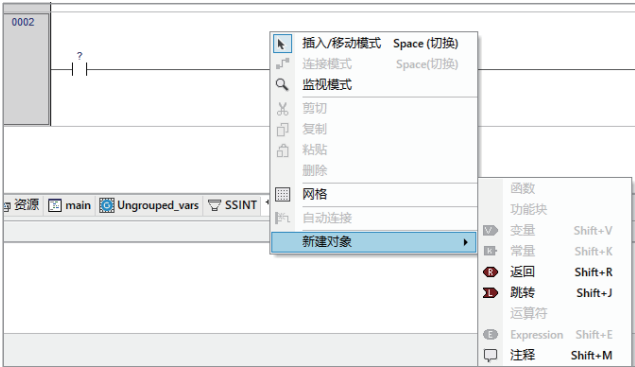


如果没有显示注释，需要去菜单栏中，点击查看，勾选显示对象的注释即可。



2.3.2.5.2 网络注释

每个网络可以拥有自己的一个注释，右击网络中空白处，新建对象，点击注释，即可为该网络添加注释。如下图所示：



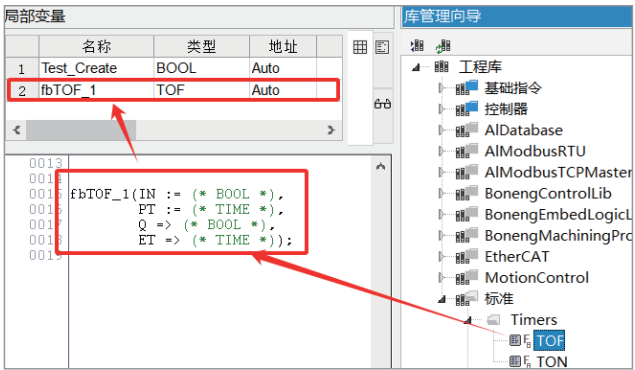
2.3.3 结构化文本编程 (ST)

2.3.3.1 编辑代码

在ST代码中添加到用户现有PLC对象的引用，则有两个选择：

- 直接输入PLC对象的名称；

- 将其拖动到一个合适的位置。可以从工作区窗口获取全局变量，可以从库窗口中拖动标准运算符和功能/功能块到目标代码位置，局部变量可以从本地变量编辑器中来选择。如下图所示：从库中直接将功能块拖拽到代码编辑区，可直接生成模板，并在局部变量内自动创建一个实例对象。



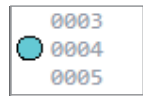
2.3.3.2 编辑功能

ST编辑器具有大多数在Windows平台上运行的编译器所共有的功能，选择文本可以对其进行复制、粘贴等操作。

2.3.3.3 书签

用户可以设置书签，以在源文件中经常访问的线条标示。设置书签标志，可以使用键盘命令移动到它。当用户不再需要，可以删除书签。

设置书签：将插入点移动到要设置书签的行，然后按 Ctrl+ F2，该行的左侧被淡蓝色圆圈标记；



跳跃到书签：反复按F2，直到达到目标行；

删除书签：将光标移动到在包含书签行上的任意位置，然后按下Ctrl+ F2

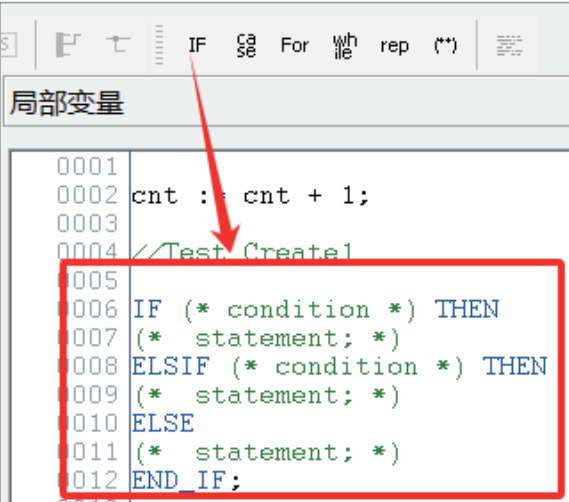
2.3.3.4 ST常用语法格式生成

在ST的编辑中，ST语法格式，如果忘记了可以通过菜单栏快捷按钮直接点击生成，在ST编辑时，如下

图所示：会高亮，表示可以点击，此时从左边到右依次是，IF，case，For while rep 注释，等



点击IF按钮直接生成如下图所示语法：



只需要补充好自己的条件内容即可。其余按钮就不再说明。

2.3.4 功能模块图 (FBD)

功能模块图的作用在这里就不再详细说明，其大多数操作方法与LD类似。

2.3.5 顺序功能图 (SFC)

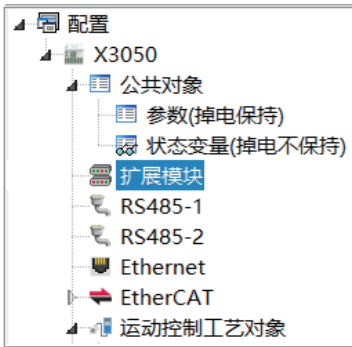
功能模块图的作用在这里就不再详细说明，其大多数操作方法与LD类似。

2.3.6 指令表 (IL)

功能模块图的作用在这里就不再详细说明，其大多数操作方法与LD类似。

2.4 扩展模块

创建一个新的工程后，在资源列表里面可以看到扩展模块，下图所示：

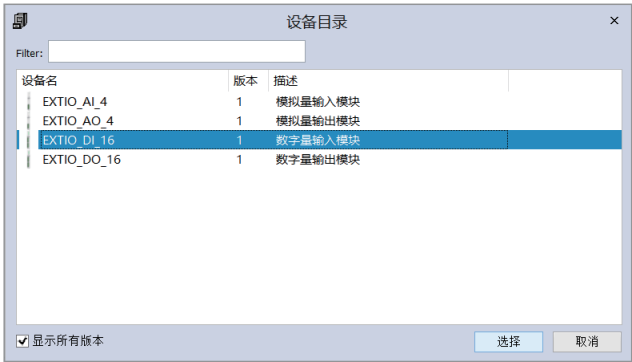


双击打开扩展模块，此时可以看到如下操作页面：

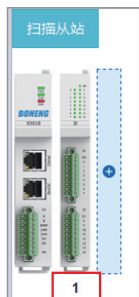


此时可以根据实际扩展模块的配置，手动或者自动添加5种扩展模块。

1、手动添加：点击上图加号，会弹出如下页面。选择一种扩展模块，该模块的选择要对应实际安装的槽号，否则需要修改槽号位置，修改扩展槽号，下面讲解。



点击选择即可添加成功，添加成功后如图下所示：

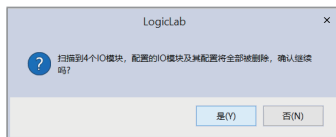


可以看到DI模块下有一个标号1，可知手动添加的DI模块在一号槽内。

2、自动添加，在下图中点击扫描从站，



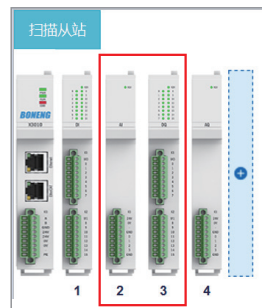
点击后，会弹出没有发现扩展RS485接口，点击确定后出现扫描到的实际配置的扩展模块数量。如下图所示：



点击是（Y），软件内从站显示界面会按照实际背板槽号的摆放顺序进行排列，与之对应。在这里添加了4个扩展模块，如下图所示：



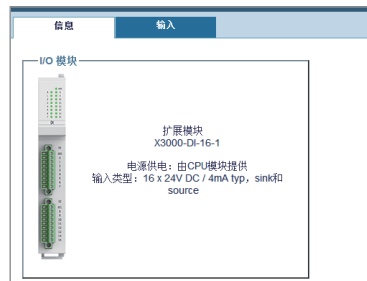
当手动添加排列顺序错乱时（与实际情况），可以看到2号槽和3号槽位置错乱如下图所示：



只需将鼠标悬停在2号槽右击选择 ，或者3号槽上右击鼠标 。即可将组态配置，与实际安装的扩展槽号对应。

2.4.1 数字量输入DI

点击1号槽的DI模块可以出现以下基本信息：



点击输入可看到如下图所示信息，该扩展模块具有16个开关量的输入，输入变量类型为BOOL。在变量列表里面可以设置其间接访问地址的变量名

信息		输入		
		分配	取消分配	
#	Data	Object Name	类型	变量
1	DI0	开关量输入 00	BOOL	
2	DI1	开关量输入 01	BOOL	
3	DI2	开关量输入 02	BOOL	
4	DI3	开关量输入 03	BOOL	
5	DI4	开关量输入 04	BOOL	
6	DI5	开关量输入 05	BOOL	
7	DI6	开关量输入 06	BOOL	
8	DI7	开关量输入 07	BOOL	
9	DI8	开关量输入 08	BOOL	
10	DI9	开关量输入 09	BOOL	
11	DI10	开关量输入 10	BOOL	
12	DI11	开关量输入 11	BOOL	

在这里，设置DI0也就是开关量输入00的变量名为

SB1设置完成后如下图所示：

#	Data	Object Name	类型	变量
1	DI0	开关量输入 00	BOOL	SB1

此时点击保存，可以看到在工程栏目中出现I_0_mappings，并在其栏目下有刚才设置的SB1变量。



点击I_0_mappings 会出现以下页面，在该页面可以看到SB1的地址是%IX1.0。（该地址是自动分配的，也可以主动修改，格式需要正确，）

1	名称	类型	地址	数组	初始化值	属性	描述
1	SB1	BOOL	%IX1.0	无			

在此可以通过，间接访问，和直接访问的办法来表示扩展端子。

1、直接访问如所示，只需要在梯形的触点中直接更改名称为地址即可，选择一个触点更改名称为%IX1.0，点击确定。

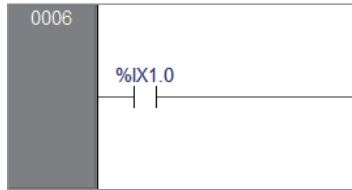


点击确定后会弹出以下对话框，在直接访问时，直接×掉即可。（点击确会报错，这是在建立变量时使用的对话框）

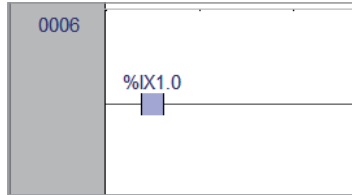


成功建立如下图所示：该地址对应的是输入变量，进入调试模式。

成功建立如下图所示：该地址对应的是输入变量，进入调试模式。



当打开，开关时该触点会导通，如下图所示：



2、间接访问是直接调用建立的变量名来表示，点击触点会出现以下对话框（LD对象属性），点击下图红框 “...”

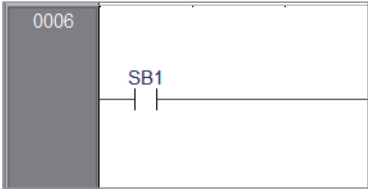


点击后会出现以下对话框。（对象浏览器）

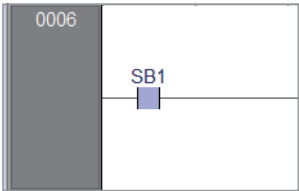


在对象浏览器中，可以看到刚才所设置的变量名。

SB1 该变量的类型是Variables。点击SB1然后点击确定。
然后会返回对象属性,再次点击确定,即可出现下图。



重新编译，下载，调试。



2.4.2 数字量输出DQ

DQ模块的使用，类似DI，在扩展模块中，选择DQ模块，点击输出按钮，设置开关量输出00的变量名称为 OUT1,点击保存，

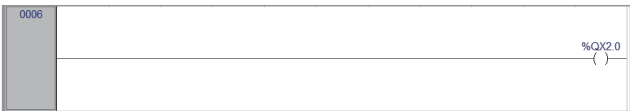
信息				
输出				
分配 取消分配				
#	Data	Object Name	类型	变量
1	DO0	开关量输出 00	BOOL	OUT1
2	DO1	开关量输出 01	BOOL	
3	DO2	开关量输出 02	BOOL	
4	DO3	开关量输出 03	BOOL	
5	DO4	开关量输出 04	BOOL	
6	DO5	开关量输出 05	BOOL	
7	DO6	开关量输出 06	BOOL	
8	DO7	开关量输出 07	BOOL	
9	DO8	开关量输出 08	BOOL	

在工程中的I_0_mappings 中可看到下图所示，可以看到OUT1的地址为%IX2.0

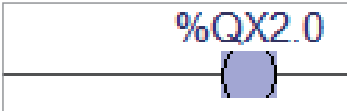
	名称	类型	地址	数组	初始化值	属性	描述
1	SB1	BOOL	%IX1.0	无			
2	OUT1	BOOL	%IX2.0	无			

使用方法也是分为直接调用和间接调用

1、直接调用如下图所示，具体调用方法参照DI模块的使用。



打开调试模式，得到如下图所示



2、间接调用如下图所示，其调用方法参见DI的调用方法



2.4.3 模拟量输入AI

点击模拟量输入模块，然后点击输入设置模拟量输入00的变量为AI_In1,可以看到模拟量输入的数据类型为WORD类型。点击保存，

模拟量输入模块				
信息 输入				
分配 取消分配				
#	Data	Object Name	类型	变量
1	AI0	模拟量输入 00	WORD	AI_In1
2	AI1	模拟量输入 01	WORD	
3	AI2	模拟量输入 02	WORD	
4	AI3	模拟量输入 03	WORD	

点击保存后，在工程中的I_0_mappings 中可看到下图所示，可以看到AI_In1的地址为%IW3.0

	名称	类型	地址	数组	初始化值	属性	描述
1	SB1	BOOL	%IX1.0	无			
2	OUT1	BOOL	%IX2.0	无			
3	AI_In1	WORD	%IW3.0	无			

将该变量用于程序中，当该扩展模块的对应通道有电压，或者电流输入时，该值会发生变化。

电压和电流模式的切换，需要使用博能的库函数实现，具体实现方法看3.2.4.1.1。若不设置是电压检测还是电流检测，则会默认电压检测。

电压输入的电压范围为0-10v 对应数值为0-10000；
电流输入的电流范围为0-20ma 对应数值为0-20000；

2.4.4 模拟量输出AQ

AQ扩展模块的使用类似AI扩展，创建变量，AQ_Out1, 点击保存

模拟量输出模块				
信息		输出		
分配		取消分配		
#	Data	Object Name	类型	变量
1	AO0	模拟量输出 00	WORD	AQ_Out1
2	AO1	模拟量输出 01	WORD	
3	AO2	模拟量输出 02	WORD	
4	AO3	模拟量输出 03	WORD	

点击保存后，在工程中的I_0_mappings 中可看到下图所示，可以看到AQ_Out1的地址为%QW4.0

	名称	类型	地址	数组	初始化值	属性	描述
1	SB1	BOOL	%X1.0	无			
2	OUT1	BOOL	%QX2.0	无			
3	AJ_In1	WORD	%IW3.0	无			
4	AQ_Out1	WORD	%QW4.0	无			

将变量AQ_Out1的值放在程序中改变其数值即可改变其对应通道的输出电压。
AQ输出的电压为0-10v对应的数值为0-10000；
注：AQ模块只有电输出，没有电流模式。

2.4.5 通讯模块CI

CI模块没有实际的软件组态信息，值得注意的是在一个PLC设备中只能挂一个CI模块，且必须紧靠CPU模块本体。X3000系列支持Modbus - RTU双主站，其中一个主站就是这个CI 模块。具体的配置信息见2.5章节串口通信。

2.5 串口通信

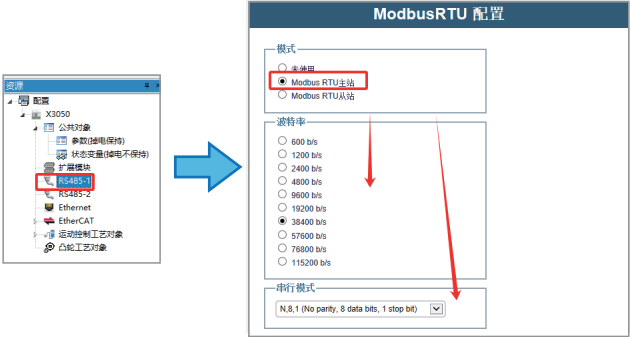
2.5.1 概述

BonnegLogicLab支持双主站通讯，其中一个主站在CPU模块上，另一个主站在扩展模块上。但是两个主站的配置方法以及使用步骤都是相同的。在资源中，进行配置。

2.5.2 Modbus-RTU主站配置

2.5.2.1 打开Modbus-RTU主站

在资源中点击RS485-1在打开的窗口页面中，选择Modbus RTU主站，会自动弹出波特率，和串行模式两个配置页面，如下图所示：

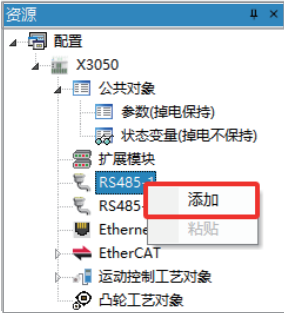


在上面的配置页面中，可以选择波特率，以及串行模式，其中串行模式选择说明如下表所示：

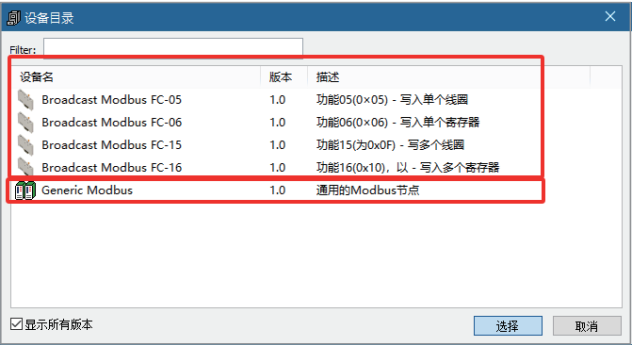
模式选择	说明
N, 8, 1 (No parity, 8 data bits, 1 stop bit)	无校验，8位数据位，1位停止位
E, 8, 1 (Even parity, 8 data bits, 1 stop bit)	偶校验，8位数据位，1位停止位
O, 8, 1 (Odd parity, 8 data bits, 1 stop bit)	奇校验，8位数据位，1位停止位
N, 8, 2 (No parity, 8 data bits, 2 stop bits)	无校验，8位数据位，2位停止位
E, 8, 2 (Even parity, 8 data bits, 2 stop bits)	偶校验，8位数据位，2位停止位
O, 8, 2 (Odd parity, 8 data bits, 2 stop bits)	奇校验，8位数据位，2位停止位

2.5.2.2 添加通讯节点

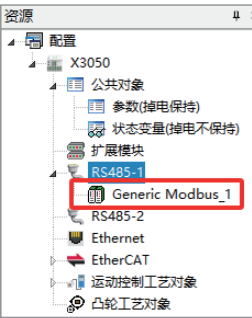
打开主站后右击RS485-1点击添加如下图所示：



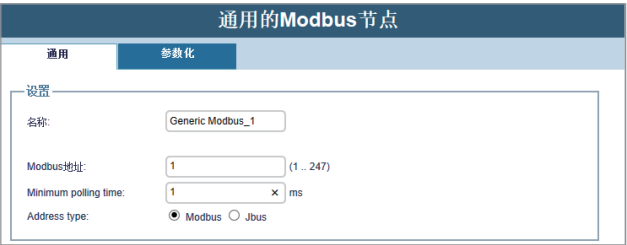
点击后会弹出设备目录窗口，如下图所示：



在设备目录窗口中选择通用的Modbus节点，（其余设备是广播功能），点击选择，在RS485-1的下方会出现这个通用节点。如下图所示：



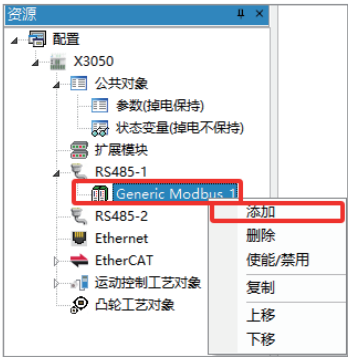
鼠标点击该节点可对节点进行配置，如下图所示：



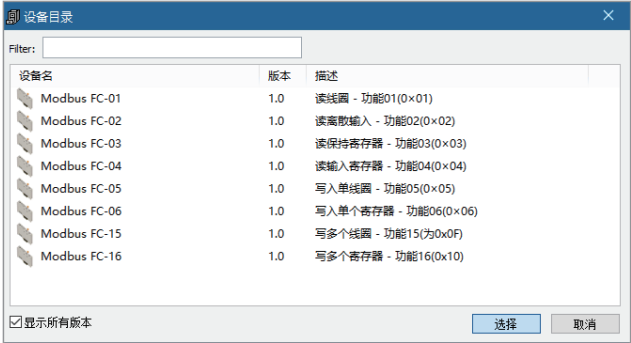
可更改该节点的名称，从站的地址（1-247），最小轮询时间等。修改最小轮询时间可以约束在该节点下的FC03，FC06等功能节点的轮询时间。

2. 5. 2. 3 添加功能节点

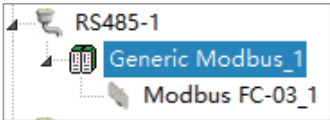
右击创建好的通用节点，点击添加如下图所示：



在弹出的设备目录窗口中，可选择许多读写操作。如下图所示：



这里以添加读保持寄存器 - 功能FC-03为例，选择FC-03，点击选择。添加好之后在通用节点下会出现该功能节点。如下图所示：



点击Modbus FC-03功能节点，在通用选项卡中，可配置寄存器地址，轮询时间，超时时间等。如下图所示：其中起始需要地址从1开始。



在上图中点击保持寄存器选项卡，可以添加，删除，一行变量列表。可以分配和取消分配一个变量。

Modbus FC 03(0x03)- 读保持寄存器

通用

保持寄存器

+ 添加

← 删除

⚙ 分配

⌛ 取消分配

#	名称	ObjType	地址	变量
1	Register	WORD	1	

该页面的主要作用是设置一个变量，在上述FC-03这个功能节点的作用下，将会去访问这个功能配置的寄存器地址，将访问到的数据赋值到该变量中。该变量可直接在变量框中直接输入如下图所示：输入名称Read_Data, 将会弹出新建PLC变量的对话框，选择该PLC变量的类型，点击确定后即可。

#	名称	ObjType	地址	变量
1	Register	WORD	1	Read_Data

新建PLC变量

名称: Read_Data

类型: WORD

确定 取消

在此之后，在工程窗口中，会出现一个Modbus_Vars 的全局变量组，在该组中，会出现上述创建的PLC的变量。如下图所示：

名称	类型	地址	数据	初始化值	属性
Read_Data	DWORD	%MD100.0	无		

2.5.2.4 注意事项

在作为Modbus-RTU主站时，设置需要访问的寄存器地址时，一般需要将该地址加1。

2.5.3 Modbus-RTU从站配置

2.5.3.1 打开Modbus RTU从站

在配置Modbus-RTU从站时，需要在资源中点击RS485-1中，在配置页面中选择Modbus RTU从站，配置

从站波特率，串行模式，以及从站站号，如下图所示：

工程

Enum_Name

ge4e

main

New_Type

SSINT

Test_ID

TEST_NEW_FC

Type_Name

GlobalVars

I_O_mappings

Modbus_Vars

Ungrouped_vars

Global shared

资源

配置

X3050

公共对象

参数(掉电保持)

状态变量(掉电不保持)

扩展模块

RS485-1

RS485-2

Ethernet

EtherCAT

运动控制工艺对象

New_Axis_0

New_Axis_1

凸轮工艺对象

ModbusRTU 配置

模式

未使用

Modbus RTU主站

Modbus RTU从站

波特率

600 b/s

1200 b/s

2400 b/s

4800 b/s

9600 b/s

19200 b/s

38400 b/s

57600 b/s

76800 b/s

115200 b/s

串行模式

E,8,1 (Even parity, 8 data bits, 1 stop bit)

从站设置

Modbus地址(1.247): 1

2.5.3.2 创建寄存器地址

创建好Modbus RTU节点后，需要设计一些寄存器地址的变量，让主站进行访问。寄存器地址的变量设计需要在资源栏中的参数（掉电保持）中设计。详细设计方法见 2.2.3和 2.2.4

2.5.4 Modbus 故障码说明

故障码可于库管理向导中打开枚举变量对照

AlModbusRTU

Definitions

Structures

Enumeratives

MB_RESULT_VALUE

MB_RTU_NETWORK_ID

MbBaudrate

MbChn

MbError

MbException

MbFunction

MbMode

MbParam

MbRqErr

MbRSMode

MbStatus

也可参照此处

Name	Value	Description	中文描述
MbRqErr_Ok	0	No error occurred	未发生错误
MbRqErr_IllegalFunction	1	Invalid request message	无效请求消息
MbRqErr_IllegalDataAddress	2	Illegal data address	非法数据地址
MbRqErr_IllegalDataValue	3	Illegal data value	非法数据值
MbRqErr_SlaveDeviceFailure	4	Slave device failure	从站设备故障
MbRqErr_Acknowledge	5	Acknowledge error	确认错误
MbRqErr_SlaveDeviceBusy	6	Slave device busy	从站设备繁忙
MbRqErr_ParityError	7	Memory parity error	内存奇偶校验错误
MbRqErr_CommChn	8	Communication channel configuration error	通信信道配置错误
MbRqErr_InvalidSlaveAddress	9	Invalid slave address (address range 0 - 247)	无效从站地址 (地址范围 0-247)
MbRqErr_GatewayPathUnavailable	10	Gateway path unavailable	网关路径不可用
MbRqErr_GatewayDeviceFailed	11	Gateway target device failed to respond	网关目标设备无法响应
MbRqErr_MsgMaxElements	12	Invalid number of element for this function	此函数的元素数无效
MbRqErr_MsgRequestInvalid	13	Cannot correctly build the request message	无法正确构建请求消息

Name	Value	Description	中文描述
MbRqErr_MsgResponseInvalid	14	Invalid response message	无效响应消息
MbRqErr_MsgSize	15	Response size error	响应大小错误
MbRqErr_Timeout	16	Timeout reached	超时到达
MbRqErr_TaskInvalid	17	Function invoked by an invalid task	无效任务调用的函数

2.6 以太网通信

2.6.1 概述

X3000系列的PLC以太网通讯部分主要是集成了Modbus-TCP协议，包括服务器与客户端。可轻松实现与支持Modbus-TCP的设备进行通讯与数据交互。

2.6.2 Modbus-TCP主机配置

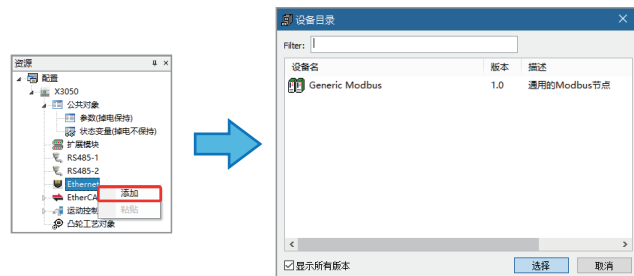
2.6.2.1 开启Modbus-TCP主站

在资源窗口中找到Ethernet, 点击Ethernet在以太网配置中勾选Modbus TCP 主站。如下图所示：



2.6.2.2 添加Modbus-TCP通用节点

勾选Modbus TCP主站后，可以添加鼠标右击Ethernet添加通用节点，在设备目录中选择通用的Modbus节点。如下图所示：



点击刚才添加好的通用Modbus TCP节点。配置该节点如下图所示：



2.6.2.3 添加功能节点

Modbus-TCP通用节点下，添加功能节点和Modbus-RTU一样，这里就不再赘述，详情见2.5.2.3

2.6.3 Modbus-TCP从机配置

2.6.3.1 打开Modbus TCP从站

Modbus TCP从站一直是默认开启的，如下图所示：只需要配置寄存器地址即可。



从站的IP地址是PLC的IP地址，端口号为502

2.6.3.2 创建寄存器地址

Modbus-TCP和Modbus-RTU从站的寄存器地址是共用的，创建寄存器变量地址见2.2.3 和2.2.4

2.6.4 Modbus故障码说明

请参考2.5.4Modbus故障码说明

2.7 EtherCAT通信

2.7.1 概述

EtherCAT是一种基于以太网的开放式工业现场技术，具有通信刷新周期短、同步抖动小、硬件成本低等特点。如需了解EtherCAT原理和相关技术，请参考书籍《工业以太网现场总线EtherCAT驱动程序

设计和应用》，或者登录EtherCAT技术委员会官网，网址：<https://www.EtherCAT.org.cn>。

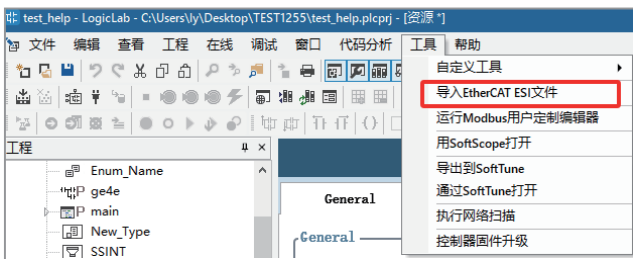
X3000系列支持标准的支持标准EtherCAT接口（1路RJ45接口），最多支持32个从站采用线性拓扑结构，EtherCAT总线周期最小可以设置为1ms。

2.7.2 EtherCAT主站配置

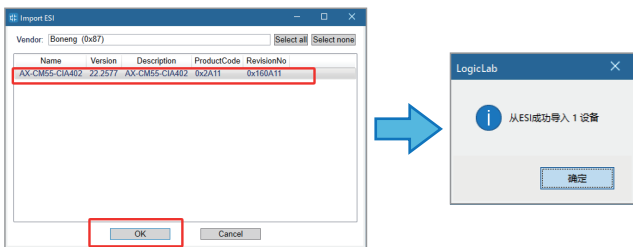
2.7.2.1 导入设备XML

在进行EtherCAT通信之前，需要先导入从站ESI文件，主站才能识别从站信息。这个操作只需要导入一次，从站信息会保存在Logiclab中，这样每次扫描从站就能识别从站信息。

选择Logiclab 工具 > 导入EtherCAT ESI 文件。



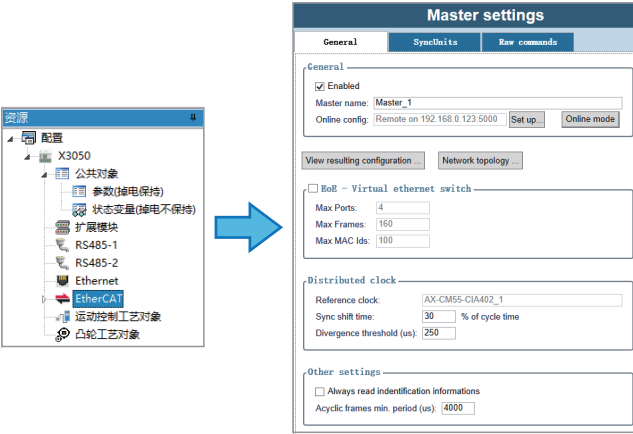
然后在打开的文件窗口中选择XML文件位置，选择文件中保存的设备即可。如下图所示：



在添加完成XML设备后，有所有的新工程都可以使用，不需要重复添加。

2.7.2.2 基本设置

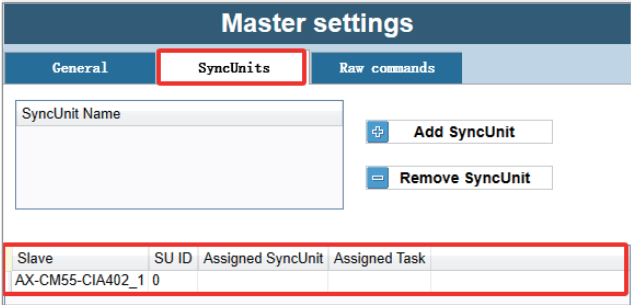
点击刚才添加好的通用Modbus TCP节点。配置该节点如下图所示：



名称	功能
Enabled	配置使能主站，
Master name	设置主站名称
Online config	PLC的IP地址，一般不用修改默认即可
Online mode	在线模式的按钮，当logiclab软件连接到控制器后，即可点击次按钮进入在线功能
View resulting configuration	浏览配置结果，包含从站相关信息，以及启动配置命令等。
Network topology	图形化显示组态网络连接
EoE- Virtual ethernet switch	虚拟网络转化相关配置，一般不涉及
Distributed clock	设置分布式时钟，默认值即可
Other settings	包含同步数据帧最小周期，以及是否一直读取身份信息。

2.7.2.3 同步单元

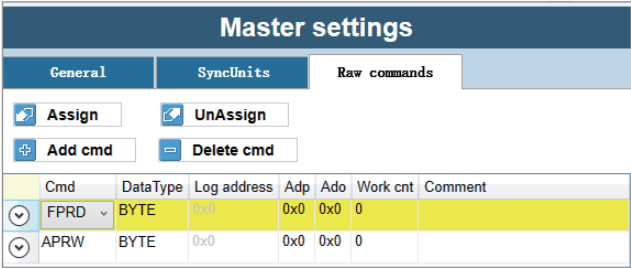
为了避免部分从站异常干扰其他设备,或者区分故障源时,可以采用同步单元数据分组功能。如下图所示



在同步单元中，可以创建一个新的同步单元，删除一个同步单元，可以将从站分配到某个同步单元中。如过没有特别的需求,可以不用管，默认即可。

2.7.2.4 自定义寻址方式

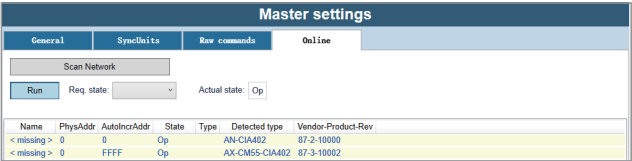
通过关联PLC变量的自定义寻址方式可以读取从站的相关寄存器或者内存。如下图所示：



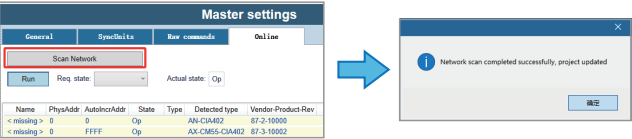
在创建寻址方式时，是通过关联PLC变量实现的，所以所读写的内容也是基于PLC变量的变化而产生。所以这里需要输入PLC变量名称，保存后在工程中则会多出一个全局变量组，EtherCAT_mappings。这里需要了解命令的功能以及报文的格式才能很好的使用这个功能。详细信息见EtherCAT寻址方式解析。如果没有特定的需求则不用关心该功能。

2.7.2.5 扫描添加EtherCAT从站设备 (Online)

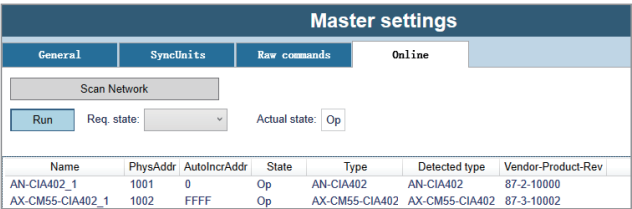
先连接好PLC和电脑的网线，和EtherCAT的网线后，在EtherCAT主站基本设置页面，点击Online按钮，可以看到会对出一个Online的选项卡。如下图所示：



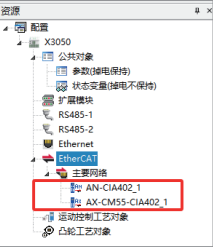
在该方框中会预扫描出连接好的从站设备（显示出设备名称，以及设备ID,表示已经导入过这个XML设备），但是该设备还没有被配置到工程中，所以需要点击Scan Network扫描。



扫描成功后如下图所示，字体由原来的蓝色，变成了黑色，表示扫描成功。



且在EtherCAT中的主要网络下可发现多出了两个设备节点。

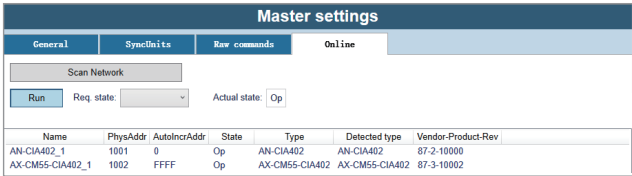


值得注意的是，如果BonengLogiclab软件中以前没有添加这个设备的XML文件的话，是无法正常的扫描到该设备的。当主站已经运行起来时，如若中途增加，或者减少从站设备，需要将主站停止，才能重新扫描到增加的设备信息。

2.7.2.6 在线监控

在主站的在线监控中，可以知己查看主站以及，从

站的状态以及从站的部分相关信息。

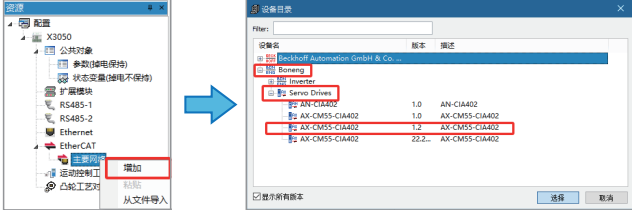


Actual state: 表示的是主站的状态
点击Run 可以开始或停止主站。

2.7.3 EtherCAT从站配置

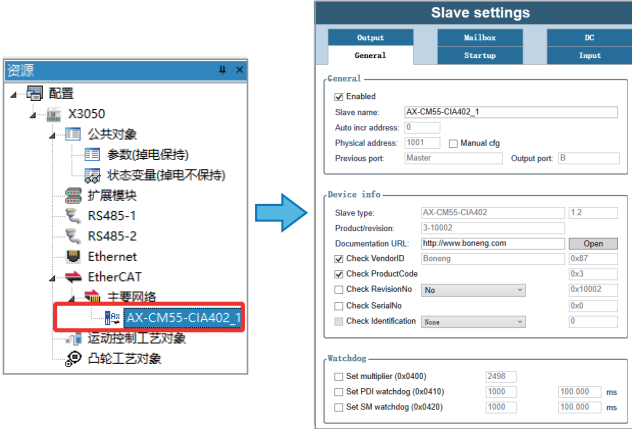
2.7.3.1 手动添加EtherCAT从站

鼠标右击主要网络，点击增加，弹出设备目录对话框，找到Boneng并展开，然后找到Servo Drives并展开，选择一个伺服驱动器设备。然后点击选择即可。如下图所示：



2.7.3.2 常规设置

点击添加好的EtherCAT从站设备，会出现从站配置页面，如下图所示：

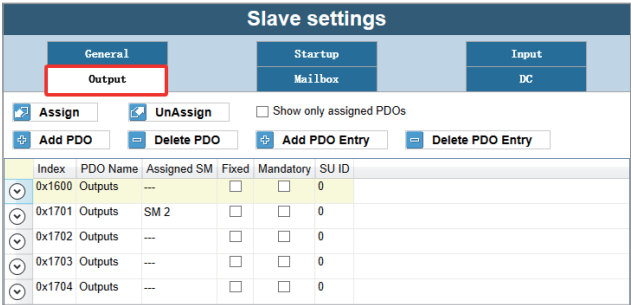


名称	描述								
Enabled	从站使能选项								
Slave name	从站名称								
Auto incr address	自动增加的地址（没有特殊要求不用更改）								
Physical address	物理地址（没有特殊要求不用更改）								
Previous port	上一个端口								
Output port	输出端口								
Device info	设备信息容器，在该设备信息中可看到从站的类型，XML版本号，产品版本号，以及产品供应商ID等如无其他特殊需求，此方框中的内容一般不需要更改。								
Watch dog（容器）	看门狗设置方框，一般不需要设置。								
Timeouts for state machine (ms)（容器）	状态机跳转超时时间，一般来说默认值即可，不用做修改。其中简写如下所示： <table><tr><td>I->P</td><td>Init到PreOP</td></tr><tr><td>P->S, S->O</td><td>PreOP到SafeOP, SafeOP到OP</td></tr><tr><td>Back to P, I</td><td>回到Init或PreOP</td></tr><tr><td>O->S</td><td>Op到SafeOP</td></tr></table>	I->P	Init到PreOP	P->S, S->O	PreOP到SafeOP, SafeOP到OP	Back to P, I	回到Init或PreOP	O->S	Op到SafeOP
I->P	Init到PreOP								
P->S, S->O	PreOP到SafeOP, SafeOP到OP								
Back to P, I	回到Init或PreOP								
O->S	Op到SafeOP								
Use LRD/LWR instead of LRW	使用LRD/LWR 代替LRW，一般不需要更改								

值得注意的是。X3000系列控制器在使用从站时，一般不直接使用地址进行管理从站，而是使用索引进行管理。规定直接与控制相连接的是0号索引，线性连接时索引依次加1。

2. 7. 3. 3 过程数据

Output选项卡和Input选项卡是在从站配置工程中经常需要配置的两部分内容。在Slave settings(从站设置)页面点击Output选项卡如下图所示：



上图中表格中的列标题解释如下表格所示：

名称	描述
Index	PDO 的索引
PDO Name	PDO的名称
Assigned SM	选择的PDO组，不同设备选择PDO时不一样，有的设备支持选多组PDO，有的设备只允许选择一组PDO（如博能的伺服设备 AX-CM55-CIA402），所以在选择PDO组时，只能选一组时，需要先取消之前选择的组，然后再重新选择。
Fixed	固定PDO组，固定的PDO组不允许修改该组中的内容，若没有固定，那么允许用户在该组下添加对象目录，删除对象目录，映射PLC变量等操作。
Mandatory	是否是强制性的
SU ID	同步单元ID，可以在主站中的“SyncUnit”选项卡中分配

2. 7. 3. 3. 1 选择PDO组

在不同的从站设备中，PDO的组别，和数量都是不一

样的, 通常根据不同的使用情况需要选择不同的PDO进行配置如下图所示: SM2所在的组别表示选中了该组

Slave settings

General

Startup

Input

Output

Mailbox

DC

Assign

UnAssign

Show only assigned PDOs

Add PDO

Delete PDO

Add PDO Entry

Delete PDO Entry

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x1600	Outputs	---	☐	☐	0
0x1701	Outputs	SM 2	☐	☐	0
0x1702	Outputs	---	☐	☐	0
0x1703	Outputs	---	☐	☐	0
0x1704	Outputs	---	☐	☐	0

2.7.3.3.2 添加PDO组

在Output选项卡中找到 Add PDO, 即可添加PDO组, 如下图所示:

Slave settings

General

Startup

Input

Output

Mailbox

DC

Assign

UnAssign

Show only assigned PDOs

Add PDO

Delete PDO

Add PDO Entry

Delete PDO Entry

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x0	---	☐	☐	☐	0
0x1600	Outputs	---	☐	☐	0
0x1701	Outputs	SM 2	☐	☐	0
0x1702	Outputs	---	☐	☐	0
0x1703	Outputs	---	☐	☐	0
0x1704	Outputs	---	☐	☐	0

2.7.3.3.3 删除PDO组

选择一个PDO组后 (光标所在行), 点击Delete PDO 即可删除该行, 也就是删除该组PDO。如下图所示:

Slave settings

General

Startup

Input

Output

Mailbox

DC

Assign

UnAssign

Show only assigned PDOs

Add PDO

Delete PDO

Add PDO Entry

Delete PDO Entry

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x0	---	☐	☐	☐	0
0x1600	Outputs	---	☐	☐	0
0x1701	Outputs	SM 2	☐	☐	0
0x1702	Outputs	---	☐	☐	0
0x1703	Outputs	---	☐	☐	0
0x1704	Outputs	---	☐	☐	0

2.7.3.3.4 PDO组内添加对象 (对象字典)

在OutPut选项卡中, 选择一个没有被固定的PDO对象组, 点击Add PDO Entry

Slave settings

General

Startup

Input

Output

Mailbox

DC

Assign

UnAssign

Show only assigned PDOs

Add PDO

Delete PDO

Add PDO Entry

Delete PDO Entry

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x1600	Outputs	---	☐	☐	0
0x1701	Outputs	SM 2	☐	☐	0
0x1702	Outputs	---	☐	☐	0
0x1703	Outputs	---	☐	☐	0
0x1704	Outputs	---	☐	☐	0
0x0	---	☐	☐	☐	0

点击后出现如下图所示对话框。

Add object

Empty object

From Dictionary

Filter:

Index	Subidx	Name	Type	Bit len	Default
0x4000	1	RPZD - R-PZD1	UINT	16	
0x4000	2	RPZD - R-PZD2	UINT	16	
0x4000	3	RPZD - R-PZD3	UINT	16	
0x4000	4	RPZD - R-PZD4	UINT	16	
0x4000	5	RPZD - R-PZD5	UINT	16	
0x4000	6	RPZD - R-PZD6	UINT	16	
0x4000	7	RPZD - R-PZD7	UINT	16	
0x4000	8	RPZD - R-PZD8	UINT	16	
0x4000	9	RPZD - R-PZD9	UINT	16	
0x4000	10	RPZD - R-PZD10	UINT	16	
0x4000	11	RPZD - R-PZD11	UINT	16	
0x4000	12	RPZD - R-PZD12	UINT	16	
0x4000	13	RPZD - R-PZD13	UINT	16	
0x4000	14	RPZD - R-PZD14	UINT	16	

OK

Cancel

在该对话框中, 一板选择来自对象字典 (From Dictionary)。在Filter 中可以输入名称或者索引相关信息, 可以筛选相关结果。

例如: 需要添加原点偏移的对象, 输入607, 即可筛选出607相关的所有的对象。如下图所示:

Add object

Empty object

From Dictionary

Filter: 607

Index	Subidx	Name	Type	Bit len	Default
0x6071		Target torque	INT	16	0 (0x0)
0x6072		Max Torque	UINT	16	3500 (i)
0x607A		Target position	DINT	32	0 (0x0)
0x607C		Home offset	DINT	32	0 (0x0)
0x607D	1	Software position limit - Min position limit	DINT	32	-21474
0x607D	2	Software position limit - Max position limit	DINT	32	21474i
0x607E		Polarity	USIN	8	0 (0x0)
0x607F		Max profile velocity	UDIN	32	10485i

OK

Cancel

选中原点偏移0x607c，点击OK即可。

展开刚才选中的PDO组，可看到下方出现了新添加的
原点偏移对象，如下图所示：

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x1600	Outputs	---	☐	☐	0
Index	Subidx	PDO Entry Name	Type	Bit len	PLC var
0x6040	0	Controlword	UINT	16	
0x607A	0	Target position	DINT	32	
0x607C	0	Home offset	DINT	32	

2.7.3.3.5 PDO组内删除对象（对象字典）

选择一个PDO组，点击左侧展开按钮，即可展开该
PDO组，选择一个对象，点击上方Delete PDO
Entry，即可删除该对象，如下图所示。

Slave settings						
General		Startup		Input		
Output		Mailbox		DC		
Assign		UnAssign		☐ Show only assigned PDOs		
Add PDO		Delete PDO		Add PDO Entry		Delete PDO Entry
Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID	
0x1600	Outputs	---	☐	☐	0	
Index	Subidx	PDO Entry Name	Type	Bit len	PLC var	Comment
0x6040	0	Controlword	UINT	16		
0x607A	0	Target position	DINT	32		
0x607C	0	Home offset	DINT	32		

点击后会询问是否需要删除该对象，点击是即可删
除该对象，如下图所示：

错误

?

Delete PDO entry 607C:0 'Home offset' ?

确定取消

2.7.3.3.6 映射PLC变量到组内对象

在PLC程序中可以直接通过PLC变量的形式去访问PDO
组内的某个对象。

在展开PDO组后可看到组内对象表，在组内对象表
中，PLC var 所在的列是映射PLC变量。

Slave settings						
General		Startup		Input		
Output		Mailbox		DC		
Assign		UnAssign		☐ Show only assigned PDOs		
Add PDO		Delete PDO		Add PDO Entry		Delete PDO Entry
Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID	
0x1600	Outputs	---	☐	☐	0	
Index	Subidx	PDO Entry Name	Type	Bit len	PLC var	Comment
0x607A	0	Target position	DINT	32		
0x6040	0	Controlword	UINT	16		

在PLC var 中可直接输入用户想生成的PLC变量的名
字，在输入名称后会选择该PLC变量的类型，一般类
型应该和该对象类型保持一致，在Comment 中可输
相关注释。如下图所示：

	PDO Name	Assigned SM	Fixed	Mandatory	SU ID		
	Outputs	---	☐	☐	0		
A	Index	Subidx	PDO Entry Name	Type	Bit len	PLC var	Comment
	0x607A	0	Target position	DINT	32		
	0x6040	0	Controlword	UINT	16	Control_word	控制字

完成输入后，在EtherCAT_mappings全局变量组中会
出现该PLC变量。

如果之前已经定义好了PLC变量，那么可以使用分配
按钮（Assign），将选择的对象分配一个PLC变量。
同理也可以使用UnAssign 取消分配。

2.7.3.4 在线监控（Online）

在主站页面开启了Online模式后，在从站页面的选
项卡，就会多出来一个Online的选项卡。点击
Online的选项卡如下图所示：

Slave settings

General

Startup

Input

Output

Mailbox

DC

Online

ESC Registers access

EEPROM access

COE SDO access

Diagnostics

Req. state: Actual state:

Op

Direction	Index	Subidx	Name	Type	Bit len	Forced	Value	PLC variable
Input	0x603F	0	Errorcode	UINT	16	☐	0 (0x0)	
Input	0x6041	0	Statusword	UINT	16	☐	0 (0x0)	
Input	0x6061	0	Modes of operation display	SINT	8	☐	0 (0x0)	
Input	0x6064	0	Position actual value	DINT	32	☐	-168139581 (0xF5FAE)	
Input	0x606C	0	Velocity actual value	DINT	32	☐	0 (0x0)	
Input	0x6077	0	Torque actual value	INT	16	☐	0 (0x0)	
Input	0x60B9	0	Touch probe status	UINT	16	☐	0 (0x0)	
Input	0x60BA	0	Touch probe pos1 pos value	DINT	32	☐	0 (0x0)	
Input	0x60BC	0	Touch probe pos2 pos value	DINT	32	☐	0 (0x0)	
Input	0x60FD	0	Digital inputs	UDINT	32	☐	0 (0x0)	
Output	0x6040	0	Controlword	UINT	16	☐	0 (0x0)	
Output	0x6060	0	Modes of operation	SINT	8	☐	0 (0x0)	
Output	0x6071	0	Target torque	INT	16	☐	0 (0x0)	
Output	0x607A	0	Target position	DINT	32	☐	-168139491 (0xF5FAE)	
Output	0x60FF	0	Target velocity	DINT	32	☐	0 (0x0)	
Output	0x6081	0	Profile velocity	UDINT	32	☐	0 (0x0)	
Output	0x60B8	0	Touch probe function	UINT	16	☐	0 (0x0)	

上述页面中出现四个按钮，具体功能如下表所示：

按钮名称	说明
ESC Registers access ...	ESC寄存器访问
EEPROM access ...	EEPROM 访问
COE SDO access ...	COE SDO 访问
Diagnostics ...	诊断信息查看

在下表中可以双击，或者右击选择，去修改某个对象的值。

Direction	Index	SubIdx	Name	Type	Bit len	Forced	Value	PLC variable
Input	0x603F	0	Errorcode	UINT	16	☐	0 (0x0)	
Input	0x6041	0	Statusword	UINT	16	☐	0 (0x0)	
Input	0x6061	0	Modes of operation display	SINT	8	☒	0 (0x0)	
Input	0x6064	0	Position actual value	DINT	32	☐	-168139585 (0xF5FA64BF)	
Input	0x606C	0	Velocity actual value	DINT	32	☐	0 (0x0)	
Input	0x6077	0	Torque actual value	INT	16	☐	0 (0x0)	
Input	0x60B9	0	Touch probe status	UINT	16	☐	0 (0x0)	
Input	0x60BA	0	Touch probe pos1 pos value	DINT	32	☐	0 (0x0)	
Input	0x60BC	0	Touch probe pos2 pos value	DINT	32	☐	0 (0x0)	
Input	0x60FD	0	Digital inputs	UDINT	32	☐	0 (0x0)	
Output	0x6040	0	Controlword	UINT	16	☐	0 (0x0)	
Output	0x6060	0	Modes of operation	SINT	8	☐	0 (0x0)	
Output	0x6071	0	Target torque	INT	16	☐	0 (0x0)	
Output	0x607A	0	Target position	DINT	32	☐	-168139491 (0xF5FA651D)	
Output	0x60FF	0	Target velocity	DINT	32	☐	0 (0x0)	
Output	0x6081	0	Profile velocity	UDINT	32	☐	0 (0x0)	
Output	0x60B8	0	Touch probe function	UINT	16	☐	0 (0x0)	

双击某个对象后会出现以下窗口

Write once value

Bit size

1

8

16

32

64

Raw

Signed:

Unsigned:

Hex:

Float:

Boolean:

String:

Raw bytes:

Enum:

0

0

0

TRUE

00 00

OK

Cancel

可以输入有符号，无符号，十六进制等类型数据，点击确定即可修改。如过修改的值被程序更改，那么可以勾选Forced 表示强制给值。

2.8 运动控制

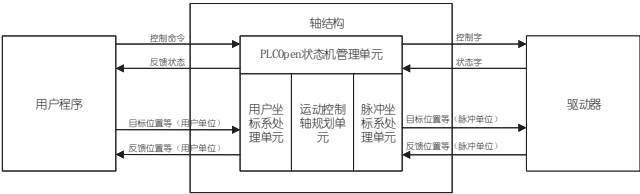
2.8.1 运动控制轴简介

2.8.1.1 概述

◆ 基本构成和逻辑控制

在运动控制系统中，将运动控制的对象称为轴。轴是连接驱动器和PLC指令间的桥梁。运动控制轴用于控制

符合402协议的EtherCAT总线驱动器，同时还可以控制编码器轴，虚拟轴等等。在PLC内部，轴的基本构和处理逻辑如下：



◆ 轴类型

支持的轴类型包括总线伺服轴，总线编码器轴，虚拟伺服轴，虚拟编码器轴。

轴类型	内容
总线伺服轴	使用EtherCAT从站伺服驱动器控制的轴。 本轴用于分配给实际的伺服驱动器加以使用。 总线伺服轴支持位置，速度，原点回归几种基本模式的控制。
总线编码器轴	使用EtherCAT从站远程IO控制的轴 本轴用于分配给实际的EtherCAT远程IO计数器加以使用。
虚拟伺服轴	虚拟的总线伺服轴，不需要关联伺服驱动器。
虚拟编码器轴	虚拟的总线编码器轴，不需要关联总线编码器。

为完成的描述轴的属性，控制轴的运动，将轴分成两部分：

轴结构	功能
轴配置参数	用于配置轴的各个参数，例如齿轮齿数比，原点回归类型，编码器模式等。
轴控制指令	在用户程序中，使用MC运动控制指令执行轴运动控制。 轴控指令分为管理类（如MC_Power）、运动类（如MC_MoveAbsolute）和状态类（MC_ReadStatus）。

2.8.1.2 轴的单位

在轴结构中用到两个单位，分别为用户单位和脉冲单位。

• 用户单位

在指令侧用的以毫米、厘米、角度等计量单位称为用户单位，通常用Unit表示。

用户坐标系根据使用工况的不同又分为线性坐标系和旋转坐标系。

线性坐标系通常包含一个零点，目标位置增大时为正向运动，目标位置减小时为反向运动。线性坐标系可以设置正负软限位。旋转坐标系包含一个零点和一个旋转周期，一个循环周期内，目标位置增大时为顺时针运动，目标位置减小时为逆时针运动。

• 脉冲单位

在驱动器侧使用的以脉冲数为计量的单位，通常用pulse表示。

驱动器通常包含脉冲零点和电机转一周编码器的脉冲数两个参数。

2.8.2 创建轴对象一般步骤

2.8.2.1 新建工程

要正确控制轴，首先根据项目需要创建组态，然后根据工况设定相关的参数，下载工程后通过在线调试进行简单的操作，判断参数设置是否合理，硬件连接是否可靠，最后编写PLC程序完成整体的控制逻辑功能。举例如下。

新建X3050的工程，在创建工程时需要选择X3050。

如下图所示：



2.8.2.2 创建组态

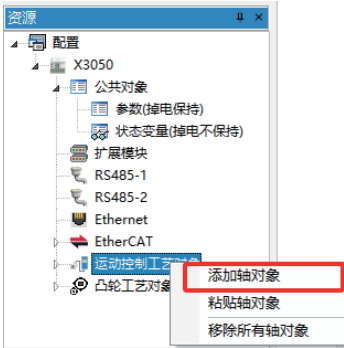
要控制AX-CM55运动，需要配置一个伺服驱动器和一个总线伺服轴，并将两者链接到一起。

2.8.2.2.1 添加伺服驱动器

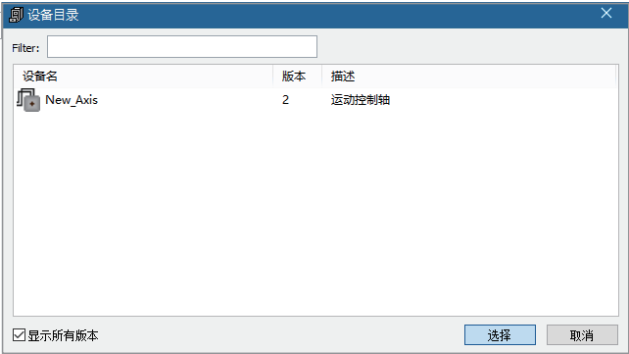
添加伺服驱动器，在EtherCAT 中添加或者扫描一个伺服驱动器。（详情见2.7节EtherCAT 通信）

2.8.2.2.2 添加工艺轴对象

在BonengLogiclab 页面，找到资源窗口，鼠标右击运动控制工艺对象，点击添加轴对象，如下图所示：



点击添加轴对象后，会弹出设备目录窗口，选择运动控制轴，如下图所示：



添加完成后，在运动控制工艺对象下面会出现一个 New_Axis_0 如下图所示：

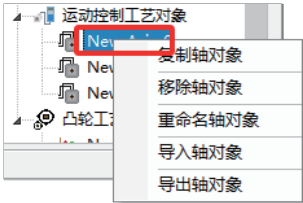


2.8.2.3 操作轴对象

右击运动控制工艺对象，可以弹出菜单，可以添加，粘贴，移除所有轴对象，如下图所示：



右击某个轴对象，可以复制，移除，重命名，导入，导出轴对象



2.8.2.4 关联伺服驱动器

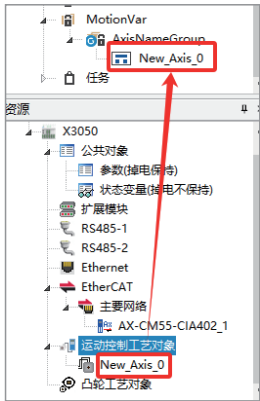
在EtherCAT节点中添加好伺服驱动器，之后便可以，在输出设备中可以找到该伺服驱动器，如下图所示：



选择该伺服驱动器，就可以把该驱动器关联到轴上了

2.8.2.5 保存或编译生成

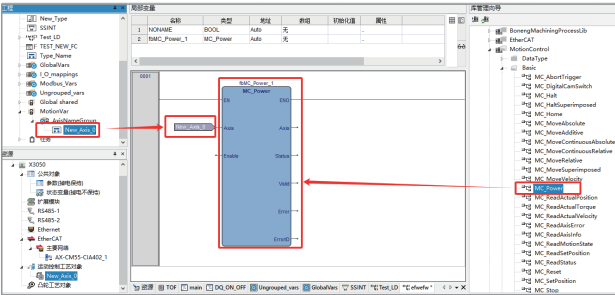
在轴配置页面当中，在单位换算，模式/参数设置原点返回设置中，将需要设置的参数设置好后，保存编译即可。在工程窗口中会生成一个变量组，在该组内，可以发现，会生成和刚才轴创建时名称相同的变量名称，在PLC程序中，调用MC_块时，需要调用这个变量，从而来标识是哪一个轴。如下图所示：



2.8.2.6 编写程序

在某个梯形图程序中，想要使能该轴，可以查看右侧库管理向导中点击展开MotionControl文件夹，点击展开Basic文件夹，找到MC_Power功能块，使用鼠标拖拽到梯形图网络上，然后再将左侧工程中的，刚才生成的变量拖拽到Axis的管脚上，即可完成轴

使能的程序编写，接下来通过控制Enable引脚就可以控制轴的使能状态。如下图所示：



2.8.3 轴页面配置

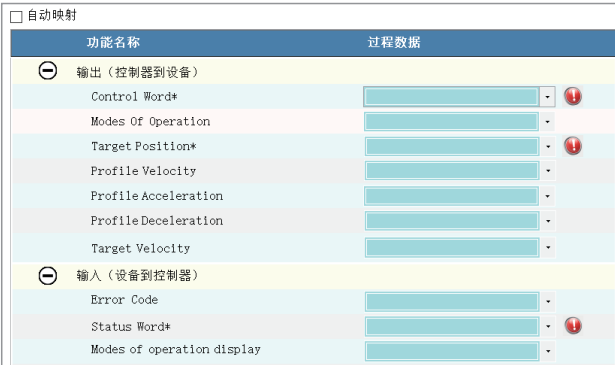
2.8.3.1 轴基本设置

2.8.3.1.1 总线伺服轴

点击刚才创建的好的轴，在轴类型中选择，可以选择所需要的轴类型。支持总线伺服轴，总线编码器轴，和虚拟轴。选择保留无效。且编译报错。



当选择的轴类型是总线伺服轴时，输入设备无法选择。当轴类型选择为编码器轴时，输出设备无法选择。选择总线伺服轴时，需要在绑定的伺服轴的PDO中至少需要控制字，状态字和目标位置，否则会有三个错误提示图标



如果没有，需要到ECAT配置页面去配置PDO

2.8.3.1.2 总线编码器轴

总线编码器中的设置与伺服类型，如果选择总线编码器轴，那么需要在下拉框中配置如下图所示的四个数据，使能，方向，预设值，当前计数值，如果不配置，那么编译会报错。



2.8.3.2 单位换算

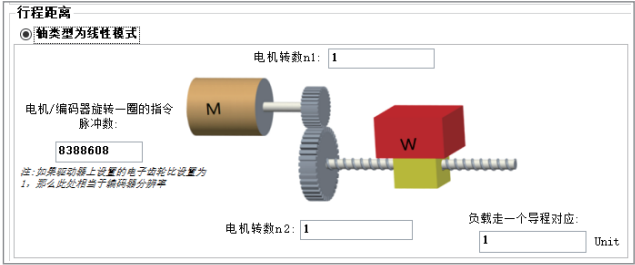
方向设置中，当勾选反向后，轴的实际的运动方向回发生改变。



在单位换算页面中需要设计电机或编码器旋转一圈的指令脉冲数，这个值和驱动器上的电子齿轮比有关，当驱动器上的电子齿轮比设置为1:1时，这里应该填入编码器分辨率（编码器在电机上）。当驱动器上的电子齿轮比不为1:1时，这里应填入编码器分辨率除以电子齿轮比。如下图所示：

$$\text{电机或编码器旋转一圈的指令脉冲数} = \frac{\text{编码器分辨率}}{\text{电子齿轮比}}$$

以博能23位编码器为例，当驱动器上设置的电子齿轮比为1:1时，此处应该填8388608



所以有

脉冲数=编码器分辨率 / 电子齿轮比 * 电机转数n1 / 电机转数n2 * 移动距离 / 单位导程移动的距离

注：单位导程对应负载旋转一圈

以上图配置为例：

当相对位置指令给定的目标位移为10时, 运动控制轴实际发送的脉冲数为83886080, 此时的电机旋转10圈
当轴类型为旋转模式时，与线性模式同理。

2. 8. 3. 3 参数模式设置

参数模式设置的界面如下：

编码器模式选择

☒ 绝对模式 ☐ 增量模式

模式设置

☒ 有限轴 ☐ 模数轴

软件限位

☐ 负向限制使能: Unit

☐ 正向限制使能: Unit

软件出招响应

限位减速速度: Unit/s²

轴故障减速速度: Unit/s²

阈值设置

跟随误差阈值: Unit

速度达到阈值: Unit

轴速度设置

最大速度: Unit/s

最大加速度: Unit/s²

Jog最大速度: Unit/s

扭矩设置

正扭矩最大值: 0.1%

负扭矩最大值: 0.1%

选项

☒ 碰撞限位后不进入ErrorStop状态

2. 8. 3. 3. 1 编码器模式

• 增量模式

PLC侧不考虑伺服驱动器编码器32位计数器的溢出导致的圈数的增加。PLC在断电重启时不保存编码器的当前位置，重新上电后，轴内部仅根据伺服驱动器反馈的单圈的位置计算出轴的当前位置。

• 绝对模式

PLC侧考虑伺服驱动器编码器32位计数器的溢出导致的圈数的增加。PLC在断电重启时保存编码器反馈的当前位置，重新上电后，轴内部在初始化过程中读取PLC内部保存的轴的编码器位置和伺服驱动器反馈的位置，推算出轴的当前绝对位置。

• 模式设置

按照实际使用的工况，运动控制轴可以设置为线性轴和模数轴。

• 有限轴

线性模式通常用于X-Y直线坐标系下的机械动作范围的装置中使用。

线性模式下通常包含一个零点。

运动时反馈位置增大则表示正向运动，反之为反向运动。

允许设置正向软件限位和反向软件限位，当使能软件限位后轴只能在限位范围内运动。

绝对定位方式: 当目标位置大于起始位置, 则正向移动目标位置-起始位置的距离; 当目标位置小于起始位置时则反向移动起始位置-目标位置的距离。

相对定位方式：当目标位移大于0时，正向移动目标位移的距离，当目标位移小于0时，反向运动“目标位移”的距离。

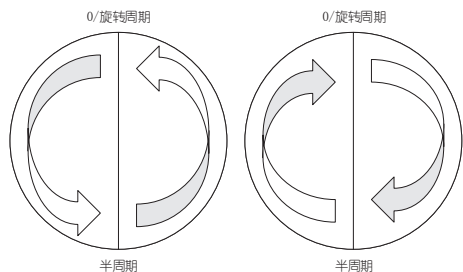
线性模式下的速度类指令处理方式：目标速度大于0则正向运动，目标速度小于0则反向运动。

• 模数轴

环形模式是在设定范围内重复无限计数的换计数器形式的模式。通常在转台或卷轴等中使用。

环形模式通常包含一个零点和一个旋转周期。环形计数器的反馈位置范围为0≤ 反馈位置<旋转周期。

环形模式下，如果反馈位置增大则认为是顺时针运动，如果反馈位置减小则认为是逆时针运动。

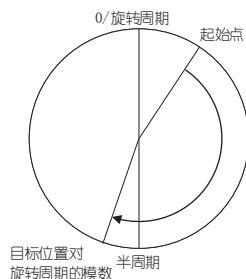


模数轴下没有软限位。

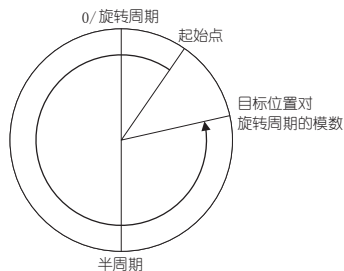
相对定位处理方式：目标位移大于0则顺时针移动目标位移的距离，目标位移小于0则逆时针移动目标位移的距离。

绝对定位处理方式：

正向：先将目标位置对旋转周期取模数，然后将轴从起始位置按照顺时针的方式运动到目标位置。



反向：先将目标位置对旋转周期取模数，然后将轴从起始位置按照逆时针的方式运动到目标位置。



最短距离：首先将目标位置对旋转周期取模得到目标位置，然后计算从起始点顺时针运动到目标位置的位移，如果该位移小于等于半周期则顺时针运动，否则采用逆时针运动到目标位置。

当前方向：按照轴最近一次的运动方向运动到目标位置，如果是第一次上电则正向运动到目标位置。
模数轴模式下的速度指令处理方式：目标速度大于0则顺时针运动，目标速度小于0则逆时针运动。

2.8.3.3.2 软件限位

在线性模式下允许设置软限位。如果软限位有效，在轴的运行过程中会时刻检测从当前速度按照设定的限位减速度做T型减速到0时轴的绝对位置，如果轴的绝对位置超出限位范围，则轴将执行软限位减速算法，同时打断当前正在执行的定位或速度指令。原点回归和力矩模式下软限位无效。

2.8.3.3.3 轴故障减速

在轴运行期间如果运动指令自身逻辑故障导致轴必须切换到errorstop状态，则轴将按照轴故障减速度设置的减速度做T型减速，直到减速到0后轴才会进入errorstop状态。

2.8.3.3.4 跟随误差

在执行定位指令和速度指令期间，伺服驱动器实际工作在CSP（周期性同步位置）模式，位置曲线的规划在PLC侧完成。PLC通过0x607A向伺服驱动器发送目标位置，伺服驱动器驱动伺服电机运动，电机编码器的位置通过0x6064反馈到PLC，由于伺服驱动器和电机本身的原因，0x607A和0x6064之间产生差值。该差值换算成用户单位后就叫做跟随误差。设置跟随误差最大值。如果轴的跟随误差的绝对值超过跟随误差最大值，则轴报跟随误差过大故障并进入errorstop状态。

2.8.3.3.5 轴速度设置

允许设置最大速度、最大加速度、最大点动速度三个参数。当定位指令或速度指令中的目标速度、加速度、减速度等参数超过速度限制值，则相关指令报故障并且轴进入ErrorStop状态。在总线伺服轴中，最大速度会通过单位换算换算成脉冲单位，通

过启动参数写入伺服驱动器的对象字典0x607f中。

2.8.3.3.6 扭矩设置

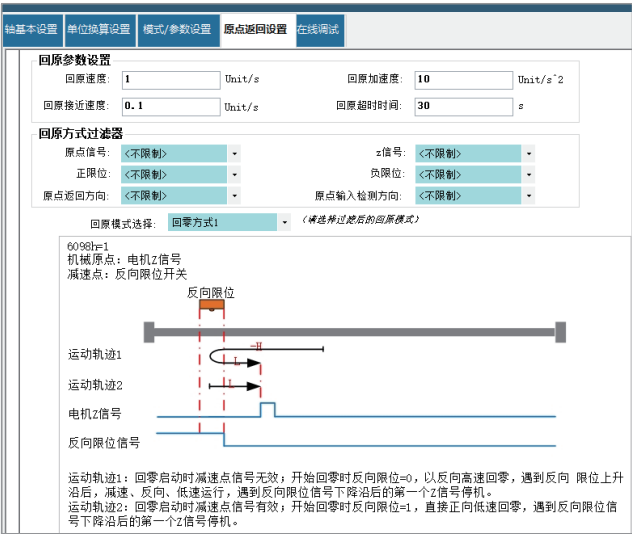
扭矩设置仅用于总线伺服轴。力矩指令中目标力矩超出最大扭矩则指令报故障且轴进入errorstop状态。正向力矩限制值将通过启动参数写入伺服驱动器的对象字典0x60e0中，反向力矩限制值将通过启动参数写入伺服驱动器的对象字典0x60e1中。

2.8.3.3.7 碰限位后不进入ErrorStop状态

选该选项之时，当运动控制轴碰到限位后仅轴控指令报故障，运动控制轴不再进入ErrorStop状态。不勾选该选项时，除MC_Jog外，由其他运动控制指令导致的碰到限位后，运动控制轴进入ErrorStop状态，同时指令报错。

2.8.3.4 原点返回设置

支持Cia402协议支持的1-35号回原方式。
原点返回设置界面如下：



配置界面中参数如下：

设置参数	描述
原点信号	用于选择是否使用原点信号 当选择未分配时,不作为强制筛选条件。 当选择不使用时，将除去必须使用原点信号的回原方式。 当选择使用时，将除去不支持原点信号的回原方式。
负限位	用于选择是否使用硬件左限位信号 当选择未分配时,不作为强制筛选条件。 当选择不使用时，将除去必须使用负限位信号的回原方式。 当选择使用时，将除去不支持负限位信号的回原方式。
正限位	用于选择是否使用硬件右限位信号 当选择未分配时,不作为强制筛选条件。 当选择不使用时，将除去必须使用正限位信号的回原方式。 当选择使用时，将除去不支持负限位信号的回原方式。
Z信号	用于选择是否使用电机z信号 当选择未分配时,不作为强制筛选条件。 当选择不使用时，将除去必须使用z信号的回原方式。 当选择使用时，将除去不支持z信号的回原方式。
原点回归方向	用于设置在原点回归开始时的运动方向 正向：限位（原点）信号输入无效时的运动方向为正，否则反向 负向：限位（原点）信号输入无效时的运动方向为负，否则反向

设置参数	描述
原点回归检测方向	到达原点信号时的运动方向 正向：在正向运动过程中碰到原点信号边沿停止。 负向：在负向运动过程中碰到原点信号的边沿停止。
原点返回列表	原点返回方式，设置范围是1-35，通过启动参数的形式写入对象字典0x6098。
回零方式	35号回零模式下设置相对模式或绝对模式，通过启动参数的形式写入对象字典0x60e6。
原点返回速度	原点返回速度，将用户单位转换为脉冲单位后通过启动参数的形式写入对象字典0x6099的1号子索引。
原点返回接近速度	原点返回接近速度，将用户单位转换为脉冲单位后通过启动参数的形式写入对象字典0x6099的2号子索引。
原点返回加速度	原点返回加速度，将用户单位转换为脉冲单位后通过启动参数的形式写入对象字典0x609A。
原点返回超时时间	博能驱动器独有，其他厂家写入无效。

在实际使用时通过原点信号、正限位、负限位、Z信号、原点回归方向、原点回归检测方向几个参数来限定原点回归方法，然后通过原点返回列表选项选出需要的模式。

需要注意的是当设置完回原筛选条件后仍然有多种回原方式，此时需要通过原点返回列表选择适当的回原方式。例如当设置参数如下表格，可以筛选出两种回原方式

信号	值
原点信号	使用
负限位	不使用
正限位	使用
Z信号	使用
原点返回方向	正向
原点返回检测方向	正向

根据以上设置，可以筛选出8号和10号两种回原方式，两者的区别为在回原完成的时候，原点输入信号是否有效。

2.8.3.5 在线调试

2.8.3.5.1 在线监控

通过PLC指令、后台的在线调试界面获取轴的状态。

1. 通过指令方式获取轴状态

通过MC_ReadStatus、MC_ReadPosition等状态类指令获取轴的状态。具体使用方法请查看第三章指令说明。

2. 通过在线调试界面获取轴状态

通过在线调试界面，可以实时查看各轴运动状态，轴的在线监控界面如下图所示。

变量	设置值	实际值
位置	-235762.0156	-235762.0156
速度	0.2384185791	0
加速度	0	0
扭矩	0	0

状态: disabled

通讯: 0p

轴错误: 0/16#0

伺服错误: 0/16#0

/	硬件正限位	硬件负限位	原点开关	软件正限位	软件负限位
状态	False	False	False	False	False

进入伺服调试>>>

可供监控的数据如下表所示：

对象	功能
设置位置	PLC执行路径规划时的目标位置（用户单位）
设置速度	PLC执行路径规划时的目标速度（用户单位）
设置加速度	PLC执行路径规划时的目标加速度（用户单位）
设置力矩	PLC执行力矩规划时的目标力矩（%）
实际位置	驱动器反馈的当前位置（用户单位）
实际加速度	根据实际速度计算出的加速度（用户单位）
实际力矩	驱动器反馈的实际力矩（%）
状态	PLCOpen状态机的状态： 0:PowerOff，去使能 1:ErrorStop，故障停机 2:Stopping，停止 3:StandStill，使能 4:DiscreteMotion，点位运动 5:ContinuousMotion，连续运动 7:Homing，原点回归 8:SynchronizedMotion，同步运动
通讯	
轴错误	运动控制轴内部错误
伺服错误	如果是本地脉冲轴为本地脉冲轴的故障码,如果是总线伺服轴则对应0x603F的值,参考使用的驱动器的手册。
硬件正限位	表示运动控制轴的硬件正限位输入是否有效
硬件负限位	表示运动控制轴的硬件负限位输入是否有效

对象	功能
原点开关	表示运动控制轴的原点开关输入是否有效
软件正限位	表示运动控制轴是否到达软件正限位
软件负限位	表示运动控制轴是否到达软件负限位

2.8.3.5.2 在线控制

可通过在线调试功能实现一些基本的伺服控制，例如使能，停止，点动，点位控制等功能。在确定基本动作正常之后可以通过运动控制指令实现复杂的逻辑控制。在线调试和PLC指令控制不能同时使用，限制如下：

值得注意的是当轴进入在线控制时PLC程序会停下来。在线控制所具有的功能如下表所示：

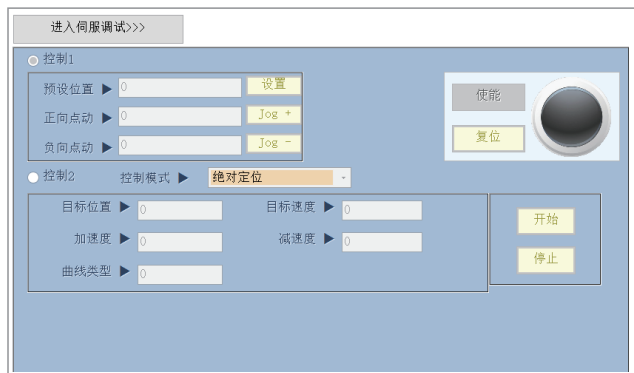
功能点	描述
进入在线调试模式	使轴进入在线调试模式，进入在线调试模式后，PLC运控类指令将不能继续执行。
使能	类似调用MC_Power指令使轴处于使能状态或去使能状态
复位	当轴发生故障后用于复位轴的故障
停止	停止轴的运动
设置当前位置	设置轴的当前位置
点动	实现点动功能
运动类型	用于选择哪种运动模式,可选项有: 1：绝对定位 2：相对定位 3：连续运动 5：往复运动

功能点	描述
相对定位	实现相对定位功能
绝对定位	实现绝对定位功能
连续运动	以一定速度连续运动
往复运动	实现两个绝对位置间的往复运动
力矩控制	实现力矩控制功能
原点回归	执行原点回归动作

在线调试步骤

1. 进入在线调试模式

如下图所示，点击“进入在线调试”按钮可以使轴进入在线调试模式。



后台在接收到进入在线调试的命令后会主动停止PLC程序

2. 基本操作

使能：进入在线调试模式后，点击“使能”按钮可以让轴进入使能状态，执行效果等同如下指令

点动：当轴处于Standstill状态时，点击“Jog+”按钮轴将按照“正向点动”框设置的目标速度往正方向运动，松开即停止；点击“Jog-”后轴将按照“负向点动”框设置的目标速度往相反方向运动，松开即停止。

复位：当轴处于errorstop状态时，点击“复位”按钮将尝试复位轴的故障，如果伺服驱动器的故障属

于不可复位故障，则有可能出现复位失败的情况。

停止：界面中有两个停止按钮，功能相同，都是用于停止轴的运动。使用的减速度是轴的模式/参数设置里面的限位减速度。

3. 运动模式

需要先使能，不使能点击运无效。

• 绝对定位

控制模式选择绝对定位后，可以设置目标位置，目标速度，加速度，减速度，曲线类型五个参数，点击开始按钮后轴将按照以上参数做绝对定位。

• 相对定位

控制模式选择相对定位后，可以设置目标位置，目标速度，加速度，减速度，曲线类型五个参数，点击开始按钮后轴将按照以上参数做相对定位。

• 连续运动

控制模式选择连续运动后，可以设置目标速度，加速度，减速度，曲线类型四个参数，点击开始按钮后轴将按照以上参数做绝对定位。

• 往复运动

控制模式选择往复运动后，可以设置两组目标位置，两组目标速度，两组加速度，两组减速度，两组曲线类型，点击开始按钮后轴将按照以上参数首先定位到目标位置1，然后定位到目标位置2，依次往复。

• 原点回归

设置原点偏移，原点返回速度，原点返回加速度，原点返回接近速度后四个参数后点击开始按钮后轴将按照以上参数做原点回归

2.9 电子齿轮

2.9.1 齿轮动作

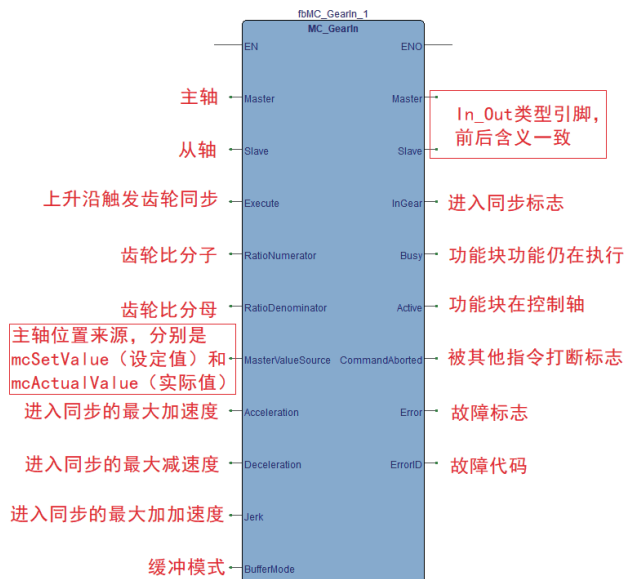
- 齿轮动作的主轴和从轴类型要求如下

- 主轴可以是总线伺服轴、虚拟轴和总线编码器轴
- 从轴：总线伺服轴和虚拟轴
- 齿轮动作通过MC_GearIn（齿轮动作开始）指令进入同步，通过MC_GearOut（齿轮动作结束）指令或MC_Stop（周停止动作）指令解除从轴与主轴的同步。
- MC_GearIn齿轮同步整体分为两个阶段，进入同步前的阶段称为过渡阶段，进入后为同步阶段。
- 齿轮过渡阶段，从轴以功能块上用户给定的加减速速度，以主轴速度乘以齿轮比得到的速度为目标速度，追赶主轴以进入同步。
- 齿轮同步阶段，从轴速度以主轴速度乘以齿轮比得到的速度为目标速度，同步动作。
- 齿轮比为正数，从轴沿主轴同方向移动；

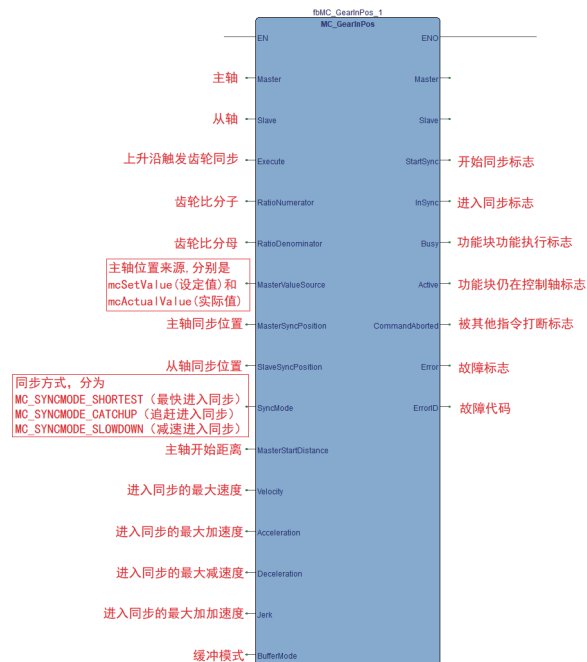
2.9.2 电子齿轮使用

2.9.2.1 MC_GearIn

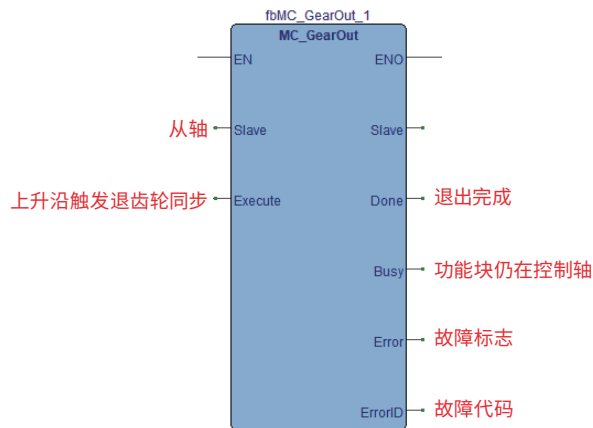
MC_GearIn功能块控制从轴速度与主轴速度符合设置的比率。



2.9.2.2 MC_GearInPos



2.9.2.3 MC_GearOut



2.9.3 轴配置/状态总览

点击运动控制工艺节点，可以查看所有轴的部分配置相关信息，其中包含轴名称，轴ID，轴类型，绑定的伺服名称，轴的通讯状态，以及轴的调试状态。如下图所示：

轴配置/状态总览					
轴名称	轴号	轴类型	指定的伺服	伺服源状态	源状态
New_Axis_0	0	伺服	AK-CIA402.1	Op	True
New_Axis_1	1	伺服	AK-CMS5-CIA402.1	Op	False

在进入调试状态后，pcl程序会停止运行，可以通过冷，热，暖三种启动方式进行启动，且当点击任意一种启动方式后所有轴都会退出伺服调试模式。

2.9.4 故障类别

轴故障分成指令故障、轴故障和驱动器故障三类。

- 指令故障是MC轴控指令本身产生的故障，例如指令参数不合理，在运行过程中轴的PLCOpen状态机发生变化导致指令报错等。可以通过查看故障指令的ErrorID获取故障码。轴故障是轴自身报出的故障，例如跟随误差过大等。轴故障码有四种查看方式。第一种可以通过后台“在线调试”界面中“轴故障”一栏获取。第二种通过MC_ReadAxisError指令中的AxisErrorID获得。
- 驱动器故障是EtherCAT总线驱动器或者本地脉冲输出轴的故障。要获取EtherCAT总线驱动器的故障，PDO映射中必须配置0x603F并关联到轴中。驱动器故障有三种查看方式。第一种可以通过后台“在线调试”界面中“伺服错误”一栏获取。第二种通过MC_ReadAxisError指令中的ServoErrorID获得。

2.10 电子凸轮

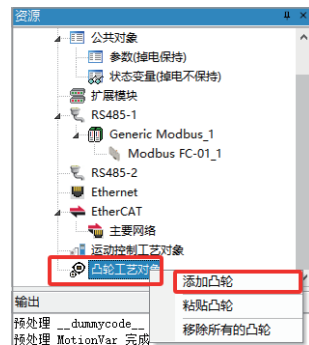
2.10.1 电子凸轮简介

- 电子凸轮是从轴跟随主轴的运动，主轴和从轴的位置对照关系可通过凸轮表来设定，
- 凸轮表可通过软件页面配置来生成，使用电子凸轮表可以建立最大32个关键点数据
- 凸轮执行过程中不允许增加，删除，修改凸轮表的关键点，修改关键点数据后需要重新下载代码生效。

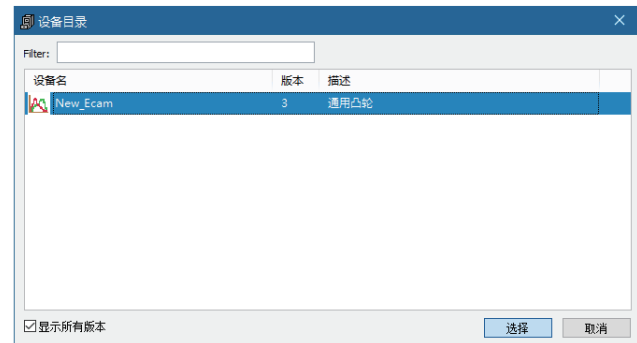
2.10.2 软件配置

2.10.2.1 添加凸轮

在资源栏中找到凸轮工艺对象的节点，右击添加凸轮如下图所示：



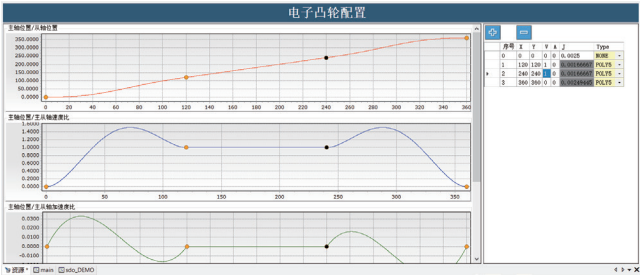
点击添加凸轮后，在弹出的设备目录框中选择通用凸轮。如下图所示：



添加好后在凸轮工艺对象下面会出现一个新的节点一般第一个是（New_Ecam_0），点击该节点即可进入该凸轮表数据的设计。

2.10.2.2 设计凸轮

点开添加好的通用凸轮，来到电子凸轮的配置页面，该页面主要分为两部分，左侧是凸轮曲线编辑部分，右边是凸轮的表格部分。凸轮表格主要是存放关键点数据。凸轮曲线可以更加直观展示凸轮运动轨迹。



2.10.2.3 编辑凸轮表格

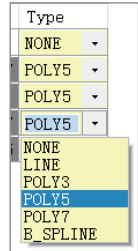
用户可以根据自己的需求在表格中添加，删除关键点，和修改一些关键点的数据。

	序号	X	Y	V	A	J	Type
	0	0	0	0	0	0.0025	NONE
	1	120	120	1	0	0.00166667	POLY5
	2	180	180	1	0	0	POLY5
	3	240	240	1	0	0.00166667	POLY5
	4	360	360	0	0	0.00249445	POLY5

点击  可以在三角形所在的行的上方添加一行数据（首行不能添加），点击  可以删除（首行不能删除，尾行不能删除），三角符号所在的行。

表格中可修改的数据一个有5列，分别为 X（主轴位置），Y（从轴位置），V（主从速度比），A（主从加速度比），J（主从加加速度比），Type（曲线类型）。

在曲线类型的一列可以选择每两个关键点之间的曲线的类型，支持直线，三次，五次，七次，如下图所示：



2.10.2.4 编辑凸轮图形

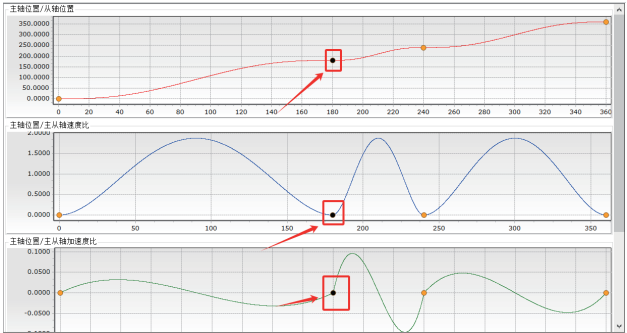
凸轮曲线的操作允许对关键点进行拖拽，对于主从轴位置图像曲线，允许对上面的点进行上下，左右

拖动，而速度，加速度，加加速度曲线只允许上下拖动，具体规则如下表所示：

曲线图 \ 曲线类型	LINE	POLY3	POLY5	POLY7
主从位置	连续	连续	连续	连续
主从速度比	不连续	连续	连续	连续
主从加速比	不连续	不连续	连续	连续
主从加加速度比	不连续	不连续	不连续	连续

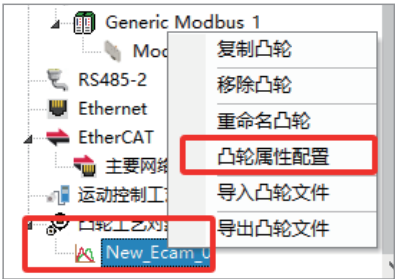
当选择的该段曲线类型不连续时，无法拖动该段曲线的关键点进行移动。

当鼠标在凸轮曲线上的关键点悬停时，该关键点会被变成黑色，而且会将凸轮表格中的数据高亮显示。且在相同的主轴位置的其余曲线的点也会变成黑色。



2.10.2.5 凸轮属性配置

右击添加好的凸轮，点击凸轮属性配置



点击后如下图所示：



这里可以设置关键点的一些相关约束。

主轴开始位置，设置后将改变第一个关键点，的主轴位置

主轴结束位置，设置后将改变最后一个的关键点，的主轴位置

从轴开始位置，设置后从轴位置不允许小于这个值
从轴结束位置，设置后从轴位置不允许大于这个值
平滑过渡，勾选后第一个点关键点的主轴位置将等于最后一个关键点的主轴位置。

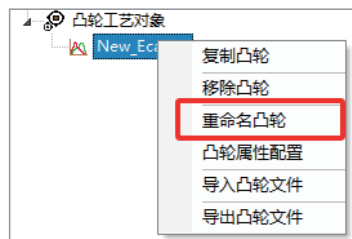
2.10.2.6 导入导出

右击凸轮设备，选择导出凸轮文件，导出的凸轮文件是以CSV的格式进行保存的。

右击凸轮设备，选择导入凸轮文件，导入的凸轮文件需是CSV的格式，并且符合导入要求。否则会导入失败。

2.10.2.7 重命名凸轮对象

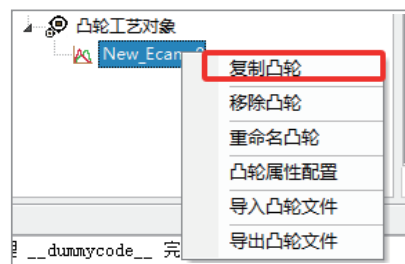
右击添加的凸轮设备，选择重命名凸轮，即可开启编辑模式，编辑凸轮的名称。



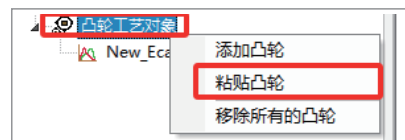
2.10.2.8 复制粘贴凸轮

鼠标右击已经编辑好的凸轮，当用户需要多个相同的凸轮时，可以使用复制粘贴的功能，

鼠标右击已经添加并编辑好的凸轮，选择复制凸轮如下图所示：

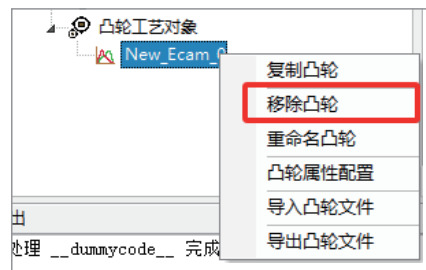


复制完凸轮后，鼠标右击凸轮工艺对象，粘贴凸轮，这样就可以得到两个拥有相同数据的凸轮了。



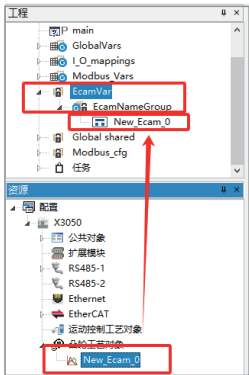
2.10.2.9 移除凸轮

鼠标右击自己不想要的凸轮，选择移除凸轮。



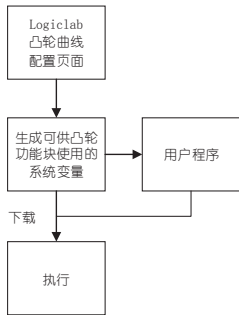
2.10.2.10 生成凸轮变量

在编辑好凸轮表格数据后，点击保存，BonengLogicLab会根据添加的凸轮设备的名称，直接产生一个，直接在工程窗口中生成一个名称相同的变量，该变量是全局变量，方便在功能块中直接调用。如下图所示：



2.10.3 电子凸轮使用

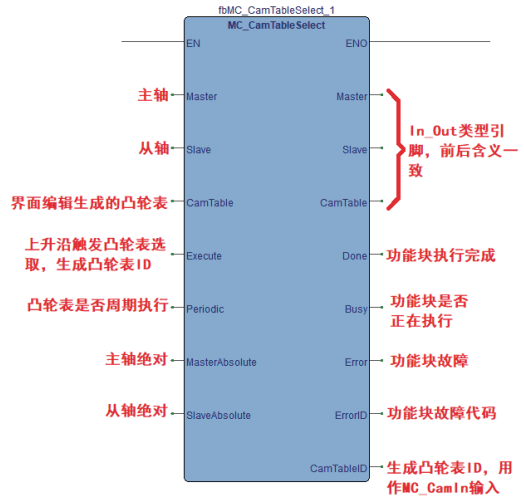
- 用户根据需求创建好凸轮表后，即可调用相应功能块进行编程。



- 凸轮相关功能块包括MC_CamTableSelect, MC_CamIn, MC_CamOut。
- 其中MC_CamTableSelect用于选取凸轮表 (MC_CAM_REF),生成可供MC_CamIn调用的凸轮ID (MC_CAM_ID)
- MC_CamIn触发后，从轴与主轴过渡后进入凸轮同步，而MC_CamOut用于接触从轴与主轴的从动关系，触发后从轴退出凸轮同步

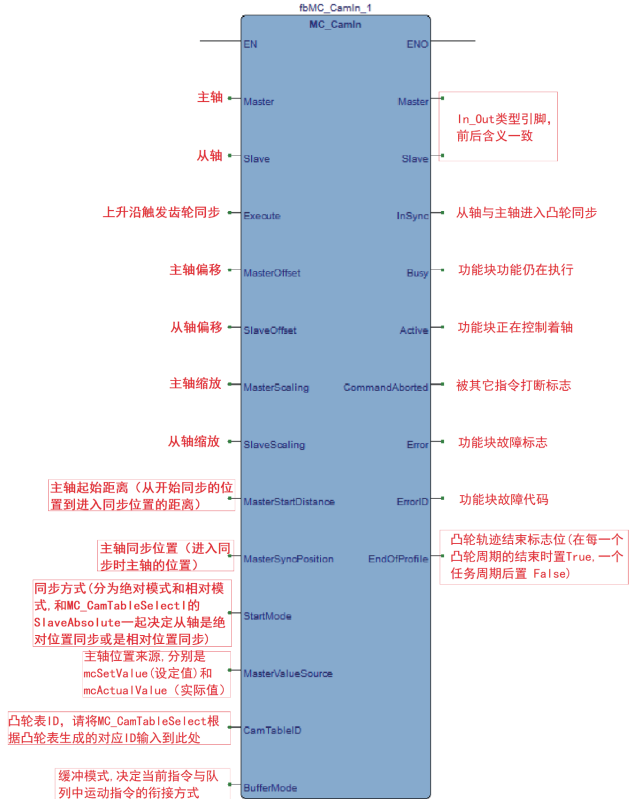
2.10.3.1 MC_CamTableSelect

MC_CamTableSelect功能块将凸轮表界面编辑生成的CamTable转化成MC_CamIn可使用的CamTableID。



2.10.3.2 MC_CamIn

MC_CamIn啮合主从轴，控制从轴按照凸轮表上主轴与从轴位置对应关系进行运动。



2.10.3.2.1 主从轴缩放与偏移

用户可根据需求,通过改变MC_CamIn功能块的输入(MasterOffset,SlaveOffset,MasterScaling,SlaveScaling)来实现凸轮曲线主从轴的缩放与偏移。

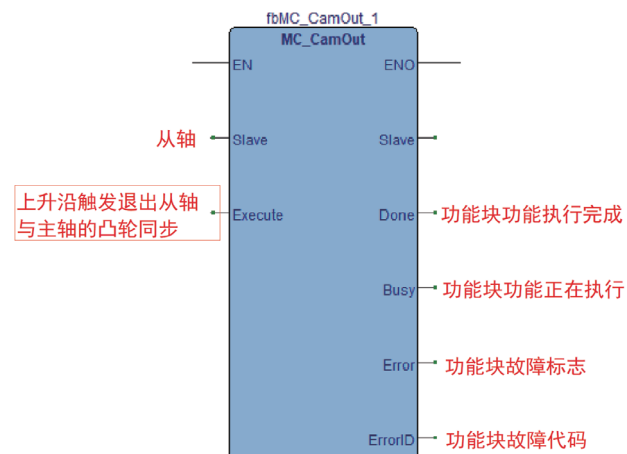
2.10.3.2.2 过渡区间

- 从轴跟随主轴进入同步,需要经过过渡区间,用户可以自定义过度区间的位置参数(MasterStartDistance, MasterSyncPosition)
- MasterSyncPosition规定了进入同步时刻的主轴位置,凸轮表主从轴的位置关系是确定的,因而从轴的同步位置也是确定的
- MasterStartDistance规定了进入同步前过渡区间的长度

2.10.3.3 MC_CamOut

MC_CamOut功能块取消从轴与主轴的啮合关系。

MC_CamOut触发后,轴会以触发时的速度持续运行,轴状态机从同步运动跳转到持续运动



2.11 轴组

2.11.1 轴组简介

- 轴组是让多个轴组成的机器人机构进行运动,可

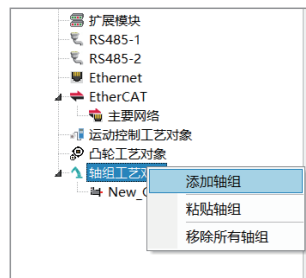
完成点到点,直线运动,圆弧运动,直线圆弧过渡运动。

- 轴组组成的机器人构型可通过软件页面配置来生成,目前可以配置三坐标等机器人,并可配置机器人关联的相关轴。
- 机器人关联的轴只可以被轴组控制,不能被单轴控制。

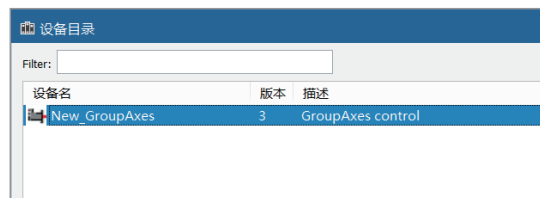
2.11.2 软件配置

2.11.2.1 添加轴组

在资源栏中找到轴组工艺对象的节点,右击添加轴组如下图所示:



点击添加轴组后,在弹出的设备目录框中选择轴组控制。如下图所示:



添加好后在轴组工艺对象下面会出现一个新的节点一般第一个是(New_GroupAxes_0),点击该节点即可进入该凸轮表数据的设计。

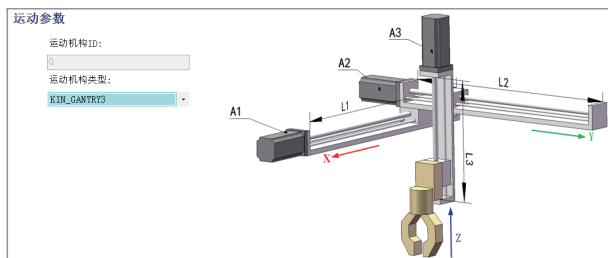
2.11.2.2 轴组配置

点开添加好的轴组,来到轴组的配置页面,该页面主要分为六个部分,分别为基本参数、互联、几何结构、动力学、对象坐标系、工具坐标系。



1. 基本参数配置

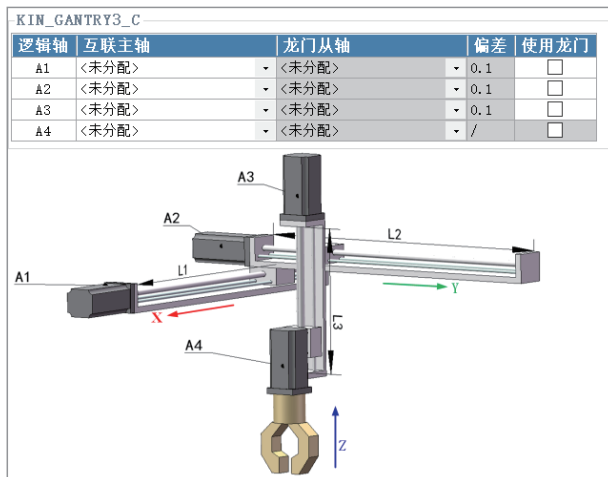
配置界面如下：



运动机构ID自动生成。运动机构类型可以选择，选择好后，右边会出现对应的构型图。

2. 互联配置

配置界面如下：



这个界面主要配置机构类型中关节电机对应哪些伺服轴。伺服轴需要提前配置好轴工艺对象。并且可以配置龙门同步轴。

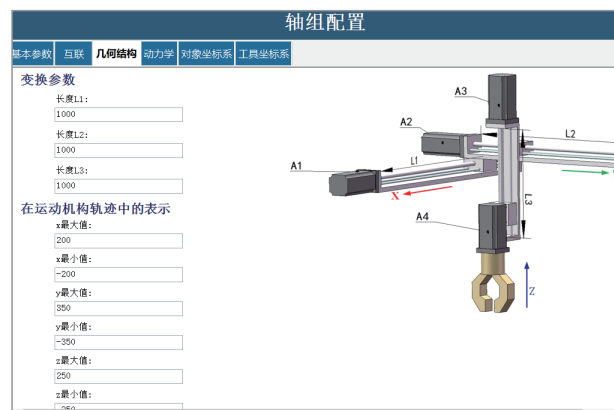
逻辑轴中A1到A3对应构型图中关节电机的位置。互联主轴下链接对应的伺服轴。龙门从轴需要点击使用龙门才可以配置龙门从轴下链接的伺服轴。勾选龙门从轴对应的构型图也会改变。

偏差表示龙门从轴和主轴的位置偏差容许值，偏差超过这个值会报错。

3. 几何结构配置

这里主要配置机构构型的几何参数。

配置界面图如下，这里的长度单位为mm。

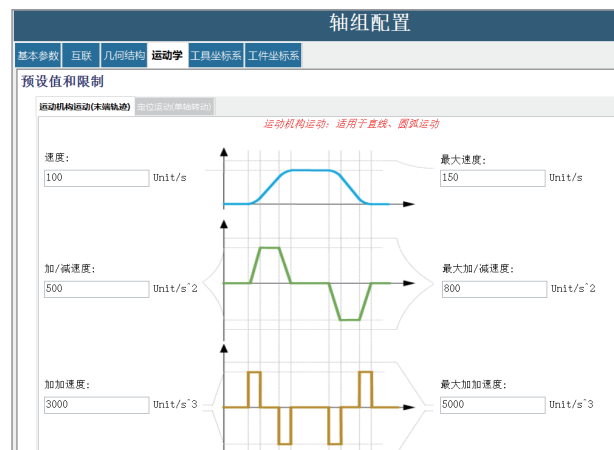


上图中的L1，L2，L3为图中构型对应的长度。X, Y, Z, C的最大值为构型可以运动到的最大位置，输入参数超过此位置系统会报错。

4. 运动学配置

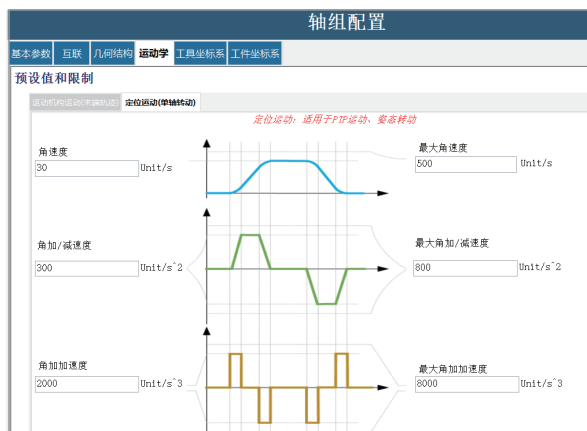
这里主要配置轴组相关的运动速度、加速度等。有两个配置选项，分别为运动机构运动配置和定位运动配置。

运动机构运动配置界面图如下：



其中运动机构运动中速度、加/减速度和加加速度表示末端的线速度、线加速度和线加加速度的输入限制。最大速度、最大加/减速度和最大加加速度表示实际运动过程中的限制。实际运动过程中末端的线速度、线加速度和线加加速度会存在一些波动。

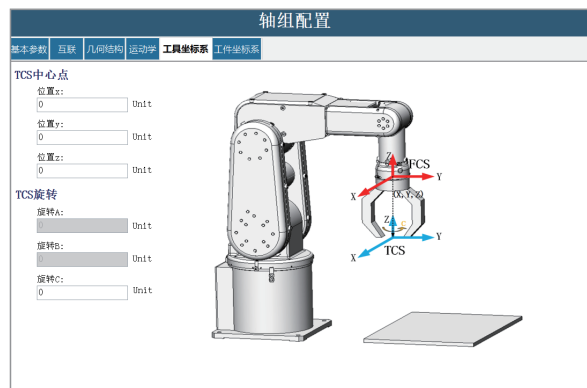
定位运动配置界面图如下：



其中定位运动中角速度、角加/减速度和角加加速度表示点到点运动的最大运动学限制,也表示第四转动轴的运动学限制。最大角速度、最大角加/减速度和最大角加加速度表示实际运动过程中的限制。实际运动过程中的角速度、角加速度和角加加速度会存在一些波动。

5. 工具坐标系配置

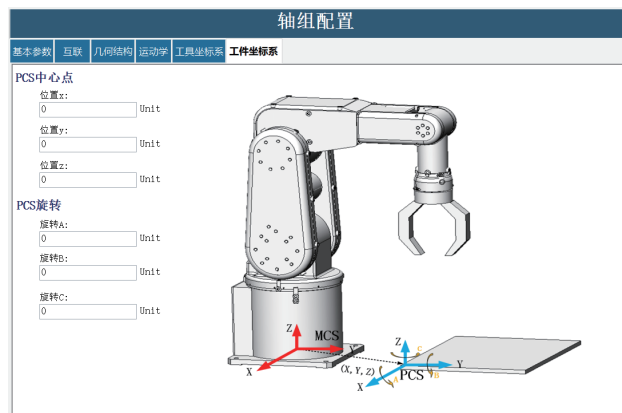
这里主要配置工具坐标系相对于末端坐标系的位置变换关系。配置界面图如下：



其中TCS中心点下面三个参数表示工具坐标系相对于末端坐标系的X, Y, Z方向偏移。TCS旋转下面三个参数表示工具坐标系绕末端坐标系的轴进行旋转。

6. 工件坐标系配置

这里主要配置工件坐标系相对于基坐标系的位置变换关系。配置界面图如下：



其中PCS中心点下面三个参数表示工件坐标系相对于基坐标系的X, Y, Z方向偏移。PCS旋转下面三个参数表示工件坐标系绕基坐标系的轴进行旋转。

2. 11. 2. 3 轴组使用

用户根据需求创建好轴组机器人后,可以调用相应的功能块进行编程。

轴组具体使用方式如下：

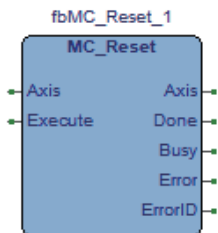
先在配置界面配置好轴组的相关参数。完成后可以得到轴组工艺对象。



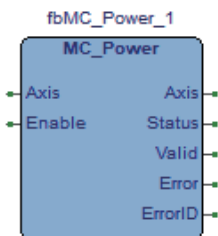
后续我们就在程序框架内调用功能块进行轴组设备的控制。

具体实现逻辑如下：

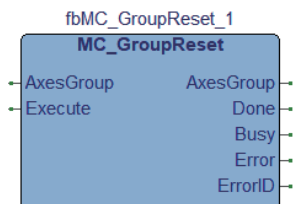
先需要调用单轴功能块进行单轴的复位操作



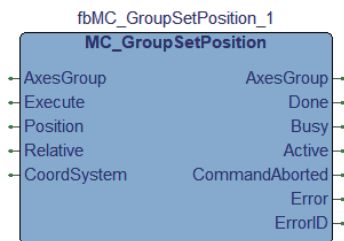
再调用单轴使能功能块对单轴进行是能上电，



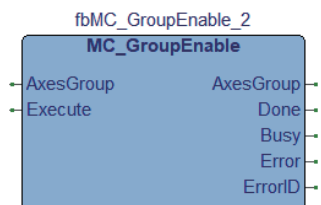
再调用轴组相关的功能块，先对轴组进行复位操作。



再调用设置当前位置命令来确定每个轴所在的位置：



再调用轴组使能模块进行轴组使能：



完成上述工作后就可以调用相关的直线，圆弧和点到点的规划命令进行轴组的运动控制。

3 指令说明

3.1 BonengControlLib

3.1.1 BN_EtherCAT

3.1.1.1 使用注意事项

1) BN_EtherCAT库的功能块用于控制EtherCAT从站，当使用X3050控制器并建立轴工艺对象时，若轴工艺对象选取了对应的EtherCAT从站作为输出设备，则该从站无法再被BN_EtherCAT库功能块使用，否则会报错FBErrorCode: 32 (16#0020)

2) BN_EtherCAT功能块分为两大类，分别控制伺服驱动器和变频驱动器，如表所示。

对应博能驱动器产品中的伺服（AX、AN）和变频器（AX、AD）

其中，AX系列产品A04.14可以进行伺服模式和变频模式的切换

A04.14	驱动器驱动模式选择	0: 伺服模式
A04.16	伺服电机编号	0: 变频模式 1: 伺服模式

若选择变频模式，C06.04需要修改为变频模式功能块，此时可以使用变频功能块

若选择伺服模式，C06.04需要修改为无，此时可以使用伺服功能块

C06.04	博能控制器功能块	0: 无
C06.05	软件版本号	0: 无 1: 变频模式功能块

	功能块名称	功能	AN / AX 产品 PDO配置	驱动器上位 机配置
变频 功能 块	BN_EtherCAT_ INV_Speed	变频模式， 速度控制	Output:1704 Input:1B04 (若选择AD产 品,则为1702 和1B02)	A04.14选择: 变频模式 C06.04选 择:变频模 式功能块
	BN_EtherCAT_ INV_Torque	变频模式， 转矩控制		
		伺服模式下 多种功能: 1: 相对位置		

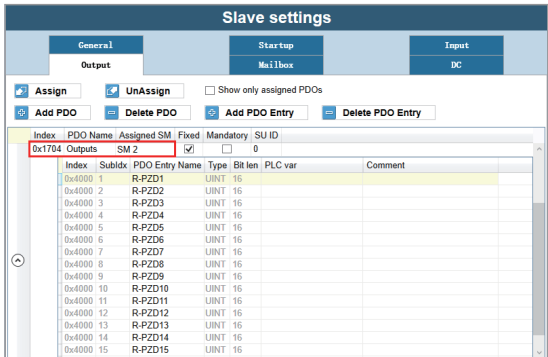
	功能块名称	功能	AN / AX 产品 PDO配置	驱动器上位 机配置
伺服功能块	BN_EtherCAT_PP_PV	2: 绝对位置 3: 速度模式 4: 主动回零 5: 设置当前位置为零点 6: **暂无 7: 速度点动 8: 位置点动	Output: 1703 Input: 1B03	A04.14 选择: 伺服模式 C06.04 选择: 无
	BN_EtherCAT_PV_Speed	伺服模式, 速度控制 (PV)	Output: 1702	
	BN_EtherCAT_PT_Torque	伺服模式, 转矩控制 (PT)	Input: 1B03	

PS: (1) 可以根据INV (Inverter) 的标签确认功能块是变频功能块
(2) 而PP、PV、PT这些属于伺服的模式, 可以根据此确认功能块为伺服功能块

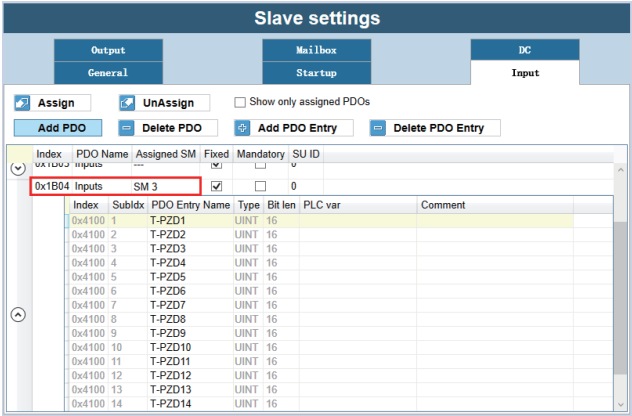
3.1.1.2 BN_EtherCAT_INV_Speed_V02模块使用

3.1.1.2.1 配置RPDO

打开output页面,取消其他PDO的同步管理输出配置SM2,并将1704添加到同步管理单元SM2中,针对博能Ether-CAT驱动器,不需要向其添加参数,默认如下图所示:



打开input页面, 选择1B04为同步管理输入配置SM3, 针对博能EtherCAT伺服驱动器, 不需要向其添加参数, 默认如下图所示:



3.1.1.2.2 引脚说明

BN EtherCAT_INV_SPEED_V02功能块引脚说明如下表所示:

名称	数据类型	描述	注意
AxisIndex	UINT	EtherCat轴ID;0—15	\
bEnable	BOOL	使能运行;0: 轴停止, 1:轴运行	\
bStopMode	BOOL	停机模式;0: 减速停机, 1:自由停机	\
bQuickStop	BOOL	快速停机;0:快速停机有效, 1:快速停机无效	\
bMotoRotation	BOOL	电机旋转方向;0: 正向, 1:反向	\
bFaultReset	BOOL	故障复位;0: 复位无效, 1:复位有效	上升沿触发
TargetSpeed	INT	目标速度;速度基准值为RefSpeed	\
RefSpeed	UINT	速度基准值	需要初始值,否则报错

名称	数据类型	描述	注意
TorqueUpLimit	INT	转矩上限,1000对应100.0%,100.0%对应电机额定转矩	\
TorqueLowLimit	INT	转矩下限,1000对应100.0%,100.0%对应电机额定转矩	\
PZD3	WORD	用户自定义PZD3	\
bReadyOK	BOOL	开机准备好状态;0:开机未准备好,1:开机准备好	\
bRunState	BOOL	运行状态;0:停机,1:运行	\
StateWord	WORD	状态字	\
bFault	BOOL	故障状态;0:无故障,1:有故障	\
FBErrorCode	WORD	功能块故障代码	\
AxisErrorCode	WORD	驱动器故障代码	\
ActSpeed	INT	反馈速度;基准值为RefSpeed	\
ActCurrent	UINT	输出电流;4096对应电机额定电流	\
ActTorque	INT	输出转矩;4096对应电机额定转矩	\
PZD6	WORD	用户自定义PZD6	\

模块介绍

- 1) AxisIndex轴名称根据拓扑依次为0—15，即与博能控制器EtherCat直接连接的为0轴，与0轴连接的为1轴，依次类推；
- 2) QuickStop要置1否则驱动器为开机封锁；
- 3) RefSpeed速度基准值，此数值不可为0；当此数值为5000时，TargetSpeed数值0—5000，对应电机0—50HZ。

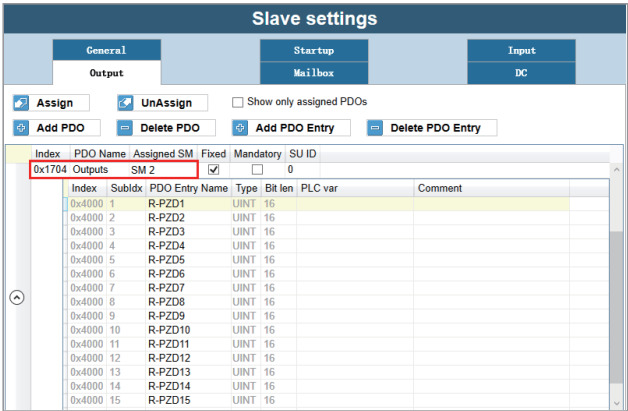
3.1.1.2.3 功能块故障说明

F B E r - rorCode	含义
16#0000	功能块无错误；
16#0001	未进入OP；
16#0002	AxisIndex 超过范围，或者未进入OP；
16#0100	参考速度未设置；
16#000A	P D O 配置不正确或不能获取 S t a - tusWord4100_01 / ControlWord4000_01 的地址；
16#000B	P D O 配置不正确或不能获取 A c t V e l o c i - ty4100_02 / TargetVelocity4000_02 的 地址；
16#000C	P D O 配置不正确或不能获取 A c t C u r - rent4100_03 / RPZD3_4000_03 的地址；
16#000D	P D O 配置不正确或不能获取 A c t - Torque4100_04 / TorqueUpLimit_4000_04 的地址；
16#000E	P D O 配置不正确或不能获取 D r i v e E r - ror4100_05 / TorqueLowLimit_4000_05 的地址；
16#000F	P D O 配置不正确或不能获取 TPZD6_4100_06 的地址；
16#0012	P D O 配置不正确或不能获取 O p e r a - tionMode_4000_09的地址；
16#0020	当前轴已被使用

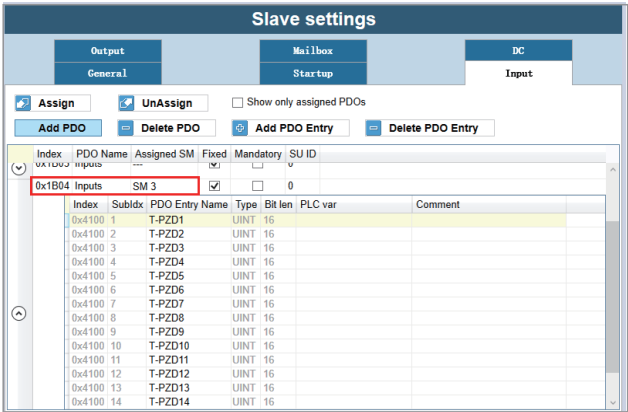
3.1.1.3 BN_EtherCAT_INV_Torque_V02模块使用

3.1.1.3.1 配置RPDO

打开output页面，取消其他PDO的同步管理输出配置SM2，并将1704添加到同步管理单元SM2中，针对博能EtherCAT驱动器，不需要向其添加参数，默认如下图所示：



打开input页面，选择1B04为同步管理输入配置SM3，针对博能EtherCAT伺服驱动器，不需要向其添加参数，默认如下图所示：



3.1.1.3.2 引脚说明

BN_EtherCAT_INV_Torque_V02功能块引脚说明如下表所示：

名称	范围/含义	数据类型	单位	描述
AxisIndex	EtherCat轴ID	UINT	\	0—15
bEnable	1:使能 0:未使能	BOOL	\	使能运行, 0: 轴停止, 1:轴运行
bMotorRotation	1:反向 0:正向	BOOL	\	电机旋转方向, 0: 正向, 1:反向

名称	范围/含义	数据类型	单位	描述
bFaultReset	1: 故障复位 0: 无故障	BOOL	上升沿触发	故障复位, 0: 复位无效, 1: 复位有效
TargetTorque	-32768 —32767	INT	\	目标转矩, 1000对应100.0%, 100.0%对应电机额定转矩
SpeedUpLimit	-32768 —32767	INT	\	速度上限, 1000对应100.0%, 100.0%对应电机额定转速
SpeedLowLimit	-32768 —32767	INT	\	速度下限, 1000对应100.0%, 100.0%对应电机额定转速
PZD3	用户自定义PZD3	WORD	\	\
bReadyOK	1: 开机已准备好 0: 开机未准备好	BOOL	\	开机准备好状态, 0: 开机未准备好, 1: 开机已准备好
bRunState	1: 运行 0: 停机	BOOL	\	运行状态, 0: 停机, 1: 运行
StateWord		WORD	\	状态字
bFault	1: 有故障 0: 无故障	BOOL	\	故障状态, 0: 无故障, 1: 有故障
FBError Code	功能块故障代码	WORD	\	功能块故障代码
AxisError Code	轴故障代码	WORD	\	驱动器故障代码
ActSpeed	实际速度	INT	指令脉冲数\ s	反馈速度, 基准为RefSpeed
ActCurrent	实际位置	UINT	指令脉冲数	输出电流, 4096对应电机额定电流
ActTorque	实际转矩	INT	电机额定转矩的百分比	输出转矩, 4096对应电机额定扭矩, 带符号

名称	范围/含义	数据类型	单位	描述
PZD6	用户自定义PZD6	WORD	\	\

AxisIndex轴名称根据拓扑依次为0—15，即与博能控制器EtherCat直接连接的为0轴，与0轴连接的为1轴，依次类推；

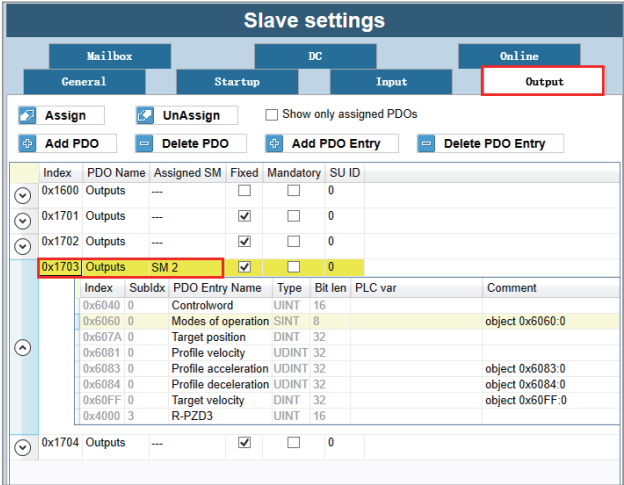
3.1.1.3.3 功能块故障说明

FBError Code	含义
16#0000	功能块无错误；
16#0001	未进入OP；
16#0002	AxisIndex 超过范围，或者未进入OP；
16#000A	PD0配置不正确或不能获取 StatusWord4100_01 / ControlWord4000_01的地址；
16#000B	PD0配置不正确或不能获取 ActVelocity4100_02 / TargetVelocity4000_02的地址；
16#000C	PD0配置不正确或不能获取 ActCurrent4100_03 / RPZD3_4000_03的地址；
16#000D	PD0配置不正确或不能获取 ActTorque4100_04 / TorqueUpLimit_4000_04的地址；
16#000E	PD0配置不正确或不能获取 DriveError4100_05 / TorqueLowLimit_4000_05的地址；
16#000F	PD0配置不正确或不能获取 TPZD6_4100_06的地址；
16#0010	PD0配置不正确或不能获取 SpeedUpLimit_4000_07的地址；
16#0011	PD0配置不正确或不能获取 SpeedLowLimit_4000_08的地址；
16#0012	PD0配置不正确或不能获取 OperationMode_4000_09的地址；
16#0020	当前轴已被使用

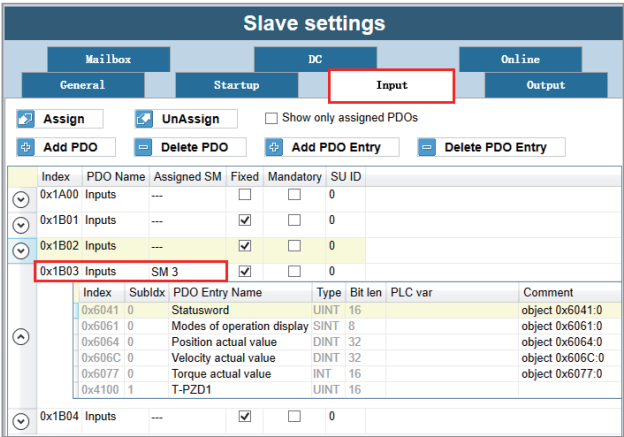
3.1.1.4 BN_EtherCAT_PP_PV模块使用

3.1.1.4.1 配置RPD0

打开output页面，取消其他PD0的同步管理输出配置SM2，并将1703添加到同步管理单元SM2中，针对博能EtherCAT伺服驱动器，不需要向其添加参数，默认如下图所示：



打开input页面，选择1B03为同步管理输入配置SM3，针对博能EtherCAT伺服驱动器，不需要向其添加参数，默认如下图所示：



3.1.1.4.2 引脚说明

BN EtherCat_PP_PV功能块引脚说明如下表所示：

名称	范围/含义	数据类型	单位	描述
AxisIndex	EtherCat轴ID	UINT	\	0—15
OperationMode	1:相对位置	INT	\	1:PP mode
	2:绝对位置			2:PP mode
	3:速度模式			3:PV mode
	4:主动回零			4:Hm mode
	5:设置当前位置为零点			5:Hm mode
	6:**暂无			
	7:速度点动			7:PP mode
	8:位置点动			8:PP mode
bAxisEnable	1:使能	BOOL	\	
	0:未使能			
bTaskCancel	1:无取消指令	BOOL	\	PP模式(OperationMode =1,2)下任务取消
	0:取消任务			
bAxisHalt	1:无暂停指令	BOOL	\	
	0:暂停任务			
bPositive	1:	BOOL	\	两BOOL量配合使用 OperationMode =3、4 当bPositive置1时，
	0:			
bNegative	1:	BOOL	\	bNegative置0时，主动回零取正向，速度模式正方向运动；
	0:			

名称	范围/含义	数据类型	单位	描述
				当bNegative置1时，bPositive置0时，主动回零取反向，速度模式反方向运动； 其余情况均取正方向
bJog1	1:置1时触发点动	BOOL	\	Jog1点动
	0:置0时取消点动			
bJog2	1:置1时触发点动	BOOL	\	Jog2点动
	0:置0时取消点动			
bAxisReset	1:故障复位	BOOL	上升沿触发	功能块无FBError时，可复位AxisError
	0:无故障			
bAxisExecute	1:触发指令	BOOL	上升沿触发	当OperationMode = 1、2、4、5时生效
	0:正常			
TargetPosition	-2147483647—— 2147483647	DINT	指令脉冲数	当OperationMode = 1、2时作为目标位置
TargetVel	-2147483647—— 2147483647	DITN	指令脉冲数/s	当OperationMode = 1、2、3时速度指令
ProfileAcc	0—— 4294967295	UDINT	指令脉冲数/s ²	当OperationMode = 1、2、3、7、8时加速度指令
ProfileDec	0—— 4294967295	UDINT	指令脉冲数/s ²	当OperationMode = 1、2、3、7、8时减速度指令
PZD3	用户自定义PZD3	WORD	\	\
bAxisEnableOK	1:轴已使能	BOOL	\	\
	0:轴未使能			

名称	范围/含义	数据类型	单位	描述
bTarget Reach	1:目标到达	BOOL	\	\
	0:未到达			
bAxis Error	1:轴有错误	BOOL	\	轴故障标志位
	0:轴无错误			
AxisErrorCode	轴错误代码	INT	\	\
bFB Error	1:功能块有错误	BOOL	\	功能块错误标志位
	0:功能块无错误			
FBError Code	功能块错误代码	UINT	\	\
ActVelocity	实际速度	DINT	指令脉冲数/s	\
ActPosition	实际位置	DINT	指令脉冲数	\
ActTorque	实际转矩	INT	电机额定转矩的百分比	\
ActCurrent	实际电流	INT	单位请参照驱动器	\
ActDiStatus	实际DI状态	UDINT	\	BIT0:负向限位(0:未到达, 1:已到达); BIT1:正向限位(0:未到达, 1:已到达); BIT2:回零开关(0:未到

名称	范围/含义	数据类型	单位	描述
				达, 1:已到达); BIT3-15:NA; BIT16-31:反映实际DI端子状态
ActMode	当前模式	INT	\	\
PZD1	用户自定义PZD1	WORD	\	\
PZD2	用户自定义PZD2	WORD	\	\

模块介绍:

- 1) AxisIndex轴名称根据拓扑依次为0—15, 即与博能控制器EtherCat直接连接的为0轴, 与0轴连接的为1轴, 依次类推;
- 2) 加速度, 单位为指令脉冲/s², 如果电子齿轮为10000脉冲对应电机一圈, 则加速度10000指令脉冲/s² = 1r/s²=60rpm/s, 相当于1s加速60转, 这个加速比较缓慢。博能小功率伺服的空载加速能力大概为2500r/s, 有些伺服可以做到5000r/s; 减速度, 同上;
- 3) 轨迹规划速度, 单位为指令脉冲/s, 同理如果电子齿轮还是按上面计算, 则10000指令脉冲/s=1r/s=60rpm, 这个值是比较小的;

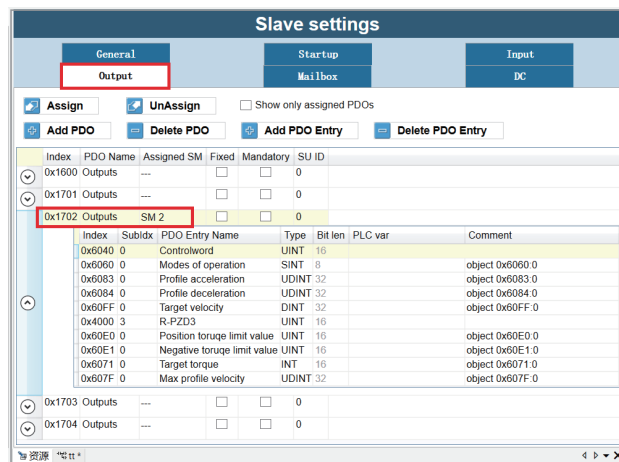
3.1.1.4.3 功能块故障说明

FBEr- rorCode	含义
0	无错误;
1	未进入OP;
2	AxisIndex 超过范围, 或者未进入OP;
3	输入操作模式错误;
7	sysECAT_Sdo_Write2功能块故障(可能是以下两种原因造成的): (1):在EtherCAT主站停止或恢复的过程中, 调用sdo_write2协议栈报错; (2):在非周期通信中和在线功能中使用, 读取寄存器出错会报这个错误;
9	PDO配置不正确或不能获取 AxisEr- rorCode603F 的地址;
10	PDO配置不正确或不能获取 StatusWord6041 / ControlWord6040 的地址;
11	PDO配置不正确或不能获取 DisMode0- f0P6061 / Mode0f0P6060 的地址;
12	PDO配置不正确或不能获取 ActPosi- tion6064 / TargetPosition607A 的地址;
13	PDO配置不正确或不能获取 ActVeloci- ty606C / ProfileVelocity6081 的地 址;
14	PDO配置不正确或不能获取 Act- Torque6077 / ProfileAcceleration6083 的地址;
15	PDO配置不正确或不能获取 ActCur- rent6078 / ProfileDeceleration6084 的地址;
16	PDO配置不正确或不能获取 ActDiSta- tus60FD / TargetVelocity60FF 的地 址;
32	当前轴已被使用

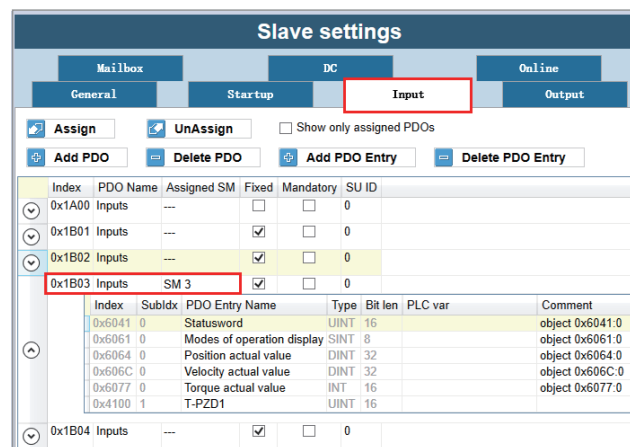
3.1.1.5 BN_EtherCAT_PV_Speed模块使用

3.1.1.5.1 配置RPDO

打开output页面, 取消其他PDO的同步管理输出配置SM2, 并将1702添加到同步管理单元SM2中, 针对博能EtherCAT伺服驱动器, 不需要向其添加参数, 默认如下图所示:



打开input页面, 选择1B03为同步管理输入配置SM3, 针对博能EtherCAT伺服驱动器, 不需要向其添加参数, 默认如下图所示:



3.1.1.5.2 引脚说明

BN_EtherCat_PV_Speed功能块引脚说明如下表所示：

名称	范围/含义	数据类型	单位	描述
AxisIndex	EtherCat轴ID	UINT	\	0—15
bAxisEnable	1:使能 0:未使能	BOOL	\	
bQuickStop	1:快速停机有效 0:快速停机无效	BOOL	\	快速停机
bMotorRotation	1:反向 0:正向	BOOL	\	电机旋转方向
bAxisReset	1:故障复位 0:无故障	BOOL	上升沿触发	功能块无FBError时，可复位AxisError
TargetSpeed	-32768 — 32767	INT	r/min	目标速度
RefSpeed	0 — 65535	UINT	目标速度	速度限制及加减速时间参考值：默认为电机额定转速
AccTime	\	REAL	s	加速时间
DecTime	\	REAL	s	减速时间
TorqueUpLimit	-32768 — 32767	INT	\	转矩上限，1000对应100.0%，100.0%对应电机额定转矩
TorqueLowLimit	-32768 — 32767	INT	\	转矩下限，1000对应100.0%，100.0%对应电机额定转矩
EncoderResolution	0—4294967295	UDINT	\	编码器分辨率，请输入对应电机编码器的分辨率

名称	范围/含义	数据类型	单位	描述
PZD3	用户自定义PZD3	WORD	\	\
bAxisEnableOK	1:轴已使能 0:轴未使能	BOOL	\	\
bFault	1:功能块有错误 0:功能块无错误	BOOL	\	功能块错误标志位
FBErrorCode	\	UINT	\	功能块错误代码
AxisErrorCode	\	WORD	\	轴错误代码
ActVelocity	\	DINT	r/min	实际速度
ActTorque	\	INT	电机额定转矩的百分比	实际转矩
ActCurrent	\	INT	单位请参考驱动器	实际电流
ActPosition	\	DINT	指令脉冲数，经过电子齿轮比换算	实际位置
				实际DI状态 BIT0:负向限位(0:未到达，1:已到达)； BIT1:正向限位(0:未到达，1:已到达)；

名称	范围/含义	数据类型	单位	描述
ActDi Status	\	UDINT	\	BIT2:回零开关(0:未 到达, 1:已到达); BIT3-15:NA; BIT16-31:反映实际DI 端子状态
PZD1	用户自定义PZD1	WORD	\	\
PZD2	用户自定义PZD2	WORD	\	\

模块介绍:

- 1) AxisIndex轴名称根据拓扑依次为0—15，即与博能控制器EtherCat直接连接的为0轴，与0轴连接的为1轴，依次类推；
- 2) 用户需要输入伺服编码器分辨率，默认8388608，需要输入参考速度（电机额定转速），默认3000，若输入错误会导致输入的加减速时间 and 目标速度无法发挥作用，输出实际速度与实际不符。
- 3) PV块运行在PV模式，此模式下，块使能引脚输入高电平，则按照用户给定限制和目标值运行，使能引脚输入低电平，则停止运行。

3. 1. 1. 5. 3 功能块故障说明

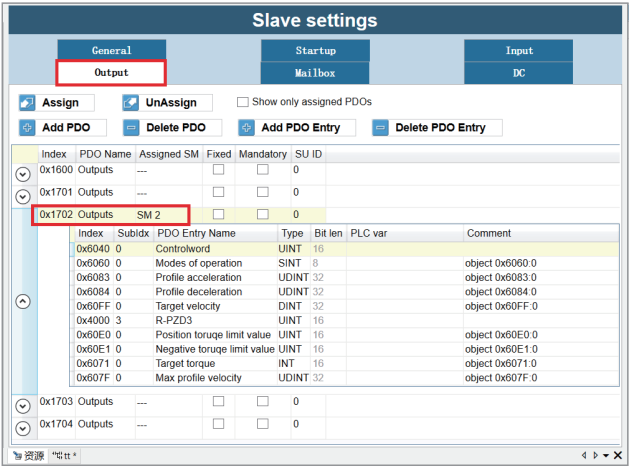
FBEr- rorCode	含义
0	无错误；
1	未进入OP；
2	AxisIndex 超过范围，或者未进入OP；
256	参考速度未设置；
7	sysECAT_Sdo_Write2功能块故障(可能是以下两种原因造成的): (1):在EtherCAT主站停止或恢复的过程中，调用sdo_write2协议栈报错； (2):在非周期通信中和在线功能中使用，读取寄存器出错会报这个错误；
9	PD0配置不正确或不能获取 AxisEr- rorCode603F 的地址；
10	PD0配置不正确或不能获取 StatusWord6041 / ControlWord6040 的地址；
11	PD0配置不正确或不能获取 DisMode0- f0P6061 / Mode0f0P6060 的地址；
12	PD0配置不正确或不能获取 ActPosi- tion6064 / TargetPosition607A 的地址；
13	PD0配置不正确或不能获取 ActVeloci- ty606C / ProfileVelocity6081 的地址；
14	PD0配置不正确或不能获取 ActTorque6077 / ProfileAcceleration6083 的地址；
15	PD0配置不正确或不能获取 ActCurrent6078 / ProfileDeceleration6084 的地址；
16	PD0配置不正确或不能获取 ActDiSta- tus60FD / TargetVelocity60FF 的地址；
17	PD0配置不正确或不能获取 Posi- tiveToruqeLimit60E0 的地址；
18	PD0配置不正确或不能获取 Nega- tiveToruqeLimit60E1 的地址；

FBErrorCode	含义
19	PDO配置不正确或不能获取 Target-Torque6071 的地址;
20	PDO配置不正确或不能获取 MaxProfileVelocity607F 的地址;
32	当前轴已被使用

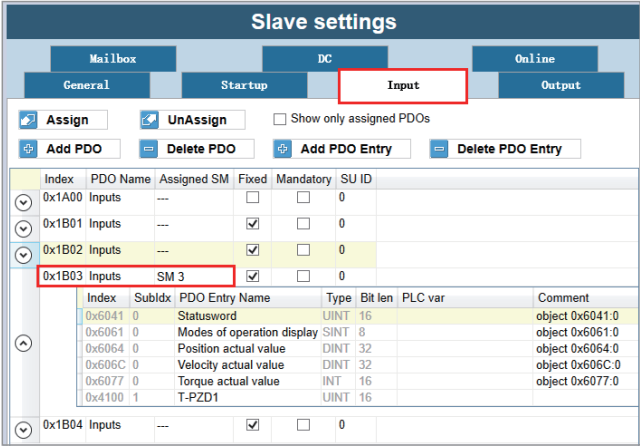
3.1.1.6 BN_EtherCAT_PT_Torque模块使用

3.1.1.6.1 配置RPDO

打开output页面，取消其他PDO的同步管理输出配置SM2，并将1702添加到同步管理单元SM2中，针对博能EtherCAT伺服驱动器，不需要向其添加参数，默认如下图所示：



打开input页面，选择1B03为同步管理输入配置SM3，针对博能EtherCAT伺服驱动器，不需要向其添加参数，默认如下图所示：



3.1.1.6.2 引脚说明

BN_EtherCat_PT_Torque功能块引脚说明如下表所示：

名称	范围/含义	数据类型	单位	描述
AxisIndex	EtherCat轴ID	UINT	\	0—15
bAxisEnable	1:使能 0:未使能	BOOL	\	
bMotorRotation	1:反向 0:正向	BOOL	\	
bAxisReset	1:故障复位 0:无故障	BOOL	上升沿触发	功能块无FBError时，可复位AxisError
TargetTorque	-32768 —— 32767	INT	\	目标转矩，1000对应100.0%，100.0%对应电机额定转矩
MaxTorque	-32768 —— 32767	INT	\	最大转矩，目标转矩受此参数限制，1000表示100.0%，最大3000
SpeedLimit	0—— 65535	UINT	r/min	速度限制 **单位：r/min

名称	范围/含义	数据类型	单位	描述
Encoder Resolution	0—4294967295	UDINT	\	编码器分辨率，请输入对应电机编码器的分辨率
PZD3	用户自定义PZD3	WORD	\	\
bAxis Enable OK	1:轴已使能 0:轴未使能	BOOL	\	\
bFault	1:功能块有错误 0:功能块无错误	BOOL	\	功能块错误标志位
FBError Code	\	UINT	\	功能块错误代码
AxisError Code	\	WORD	\	轴错误代码
ActVelocity	\	DINT	r/min	实际速度
ActTorque	\	INT	电机额定转矩的百分比	实际转矩
ActCurrent	\	INT	单位请参考驱动器	实际电流
ActPosition	\		指令脉冲数，经过电子齿轮比换算	实际位置
				实际DI状态 BIT0:负向限位(0:未

名称	范围/含义	数据类型	单位	描述
ActDi Status	\	UDINT	\	到达，1:已到达)； BIT1:正向限位(0:未到达，1:已到达)； BIT2:回零开关(0:未到达，1:已到达)； BIT3-15:NA； BIT16-31:反映实际DI端子状态
PZD1	用户自定义PZD1	WORD	\	\
PZD2	用户自定义PZD2	WORD	\	\

模块介绍：

- 1) AxisIndex轴名称根据拓扑依次为0—15，即与博能控制器EtherCat直接连接的为0轴，与0轴连接的为1轴，依次类推；
- 2) 用户需要输入伺服编码器分辨率，默认8388608，此引脚输入错误会导致输入速度限制和输出实际速度与实际不符。
- 3) PT块运行在PT模式，此模式下，块使能引脚输入高电平，则按照用户给定限制和目标值运行，使能引脚输入低电平，则停止运行。

3. 1. 1. 6. 3 功能块故障说明

FBError Code	含义
0	无错误；
1	未进入OP；
2	AxisIndex 超过范围，或者未进入OP；
7	sysECAT_Sdo_Write2功能块故障(可能是以下两种原因造成的)： (1):在EtherCAT主站停止或恢复的过程中，调用sdo_write2协议栈报错； (2):在非周期通信中和在线功能中使用，

FBErrorCode	含义
	读取寄存器出错会报这个错误；
9	PD0配置不正确或不能获取 AxisErrorCode603F 的地址；
10	PD0配置不正确或不能获取 StatusWord6041 / ControlWord6040 的地址；
11	PD0配置不正确或不能获取 DisMode0f0P6061 / Mode0f0P6060 的地址；
12	PD0配置不正确或不能获取 ActPosition6064 / TargetPosition607A 的地址；
13	PD0配置不正确或不能获取 ActVelocity606C / ProfileVelocity6081 的地址；
14	PD0配置不正确或不能获取 ActTorque6077 / ProfileAcceleration6083 的地址；
15	PD0配置不正确或不能获取 ActCurrent6078 / ProfileDeceleration6084 的地址；
16	PD0配置不正确或不能获取 ActDiStatus60FD / TargetVelocity60FF 的地址；
17	PD0配置不正确或不能获取 PositiveTorqueLimit60E0 的地址；
18	PD0配置不正确或不能获取 NegativeTorqueLimit60E1 的地址；
19	PD0配置不正确或不能获取 TargetTorque6071 的地址；
20	PD0配置不正确或不能获取 MaxProfileVelocity607F 的地址；
32	当前轴已被使用

3.1.2 BN_Basic

3.1.2.1 BN_PID

PID控制器：PID控制器可实现温度，压力，流量，频率等的闭环调节功能，一般需要配合AI和AO模块进行使用，AI模块及线性矫正模块对反馈值进行标

定，输出值通过AO模块输出。默认PID参数比较小，请根据实际情况进行调整。

3.1.2.1.1 引脚说明

Name	DataType	Explain
set_point	REAL	Controller setpoint
feedback	REAL	Signal actual value (feedback)
Kp	REAL	Proportional gain
Ki	REAL	Integral gain
Kd	REAL	Derivative gain
rst	BOOL	Controller reset
OutMax	REAL	Output signal upper limit
OutMin	REAL	Output signal lower limit
Magnification	BOOL	Adjusting the magnification
out	REAL	Output signal
limitFlag	BOOL	In-limit output status

3.1.2.2 BN_Ramp

BN_Ramp斜坡函数，用户可根据需求实现需要的斜坡输出。

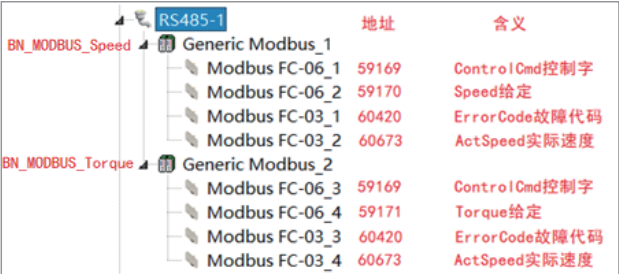
3.1.2.2.1 引脚说明

Name	DataType	Explain
Input	REAL	输入给定值
RefValue	REAL	参考值
RiseTime	UDINT	上升时间 **单位：ms
FallTime	UDINT	下降时间 **单位：ms
InputMaxLimit	REAL	给定上限值
InputMinLimit	REAL	给定下限值
ResetValue	REAL	RAMP复位后的值
Reset	BOOL	复位:TRUE:将输出复位成0；

Name	DataType	Explain
		FALSE:无复位;
bFBError	BOOL	功能块输入错误
FBError Code	WORD	功能块故障码 FBErrorCode: 16#0000:功能块无错误; 16#0001:上升时间或下降时间 设置错误; 16#0002:参考值设置错误;
Output	REAL	输出值

3. 1. 3 BN_Modbus

BN_Modbus库包含BN_MODBUS_Speed和BN_MODBUS_Torque两功能块，用于控制变频器从站
需要配置Modbus RTU通信，X3000系列控制器做主站，变频器做从站，配置可参考下图



3. 1. 3. 1 BN_MODBUS_Speed

Name	DataType	Explain
bEnable	BOOL	使能运行, 0:轴停止, 1:轴运行
bStopMode	BOOL	停机模式, 0:减速停机, 1:自由停机
bJog	BOOL	点动控制
bMotor Rotation	BOOL	电机旋转方向, 0:正向, 1:反向
bFault Reset	BOOL	故障复位, 0:复位无效, 1:复位有效
Slave ErrorCode	WORD	轴反馈故障代码

Name	DataType	Explain
SlaveSpeed ActPer	WORD	轴反馈速度标么值
SpeedSet	INT	目标速度, 基准为SpeedRef
SpeedRef	UINT	速度值的参考基准
bFault	BOOL	故障状态, 0:无故障, 1:有故障
FBError Code	WORD	功能块故障代码
Drive ErrorCode	WORD	驱动器故障代码
Slave SpeedAct	INT	从轴反馈的速度值, 参考基准为 SpeedRef
SpeedSet- ToSlave	WORD	给从站轴的速度指令
CtrlWord ToSlave	WORD	给从轴的控制命令, 1:运行, 2:点 动, 3:自由停机, 4:减速停机, 5: 故障复位

使用时，需要将60420故障代码输入到SlaveErrorCode，60673实际速度输入到SlaveSpeedActPer
输出SpeedSetToSlave到59170速度给定，输出CtrlWordToSlave到59169控制字给定

3. 1. 3. 2 BN_MODEBUS_Torque

Name	DataType	Explain
bEnable	BOOL	使能运行, 0:轴停止, 1:轴运行
bMotor Rotation	BOOL	电机旋转方向, 0:正向, 1:反向
bFault Reset	BOOL	故障复位, 0:复位无效, 1:复位有效
Slave ErrorCode	WORD	轴反馈故障代码
SlaveSpeed ActPer	WORD	轴反馈速度标么值
SpeedRef	UINT	速度反馈值的参考基准

Name	DataType	Explain
TorqueSet	INT	目标转矩, 1000对应100.0%, 100.0%对应电机额定转矩
bFault	BOOL	故障状态, 0:无故障, 1:有故障
FBError Code	WORD	功能块故障代码
Drive ErrorCode	WORD	驱动器故障代码
Slave SpeedAct	INT	从轴反馈的速度值, 参考基准为SpeedRef
Torque SetToSlave	WORD	给从站轴的速度指令
CtrlWord ToSlave	WORD	给从轴的控制命令, 1:运行, 2:点动, 3:自由停机, 4:减速停机, 5:故障复位

使用时, 需要将60420故障代码输入到SlaveErrorCode, 60673实际速度输入到SlaveSpeedActPer, 输出TorqueSetToSlave到59171转矩给定, 输出CtrlWordToSlave到59169控制字给定

3.2 BonengEmbedLogicLib

博能嵌入式库包含工程师常用的一些函数和功能块。

3.2.1 ArithmeticOperationClassExtLib

3.2.1.1 GetFraction

GetFraction函数用于获取REAL类型数据的小数部分。

函数输入部分:

名称	类型	描述
DATA_REAL	REAL	Real类型输入数据

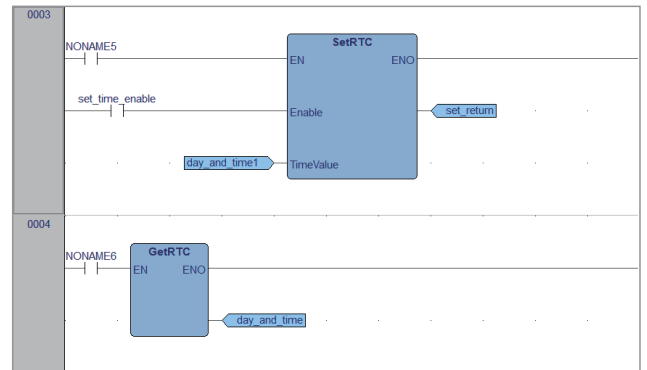
3.2.2 ClockClassExtLib

3.2.2.1 设置实时时间与获取实时时间

首先在库管理向导中找到BonengEmbedLogicLib, 展开ClockClassExtLib文件夹可看到SetRTC 和 GetRTC



将这两个函数都添加到程序中, 添加完成后如下图所示:



在SetRTC中EN引脚是该函数块的使能, Enable是该函数的写入数据的触发引脚, 强调: (该引脚需要上升沿触发), TimeValue是该函数的时间设置引脚, 该引脚的变量类型是TIME_AND_DAY, 该引脚可设置如→1920-01-01-00:00:00格式的时间, 该时间的设置一般来说是当前的北京时间, 该函数的返回值不用关注,

在GetRTC中EN引脚是该函数块的使能, 该函数的返回值是一个TIME_AND_DAY类型的数据, 当使能该函数后可以在输出引脚看到时间在变化。

3.2.3 DataConversionClassExtLib

3.2.3.1 DATA16_SWAP

DATA16_SWAP函数用于交换16位输入数据的高低字节。

名称	类型	描述
DATA_IN	UINT	要进行高低字节交换的16位输入数据

3.2.3.2 DATA2BIT

DATA2BIT函数用于选定16位输入数据的对应BIT位。

名称	类型	描述
DATA_IN	UINT	16位输入数据
BIT_NUM	UINT	选取的BIT位，范围0-15

3.2.3.3 DATA32_SWAP

DATA32_SWAP函数用于交换32位输入数据的高低字节。

名称	类型	描述
DATA_IN	UINT	要进行高低字节交换的32位输入数据

3.2.3.4 REAL_ENDIAN_SWAP

REAL_ENDIAN_SWAP函数用于交换32位REAL类型数据的大小端，用于转换西门子PLC传输的REAL类型数据。

名称	类型	描述
DATA_IN	REAL	要进行大小端交换的REAL类型输入数据

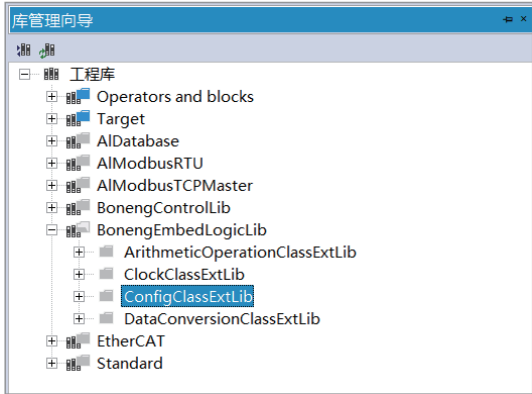
3.2.4 ConfigClassExtLib

3.2.4.1 模拟量设置

3.2.4.1.1 AI_ModeConfig

模拟量输入电压电流的切换

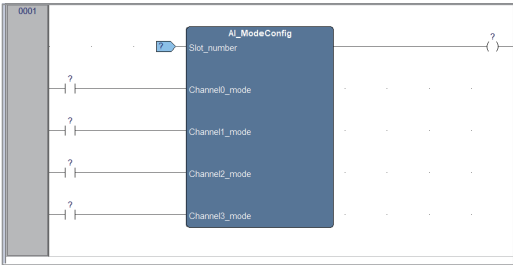
首先找到BonengEmbedLogicLib，点击加号展开文件夹找到ConfigClassExtLib点击加号继续展开文件夹



找到如右图所示函数 AI_ModeConfig 将该函数添加到程序中，可以使用5种语言进行调用，在这里仅演示使用LD语言进行调用；首先建立一个空白LD语言程序如图所示；



将上述蓝色光标AI_ModeConfig拖拽到LD程序的触点上即可添加。添加成功后如下图所示；



该函数共有5个输入参数，需要设置对应的槽号，和通道的模式即可切换电压，电流之间的输入检测转化，如果不调用该函数，默认电压模式输入。
Slot_number:UINT; (* AI04扩展模块的槽号，在背板组态图中查询，范围1-15 *)
Channel0_mode:BOOL; (* AI0模式选择，0：电压模式，1：电流模式 *)
Channel1_mode:BOOL; (* AI1模式选择，0：电压模式，1：电流模式 *)

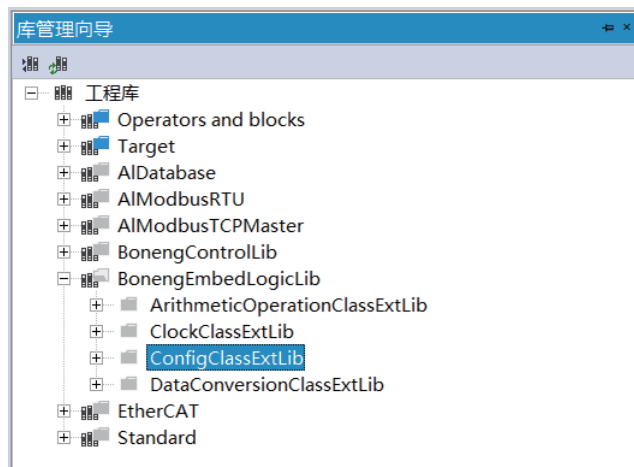
Channel2_mode:BOOL; (* AI2模式选择, 0: 电压模式, 1: 电流模式 *)

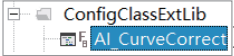
Channel3_mode:BOOL; (* AI3模式选择, 0: 电压模式, 1: 电流模式 *)

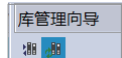
3.2.4.1.2 AI_CurveCorrect

用户自定义模拟量输入曲线

首先找到BonengEmbedLogicLib, 点击加号展开文件夹找到ConfigClassExtLib, 点击加号继续展开文件夹



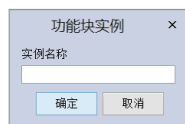
找到如右图所示功能块  如果没有显示该功能块, 请重新安装最新的库文件, 具体操作方法看2.1节, 点击库文件的刷新按钮如右图,



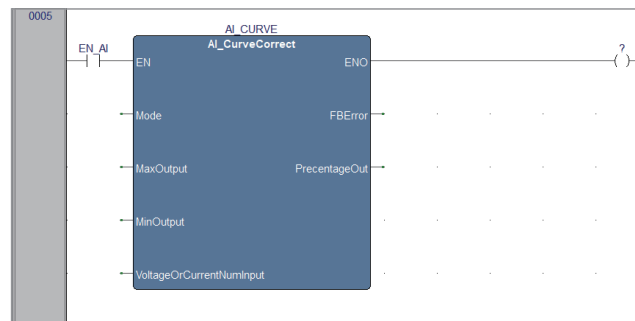
刷新库文件。将AI_CurveCorrectr

拖拽到程序中, 具体操作方法见2.10.1.1。与添加函数不同的是, 当添加功能块到程序中时, 需要设

置一个实例名称, 如图所示:



添加完成后, 才可以调用该功能块。添加完成如下图所示:



该功能块有四个输入的参数:

Mode:BOOL; (* 模式选择, 0: 电压型, 1: 电流型 *)

MaxOutput:INT; (* 输出的最大值 *) (最大值应该大于最小值, 否则会报错)

MinOutput:INT; (* 输出的最小值 *)

VoltageOrCurrentNumInput:UINT; (* 电压或电流模拟量输入值 *)

电压的输入值不应超过10000, 电流的输入值不应超过20000。

该功能块有两个输出的参数:

FBError:UINT; (* 功能块错误码 *) 当输入的值不符合要求时会报错。

PercentageOut:DINT; (* 特性曲线数值输出 *)

3.2.4.2 数字量设置

3.2.4.2.1 Control_IO_By_Word

Control_IO_By_Word功能块可选择对应槽号, 通过输入Word控制开关量输出。

名称	类型	描述
Execute	BOOL	执行按字控制IO
Slot_Index	USINT	扩展DQ的槽号范围: 1-15
Word_Data	WORD	输入的控制字
ERROR	USINT	错误代码 bit0=1: 扩展槽号超出范围, bit1=1: 输入控制字超出范围

3.2.4.2.2 Reset_IO_By_Index

Reset_IO_By_Index功能块可选择对应槽号，并选择开始和结束点位，执行对应开关量的复位。

名称	类型	描述
Execute	BOOL	执行复位功能块
Slot_Index	USINT	扩展DQ的槽号范围：1-15
Start_IO	USINT	开始点位 ， 范围0-15
End_IO	USINT	结束点位， 范围0-15
ERROR	USINT	错误代码 bit0=1 槽号超出范围，bit1=1开始点位超出范围，bit2=1结束点位超出范围，bit3=1开始点位大于结束点位

3.2.4.2.3 Set_IO_By_Index

Set_IO_By_Index功能块可选择对应槽号，并选择开始和结束点位，执行对应开关量的置位。

名称	类型	描述
Execute	BOOL	执行置位功能块
Slot_Index	USINT	扩展DQ的槽号范围：1-15
Start_IO	USINT	开始点位 ， 范围0-15
End_IO	USINT	结束点位， 范围0-15
ERROR	USINT	错误代码 bit0=1 槽号超出范围，bit1=1开始点位超出范围，bit2=1结束点位超出范围，bit3=1开始点位大于结束点位

3.3 BonengMachiningProcessLib

3.3.1 BN_TensionControl

BN_TensionControl库是卷曲工艺库。

3.3.1.1 BN_TensionTaper

BN_TensionTaper张力锥度功能块，输入为设定张

力，输出为经过锥度处理的张力值，锥度模式支持抛物线和线性两种模式，通过锥度衰减系数设定张力衰减的比例。

名称	类型	描述
TensionSet	REAL	收卷张力， 单位 (N)
ActDiameterOfRoller	UINT	收卷辊实际直径， 最小\50mm， 最大2000mm
TaperCoff	REAL	锥度衰减系数， 此值越大， 锥度越大， 默认0， 范围0.0~1.0
TaperMode	INT	锥度模式， 0:抛物线锥度， 1: 线性锥度， 默认0
TaperDiameterStart	UINT	锥度起始直径， 单位 (mm)
TaperDiameterMax	UINT	锥度最大直径， 单位 (mm)
bError	BOOL	模块错误标志
FBError Code	UINT	模块错误代码
TensionBy Taper	REAL	经过锥度处理的张力值, 单位 (Nm)

3.3.1.2 BN_TractionSpeed

BN_TractionSpeed牵引轴速度控制功能块，根据设定线速度，牵引辊直径，及减速比来计算牵引电机转速。牵引轴可以是moudbus速度轴，也可以是EtherCAT速度轴。

名称	类型	描述
LineVelocitySet	REAL	线速度设定值 (m/min)
TractionMotorSpeed Act	INT	牵引电机转速 (rpm)
DiameterOfTraction Roller	UINT	牵引辊直径, 最小\10mm, 最大1000mm
GearRatio	REAL	齿轮速比， 默认为1.0

名称	类型	描述
bError	BOOL	模块错误标志
FBErrorCode	UINT	模块错误代码
LineVelocityAct	REAL	牵引辊线速度实际值, m/min
TractionMotorSpeed Set	INT	牵引辊电机转速 (rpm)

3.3.1.3 BN_Unwinder_CTorque

BN_Unwinder_CTorque放卷辊闭环张力转矩控制功能块, 和开环张力转矩控制的区别, 在于可通过PID进行张力闭环控制, 要求设备支持张力检测和反馈。此模块建议在fast任务中执行, 否则PID很难调试。

名称	类型	描述
LineVelocityAct	REAL	放卷线速度 (m/min)
UnwinderMotorSpeedAct	INT	放卷电机转速 (rpm)
DiameterOfFullRoller	UINT	放卷辊满辊直径, 最小10mm, 最大1000mm
GearRatio	REAL	齿轮速比, 默认为1.0
UnwinderMotorRated Torque	REAL	放卷辊电机额定扭矩: $9550 \cdot P/n$, (Nm)
FrictionCompensation	INT	放卷辊摩擦补偿, 基值为电机额定扭矩, (0.1%)
TensionSet	REAL	放卷设定张力 (N), 也是PID参考值
TensionAct	REAL	张力实际反馈值, PID反馈值
Kp	REAL	PID调节器比例系数
Ki	REAL	PID调节器积分系数
Kd	REAL	PID调节器微分系数
PidOutLimit	REAL	PID调节器输出限制, rpm
PidReset	BOOL	PID调节结果复位

名称	类型	描述
bError	BOOL	模块错误标志
FBErrorCode	UINT	模块错误代码
DiameterOfRollerAct	REAL	放卷辊实时卷径, 直径, (mm)
PidOut	INT	摆杆张力PID调节输出值, 电机额定扭矩百分比 (0.1%)
UnwinderMotorSpeed Set	INT	放卷辊电机转速 (rpm)
UnwinderMotor TorqueSet	INT	放卷辊电机转矩, 1000表示100.0%

3.3.1.4 BN_Unwinder_OTorque

BN_Unwinder_OTorque放卷辊开环张力转矩控制功能块, 用于设备无张力反馈的场合, 一般张力要求不高的场合, 无PID闭环张力调节。

名称	类型	描述
TensionSet	REAL	放卷张力 (N)
LineVelocityAct	REAL	放卷线速度 (m/min)
UnwinderMotorSpeedAct	INT	放卷电机转速 (rpm)
DiameterOfFullRoller	UINT	放卷辊满辊直径, 最小10mm, 最大1000mm
GearRatio	REAL	齿轮速比, 默认为1.0
UnwinderMotorRated Torque	REAL	放卷辊电机额定扭矩: $9550 \cdot P/n$, (Nm)
FrictionCompensation	INT	放卷辊摩擦补偿, 基值为电机额定扭矩, (0.1%)
MassOfFullRoller	REAL	放卷辊满辊时的重量 (kg), 用于加减速力矩补偿
DensityRadioOfMaterial	REAL	方卷材料和辊筒的密度比例, 范围0.0~1.0

名称	类型	描述
bError	BOOL	模块错误标志
FBErrorCode	UINT	模块错误代码
DiameterOfRollerAct	REAL	放卷辊实时卷径, 直径, (mm)
UnwinderMotorSpeedSet	INT	放卷辊电机转速 (rpm)
UnwinderMotorTorqueSet	INT	放卷辊电机转矩, 1000 表示100.0%

3.3.1.5 BN_Winder_CTorque

BN_Winder_CTorque收卷辊闭环张力转矩控制功能块, 和开环张力转矩控制的区别, 在于可通过PID进行张力闭环控制, 要求设备支持张力检测和反馈。此模块建议在fast任务中执行, 否则PID很难调试。

名称	类型	描述
LineVelocityAct	REAL	收卷线速度 (m/min)
WinderMotorSpeedAct	INT	收卷电机转速 (rpm)
DiameterOfEmptyRoller	UINT	收卷辊初始直径, 最小10mm, 最大1000mm
GearRatio	REAL	齿轮速比, 默认为1.0
WinderMotorRatedTorque	REAL	收卷辊电机额定扭矩: $9550 \cdot P/n$, (Nm)
FrictionCompensation	INT	收卷辊摩擦补偿, 基值为电机额定扭矩, (0.1%)
TensionSet	REAL	收卷设定张力(N), 也是PID参考值
TensionAct	REAL	张力实际反馈值, PID反馈值
Kp	REAL	PID调节器比例系数
Ki	REAL	PID调节器积分系数
Kd	REAL	PID调节器微分系数
PidOutLimit	REAL	PID调节器输出限制, rpm

名称	类型	描述
PidReset	BOOL	PID调节结果复位
bError	BOOL	模块错误标志
FBErrorCode	UINT	模块错误代码
DiameterOfRollerAct	REAL	放卷辊实时卷径, 直径, (mm)
PidOut	INT	摆杆张力PID调节输出值, 电机额定扭矩百分比 (0.1%)
WinderMotorSpeedSet	INT	放卷辊电机转速 (rpm)
WinderMotorTorqueSet	INT	收卷辊电机转矩, 1000 表示100.0%

3.3.1.6 BN_Winder_OTorque

BN_Winder_OTorque收卷辊开环张力转矩控制功能块, 用于设备无张力反馈的场合, 一般张力要求不高的场合, 无PID闭环张力调节。

名称	类型	描述
TensionSet	REAL	收卷张力(N)
LineVelocityAct	REAL	收卷线速度 (m/min)
WinderMotorSpeedAct	INT	收卷电机转速 (rpm)
DiameterOfEmptyRoller	UINT	收卷辊初始直径, 最小10mm, 最大1000mm
GearRatio	REAL	齿轮速比, 默认为1.0
WinderMotorRatedTorque	REAL	收卷辊电机额定扭矩: $9550 \cdot P/n$, (Nm)
FrictionCompensation	INT	收卷辊摩擦补偿, 基值为电机额定扭矩, (0.1%)
MassOfEmptyRoller	REAL	收卷辊空辊时的重量(kg), 用于加减速度矩补偿
DensityRadioOfMaterial	REAL	收卷材料和辊筒的密度比例, 范围0.0~1.0

名称	类型	描述
bError	BOOL	模块错误标志
FBErrorCode	UINT	模块错误代码
DiameterOfRollerAct	REAL	放卷辊实时卷径, 直径, (mm)
WinderMotorSpeedSet	INT	放卷辊电机转速 (rpm)
WinderMotorTorque Set	INT	收卷辊电机转矩, 1000表示100. 0%

3.3.1.7 BN_WinderUnwinderSpeed

BN_WinderUnwinderSpeed收卷, 放卷辊闭环张力速度控制功能块, 要求设备有张力检测和反馈, 通过PID闭环张力调节, 使收放卷辊线速度保持恒定来控制张力恒定。此模块建议在fast任务中执行, 否则PID很难调试。

名称	类型	描述
LineVelocityAct	REAL	收卷线速度 (m/min)
WinderMotorSpeedAct	INT	收卷电机转速 (rpm)
DiameterOfEmptyRoller	UINT	收卷辊初始直径, 最小10mm, 最大1000mm
GearRatio	REAL	齿轮速比, 默认为1. 0
TensionSet	REAL	张力设定值, PID参考值
TensionAct	REAL	张力反馈值, PID反馈值
Kp	REAL	PID调节器比例系数
Ki	REAL	PID调节器积分系数
Kd	REAL	PID调节器微分系数
PidOutLimit	REAL	PID调节器输出限制, rpm
PidReset	BOOL	PID调节结果复位
bError	BOOL	模块错误标志
FBErrorCode	UINT	模块错误代码
DiameterOfRollerAct	REAL	放卷辊实时卷径, 直径, (mm)
PidOut	INT	摆杆张力PID调节输出值 (rpm)

名称	类型	描述
WinderMotorSpeedSet	INT	放卷辊电机转速 (rpm)

3.4 MotionControl轴控

3.4.1 MotionControl库介绍

MotionControl库包含博能自主研发的MC功能块, 符合PLCOpen规范, 实现了基础指令和主从指令部分

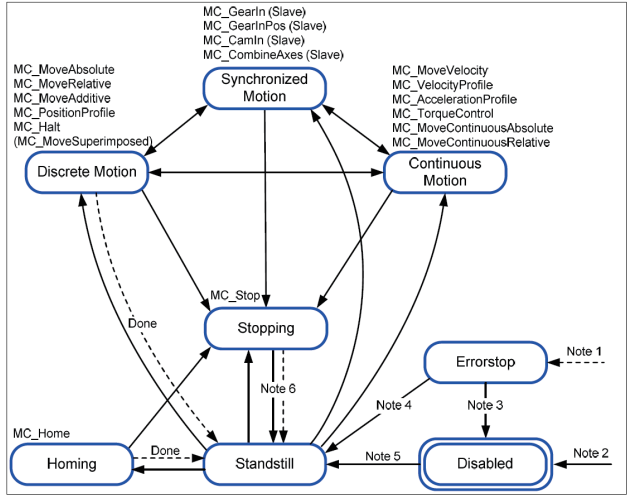
3.4.1.1 指令列表

运动控制基本指令包含如下:

指令		功能
基础指令	MC_Power	使能控制指令
	MC_Home	原点回归指令
	MC_Stop	停止指令
	MC_Halt	暂停指令
	MC_MoveAbsolute	绝对定位指令
	MC_MoveRelative	相对定位指令
	MC_MoveAddtive	增量定位
	MC_MoveSuperImposed	运动叠加指令
	MC_HaltSuperImposed	暂停叠加运动指令
	MC_MoveVelocity	速度指令
	MC_MoveContinuous Absolute	绝对定位后持续运动
	MC_MoveContinuous Relative	相对定位后持续运动
	MC_SetPosition	设置定位指令
	MC_ReadActualPosition	读取实际位置指令
	MC_ReadActualVelocity	读取实际速度指令
	MC_ReadActualTorque	读取实际力矩指令
	MC_ReadStatus	读取轴PLCOpen状态机
	MC_ReadMotionState	读取当前轴运动状态
	MC_ReadAxisInfo	读取轴限位/通信等信息
	MC_ReadAxisError	读取轴故障指令

指令		功能
	MC_Reset	复位故障指令
	MC_DigitalCamSwitch	设置指定位置上的开关信号量
	MC_TouchProbe	探针指令
	MC_AbortTrigger	打断指令
电子 凸轮 & 齿轮	MC_CamTableSelect	选取凸轮表输出可供MC_CamIn适用的CamID
	MC_CamIn	凸轮同步指令
	MC_CamOut	退出凸轮同步指令
	MC_GearIn	齿轮同步指令
	MC_GearOut	退出齿轮同步指令
	MC_GearInPos	齿轮同步指令（增加限定进入同步前的过渡过程参数）

3. 4. 1. 2 PLCopen状态机



状态机详细描述如下：

状态	功能描述
Disabled	未使能状态
ErrorStop	故障停机状态
Standstill	使能状态

状态	功能描述
Homing	原点回归状态
Stopping	停止状态
Discrete Motion	离散运动
Continuous Motion	连续运行状态
Synchronized Motion	同步运行状态

状态迁移条件：

Note1: 当轴的故障检测逻辑检测到故障时立即进入该状态

Note2: 当轴无故障且MC_Power.Enable=FALSE时

Note3: 当调用MC_Reset复位轴故障且MC_Power.Status=FALSE时

Note4: 当调用MC_Reset复位轴故障且MC_Power.Status=TRUE时

Note5: 当MC_Power.Enable=TRUE且MC_Power.Status=TRUE时

Note6: 当MC_Stop.Done=TRUE且MC_Stop.Execute=FALSE时

3. 4. 1. 3 BufferMode说明

部分运动功能块具有BufferMode的引脚，其意为，缓冲模式，规定了两运动功能块之间的衔接模式。

Name	Value	Description
MC_BUFFERMODE_ABORTING	0	立即启动FB(默认模式)
MC_BUFFERMODE_BUFFERED	1	在当前运动完成后开始FB
MC_BUFFERMODE_BLENDINGLOW	2	速度与两个FB的最低速度混合
MC_BUFFERMODE_BLENDINGPREVIOUS	3	速度与第一个FB的速度混合
MC_BUFFERMODE_BLENDINGNEXT	4	速度与第二个FB的速度混合

Name	Value	Description
MC_BUFFERMODE_ BLENDINGHIGH	5	速度与两个FBS的最高速度混合

3.4.1.4 故障代码说明

用户可通过MotionControl库的数据类型中的枚举类型找到故障代码的说明，通过属性窗口，或双击打开查看

MotionControl

Data Type

Alias

Structures

Numeratives

MC_BUFFER_MODE

MC_COMBINE_MODE

MC_DIRECTION

MC_ERRORCODE

MC_SOURCE

MC_START_MODE

MC_SYNC_MODE

Enumeration: MC_ERRORCODE (ver. 1.0.0)

Location: (库) MotionControl

Name	Value	Description
NO_ERROR	0	无错误
AXIS_NOT_EXIST	1	轴ID号不存在
AXIS_TYPE_NOT_MATCH	2	轴类型不匹配
AXIS_QUEUE_FULL	3	轴队列已满
AXIS_ENCODER_OVERFLOW	4	轴编码器溢出

名称	初始化值	描述
1 NO_ERROR	0	无错误
2 AXIS_NOT_EXIST1		轴ID号不存在
3 AXIS_TYPE_NOT2		轴类型不匹配
4 AXIS_QUEUE_FU3		轴队列已满
5 AXIS_ENCODER_4		轴编码器溢出
6 AXIS_POWER_O5		轴未使能
7 AXIS_POWER_O6		轴已使能
8 FREQUENCY_IL7		频率不合法
9 AXIS_BUSY		轴正忙，有功能块正在控制轴运动
10 FAILED_TO_BUF9		不支持以buffer形式添加
11 BUFFER_MODE_16		BufferMode值非法
12 DIRECTION_ILLE17		Direction值非法
13 EXCUTION_MOD18		ExcutionMode值非法
14 SOURCE_ILLEG19		Source值非法
15 OVERRIDE_ILLE20		Override值非法
16 SHIFTING_MODE21		Shifting模式非法
17 CONTROL_MODI22		控制模式设置错误
18 POS_ILLEGAL	256	位置不合法
19 ACC_DEC_ILLEG257		加/减速度不合法
20 VEL_ILLEGAL	258	速度不合法
21 CMD_P_POS_OV259		指令位置超出配置文件正向限制
22 CMD_N_POS_OV260		指令位置超出配置文件负向限制
23 FORBIDDEN_P_261		禁止正向移动
24 FORBIDDEN_N_262		禁止负向移动

也可查询下表

故障名称	对应值	含义
NO_ERROR	0	无错误
AXIS_NOT_EXIST	1	轴ID号不存在
AXIS_TYPE_NOT_MATCH	2	轴类型不匹配
AXIS_QUEUE_FULL	3	轴队列已满
AXIS_ENCODER_OVERFLOW	4	轴编码器溢出
AXIS_POWER_OFF	5	轴未使能
AXIS_POWER_ON	6	轴已使能
FREQUENCY_ILLEGAL	7	频率不合法

故障名称	对应值	含义
AXIS_BUSY	8	轴正忙，有功能块正在控制轴运动
FAILED_TO_BUFFER	9	不支持以buffer形式添加
BUFFER_MODE_ILLEGAL	16	BufferMode值非法
DIRECTION_ILLEGAL	17	Direction值非法
EXCUTION_MODE_ILLEGAL	18	ExcutionMode值非法
SOURCE_ILLEGAL	19	Source值非法
OVERRIDE_ILLEGAL	20	Override值非法
SHIFTING_MODE_ILLEGAL	21	Shifting模式非法
CONTROL_MODE_ILLEGAL	22	控制模式设置错误
POS_ILLEGAL	256	位置不合法
ACC_DEC_ILLEGAL	257	加/减速度不合法
VEL_ILLEGAL	258	速度不合法
CMD_P_POS_OVER_LIMIT	259	指令位置超出配置文件正向限制
CMD_N_POS_OVER_LIMIT	260	指令位置超出配置文件负向限制
FORBIDDEN_P_POS_MOVE	261	禁止正向移动
FORBIDDEN_N_POS_MOVE	262	禁止负向移动
POS_LAG_OVER_LIMIT	263	轴位置跟随误差超限
VEL_LAG_OVER_LIMIT	264	轴速度跟随误差超限
CMD_VEL_OVER_LIMIT	265	轴指令速度超出限制
CMD_ACC_OVER_LIMIT	266	轴指令加速度超出限制
POS_INFINITY	267	轴设定位置不合法
SOFTWARE_EMGS	494	用户急停
SYSTEME_MGS	495	系统急停
COMMUNICATION_EXCEPTION	496	通信异常
HOMING_VEL_ILLEGAL	608	回零速度非法
HOMING_ACC_ILLEGAL	609	回零加速度非法
HOMING_APPR_ILLEGAL	610	回零接近速度非法
HOMING_METHOD_ILLEGAL	611	回零方式非法

故障名称	对应值	含义
HOME_POSITION_ILLEGAL	612	home位置非法
HOMING_INTERRUPT_OR_NSTART	613	回零中断/未开始
HOMING_TARGET_NOT_REACH	614	回零目标未到达
HOMING_ERR	615	回零错误
HOMING_ERR_VO	616	回零错误速度为0
HOMING_TIMEOUT	617	回零超时
HOMING_MODE_CFG_FAILED	618	回零模式设置失败
HOMING_VEL_CFG_FAILED	619	回零速度设置失败
HOMING_ACC_CFG_FAILED	620	回零加速度设置失败
HOMING_APPR_CFG_FAILED	621	回零接近速度设置失败
HOMING_OPMODE_CFG_FAILED	622	回零操作模式设置失败
ENCODER_PRESET_OVERFLOW	650	编码器预设位置溢出
PLCOPEN_DISABLED	1280	PLCOpen状态机：DISABLED
PLCOPEN_STANDSTILL	1281	PLCOpen状态机：STANDSTILL
PLCOPEN_AXISHOMING	1282	PLCOpen状态机：HOMING
PLCOPEN_DISCRETEMOTION	1283	PLCOpen状态机：DISCRETEMOTION
PLCOPEN_CONTINUOUSMOTION	1284	PLCOpen状态机：CONTINUOUSMOTION
PLCOPEN_SYNCHRONIZEDMOTION	1285	PLCOpen状态机：SYNCHRONIZEDMOTION
PLCOPEN_STOPPING	1286	PLCOpen状态机：STOPPING
PLCOPEN_ERRORSTOP	1287	PLCOpen状态机：ERRORSTOP
PARAMETER_ILLEGAL	1536	参数不合法
CURRENT_NOT_SUPPORTED	1537	暂时不支持

故障名称	对应值	含义
CAMSWITCH_MODE_ILLEGAL	1616	CamSwitchMode值不合法
CAMSWITCH_DIRECTION_ILLEGAL	1617	CamSwitchDirection值不合法
TRIGGERINPUT_USED	1621	Trigger input 已被使用
TRIGGEREVENT_FULL	1622	Trigger event 已满
RIGGEREVENT_BUSY	1623	Trigger event已被使用
USED_UPLICATE_AXIS	1632	主从轴为同一轴
DENOMINATOR_0	1633	齿轮比分母为0
MASTER_VALUE_SOURCE_ILLEGAL	1634	主轴源数据非法
NO_CAMSELECTED	1637	CamTableID无数据
NOT_CAMREFTYPE	1638	非凸轮表数据
CAMTABLE_EMPTY_MASTER_RANGE	1639	凸轮表主轴数据异常
CAM_NO_POINTER	1640	凸轮表数据为空
MASTER_START_DISTANCE_ILLEGAL	1641	凸轮MasterStart-Distance为0，非法
MASTERNOTMATCH	1642	Phasing功能块主轴与实际主轴不匹配
ECT_SDO_READ_ERR	2048	SDO读错误
ECT_SDO_WRITE_ERR	2049	SDO写错误
AXIS_UNASSOCIATED_DRIVER	4096	轴未关联设备
DRIVER_NOT_READY	4097	驱动设备未准备好
DRIVER_ERROR	4098	驱动设备报错
DRIVER_ARRAY_FULL	4099	驱动设备个数超限
DRIVER_INDEX_OUTRANGE	4100	驱动设备索引超出范围
DRIVER_INDEX_USED	4101	驱动设备索引已被使用
DRIVER_CONTROL_MODE_UNSUPPORTED	4102	驱动设备控制模式不支持

故障名称	对应值	含义
FSM_NOT_READY_TO_SWITCH_ON	4368	402状态机:NOT_READY_TO_SWITCH_ON
FSM_SWITCH_ON_DISABLE	4369	402状态机:SWITCH_ON_DISABLE
FSM_READY_TO_SWITCH_ON	4370	402状态机:READY_TO_SWITCH_ON
FSM_SWITCH_ON	4371	402状态机:SWITCH_ON
FSM_OPERATION_ENABLE	4372	402状态机:OPERATION_ENABLE
FSM_QUICK_STOP_ACTIVE	4373	402状态机:QUICK_STOP_ACTIVE
FSM_FAUT_REACTION_ACTIVE	4374	402状态机:FAUT_REACTION_ACTIVE
FSM_FAUT	4375	402状态机:FAUT
FSM_UNKNOW	4376	402状态机:UNKNOW
PERMISSION_DENIED	5000	权限不允许
GROUP_NOT_EXIST	5001	轴组不存在
GROUP_AXIS_ARRAY_FULL	5002	轴组空间已满
AXIS_ALREADY_IN_GROUP	5003	轴已存在于组中
AXIS_NOT_IN_GROUP	5004	轴组不包含该轴
GROUP_QUEUE_FULL	5005	轴组队列已满
GROUP_DISABLED	5006	轴组未使能
GROUP_AXIS_DISABLED	5007	轴组的轴成员未使能
GROUP_AXIS_ERR	5008	轴组的轴成员错误
GROUP_AXIS_0_ERR	5009	轴组的轴 0 成员错误
GROUP_AXIS_1_ERR	5010	轴组的轴 1 成员错误
GROUP_AXIS_2_ERR	5011	轴组的轴 2 成员错误
GROUP_AXIS_3_ERR	5012	轴组的轴 3 成员错误
GROUP_SYNC_AXIS_0_ERR	5013	轴组的同步轴 0 成员错误
GROUP_SYNC_AXIS_1_ERR	5014	轴组的同步轴 1 成员错误
GROUP_SYNC_AXIS_2_ERR	5015	轴组的同步轴 2 成员错误

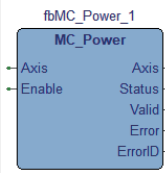
故障名称	对应值	含义
GROUP_AXIS_0_HOME_FAILED	5016	轴组的轴 0 回零失败
GROUP_AXIS_1_HOME_FAILED	5017	轴组的轴 1 回零失败
GROUP_AXIS_2_HOME_FAILED	5018	轴组的轴 2 回零失败
GROUP_AXIS_3_HOME_FAILED	5019	轴组的轴 3 回零失败
GROUP_SYNC_AXIS_0_HOME_FAILED	5020	轴组的同步轴 0 回零失败
GROUP_SYNC_AXIS_1_HOME_FAILED	5021	轴组的同步轴 1 回零失败
GROUP_SYNC_AXIS_2_HOME_FAILED	5022	轴组的同步轴 2 回零失败
GROUP_AXIS_MODULO_NOT_ALLOWED	5023	轴组成员不允许为模态轴
GROUP_FSM_DISABLED	5100	PLCOpen 状态机:DISABLED
GROUP_FSM_STANDBY	5101	PLCOpen 状态机:STANDBY
GROUP_FSM_HOMING	5102	PLCOpen 状态机:HOMING
GROUP_FSM_MOVING	5103	PLCOpen 状态机:MOVING
GROUP_FSM_STOPPING	5104	PLCOpen 状态机:STOPPING
GROUP_FSM_ERRORSTOP	5105	PLCOpen 状态机:ERRORSTOP
KINEMATIC_DEFINED	5140	运动学变换已定义
KINEMATIC_NOT_ALLOWED	5141	运动机构类型不允许
KIN_TRANSFORM_NOT_CONSTANT	5142	运动机构变换不唯一
GROUP_AIMPOUTLIMIT	5150	目标位置超限制

故障名称	对应值	含义
GROUP_KINPARAMERROR	5151	运动学参数限制不合理
GROUP_TRANMODEERROR	5152	过渡模式错误
GROUP_CIRCLEPARAERROR	5153	圆弧模式参数错误
GROUP_PLANFAIL	5154	S 曲线规划失败
GROUP_PARAMERROR	5155	参数错误
GROUP_ACC_DEC_ILLEGAL	5156	加减速不合法
GROUP_JERK_ILLEGAL	5157	加加速度不合法
GROUP_AIMEQUAL	5158	目标位置重合
GROUP_BLENDING_ERROR	5159	轨迹混合错误
GROUP_MOTION_LIMIT_AVOUT	5160	运动过程中关节角速度和角加速度超过限制
GROUP_MOTION_LIMIT_POSOUT	5161	运动过程中关节位置超限
GROUP_PLAN_ERROR	5162	轴组规划错误
GROUP_INTERPOLATE_ERROR	5180	轴组插补器错误
GROUP_SYNC_POSLAGOVERLIMIT	5181	同步轴跟随误差超限
GROUP_SYNC_CMDPOSLAGOVERLIMIT	5182	同步轴跟随误差超限
GROUP_UNAVALID_COMBINE_PARAM	5183	不支持的参数组合
GROUP_INTERRUPT_NODE_NOT_EXIST	5184	轴组中断运动不存在

3.4.2 Basic基础指令

3.4.2.1 MC Power

3.4.2.1.1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_Power		<pre>fbMC_Power_1(Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Status=>(* BOOL *), Valid=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Enable	BOOL	使能, 高电平时功能块生效
VAR_OUTPUT	Status	BOOL	轴使能状态, true为使能, false为未使能
	Valid	BOOL	有效标志位, true时, 功能块输出状态可信
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

3.4.2.1.2 功能块使用说明

MC_Power指令适用于总线伺服轴和虚拟轴, 用于控制轴的使能状态

• 使用说明

• 引脚说明:

使能Enable, 置True后, 轴进入使能状态, 此时轴的PLCOpen状态机由Disabled状态进入Stand-Still状态, 可以执行轴运动指令

置False, 轴退出使能状态, 会中断当前正在执行

的运动指令，轴解除使能状态后，无法执行运动指令，仍可以执行管理指令，进行故障的复位，参数的写入和读取等；

轴存在故障时，重新使能MC_Power不能控制轴进入使能，需要先调用MC_Reset复位轴故障

有效Valid，信号短暂置False后置True，意为当前功能块输出引脚状态可信；

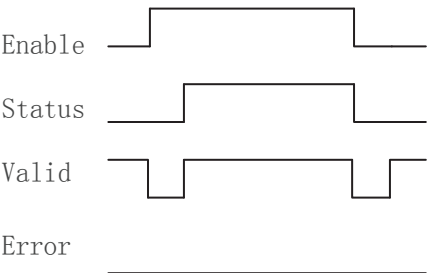
使能状态Status，若轴使能成功，则在Valid置True后置True；

• 多重调用

当多个MC_Power指令控制同一轴对象时，以一周期内最后执行的MC_Power的输入输出为准。

• 时序图

使用一条MC_Power指令，轴正常使能的时序图如下：



3. 4. 2. 2 MC_Home
3. 4. 2. 2. 1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_Home		<pre>fbMC_Home_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Position:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *),</pre>

Instruc-tions	LD Per-formance	ST Performance
		<pre>Done =>(* BOOL *), Busy =>(* BOOL *), Active =>(* BOOL *), CommandAborted =>(* BOOL *), Error =>(* BOOL *), ErrorID =>(* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Position	REAL	零点位置
	Buffer Mode	MC_BufferMode	缓冲模式给定, 请参考3. 4. 1. 3 Buffer-Mode说明
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成, 且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

3. 4. 2. 2. 2 功能块使用说明

MC_Home指令适用于总线伺服轴和虚拟轴，实现原点回归功能，指令上升沿有效

• 使用说明

轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。

总线伺服轴调用本指令时，用户可以在轴配置界

面设定Homing Method

虚拟轴调用本指令时,采用CiA402协议中35号方式回零

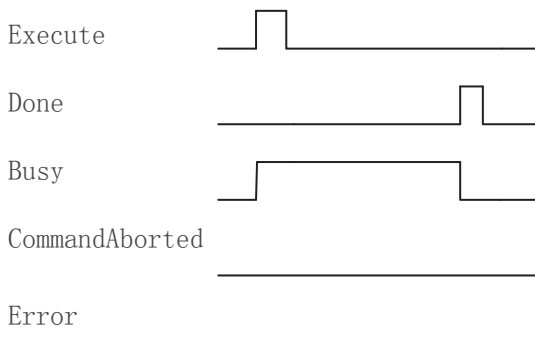
• 引脚说明:

触发Execute, 输入上升沿触发回零指令, 轴PLCOpen状态机进入Homing状态

零点偏移Position, 用户给定的零点偏移, 回零执行完成后, 会将当前位置设置为用户给定值。

• 时序图

MC_Home功能块触发, 轴对象执行回零动作, 功能正常执行的时序图如下:



3.4.2.3 MC_Stop

3.4.2.3.1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_Stop		<pre>fbMC_Stop_1 (Axis:=(* AXIS_REF *), Execute:= (* BOOL *), Deceleration:= (* REAL *), Jerk:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Decel-eration	REAL	减速度 [u/s2]
	Jerk	REAL	加加速度 [u/s3]
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

3.4.2.3.2 功能块使用说明

MC_Stop指令适用于总线伺服轴和虚拟轴, 实现停止功能, 指令上升沿有效

• 使用说明

轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。

用户给定的减速度和加加速度将在Execute上升沿被锁存, 作为此次停止动作的参数。

Execute输入上升沿后, 若未置Flase, 即一直处于高电平的状态, 则减速完成后Done信号持续有效, 且当前轴的PLCOpen状态机一直处于Stopping状态 此后将Execute置False, 则轴状态机从Stopping切换至Standstill状态

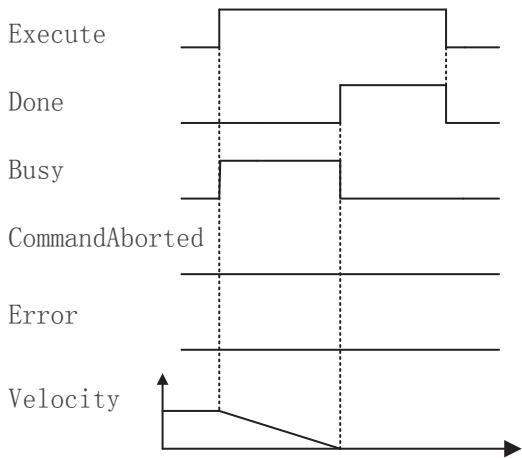
• 引脚说明:

减速度Deceleration: 暂停动作的减速度

加加速度Jerk: 加加速度

• 时序图

MC_Stop打断MC_MoveVelocity运动时序可参考下图



3.4.2.4 MC_Halt

3.4.2.4.1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_Halt	fbMC_Halt_1 MC_Halt Axis Execute Deceleration Jerk BufferMode CommandAborted Error ErrorID	fbMC_Halt_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Deceleration:= (* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), Done =>(* BOOL *), Busy =>(* BOOL *), Active =>(* BOOL *), CommandAborted => (* BOOL *), Error =>(* BOOL *), ErrorID =>(* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Decel-eration	REAL	减速度 [u/s2]
	Jerk	REAL	加加速度 [u/s3]

引脚类型	名称	类型	描述
	Buffer Mode	MC_BUFFER_MODE	缓冲模式给定，请参考3.4.1.3 Buffer-Mode说明
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成，且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号，true为故障，false为无故障
	ErrorID	MC_ERRORCODE	轴故障代码

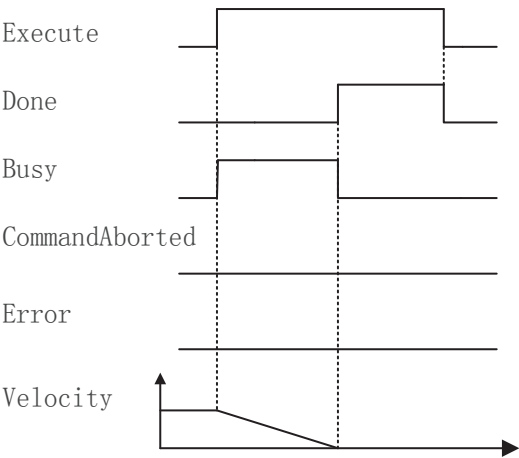
- 3.4.2.4.2 功能块使用说明
- MC_Halt指令适用于总线伺服轴和虚拟轴，实现暂停功能，指令上升沿有效
- 使用说明

轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。

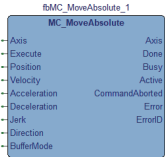
指令上升沿锁存，减速度，加加速度等

功能块触发后生效，轴状态机切换至Discrete motion状态，这一状态，意为其它符合PLCOpen状态机切换的指令均可打断该指令。
- 时序图

MC_Halt打断MC_MoveVelocity指令时序图如图所示



3.4.2.5 MC_MoveAbsolute
3.4.2.5.1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_MoveAbsolute		fbMC_MoveAbsolute_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Position:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), Direction:=(* MC_DIRECTION *), BufferMode:= (* MC_BUFFER_MODE *), Done =>(* BOOL *), Busy =>(* BOOL *), Active =>(* BOOL *), CommandAborted =>(* BOOL *), Error =>(* BOOL *), ErrorID =>(* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Position	REAL	绝对定位的目标位置[u]
	Velocity	REAL	最大速度 [u/s].
	Acceleration	REAL	最大加速度[u/s2]
	Deceleration	REAL	最大减速度 [u/s2]
	Jerk	REAL	加加速度 [u/s3]
	Direction	MC_DIRECTION	运行方向, 枚举类型 (分别是: mcPositiveDirection, mcShortestWay, mcNegativeDirection, mcCurrentDirection)
VAR_OUTPUT	BufferMode	MC_BUFFER_MODE	缓冲模式给定, 请参考 3.4.1.3 BufferMode说明
	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成, 且将生成新的输出值
	CommandAborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

3.4.2.5.2 功能块使用说明
MC_MoveAbsolute指令适用于总线伺服轴和虚拟轴, 实现绝对定位功能, 指令上升沿有效

- 使用说明
轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。
Execute上升沿触发时, 锁存输入参数, 触发绝对定位功能, 轴状态机切换至Discrete motion状态

• 引脚说明:

Position目标位置: 绝对定位的目标位置, 轴到达目标位置的方式由Direction输入决定

Velocity速度: 绝对定位过程中的最大速度

Acceleration加速度: 绝对定位过程中的最大加速度

Deceleration减速度: 绝对定位过程中的最大减速度

Jerk加加速度: 定位过程中的最大加加速度

轴速度曲线的形状由Velocity、Acceleration、Deceleration、Jerk共同决定

Direction方向: 包含四种选项, 在轴为线性轴时不发挥作用(绝对模式下, 线性轴的运动方向是根据当前位置和目标位置确定的), 而在模数轴的情况下, 同一目标位置可通过不同方向到达, 分为以下四种情况

0-mcPositiveDirection: 正向运动, 沿轴运动的正方向到达目标位置

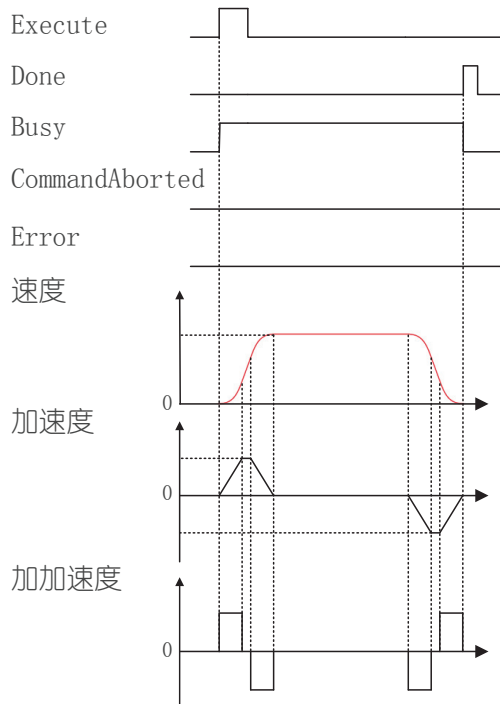
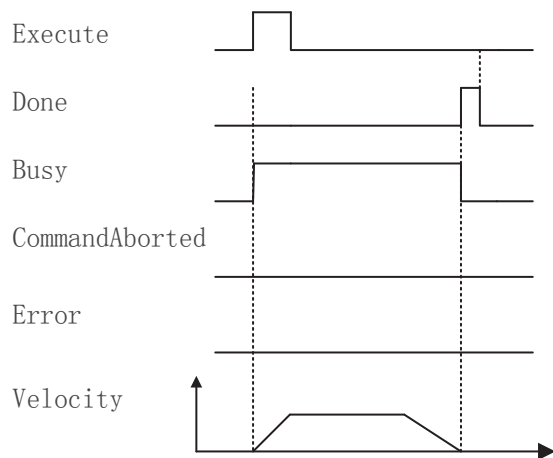
1-mcShortestWay: 最短距离, 轴以到达目标位置距离最短的方向运动

2-mcNegativeDirection: 反向运动, 沿轴运动的反方向到达目标位置

3-mcCurrentDirection: 沿当前方向到达目标位置
若模数轴的目标位置大于模数值, 则将目标位置对模数值求模, 得到可用的目标位置

• 时序图

轴Standstill状态下调用MC_MoveAbsolute指令
时序图可参考如图



3.4.2.6 MC_MoveRelative

3.4.2.6.1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_MoveRelative		<pre> fbMC_MoveRelative_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *)); </pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Distance	REAL	相对定位的目标距离[u]
	Velocity	REAL	最大速度 [u/s].
	Acceleration	REAL	最大加速度[u/s2]
	Deceleration	REAL	最大减速度 [u/s2]
	Jerk	REAL	加加速度 [u/s3]
	Buffer Mode	MC_BUFFER_MODE	缓冲模式给定, 请参考 3.4.1.3 BufferMode说明
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成, 且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

3.4.2.6.2 功能块使用说明

MC_MoveRelative指令适用于总线伺服轴和虚拟轴, 实现相对定位, 指令上升沿有效

• 使用说明

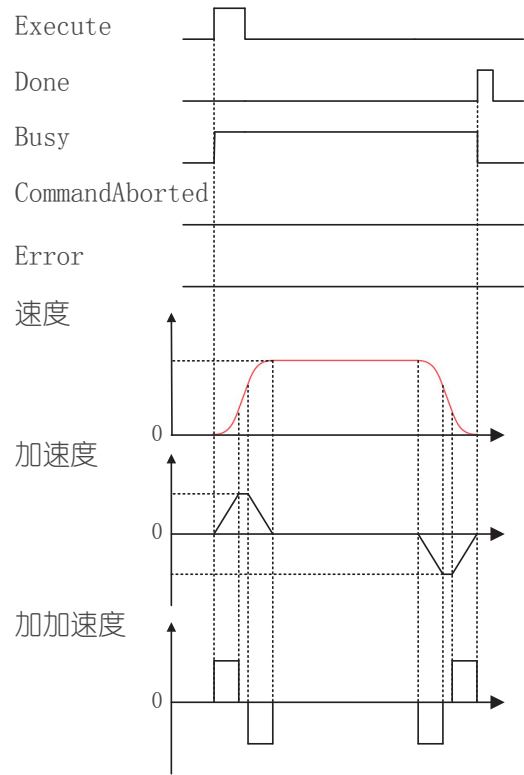
轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。
Execute上升沿触发时, 锁存输入参数, 触发绝对定位功能, 轴状态机切换至Discrete motion状态

• 引脚说明:

Distance目标距离: 相对定位的移动距离, 轴的线性或是模数模式下, Distance为正, 轴正向运行, Distance为负, 轴反向运行, 运行距离都是|Distance|指定的距离
Velocity速度: 定位过程中的最大速度

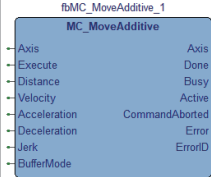
Acceleration加速度: 定位过程中的最大加速度
Deceleration减速度: 定位过程中的最大减速度
Jerk加加速度: 定位过程中的最大加加速度
轴速度曲线的形状由Velocity、Acceleration、Deceleration、Jerk共同决定

• 时序图



3.4.2.7 MC_MoveAdditive

3.4.2.7.1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_MoveAdditive		fbMC_MoveAdditive_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *),

Instruc-tions	LD Performance	ST Performance
		Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Distance	REAL	增量定位目标距离[u]
	Velocity	REAL	最大速度 [u/s].
	Accel-eration	REAL	最大加速度[u/s2]
	Decel-eration	REAL	最大减速度 [u/s2]
	Jerk	REAL	加加速度 [u/s3]
	Buffer Mode	MC_BUFF-ER_MODE	缓冲模式给定, 请参考 3.4.1.3 BufferMode说明
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成, 且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

3.4.2.7.2 功能块功能和指令说明

MC_MoveAddtive指令适用于总线伺服轴和虚拟轴，实现Discrete motion状态机下，在上一离散运动目标位置基础上叠加距离，如果打断上一离散运动，其运行结果仍是上一指令目标位置与MC_MoveAddtive目标距离的叠加，指令上升沿有效

• 使用说明

轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。

• 引脚说明:

Distance目标距离: 该距离将在上一离散运动目标位置上叠加, 不论是打断上一运动, 或是在上一运动结束后进行叠加运动, 当功能执行完毕时, 轴运动距离是上一离散运动目标位置与Distance的叠加值。

Velocity速度: 定位过程中的最大速度

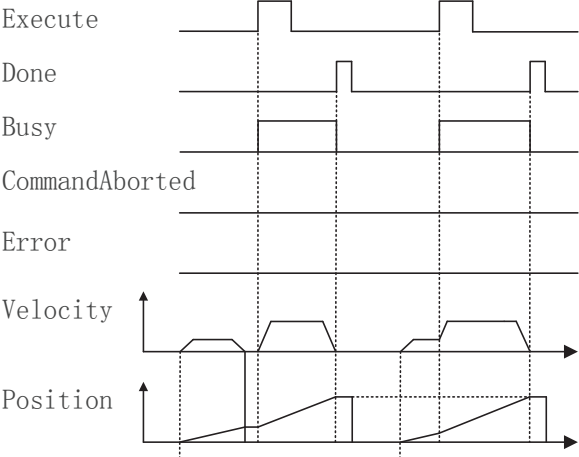
Acceleration加速度: 定位过程中的最大加速度

Deceleration减速度: 定位过程中的最大减速度

Jerk加加速度: 定位过程中的最大加加速度

轴速度曲线的形状由Velocity、Acceleration、Deceleration、Jerk共同决定

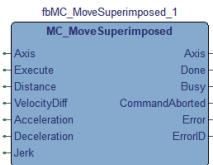
• 时序图



MC_MoveAbsolute 触发 MC_MoveAbsolute 触发

3.4.2.8 MC_MoveSuperImposed

3.4.2.8.1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_MoveSuperImposed		<pre>fbMC_MoveSuperimposed_1(Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), VelocityDiff:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Distance	REAL	叠加的距离[u]
	VelocityDiff	REAL	叠加运动的速度差值[u/s].
	Acceleration	REAL	加速度[u/s ²]
	Deceleration	REAL	减速度 [u/s ²]
	Jerk	REAL	加加速度 [u/s ³]
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	CommandAborted	BOOL	被打断标志位

引脚类型	名称	类型	描述
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码
	CoveredDistance	REAL	从叠加运动开始持续输出本功能块叠加的距离

3.4.2.8.2 功能块使用说明

MC_MoveSuperimposed指令适用于总线伺服轴和虚拟轴, 通常用于在基础运动控制功能块(如MC_MoveRelative)的基础上叠加一个增量式位置控制运动, 指令上升沿有效

• 使用说明

轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。

如果MC_MoveSuperimposed处于活动状态, 那么除了MC_MoveSuperimposed之外, 任何其他处于中止模式的命令都将中止两个运动命令: MC_MoveSuperimposed和底层的运动命令。在任何其他模式下, 底层的运动命令不会被中止。

如果MC_MoveSuperimposed处于活动状态, 并且命令了另一个MC_MoveSuperimposed, 只有正在进行的MC_MoveSuperimposed命令会被中止, 并被新的MC_MoveSuperimposed替换, 但不会中止底层的运动命令。

MC_MoveSuperimposed功能会导致正在进行的运动在所有相关状态下的速度变化, 如果适用, 还有命令的位置变化。

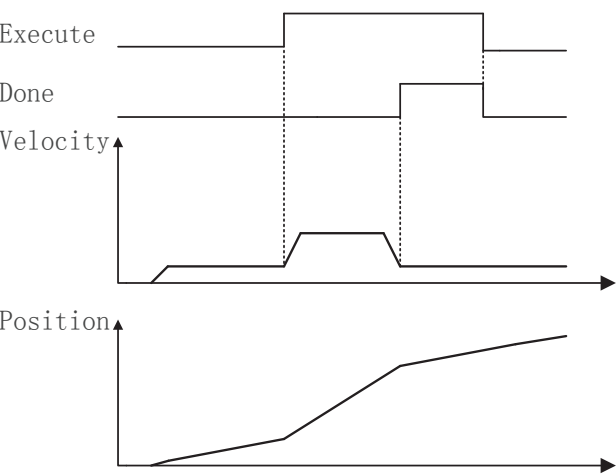
在“静止”状态下, FB MC_MoveSuperimposed的作用类似于MC_MoveRelative。

“加速度”、“减速度”和“急动度”的值是正在进行的运动的额外值, 而不是绝对值。这样, 无论是否同时发生MC_MoveSuperimposed FB, 底

层FB总是在同一时间段内完成其工作。
输出“活动”的行为与缓冲FBs不同。

- 引脚说明：
Distance目标距离： 叠加运动的额外距离。
VelocityDiff速度： 叠加运动过程中的最大速度差值
Acceleration加速度： 定位过程中的最大加速度
Deceleration减速度： 定位过程中的最大减速度
Jerk加加速度： 定位过程中的最大加加速度
轴速度曲线的形状由Velocity、Acceleration、Deceleration、Jerk共同决定

• 时序图



3.4.2.9 MC_HaltSuperImposed

3.4.2.9.1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_Halt Super Imposed	fbMC_HaltSuperimposed_1 MC_HaltSuperimposed Axis Execute Deceleration Jerk Axis Done Busy CommandAborted Error ErrorID	fbMC_HaltSuperimposed_1(Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Deceleration:=(* REAL *), Jerk:=(* REAL *),

Instruc-tions	LD Performance	ST Performance
		Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Decel-eration	REAL	减速度 [u/s2]
	Jerk	REAL	加加速度 [u/s3]
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	MC_ERRORCODE	轴故障代码

3.4.2.9.2 功能块使用说明

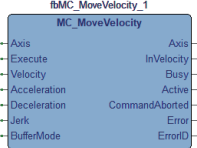
指令适用于总线伺服轴和虚拟轴，执行MC_HaltSuperimposed 可以单独停止MC_MoveSuperimposed 的运动，而不影响基础运动的进行。这与执行MC_Stop不同，后者会停止所有运动，包括基础运动和叠加运动，指令上升沿有效

• 使用说明

轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。

3.4.2.10 MC_MoveVelocity

3.4.2.10.1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_MoveVelocity		<pre>fBMC_MoveVelocity_1(Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), InVelocity=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_OUTPUT	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成，且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号，true为故障，false为无故障
	ErrorID	WORD	轴故障代码

3.4.2.10.2 功能块使用说明

MC_MoveVelocity指令适用于总线伺服轴和虚拟轴，实现用户指定速度的持续运动，指令上升沿有效。

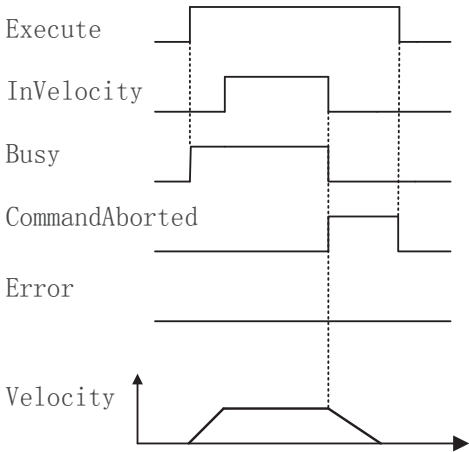
• 使用说明

轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。

Execute输入上升沿,指令锁存Velocity、Accceleration等输入参数

Velocity、Accceleration、Jerk共同决定曲线的形状
轴状态机切换至ContinuousMotion状态

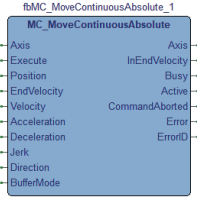
• 时序图



引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Velocity	REAL	目标速度 [u/s].
	Accel-eration	REAL	最大加速度[u/s2]
	Decel-eration	REAL	最大减速度 [u/s2]
	Jerk	REAL	加加速度 [u/s3]
	Buffer-Mode	MC_BUFF-ER_MODE	缓冲模式给定，请参考3.4.1.3 Buffer-Mode说明
	In Velocity	BOOL	到达指定目标速度

3.4.2.11 MC_MoveContinuousAbsolute

3.4.2.11.1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_MoveContinuousAbsolute		<pre>fbMC_MoveContinuousAbsolute_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Position:=(* REAL *), EndVelocity:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), Direction:= (* MC_DIRECTION *), BufferMode:= (* MC_BUFFER_MODE *), InEndVelocity=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Position	REAL	相对定位的目标位置 [u]
	EndVelocity	REAL	定位结束后的持续运动速度 [u/s].
	Velocity	REAL	定位过程的最大速度 [u/s].

引脚类型	名称	类型	描述
	Acceleration	REAL	定位过程的最大加速度 [u/s²]
	Deceleration	REAL	定位过程的最大减速度 [u/s²]
	Jerk	REAL	加加速度 [u/s³]
	Direction	MC_DIRECTION	运行方向, 枚举类型 (分别是: mcPositiveDirection, mcShortestWay, mcNegativeDirection, mcCurrentDirection)
	Buffer Mode	MC_BUFFER_MODE	缓冲模式给定, 请参考3.4.1.3 Buffer-Mode说明
VAR_OUTPUT	InEnd Velocity	BOOL	绝对定位完成且进入持续运动的目标速度
	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成, 且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

3.4.2.11.2 功能块使用说明

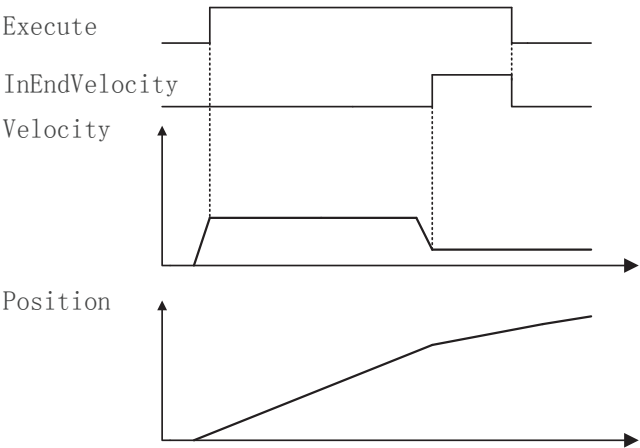
MC_MoveContinuousAbsolute指令适用于总线伺服轴和虚拟轴, 实现一段绝对定位, 完成后将以设定速度持续运动, 指令上升沿有效

• 使用说明

轴对象需要先使用MC_Power指令进入使能状态才

可以调用该指令。
轴状态机从Stopping切换至Standstill状态后调用该指令，进入ContinuousMotion状态

• 时序图



3. 4. 2. 12 MC_MoveContinuousRelative
3. 4. 2. 12. 1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_MoveContinuousRelative	fbMC_MoveContinuousRelative_1 MC_MoveContinuousRelative Axis Execute Distance EndVelocity Velocity Acceleration Deceleration Jerk BufferMode Axis InEndVelocity Busy Active CommandAborted Error ErrorID	fbMC_MoveContinuousRelative_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), EndVelocity:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), InEndVelocity=>(* BOOL *), Busy=>(* BOOL *),

Instructions	LD Performance	ST Performance	
		Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));	

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Distance	REAL	相对定位的目标距离 [u]
	End Velocity	REAL	定位结束后的持续运动速度 [u/s].
	Velocity	REAL	定位过程的最大速度 [u/s].
	Acceleration	REAL	定位过程的最大加速度 [u/s2]
	Deceleration	REAL	定位过程的最大减速度 [u/s2]
	Jerk	REAL	加加速度 [u/s3]
VAR_OUTPUT	Buffer Mode	MC_BUFFER_MODE	缓冲模式给定, 请参考3. 4. 1. 3 BufferMode说明
	InEnd Velocity	BOOL	相对定位完成且进入持续运动的目标速度
	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成, 且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

3.4.2.12.2 功能块使用说明

MC_MoveContinuousRelative指令适用于总线伺服轴和虚拟轴，实现一段相对定位，完成后按照给定速度持续运动，指令上升沿有效

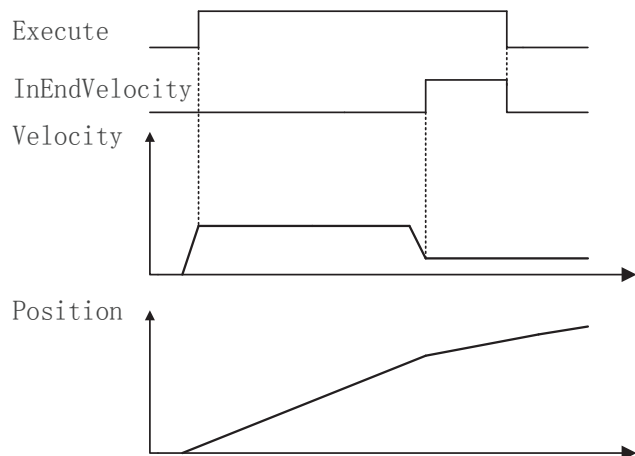
• 使用说明

轴对象需要先使用MC_Power指令进入使能状态才可以调用该指令。

轴状态机从Stopping切换至Standstill状态后调用该指令，进入ContinuousMotion状态

• 引脚说明：

• 时序图



3.4.2.13 MC_SetPosition

3.4.2.13.1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_SetPosition	fbMC_SetPosition_1 MC_SetPosition Axis Execute Position Relative Axis Done Busy Error ErrorID	fbMC_SetPosition_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Position:=(* REAL *), Relative:=(* BOOL *), Done=>(* BOOL *),

Instructions	LD Performance	ST Performance
		Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *);

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Position	REAL	设定位置[u]
	Relative	BOOL	True则为相对位置设置模式, False则为绝对位置设置模式
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号, True为故障, False为无故障
	ErrorID	WORD	轴故障代码

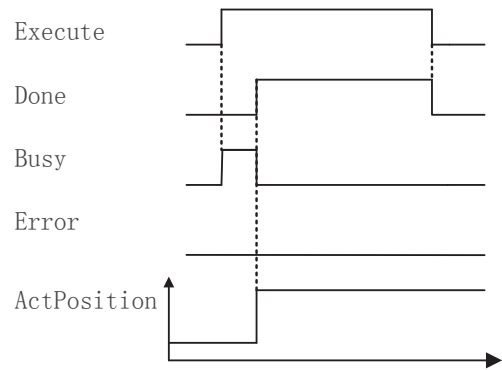
3.4.2.13.2 功能块使用说明

MC_SetPosition指令适用于总线伺服轴和虚拟轴，用于设置运动控制轴的位置，指令上升沿有效

• 使用说明

MC_SetPosition设置位置后，当前位置改变，但这一位置并不会改变伺服编码器计数值，仅作为当前motion运行的位置参考值，若断电上电，上一位置参考值会丢失，需要用户重新设置参考位置
功能块无需轴使能也可触发生效

• 时序图



3. 4. 2. 14 MC_ReadActualPosition

3. 4. 2. 14. 1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_Read Actual Position		fbMC_ReadActualPosi-tion_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *), Position=>(* REAL *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Enable	BOOL	使能, 高电平时功能块生效
VAR_OUTPUT	Valid	BOOL	有效标志位, true时, 功能块输出状态可信
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号, true为故障, false为无故障

引脚类型	名称	类型	描述
	ErrorID	WORD	轴故障代码
	Position	REAL	读取的实际位置[u]

3. 4. 2. 14. 2 功能块使用说明

MC_ReadActualPosition指令适用于总线伺服轴和虚拟轴, 读取实际位置, 指令高电平有效

• 使用说明

MC_ReadActualPosition功能块需要在轴初始化完成后才可正常读取位置值, 未初始化完成读取值为NaN

功能块无需轴使能也可使能生效

3. 4. 2. 15 MC_ReadActualVelocity

3. 4. 2. 15. 1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_Read Actual Velocity		fbMC_ReadActualVeloci-ty_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *), Velocity=>(* REAL *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Enable	BOOL	使能, 高电平时功能块生效
VAR_OUTPUT	Valid	BOOL	有效标志位, true时, 功能块输出状态可信
	Busy	BOOL	输入参考已知并设置完成

引脚类型	名称	类型	描述
	Error	BOOL	轴故障信号，true为故障，false为无故障
	ErrorID	WORD	轴故障代码
	Velocity	REAL	读取的实际速度 [u/s].

3. 4. 2. 15. 2 功能块使用说明

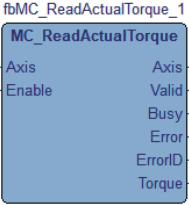
MC_ReadActualVelocity指令适用于总线伺服轴和虚拟轴，读取轴实际速度，指令高电平有效

• 使用说明

MC_ReadActualVelocity功能块需要在轴初始化完成后才可正常读取速度值，未初始化完成读取值为NAN
功能块无需轴使能也可使能生效

3. 4. 2. 16 MC_ReadActualTorque

3. 4. 2. 16. 1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_Read Actual Torque		fbMC_ReadActualTorque_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), Torque=>(* REAL *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Enable	BOOL	使能，高电平时功能块生效

引脚类型	名称	类型	描述
VAR_OUTPUT	Valid	BOOL	有效标志位，true时，功能块输出状态可信
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号，true为故障，false为无故障
	ErrorID	MC_ERRORCODE	轴故障代码
	Torque	REAL	读取的实际转矩 (in technical units)

3. 4. 2. 16. 2 功能块使用说明

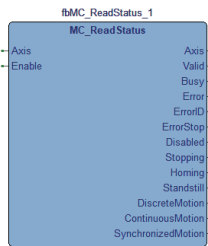
MC_ReadActualTorque指令适用于总线伺服轴，能够实时读取实际转矩，指令高电平有效

• 使用说明

MC_ReadActualTorque功能块需要在轴初始化完成后才可正常读取转矩，未初始化完成读取值为NAN
指令应用于虚拟轴时，功能块不会报错，但输出转矩始终为0。
功能块无需轴使能也可使能生效

3. 4. 2. 17 MC_ReadStatus

3. 4. 2. 17. 1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_Read Status		fbMC_ReadStatus_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *),

Instruc-tions	LD Performance	ST Performance
		ErrorStop=>(* BOOL *), Disabled=>(* BOOL *), Stopping=>(* BOOL *), Homing=>(* BOOL *), Standstill=>(* BOOL *), DiscreteMotion=>(* BOOL *), ContinuousMotion=>(* BOOL *), SynchronizedMotion=>(* BOOL *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Enable	BOOL	使能，高电平时功能块生效
VAR_OUTPUT	Valid	BOOL	有效标志位，true时，功能块输出状态可信
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号，true为故障，false为无故障
	ErrorID	WORD	轴故障代码
	Error Stop	BOOL	请参考PLCopen状态机
	Disabled	BOOL	请参考PLCopen状态机
	Stopping	BOOL	请参考PLCopen状态机
	Homing	BOOL	请参考PLCopen状态机
	Standstill	BOOL	请参考PLCopen状态机
	Discrete Motion	BOOL	请参考PLCopen状态机
	ContinuousMotion	BOOL	请参考PLCopen状态机

引脚类型	名称	类型	描述
	Synchr-onizedMo-tion	BOOL	请参考PLCopen状态机

3.4.2.17.2 功能块使用说明

MC_ReadStatus指令适用于总线伺服轴和虚拟轴，用于读取轴的PLCopen状态机，指令高电平有效

3.4.2.18 MC_ReadMotionState

3.4.2.18.1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_Read Motion State		fbMC_ReadMotionState_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Source:=(* MC_SOURCE *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *), ConstantVelocity=>(* BOOL *), Accelerating=>(* BOOL *), Decelerating=>(* BOOL *), DirectionPositive=>(* BOOL *), DirectionNegative=>(* BOOL *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Enable	BOOL	使能，高电平时功能块生效

引脚类型	名称	类型	描述
	Source	MC_SOURCE	枚举类型，数据来源为实际值/设定值：可选 mcSetValue, mcActualValue.
VAR_OUTPUT	Valid	BOOL	有效标志位, true时, 功能块输出状态可信
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	MC_ERRORCODE	轴故障代码
	Constant Velocity	BOOL	速度是否恒定
	Accelerating	BOOL	速度绝对值在增加
	Decelerating	BOOL	速度绝对值在减小
	Direction Positive	BOOL	位置值在增加
	Direction Negative	BOOL	位置值在减小

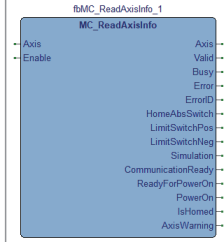
3. 4. 2. 18. 2 功能块使用说明

MC_ReadMotionState指令适用于总线伺服轴和虚拟轴，返回轴当前进行运动的状态，指令高电平有效

3. 4. 2. 19 MC_ReadAxisInfo

3. 4. 2. 19. 1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
		fbMC_ReadAxisInfo_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *),

Instruc-tions	LD Performance	ST Performance
MC_Read AxisInfo		Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), HomeAbsSwitch=> (* BOOL *), LimitSwitchPos=> (* BOOL *), LimitSwitchNeg=> (* BOOL *), Simulation=> (* BOOL *), CommunicationReady=> (* BOOL *), ReadyForPowerOn=> (* BOOL *), PowerOn=>(* BOOL *), IsHomed=>(* BOOL *), AxisWarning=> (* BOOL *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Enable	BOOL	使能，高电平时功能块生效
VAR_OUTPUT	Valid	BOOL	有效标志位，true时，功能块输出状态可信
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号，true为故障，false为无故障

引脚类型	名称	类型	描述
	ErrorID	MC_ERRORCODE	轴故障代码
	HomeAbsSwitch	BOOL	数字原点开关激活
	LimitSwitchPos	BOOL	正向硬件限位开关激活
	LimitSwitchNeg	BOOL	反向硬件限位开关激活
	Simulation	BOOL	轴处于模拟模式
	CommunicationReady	BOOL	网络初始化完成
	ReadyForPowerOn	BOOL	伺服可以使能
	PowerOn	BOOL	轴使能状态
	IsHomed	BOOL	轴的绝对参考位置已知
	AxisWarning	BOOL	轴存在警告

3.4.2.19.2 功能块使用说明

MC_ReadAxisInfo指令适用于总线伺服轴和虚拟轴，读取与轴有关的信息，如模式、使能状态等，指令高电平有效

3.4.2.20 MC_ReadAxisError

3.4.2.20.1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_ReadAxisError	fbMC_ReadAxisError_1 MC_ReadAxisError Axis Enable Axis Valid Busy Error ErrorID AxisErrorID	fbMC_ReadAxisError_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Valid=>(* BOOL *),

Instruc-tions	LD Performance	ST Performance
		Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *), AxisErrorID=>(* WORD *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Enable	BOOL	使能，高电平时功能块生效
VAR_OUTPUT	Valid	BOOL	有效标志位，true时，功能块输出状态可信
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号，true为故障，false为无故障
	ErrorID	MC_ERRORCODE	轴故障代码
	AxisErrorID	WORD	伺服故障代码

3.4.2.21 MC_Reset

3.4.2.21.1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_Reset	fbMC_Reset_1 MC_Reset Axis Execute Axis Done Busy Error ErrorID	fbMC_Reset_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

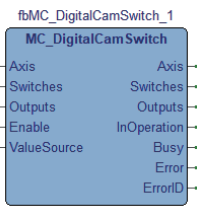
引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号,上升沿有效
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

3. 4. 2. 21. 2 功能块使用说明

MC_Reset指令适用于总线伺服轴和虚拟轴, 用于复位motion轴故障和伺服故障, 指令上升沿有效

3. 4. 2. 22 MC_DigitalCamSwitch

3. 4. 2. 22. 1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_DigitalCamSwitch		<pre>fbMC_DigitalCamSwitch_1 (Axis:=(* AXIS_REF *), Switches:= (* MC_CAMSWITCH_REF *), Outputs:=(* BOOL *), Enable:=(* BOOL *), ValueSource:= (* MC_SOURCE *), InOperation=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
	Switches	MC_CAMSW-ITCH_REF	开关动作结构对象

引脚类型	名称	类型	描述
	Outputs	BOOL	输出信号量
VAR_INPUT	Enable	BOOL	使能, 高电平时功能块生效
	Value-Source	MC_SOURCE	轴信息来源, 枚举类型, 可选:mcSetValue-Synchronization on set value mcActual-Value Synchroniza-tion on actual value
VAR_OUTPUT	InOper-ation	BOOL	设定曲线追踪已激活
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	MC_ER-RORCODE	轴故障代码

3. 4. 2. 22. 2 功能块使用说明

MC_DigitalCamSwitch指令适用于总线伺服轴和虚拟轴, 功能类似轴上的位置开关, 在给定位置、方向, 输出给定信号, 指令高电平有效

• 使用说明

用户可根据实际需要设置输入Switches, 以此来完成开关点所在位置和运行方向的设置
在Outputs的输出可以得到输出开关信号
Switchs处输入类型MC_CAMSWITCH_REF结构如下

Name	Type	Description
NoOfSwitches	BYTE	开关信号组的数量
CamSwitchPtr	@MC_CAMSWITCH_TR	开关信号指针

MC_CAMSWITCH_TR结构如下

Name	Type	Description
FirstOnPosition	REAL	Lower boundary where the switch is ON
LastOnPosition	REAL	Upper boundary where the switch is ON
AxisDirection	INT	Both (=0; Default); Positive (1); Negative (2)
CamSwitchMode	INT	Position based (=0; Default); Time based (=1)
Duration	TIME	Coupled to time based CamSwitchMode
bOn	BOOL	internal; used by MC_DigitalCamSwitch
CounterOff	UDINT	internal; used by MC_DigitalCamSwitch

用户可设置一MC_CAMSWITCH_TR结构体数组类型的变量，再将MC_CAMSWITCH_REF的CamSwitchPtr指针指向这一数组，开关点数量填入NoOfSwitches，即可完成开关点的设置。

使能后可实现给定方向和位置上的开关量输出

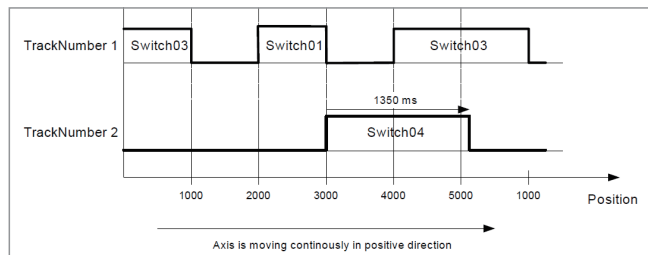
• 时序图

以表中数据为例

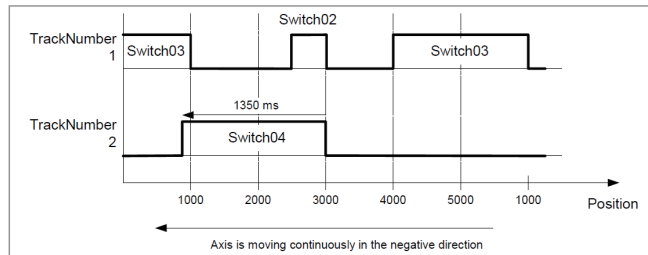
Parameter	Type	Switch 01	Switch 02	Switch 03	Switch 04	...	Switch N
TrackNumber	INTEGER	1	1	1	2		
FirstOnPosition [u]	REAL	2000	2500	4000	3000	—	
LastOnPosition [u]	REAL	3000	3000	1000	—		

Parameter	Type	Switch 01	Switch 02	Switch 03	Switch 04	...	Switch N
AxisDirection	INTEGER	1=Pos	2=Neg	0=Both	0=Both		
CamSwitchMode	INTEGER	0=Position	0=Position	0=Position	1=Time		
Duration	TIME	—	—	—	1350		

若运行于正方向可参考如图



若运行于反方向

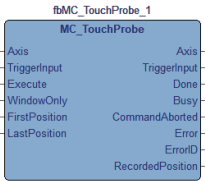


实际运行时，由于轴速度不同，开关量信号可能存在不同程度误差

3. 4. 2. 23 MC_TouchProbe

3. 4. 2. 23. 1 功能块接口说明

Instructions	LD Performance	ST Performance
		fbMC_TouchProbe_1 (Axis:=(* AXIS_REF *), TriggerInput:=(* MC_TRIGGER_REF *), Execute:=(* BOOL *),

Instruc-tions	LD Performance	ST Performance
MC_TouchProbe		WindowOnly:=(* BOOL *), FirstPosition:=(* REAL *), LastPosition:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *), RecordedPosition=>(* REAL *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
	TriggerInput	MC_TRIGGER_REF	探针来源
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Window Only	BOOL	是否仅检测位置窗口内的探针源
	First Position	REAL	窗口位置起始值[u].
	Last Position	REAL	窗口位置终止值[u].
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码
	Recorded Position	REAL	探针触发时的记录位置 [u]

3.4.2.23.2 功能块使用说明

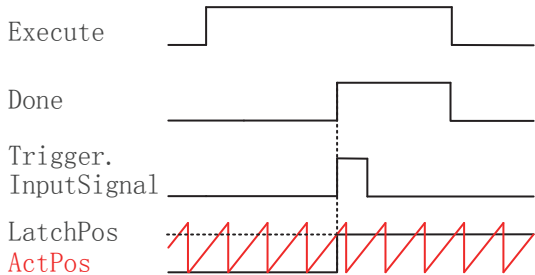
MC_TouchProbe指令适用于总线伺服轴和虚拟轴，用于记录触发事件的轴位置，指令上升沿有效

• 使用说明

功能块触发时，功能块锁存的位置清零，当检测到输入Trigger.InputSingal上升沿时，记录此时轴位置，功能块执行完毕

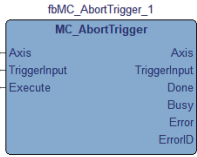
输入引脚WindowOnly开启触发事件窗口限制，具体的限制由输入引脚FirstPosition和LastPosition设定

• 时序图



3.4.2.24 MC_AbortTrigger

3.4.2.24.1 功能块接口说明

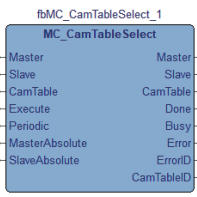
Instruc-tions	LD Performance	ST Performance
MC_AbortTrigger		fbMC_AbortTrigger_1 (Axis:=(* AXIS_REF *), TriggerInput:=(* MC_TRIGGER_REF *), Execute:=(* BOOL *), Done => (* BOOL *), Busy => (* BOOL *), Error => (* BOOL *), ErrorID =>(* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	轴工艺对象
	Trigger Input	MC_TRIG- GER_REF	探针来源
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	MC_ER- RORCODE	轴故障代码

3.4.3 Cam and Gear 凸轮齿轮指令

3.4.3.1 MC_CamTableSelect

3.4.3.1.1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_CamTableSelect		<pre>fbMC_CamTableSelect_1 (Master:=(* AXIS_REF *), Slave:=(* AXIS_REF *), CamTable:= (* MC_CAM_REF *), Execute:=(* BOOL *), Periodic:=(* BOOL *), MasterAbsolute:= (* BOOL *), SlaveAbsolute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), CamTableID=> (* MC_CAM_ID *));</pre>

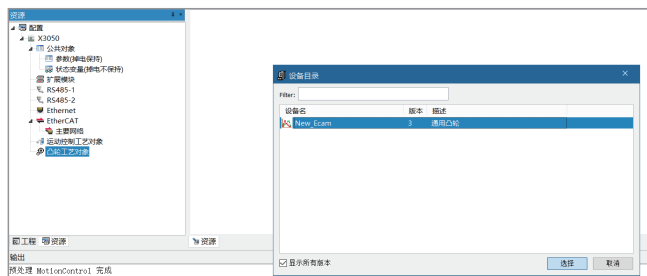
引脚类型	名称	类型	描述
VAR_IN_OUT	Master	AXIS_REF	主轴轴工艺对象
	Slave	AXIS_REF	从轴轴工艺对象
	CamTable	MC_CAM_REF	凸轮表对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
	Periodic	BOOL	周期运行1 = periodic, 0 = non periodic (single-shot)
	Master Absolute	BOOL	主轴绝对1 = absolute; 0 = relative coordinates
	Slave Absolute	BOOL	从轴绝对1 = absolute; 0 = relative coordinates
	Done	BOOL	功能执行完成
VAR_OUTPUT	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码
	CamTable ID	MC_CAM_ID	可供MC_CamIn使用的凸轮表ID

3.4.3.1.2 功能块使用说明

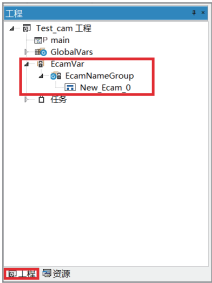
MC_CamTableSelect指令将凸轮表界面编辑生成的CamTable转化为MC_CamIn可使用的CamTableID, 指令上升沿触发。

• 使用说明

功能块CamTable引脚, 输入凸轮表变量, 凸轮表可由界面创建



创建后在工程栏会对应生成可供用户使用的凸轮表变量



相对绝对位置模式选择

主轴绝对MasterAbsolute: 此选项为True时，从轴参考主轴的实际位置对应凸轮表上的主轴位置来获取目标位置，否则参考主轴的相对位置来获取目标位置

从轴绝对SlaveAbsolute: 此选项为True且对应MC_CamIn的Startmode选择mcAbsolute时，从轴的目标位置与凸轮表对应位置一致，否则为凸轮表对应位置的相对值

3. 4. 3. 2 MC_CamIn

3. 4. 3. 2. 1 功能块接口说明

Instructions	LD Performance	ST Performance
		fbMC_CamIn_1 (Master:=(* AXIS_REF *), Slave:=(* AXIS_REF *), Execute:=(* BOOL *), MasterOffset:=(* REAL *), SlaveOffset:=(* REAL *), MasterScaling:=(* REAL *), SlaveScaling:=(* REAL *), MasterStartDistance:= (* REAL *),

Instructions	LD Performance	ST Performance
MC_CamIn		MasterSyncPosition:= (* REAL *), StartMode:= (* MC_START_MODE *), MasterValueSource:= (* MC_SOURCE *), CamTableID:= (* MC_CAM_ID *), BufferMode:= (* MC_BUFFER_MODE *), InSync=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), EndOfProfile=>(* BOOL *);

引脚类型	名称	类型	描述
VAR_IN_OUT	Master	AXIS_REF	主轴轴工艺对象
	Slave	AXIS_REF	从轴轴工艺对象
	Execute	BOOL	触发信号，上升沿有效
	Master Offset	LREAL	主轴偏移[u]
	Slave Offset	LREAL	从轴偏移[u]
	Master Scaling	LREAL	主轴缩放比例
	Slave Scaling	LREAL	从轴缩放比例

引脚类型	名称	类型	描述
VAR_INPUT	Master Start Distance	LREAL	主轴过渡区间 (进入同步过程) 的距离
	Master Sync Position	LREAL	主轴的同步位置, 对应凸轮表上的实际位置
	Start Mode	MC_START_MODE	枚举类型, 同步模式, 可选: mcAbsolute, mcRelative, or mcRampIn
	Master Value Source	MC_SOURCE	主轴信息来源, 可选: mcSetValue - Synchronization on master set value mcActualValue - Synchronization on master actual value
	Cam TableID	MC_CAM_ID	MC_CamTableSelect输出凸轮表ID
VAR_OUTPUT	Buffer Mode	MC_BUFFER_MODE	缓冲模式给定, 请参考 3.4.1.3 BufferMode说明
	InSync	BOOL	从轴与主轴进入同步
	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成, 且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码
	EndOf Profile	BOOL	每一段凸轮曲线的结束位置标志

3.4.3.2.2 功能块使用说明

MC_CamIn指令啮合啮合凸轮, 使轴进入凸轮同步,

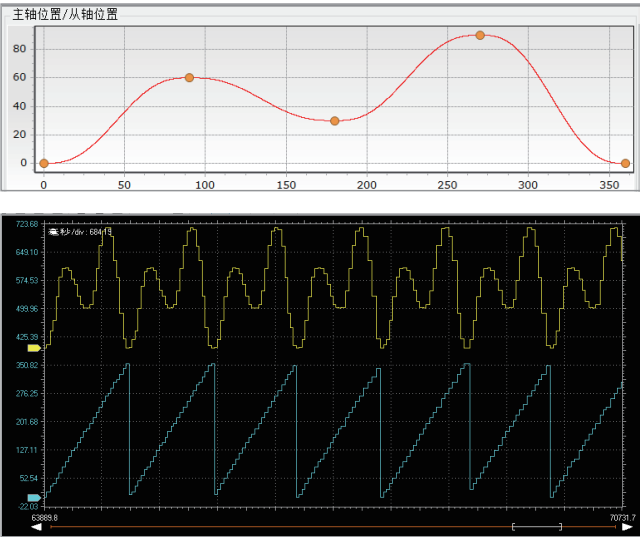
- 使用说明
- 引脚说明:

MC_CamIn功能块执行对主轴状态无要求
执行MC_CamIn功能块时, 若主从位置不对应用户设置偏移量, 将在过渡阶段自动处理偏差
- 相对与绝对:

请参考MC_CamTableSelect中的说明
- 缩放与偏移:

主轴缩放和偏移: 此时的主轴位置乘以主轴缩放, 并加上主轴偏移量后, 即可根据凸轮表, 得到从轴对应的位置
从轴缩放和偏移: 根据凸轮表初步得到从轴位置后, 对此位置值进行处理, 进行缩放和偏移, 得到输出从轴位置
- MasterValueSource: 主轴位置来源, 分别为 mcSetValue设定值和mcActualValue实际值

• 时序图



3. 4. 3. 3 MC_CamOut

3. 4. 3. 3. 1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_CamOut	fbMC_CamOut_1 MC_CamOut Slave Execute Slave Done Busy Error ErrorID	fbMC_CamOut_1 (Slave:=(* AXIS_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Slave	AXIS_REF	脱离同步的从轴轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号，上升沿有效
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

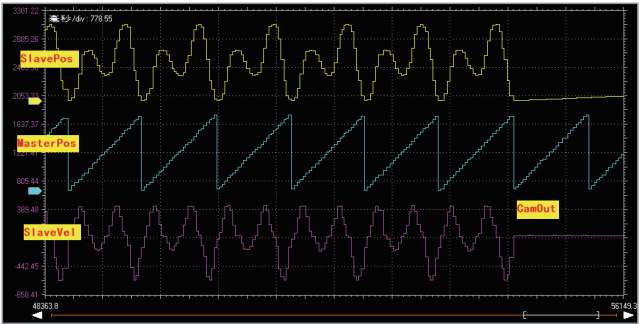
3. 4. 3. 3. 2 功能块使用说明

• 使用说明

MC_CamOut指令适用于总线伺服轴和虚拟轴，用于打断从轴的凸轮同步状态，Execute上升沿触发，从轴会以触发时的速度持续运行，轴状态机从同步运动跳转到持续运动。

注意:MC_CamOut触发后，尽管从轴还在以一定速度运动，但实际上MC_CamOut功能块功能执行完毕后将不会继续控制轴运动。

• 时序图



3. 4. 3. 4 MC_GearIn

3. 4. 3. 4. 1 功能块接口说明

Instructions	LD Performance	ST Performance
MC_GearIn	fbMC_GearIn_1 MC_GearIn Master Slave Execute InGear RatioNumerator RatioDenominator MasterValueSource Acceleration Deceleration Jerk BufferMode Master Slave InGear Active Busy CommandAborted Error ErrorID	fbMC_GearIn_1 (Master:=(* AXIS_REF *), Slave:=(* AXIS_REF *), Execute:=(* BOOL *), RatioNumerator:=(* DINT *), RatioDenominator:= (* UDINT *), MasterValueSource:= (* MC_SOURCE *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), InGear=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Master	AXIS_REF	主轴轴工艺对象
	Slave	AXIS_REF	从轴轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号,上升沿有效
	Ratio Numerator	LREAL	齿轮比分子（从轴/主轴）
	RatioDenominator	LREAL	齿轮比分母（从轴/主轴）
	MasterValueSource	MC_SOURCE	枚举类型，主轴信息来源： mcSetValue - Synchronization on master set value mcActualValue - Synchronization on master actual value
	Acceleration	LREAL	进入同步前的加速度 [u/s2]
	Deceleration	LREAL	进入同步前的减速度 [u/s2]
	Jerk	LREAL	进入同步前的加加速度 [u/s3]
	Buffer Mode	MC_BUFFER_MODE	缓冲模式给定, 请参考 3.4.1.3 BufferMode说明
VAR_OUTPUT	InGear	BOOL	从轴与主轴进入同步
	Busy	BOOL	输入参考已知并设置完成
	Active	BOOL	功能块功能未执行完成，且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

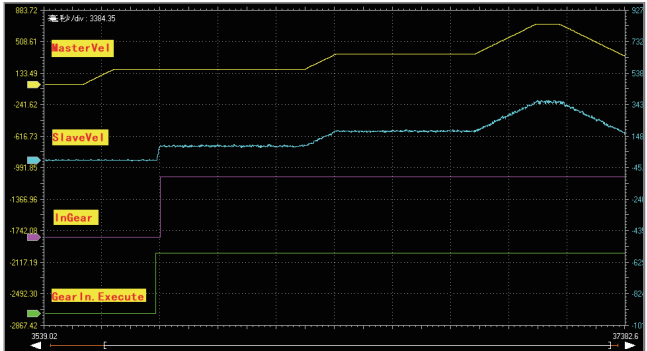
3.4.3.4.2 功能块使用说明

MC_GearIn指令适用于总线伺服轴和虚拟轴

• 使用说明

开始动作后，从轴以主轴速度乘以齿轮比得到的速度为目标速度，进行加减速动作。
从轴速度与主轴速度满足齿轮比前的阶段是过渡阶段，主从轴速度满足齿轮比后进入同步阶段
齿轮比为正，从轴与主轴同方向运动，齿轮比为负，从轴与主轴反方向运动
块输入加减速及加加速度均规定的是进入同步前的最大加减速速度，也即过渡阶段的加减速速度
进入同步上从轴跟随主轴运动，此时的加减速不应收到限制

• 时序图



3.4.3.5 MC_GearInPos

3.4.3.5.1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
		fbMC_GearInPos_1 (Master:=($\ast$ AXIS_REF $\ast$), Slave:=($\ast$ AXIS_REF $\ast$), Execute:=($\ast$ BOOL $\ast$), RatioNumerator:=($\ast$ DINT $\ast$),

Instructions	LD Performance	ST Performance
MC_Gear InPos		<pre>RatioDenominator:= (* UDINT *), MasterValueSource:= (* MC_SOURCE *), MasterSyncPosition:= (* REAL *), SlaveSyncPosition:= (* REAL *), SyncMode:= (* MC_SYNC_MODE *), MasterStartDistance:= (* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), StartSync=>(* BOOL *), InSync=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error => (* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Master	AXIS_REF	主轴轴工艺对象
	Slave	AXIS_REF	从轴轴工艺对象
	Execute	BOOL	触发信号,上升沿有效
	RatioNumerator	LREAL	齿轮比分子（从轴/主轴）
	RatioDenominator	LREAL	齿轮比分母（从轴/主轴）

引脚类型	名称	类型	描述
VAR_INPUT	MasterValueSource	MC_SOURCE	枚举类型，主轴信息来源： mcSetValue - Synchronization on master set value mcActualValue - Synchronization on master actual value
	MasterSyncPosition	LREAL	主轴的同步位置
	SlaveSyncPosition	LREAL	从轴的同步位置
	SyncMode	MC_SYNC_MODE	枚举类型，同步模式，可选 (like 'mcShortest'; 'mcCatchUp'; 'mcSlowDown').
	MasterStartDistance	LREAL	主轴过渡区间（进入同步过程）的距离
	Velocity	LREAL	开始同步到同步阶段的最大速度 [u/s].
	Acceleration	LREAL	开始同步到同步阶段的最大加速度 [u/s2]
	Deceleration	LREAL	开始同步到同步阶段的最大减速度 [u/s2]
	Jerk	LREAL	开始同步到同步阶段的最大加加速度 [u/s3]
	BufferMode	MC_BUFFER_MODE	缓冲模式给定,请参考 3.4.1.3 BufferMode说明
	StartSync	BOOL	开始同步
	InGear	BOOL	从轴与主轴进入同步
	Busy	BOOL	输入参考已知并设置完成

引脚类型	名称	类型	描述
VAR_OUTPUT	Active	BOOL	功能块功能未执行完成, 且将生成新的输出值
	Command Aborted	BOOL	被打断标志位
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

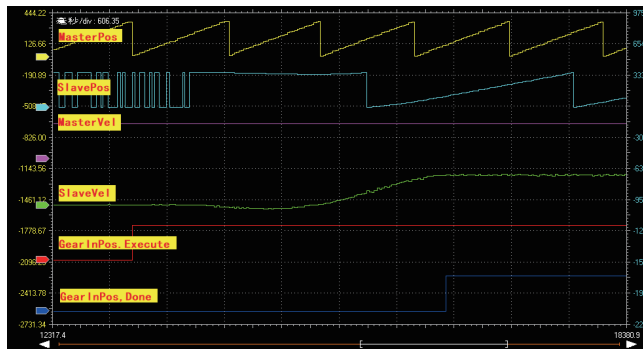
3. 4. 3. 5. 2 功能块使用说明

MC_GearInPos指令适用于总线伺服轴和虚拟轴

• 使用说明

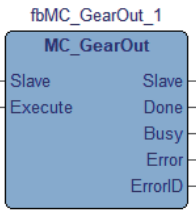
与MC_GearIn类似, MC_GearInPos目标是控制从轴与主轴进入电子齿轮耦合, 按照给定齿轮比运行相比MC_GearIn, MC_GearInPos指定了进入同步的主轴位置, 从轴位置, 主轴开始同步距离, 以此来完成切入电子齿轮动作

• 时序图



3. 4. 3. 6 MC_GearOut

3. 4. 3. 6. 1 功能块接口说明

Instruc-tions	LD Performance	ST Performance
MC_GearOut		fbMC_GearOut_1 (Slave:=(* AXIS_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Slave	AXIS_REF	脱离同步的从轴轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号, 上升沿有效
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号, true为故障, false为无故障
	ErrorID	WORD	轴故障代码

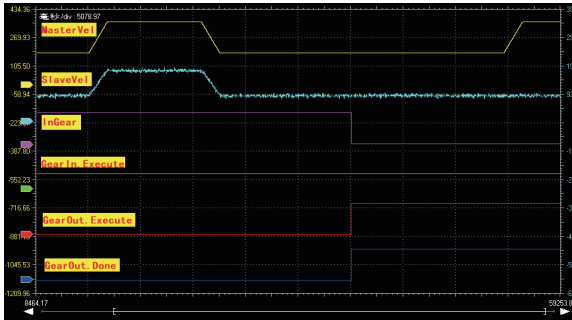
3. 4. 3. 6. 2 功能块使用说明

MC_GearOut指令适用于总线伺服轴和虚拟轴

• 使用说明

MC_GearOut指令用于分离以电子齿轮方式相互耦合的主从轴, 从轴与主轴分离后, 会按照MC_GearOut指令触发时的速度持续运行, 在后续持续运行期间, MC_GearOut功能块将不再控制轴的动作

• 时序图



3.4.4 Encoder编码器轴

编码器轴作为轴对象使用MC基础功能块，仅可作为主从（Cam and Gear）功能块的主轴，或使用专用的ENC功能块，实现位置的设置与读取。

3.4.4.1 ENC_Counter

ENC_Counter功能块用于读取编码器轴位置，速度和限位信息，可以设置位置变化的方向，高电平有效

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	编码器轴轴工艺对象
VAR_INPUT	Enable	BOOL	使能，高电平时功能块生效
	Direction	BOOL	true则计数值增加，false则计数值减小
VAR_OUTPUT	Valid	BOOL	有效标志位，true时，功能块输出状态可信
	Busy	BOOL	输入参考已知并设置完成
	Position	REAL	实际位置值[u]
	Velocity	REAL	实际速度[u/s]
	Positive Limit	BOOL	线性模式到达正向限位
	Negative Limit	BOOL	线性模式到达反向限位
	Error	BOOL	轴故障信号，true为故障，false为无故障
	ErrorID	MC_ER-RORCODE	轴故障代码

3.4.4.2 ENC_Preset

ENC_Preset功能块用于设置编码器轴的初始值预设值，上升沿触发后，这一位置值设置进入编码器轴的初始值预设值，此时，这一值仍未生效，需要用户调用ENC_Reset功能块后方可将当前位置复位为预设的初始位置值。

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	编码器轴轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号,上升沿有效
	Position	REAL	预设位置值[u]
VAR_OUTPUT	Done	BOOL	设定预设值完成
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号，true为故障，false为无故障
	ErrorID	MC_ER-RORCODE	轴故障代码

3.4.4.3 ENC_Reset

ENC_Reset功能块用于复位编码器轴位置，将当前位置设为预设位置值，注意，这一功能块上升沿触发，且需要在ENC_Counter功能块的Enable引脚处于低电平时触发才有效，也就是说，复位编码器轴位置时，不可以读取编码器轴的位置。

引脚类型	名称	类型	描述
VAR_IN_OUT	Axis	AXIS_REF	编码器轴轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号,上升沿有效
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	输入参考已知并设置完成
	Error	BOOL	轴故障信号，true为故障，false为无故障
	ErrorID	MC_ERRORCODE	轴故障代码

3.4.5 MotionGroupControl轴组控制

3.4.5.1 MotionGroupControl库介绍

MotionGroupControl库包含博能自主研发的MC功能块，符合PLCopen规范，实现了基础指令和主从指令部分

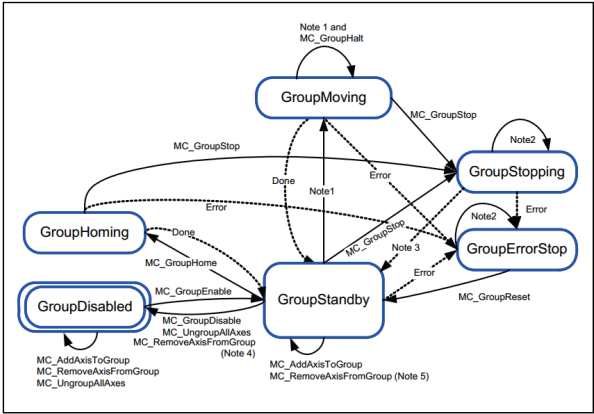
3.4.5.2 指令列表

运动控制基本指令包含如下：

指令	功能	
管理类指令	MC_AddAxisToGroup	将轴添加到轴组中
	MC_GroupReadActualPosition	读取轴组位置指令
	MC_GroupReadActualVelocity	读取轴组速度指令
	MC_GroupSetPosition	设置轴组定位指令
	MC_GroupReset	轴组复位指令
	MC_ReadCoordinateTransform	读取MCS和PCS之间活动的坐标变换
	MC_ReadKinTransform	读取ACS和MCS之间活动的运动学变换
	MC_GroupReadConfiguration	读取轴组配置指令
	MC_GroupReadError	读取轴组错误码指令
	MC_GroupReadStatus	读取轴组状态指令
	MC_SetCoordinateTransform	设置MCS和PCS之间的坐标变换。
	MC_SetkinTransform	设置ACS和MCS之间的运动学变换
	MC_UngroupAllAxes	从轴组中删除所有轴
	MC_RemoveAxisFromGroup	从轴组中删除对应的轴
	MC_GroupContinue	轴组按照原来的轨迹继续运行
	MC_GroupDisable	失能轴组指令

指令	功能	
运动类指令	MC_GroupEnable	使能轴组指令
	MC_GroupHalt	轴组运动中中止指令
	MC_GroupStop	轴组停止
	MC_GroupHome	轴组回到原点指令
	MC_Groupinterrupt	中断原来的运动，并执行新的运动
	MC_MoveCircularAbsolute	轴组在绝对模式下进行圆弧运动指令
	MC_MoveCircularRelative	轴组在相对模式下进行圆弧运动指令
	MC_MoveDirectAbsolute	轴组在绝对模式下进行点到点运动指令
	MC_MoveDirectRelative	轴组在相对模式下进行点到点运动指令
	MC_MoveLinearAbsolute	轴组在绝对模式下进行直线运动指令
	MC_MoveLinearRelative	轴组在相对模式下进行直线运动指令

3.4.5.3 PLCopen状态机



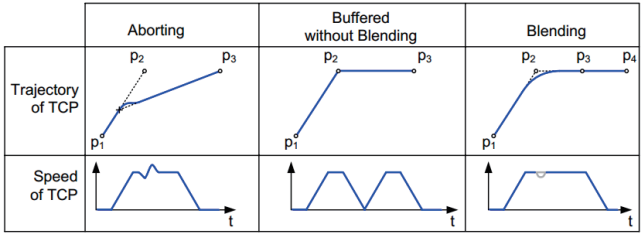
状态机详细描述如下：

状态	功能描述
GroupDisabled	轴组未使能状态
GroupErrorStop	轴组故障停机状态
GroupStandby	轴组使能状态
GroupHoming	轴组原点回归状态
GroupStopping	轴组停止状态
GroupMoving	轴组运动状态

- 注1：适用于所有非管理（移动）功能块。
- 注2：在GroupErrorStop或GroupStopping状态下，可以调用所有功能块，尽管它们不会被执行，但GroupErrorStop和任何发生的错误的MC_GroupReset除外——它们将分别生成到GroupStandby或GroupErrorStop的转换。
- 注3：MC_GroupStop。已完成，而非MC_GroupStop。
- 注4：如果从组中删除最后一个轴，则转换适用。
- 注5：当组不为空时，转换适用。
- 注6：MC_GroupDisable和MC_UngroupAllAxes可以在所有状态下发出，并将状态更改为GroupDisabled。

3.4.5.4 BufferMode和TransitionModes说明

轴组之间的两个功能块存在过度模式，分别由BufferMode和TransitionModes两个接口影响。主要产生的效果如



BufferMode接口说明：

部分运动功能块具有BufferMode的引脚，其意为，缓冲模式，规定了两运动功能块之间的衔接模式。

Name	Value	Description
Aborting	0	立即启动FB（默认模式）
Buffered	1	在当前运动完成后开始FB
BlendingLow	2	速度与两个FB的最低速度混合
BlendingPrevious	3	速度与第一个FB的速度混合
BlendingNext	4	速度与第二个FB的速度混合
BlendingHigh	5	速度与两个FBS的最高速度混合

TransitionModes接口说明：

根据过渡模式，可以给出不同的过渡参数，这些参数表征了轮廓曲线。

Name	Value	Description
TMNone	0	不插入过渡曲线（默认模式）
TMStartVelocity	1	给定起动速度下的过渡
TMConstantVelocity	2	给定恒速下的过渡
TMCornerDistance	3	给定转角距离的过渡
TMCornerDistance	4	给定最大转角偏差的过渡

上述接口配合下的可用转换模式如下：

Buffer Transi- Mode tion Mode	Abort- ing	Buff- ered	Blend- ing Low	Blending Previous	Blend- ing Next	Blend- ing High
TMNone	A	A	N	N	N	N
TMMax Velocity	D	D	D	D	D	D
TMDefined Velocity	A	N	A	A	A	A
TMCorner Distance	N	N	A	A	A	A
TMMaxCorner Deviation	N	N	A	A	A	A

代表可以使用

N:代表不能使用

D:代表TransitionModes接口无作用

3.4.5.5 CoordSystem接口说明

这里为坐标系接口,表示输入的参数相对于此坐标系。

ACS: 即输入的参数为相对于关节轴参数;

MCS: 即输入的参数为相对于轴组基坐标系的参数;

PCS: 即输入的参数为相对于工件坐标系的参数;

3.4.5.6 CircMode和PathChoice接口说明

这个接口针对圆弧规划会提供。主要对AuxPoint和EndPoint的定义使用。其中轴组末端当前点和Aux-Point、EndPoint三个点不能共线,共线会报错。CircMode提供如下参数: BORDER:三点圆弧模式,其中AuxPoint为圆弧经过点,三点构建的圆弧会经过这个点,此时PathChoice失效。

CENTER:圆心圆弧模式,其中AuxPoint为圆心,EndPoint为终止点,PathChoice为确定圆弧为顺时针还是逆时针方向。

SEMI:半圆模式,其中AuxPoint为确定圆所在的平面,圆弧不会经过此点,也不为圆心,圆心为起点和EndPoint的中点,PathChoice为确定圆弧为顺时针还是逆时针方向。

FULL:整圆模式,其中AuxPoint为确定圆心,EndPoint为确定圆所在的平面,PathChoice为确定圆弧为顺时针还是逆时针方向。

3.4.5.7 故障代码说明

用户可通过MotionControl库的数据类型中的枚举类型找到故障代码的说明,通过属性窗口,或双击打开查看

Enumeration: MC_ERRORCODE (ver. 1.0.0) Location: (库) MotionControl			
Name	Value	Description	
NO_ERROR	0	无错误	
AXIS_NOT_EXIST	1	轴ID号不存在	
AXIS_TYPE_NOT_MATCH	2	轴类型不匹配	
AXIS_QUEUE_FULL	3	轴队列已满	
AXIS_ENCODER_OVERFLOW	4	轴编码器溢出	

96	GROUP_QUEUE_5005	轴组队列已满
97	GROUP_DISABLED_5006	轴组未使能
98	GROUP_AXIS_DISABLED_5007	轴组的轴成员未使能
99	GROUP_AXIS_ERR_5008	轴组的轴成员错误
100	GROUP_AXIS_0_ERR_5009	轴组的轴0成员错误
101	GROUP_AXIS_1_ERR_5010	轴组的轴1成员错误
102	GROUP_AXIS_2_ERR_5011	轴组的轴2成员错误
103	GROUP_AXIS_3_ERR_5012	轴组的轴3成员错误
104	GROUP_SYNC_AXIS_0_ERR_5013	轴组的同步轴0成员错误
105	GROUP_SYNC_AXIS_1_ERR_5014	轴组的同步轴1成员错误
106	GROUP_SYNC_AXIS_2_ERR_5015	轴组的同步轴2成员错误
107	GROUP_AXIS_0_HOME_FAILED_5016	轴组的轴0回零失败
108	GROUP_AXIS_1_HOME_FAILED_5017	轴组的轴1回零失败
109	GROUP_AXIS_2_HOME_FAILED_5018	轴组的轴2回零失败
110	GROUP_AXIS_3_HOME_FAILED_5019	轴组的轴3回零失败
111	GROUP_SYNC_AXIS_0_HOME_FAILED_5020	轴组的同步轴0回零失败
112	GROUP_SYNC_AXIS_1_HOME_FAILED_5021	轴组的同步轴1回零失败
113	GROUP_SYNC_AXIS_2_HOME_FAILED_5022	轴组的同步轴2回零失败
114	GROUP_AXIS_MODE_ERR_5023	轴组成员不允许为模态轴
115	GROUP_FSM_DISABLED_5100	PLCOpen状态机.DISABLED
116	GROUP_FSM_STANDBY_5101	PLCOpen状态机.STANDBY
117	GROUP_FSM_HOMING_5102	PLCOpen状态机.HOMING

5 MC_CIRC_MODE MC_CIRC_PATHCHOICE MC_COORD_SYSTEM MC_TRANSITION

也可查询下表

故障名称	对应值	含义
PERMISTION_DENIED	5000	权限不允许
GROUP NOT EXIST	5001	轴组不存在
GROUP AXIS ARRAY FULL	5002	轴组空间已满
AXIS ALREADY IN GROUP	5003	轴已存在于组中
AXIS NOT IN GROUP	5004	轴组不包含该轴
GROUP QUEUE FULL	5005	轴组队列已满
GROUP DISABLED	5006	轴组未使能
GROUP_AXIS_DISABLED	5007	轴组的轴成员未使能
GROUP AXIS ERR	5008	轴组的轴成员错误
GROUP_AXIS_0_ERR	5009	轴组的轴0成员错误
GROUP_AXIS_1_ERR	5010	轴组的轴1成员错误
GROUP_AXIS_2_ERR	5011	轴组的轴2成员错误
GROUP_AXIS_3_ERR	5012	轴组的轴3成员错误
GROUP_SYNC_AXIS_0_ERR	5013	轴组的同步轴0成员错误
GROUP_SYNC_AXIS_1_ERR	5014	轴组的同步轴1成员错误
GROUP_SYNC_AXIS_2_ERR	5015	轴组的同步轴2成员错误
GROUP_AXIS_0_HOME_FAILED	5016	轴组的轴0回零失败
GROUP_AXIS_1_HOME_FAILED	5017	轴组的轴1回零失败

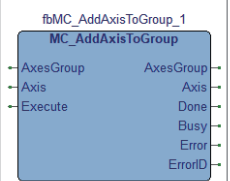
故障名称	对应值	含义
GROUP_AXIS_2_HOME_FAILED	5018	轴组的轴2回零失败
GROUP_AXIS_3_HOME_FAILED	5019	轴组的轴2回零失败
GROUP_SYNC_AXIS_0_HOME_FAILED	5020	轴组的同步轴0回零失败
GROUP_SYNC_AXIS_1_HOME_FAILED	5021	轴组的同步轴1回零失败
GROUP_SYNC_AXIS_2_HOME_FAILED	5022	轴组的同步轴2回零失败
GROUP_AXIS_MODULO_NOT_ALLOWED	5023	轴组成员不允许为模态轴
GROUP_FSM_DISABLED	5100	PLCopen状态机:DISABLED
GROUP_FSM_STANDBY	5101	PLCopen状态机:STANDBY
GROUP_FSM_HOMING	5102	PLCopen状态机:HOMING
GROUP_FSM_MOVING	5103	PLCopen状态机:MOVING
GROUP_FSM_STOPPING	5104	PLCopen状态机:STOPPING
GROUP_FSM_ERRORSTOP	5105	PLCopen状态机:ERRORSTOP
KINEMATIC_DEFINED	5140	运动学变换已定义
KINEMATIC_NOT_ALLOWED	5141	运动机构类型不允许
KIN_TRANSFORM_NOT_CONSTANT	5142	运动机构变换不唯一
GROUP_AIMPOUTLIMIT	5150	目标位置超限制
GROUP_KINPARAMERROR	5151	运动学参数限制不合理
GROUP_TRANMODEERROR	5152	过渡模式错误
GROUP_CIRCLEPARAERROR	5153	圆弧模式参数错误
GROUP_PLANFAIL	5154	S曲线规划失败
GROUP_PARAMERROR	5155	参数错误
GROUP_ACC_DEC_ILLEGA	5156	加减速不合法
GROUP_JERK_ILLEGAL	5157	加加速度不合法
GROUP_AIMEQUAL	5158	目标位置重合
GROUP_BLENDED_ERROR	5159	轨迹混合错误
GROUP_MOTION_LIMIT_AVOUT	5160	运动过程中关节角速度和角加速度超过限制

故障名称	对应值	含义
GROUP_MOTION_LIMIT_POSOUT	5161	运动过程中关节位置超限
GROUP_PLAN_ERROR	5162	轴组规划错误
GROUP_INTERPOLATE_ERROR	5180	轴组插补器错误
GROUP_SYNC_POSLAGOVERLIMIT	5181	同步轴跟随误差超限
GROUP_SYNC_CMDPOSLAGOVERLIMIT	5182	
GROUP_UNVALID_COMBINE_PARAMETER	5183	不支持的参数组合
GROUP_INTERRUPT_NODE_NOT_EXIST	5184	轴组中断运动不存在

3.4.5.8 管理类指令

3.4.5.8.1 MC_AddAxisToGroup

把轴添加到轴组

Instruc-tions	LD Performance	ST Performance
MC_AddAxisToGroup		<pre>fbMC_AddAxisToGroup_1 (AxesGroup:= (* AXES_GROUP_REF *), Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

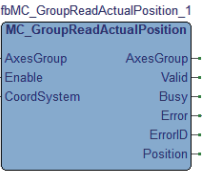
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	触发信号,上升沿有效
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	功能块执行还未结束
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块将一个轴添加到结构AxesGroup中的组中。这是一个管理FB，因为没有产生任何移动。无法缓冲该命令。

3. 4. 5. 8. 2 MC_GroupReadActualPosition

获取对应坐标系下的位置

Instruc-tions	LD Performance	ST Performance
MC_GroupReadActualPosition		<pre>fbMC_GroupReadActual- Position_1(AxesGroup:= (* AXES_GROUP_REF *), Enable:=(* BOOL *), CoordSystem:= (* MC_COORD_SYSTEM *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), Position=>(* REAL *));</pre>

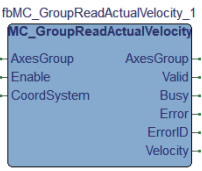
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Enable	BOOL	启用时，连续获取轴组在所选坐标系中的实际位置
	Coord System	MC_COORD_SYSTEM	参考坐标系（ACS、MCS、PCS）
VAR_OUTPUT	Valid	BOOL	如果有效输出可用，则为True
	Busy	BOOL	功能块执行还未结束
	Error	BOOL	轴组故障信号，true为故障，false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码
	Position	REAL	轴组的当前位置

功能块使用说明

此功能块返回轴组在选定坐标系中的实际位置。这是一个管理FB，因为没有产生任何移动。

3. 4. 5. 8. 3 MC_GroupReadActualVelocity

获取对应坐标系下的速度

Instruc-tions	LD Performance	ST Performance
MC_GroupReadActualPosition		<pre>fbMC_GroupReadActu- alVelocity_1(AxesGroup:= (* AXES_GROUP_REF *), Enable:=(* BOOL *), CoordSystem:= (* MC_COORD_SYSTEM *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), Velocity=>(* REAL *));</pre>

Instruc-tions	LD Performance	ST Performance
		Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), Velocity=>(* REAL *));

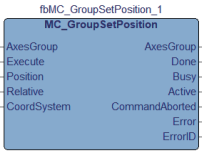
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Enable	BOOL	启用时，连续获取轴组在所选坐标系中的实际速度
	Coord System	MC_COORD_SYSTEM	参考坐标系（ACS、MCS、PCS）
VAR_OUTPUT	Valid	BOOL	如果有效输出可用，则为True
	Busy	BOOL	功能块执行还未结束
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码
	Velocity	REAL	轴组的当前速度

功能块使用说明

此功能块返回轴组在选定坐标系中的实际速度。这是一个管理FB，因为没有产生任何移动。

3. 4. 5. 8. 4 MC_GroupSetPosition

Instruc-tions	LD Performance	ST Performance
		fbMC_GroupSetPosition_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *),

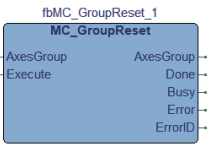
Instruc-tions	LD Performance	ST Performance
MC_Group Set Position		Position:=(* REAL *), Relative:=(* BOOL *), CoordSystem:= (* MC_COORD_SYSTEM *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	Position	REAL	设置的位置
	Relative	BOOL	位置输入模式-相对=true，绝对=false（默认）
	Coord System	MC_COORD_SYSTEM	参考坐标系（ACS、MCS、PCS）
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表示FB对轴组具有控制权
	Command Aborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块在不移动轴的情况下设置组中所有轴的位置。新坐标以数组形式描述。输入坐标系后，选择相应的坐标系。它可以被视为一种引用或转换的方式。MC_GroupSetPosition会移动所寻址坐标系的位置，并影响更高级别的坐标系（因此，如果选择ACS，MCS和PCS会受到影响）。

3. 4. 5. 8. 5 MC_GroupReset

Instruc-tions	LD Performance	ST Performance
MC_Group Reset		<pre>fbMC_GroupReset_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *)</pre>

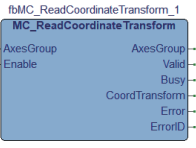
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
VAR_OUTPUT	Done	BOOL	功能执行完成
	Busy	BOOL	功能块执行还未结束
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块通过重置所有内部组相关错误，从状态GroupErrorStop转换为GroupStandby——它不会影

响FB实例的输出。此功能块还会重置此组中的所有轴，如MC_Reset。

3. 4. 5. 8. 6 MC_ReadCoordinateTransform

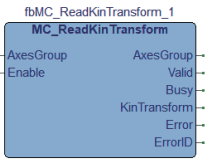
Instruc-tions	LD Performance	ST Performance
MC_Group Reset		<pre>fbMC_ReadCoordinateTr- ansform_1(AxesGroup:= (* AXES_GROUP_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), CoordTransform=> (* MC_COORD_REF *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *)</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Enable	BOOL	启用时,连续获取轴组的实际坐标变换参考
VAR_OUTPUT	Valid	BOOL	如果有效输出可用,则为True
	Busy	BOOL	功能块执行还未结束
	Coord Transform	MC_COORD_REF	参考坐标变换
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块读取MCS和PCS之间活动的坐标变换。

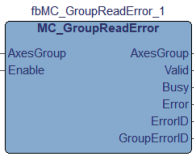
3. 4. 5. 8. 7 MC_ReadKinTransform

Instruc- tions	LD Performance	ST Performance
MC_Group Reset		fbMC_ReadKinTrans- form_1(AxesGroup:= (* AXES_GROUP_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), KinTransform=> (* MC_KIN_REF *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_ GROUP_REF	轴组工艺对象
VAR_INPUT	Enable	BOOL	启用时,连续获取轴组 的实际坐标变换参考
VAR_OUTPUT	Valid	BOOL	如果有效输出可用, 则为True
	Busy	BOOL	功能块执行还未结束
	Coord Transform	MC_COORD_ REF	参考坐标变换
	Error	BOOL	轴组故障信号,true为 故障,false为无故障
	ErrorID	MC_ ERRORCODE	轴组故障代码

功能块使用说明
此功能块读取ACS和MCS之间的运动学变换。

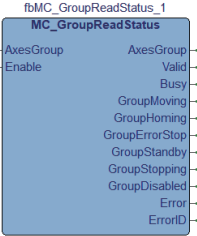
3. 4. 5. 8. 8 MC_GroupReadError

Instruc- tions	LD Performance	ST Performance
MC_Group Read Error		fbMC_GroupReadError_1 (AxesGroup:=(* AXES_GROUP_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), GroupErrorID=> (* WORD *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_ GROUP_REF	轴组工艺对象
VAR_INPUT	Enable	BOOL	启用时,连续获取轴 组的错误
VAR_OUTPUT	Valid	BOOL	如果有效输出可用, 则为True
	Busy	BOOL	功能块执行还未结束
	Error	BOOL	轴组故障信号,true为 故障,false为无故障
	ErrorID	MC_ ERRORCODE	轴组故障代码
	Group ErrorID	WORD	轴组错误的值。

功能块使用说明
此功能块描述了与功能块无关的一般轴组错误。这是一个管理FB, 因为没有产生任何移动。

3. 4. 5. 8. 9 MC_GroupReadStatus

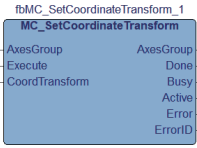
Instructions	LD Performance	ST Performance
MC_Group Read Status		<pre>fbMC_GroupReadStatus_1(AxesGroup:= (* AXES_GROUP_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), GroupMoving=>(* BOOL *), GroupHoming=>(* BOOL *), GroupErrorStop=> (* BOOL *), GroupStandby=>(* BOOL *), GroupStopping=>(* BOOL *), GroupDisabled=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_ GROUP_REF	轴组工艺对象
VAR_INPUT	Enable	BOOL	启用时连续获取轴组的状态
VAR_OUTPUT	Valid	BOOL	如果有效输出可用,则为True
	Busy	BOOL	功能块执行还未结束
	Group Moving	BOOL	参见组状态图
	Group Homing	BOOL	参见组状态图
	Group ErrorStop	BOOL	参见组状态图
	Group Standby	BOOL	参见组状态图

引脚类型	名称	类型	描述
	Group Stopping	BOOL	参见组状态图
	Group Disabled	BOOL	参见组状态图
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ ERRORCODE	轴组故障代码

功能块使用说明
此功能块根据活动的组FB返回轴组的状态。这是一个管理FB, 因为没有产生任何移动。

3. 4. 5. 8. 10 MC_SetCoordinateTransform

Instructions	LD Performance	ST Performance
MC_Set Coordinate Transform		<pre>fbMC_SetCoordinateTransform_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), CoordTransform:= (* MC_COORD_REF *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_ GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发

引脚类型	名称	类型	描述
	Coord Transform	MC_COORD_REF	参考坐标变换。
VAR_OUTPUT	Done	BOOL	转换设置成功
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表示FB已处理
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块设置MCS和PCS之间的坐标变换。

3. 4. 5. 8. 11 MC_SetkinTransform

Instruc-tions	LD Performance	ST Performance
MC_Setkin Transform		<pre>fbMC_SetkinTransform_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), KinTransform:= (* MC_KIN_REF *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

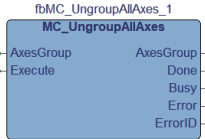
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发

引脚类型	名称	类型	描述
	KinTransform	MC_KIN_REF	参考运动学模型。
VAR_OUTPUT	Done	BOOL	转换设置成功
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表示FB已处理
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块根据预定义的运动学模型设置ACS和MCS之间的运动学变换。

3. 4. 5. 8. 12 MC_UngroupAllAxes

Instruc-tions	LD Performance	ST Performance
MC_Setkin Transform		<pre>fbMC_UngroupAllAxes_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

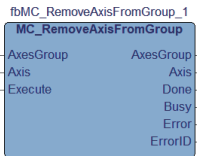
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	Done	BOOL	转换设置成功
	Busy	BOOL	功能块执行还未结束
	Error	BOOL	轴组故障信号,true为故障,false为无故障

引脚类型	名称	类型	描述
	ErrorID	MC_ ERRORCODE	轴组故障代码

功能块使用说明

此功能块从AxesGroup组中删除所有轴。这是一个管理FB，因为没有产生任何移动。无法缓冲该命令。完成后，状态更改为GroupDisabled

3. 4. 5. 8. 13 MC_RemoveAxisFromGroup

Instruc- tions	LD Performance	ST Performance
MC_ Remove AxisFrom Group		<pre>fbMC_RemoveAxis- FromGroup_1(AxesGroup:= (* AXES_GROUP_REF *), Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

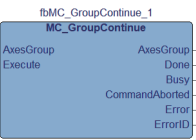
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_ GROUP_REF	轴组工艺对象
	Axis	AXIS_REF	轴工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
VAR_OUTPUT	Done	BOOL	转换设置成功
	Busy	BOOL	功能块执行还未结束
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ ERRORCODE	轴组故障代码

功能块使用说明

此功能块从AxesGroup组中删除一个轴。这是一个管理FB，因为没有产生任何移动。无法缓冲该命令。如果组中没有剩余轴，则状态将更改为GroupDisabled。

3. 4. 5. 9 运动类指令

3. 4. 5. 9. 1 MC_GroupContinue

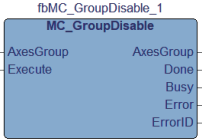
Instruc- tions	LD Performance	ST Performance
MC_Group Continue		<pre>fbMC_GroupContinue_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_ GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Command Aborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ ERRORCODE	轴组故障代码

功能块使用说明

此功能块将程序转移回发出MC_GroupInterrupt时的情况。它在内部使用在发出MC_GroupInterrupt时存储的数据集，并在最后（输出DONE集）将组上的控制权转移回原始FB，在轴组上进行移动，这也意味着在原始中断的FB上，输出Busy仍然很高，并且再次设置为Active。

3.4.5.9.2 MC_GroupDisable

Instructions	LD Performance	ST Performance
MC_Group Disable		<pre>fbMC_GroupDisable_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

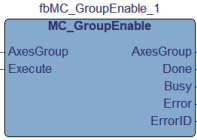
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块将组的状态更改为GroupDisabled，尽管它是一个管理FB，因为不会生成任何移动。如果在发出此命令时轴没有静止，则由应用程序采取必要的

预防措施。该命令不会影响组中任何单个轴的电源状态。

3.4.5.9.3 MC_GroupEnable

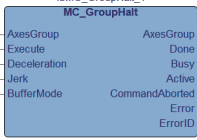
Instructions	LD Performance	ST Performance
MC_Group Enable		<pre>fbMC_GroupEnable_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Command Aborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块将组的状态从GroupDisabled更改为GroupStandby。这是一个管理FB，因为没有产生任何移动。

3. 4. 5. 9. 4 MC_GroupHalt

Instructions	LD Performance	ST Performance
MC_GroupHalt		<pre>fbMC_GroupHalt_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	AxesGroup	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	Deceleration	REAL	停止过程中的加速度值
	Jerk	REAL	停止过程中的加加速度值
	BufferMode	MC_BUFFER_MODE	参考上面BufferMode介绍
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表示FB对轴组具有控制权

引脚类型	名称	类型	描述
	CommandAborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块命令受控运动停止。它中止任何正在进行的功能块执行。

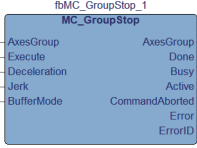
AxesGroup将移动到“GroupMoving”状态，直到速度为零。在设置了DONE输出后，状态将转移到GroupStandby。

MC_GroupHalt用于在正常操作条件下停止轴组。在非缓冲模式下：在轴组减速期间，可以设置另一个运动命令，这将中止MC_GroupHalt并立即执行。

- 如果此命令处于活动状态，则可以发出下一个命令。例如，无人驾驶车辆检测到障碍物并需要停车。MC_GroupHalt已发出。在达到静止状态之前，障碍物被移除，可以通过设置另一个运动命令继续运动，因此机器不会停止。
- 相关轴保持在未发出MC_GroupHalt的情况下执行的相同路径上。

3. 4. 5. 9. 5 MC_GroupStop

Instructions	LD Performance	ST Performance
		<pre>fbMC_GroupStop_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Deceleration:=(* REAL *), Jerk:=(* REAL *),</pre>

Instructions	LD Performance	ST Performance
MC_Group Halt		<pre> BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *)); </pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	Deceleration	REAL	停止过程中的加速度值
	Jerk	REAL	停止过程中的加加速度值
	Buffer Mode	MC_BUFFER_MODE	参考上面BufferMode介绍
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表示FB对轴组具有控制权
	Command Aborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

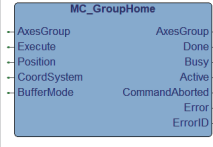
功能块使用说明

此功能块命令受控运动停止，并将轴组转换为

“GroupStopping” 状态。

它中止任何正在进行的功能块执行。当轴组处于 GroupStopping状态时，其他FB无法在同一轴组上执行任何运动。轴组达到速度零时，Done输出立即设置为TRUE。只要Execute仍然为TRUE或速度尚未达到零，轴组就会保持在“GroupStopping” 状态。一旦“完成” 设置为SET，“执行” 设置为FALSE，轴组就会进入“GroupStandBy” 状态。该命令只能通过MC_GroupDisable中止。

3. 4. 5. 9. 6 MC_GroupHome

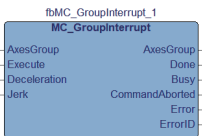
Instructions	LD Performance	ST Performance
MC_Group Home		<pre> fbMC_GroupHome_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Position:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *)); </pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
	Execute	BOOL	上升沿触发

引脚类型	名称	类型	描述
VAR_INPUT	Position	REAL	坐标数组，包括位置和方向
	Coord System	MC_COORD_SYSTEM	参考适用的坐标系：ACS、MCS、PCS
	Buffer Mode	MC_BUFFER_MODE	参考上面BufferMode介绍
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表示FB对轴组具有控制权
	Command Aborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障, false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明
将轴组回到原点。

3. 4. 5. 9. 7 MC_GroupInterrupt

Instruc-tions	LD Performance	ST Performance
MC_Group Interrupt		fbMC_GroupInterrupt_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	Deceleration	REAL	停止过程中的加速度值
	Jerk	REAL	停止过程中的加加速度值
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Command Aborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障, false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明
此功能块中断正在进行的运动并停止组移动，但不会中止中断的运动（这意味着在中断的FB上，输出CommandAborted将不会被设置，Busy仍然很高，Active被重置）。它在激活时在内部存储所有相关的轨迹或路径信息。

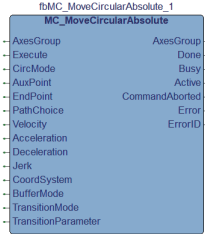
即使达到速度零和完成输出设置，轴组也会保持原始状态。

此FB与MC_GroupContinue耦合。发出MC_GroupContinue会将程序转移回发出MC_Group Interrupt时的状态。

该组可能会接受进一步的运动命令。

3. 4. 5. 9. 8 MC_MoveCircularAbsolute

Instruc-tions	LD Performance	ST Performance
		fbMC_MoveCircularAbsolute_1(

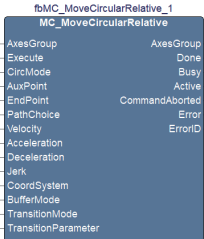
Instruc-tions	LD Performance	ST Performance
MC_MoveCircularAbsolute		<pre>AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), CircMode:= (* MC_CIRC_MODE *), AuxPoint:=(* REAL *), EndPoint:=(* REAL *), PathChoice:= (* MC_CIRC_PATHCHOICE *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), TransitionMode:= (* MC_TRANSITION_MODE *), TransitionParameter:= (* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

引脚类型	名称	类型	描述
VAR_IN_OUT	AxesGroup	AXES_GROUP_REF	轴组工艺对象
	Execute	BOOL	上升沿触发
	CircMode	MC_CIRC_MODE	参考CircMode介绍

引脚类型	名称	类型	描述
VAR_INPUT	AuxPoint	REAL	辅助点
	EndPoint	REAL	终止点
	PathChoice	MC_CIRC_PATHCHOICE	路径为顺时针还是逆时针
	Velocity	REAL	运动过程中的最大线速度
	Acceleration	REAL	运动过程中的最大线加速度
	Deceleration	REAL	运动过程中的最大线减速度
	Jerk	REAL	运动过程中的最大线加加速度
	CoordSystem	MC_COORD_SYSTEM	参考适用的坐标系：ACS、MCS、PCS
	BufferMode	MC_BUFFER_MODE	参考上面BufferMode介绍
	TransitionMode	MC_TRANSITION_MODE	参考上面TransitionMode介绍
VAR_OUTPUT	TransitionParameter	REAL	参考上面TransitionMode介绍
	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表明FB对轴组拥有控制权
	CommandAborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明
此功能块命令从TCP的实际位置在轴组上进行插值圆周运动。在绝对坐标系下进行。

3.4.5.9.9 MC_MoveCircularRelative

Instructions	LD Performance	ST Performance
MC_MoveCircularRelative		<pre>fbMC_MoveCircularRelative_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), CircMode:= (* MC_CIRC_MODE *), AuxPoint:=(* REAL *), EndPoint:=(* REAL *), PathChoice:= (* MC_CIRC_PATHCHOICE *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), TransitionMode:= (* MC_TRANSITION_MODE *), TransitionParameter:= (* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

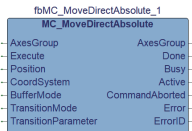
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	CircMode	MC_CIRC_MODE	参考CircMode介绍
	AuxPoint	REAL	辅助点
	EndPoint	REAL	终止点
	Path Choice	MC_CIRC_PATHCHOICE	路径为顺时针还是逆时针
	Velocity	REAL	运动过程中的最大线速度
	Acceleration	REAL	运动过程中的最大线加速度
	Deceleration	REAL	运动过程中的最大线减速度
	Jerk	REAL	运动过程中的最大线加加速度
	Coord System	MC_COORD_SYSTEM	参考适用的坐标系：ACS、MCS、PCS
	Buffer Mode	MC_BUFFER_MODE	参考上面BufferMode介绍
	TransitionMode	MC_TRANSITION_MODE	参考上面TransitionMode介绍
VAR_OUTPUT	TransitionParameter	REAL	参考上面TransitionMode介绍
	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表明FB对轴组拥有控制权
	Command Aborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障

引脚类型	名称	类型	描述
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块命令从TCP的实际位置在轴组上进行插值圆周运动。在相对坐标系下进行。

3. 4. 5. 9. 10 MC_MoveDirectAbsolute

Instruc-tions	LD Performance	ST Performance
MC_Move Direct Absolute		<pre>fbMC_MoveDirectAbso- lute_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Position:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), TransitionMode:= (* MC_TRANSITION_MODE *), TransitionParameter:= (* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

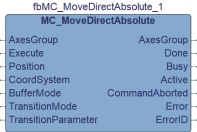
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	Position	REAL	需要运动到的绝对坐标系下的目标位置
	Coord System	MC_COORD_SYSTEM	参考适用的坐标系：ACS、MCS、PCS
	Buffer Mode	MC_BUFFER_MODE	参考上面BufferMode介绍
	TransitionMode	MC_TRANSITION_MODE	参考上面TransitionMode介绍
VAR_OUTPUT	TransitionParameter	REAL	参考上面TransitionMode介绍
	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表明FB对轴组拥有控制权
	Command Aborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块命令将轴组移动到指定坐标系中的指定绝对位置，而不考虑如何（在哪条路径上）到达目标位置。

3. 4. 5. 9. 11 MC_MoveDirectRelative

Instruc-tions	LD Performance	ST Performance
		fbMC_MoveDirectRelative_1(

Instructions	LD Performance	ST Performance
MC_Move Direct Relative		<pre>AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), TransitionMode:= (* MC_TRANSITION_MODE *), TransitionParameter:= (* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

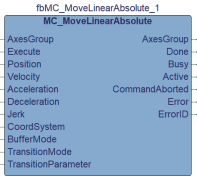
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_ GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	Distance	REAL	需要运动到的相对坐标系下的目标位置
	Coord System	MC_COORD_ SYSTEM	参考适用的坐标系： ACS、MCS、PCS
	Buffer Mode	MC_BUFFER_ MODE	参考上面BufferMode 介绍
	Transi- tionMode	MC_TRAN- SITION_ MODE	参考上面Transi- tionMode介绍

引脚类型	名称	类型	描述
	Transition Parameter	REAL	参考上面Transi- tionMode介绍
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表明FB对轴组拥有控 制权
	Command Aborted	BOOL	命令被另一个命令中 止
	Error	BOOL	轴组故障信号,true为 故障,false为无故障
	ErrorID	MC_ ERRORCODE	轴组故障代码

功能块使用说明

此功能块命令轴组移动到相对位置，而不考虑如何（在哪条路径上）到达目标位置。起始位置是TCP的实际位置。

3.4.5.9.12 MC_MoveLinearAbsolute

Instructions	LD Performance	ST Performance
MC_Move Linear Absolute		<pre>fbMC_MoveLinearAbso- lute_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Position:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *),</pre>

Instruc-tions	LD Performance	ST Performance
		TransitionMode:= (* MC_TRANSITION_MODE *), TransitionParameter:= (* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

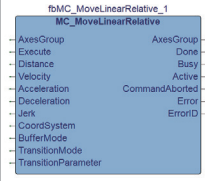
引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	Position	REAL	需要运动到的绝对坐标系下的目标位置
	Velocity	REAL	运动过程中的最大线速度
	Acceleration	REAL	运动过程中的最大线加速度
	Deceleration	REAL	运动过程中的最大线减速度
	Jerk	REAL	运动过程中的最大线加加速度
	Coord System	MC_COORD_SYSTEM	参考适用的坐标系：ACS、MCS、PCS
	Buffer Mode	MC_BUFFER_MODE	参考上面BufferMode介绍
	TransitionMode	MC_TRANSITION_MODE	参考上面TransitionMode介绍

引脚类型	名称	类型	描述
	TransitionParameter	REAL	参考上面TransitionMode介绍
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表明FB对轴组拥有控制权
	CommandAborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块命令在轴组上从TCP的实际位置到指定坐标系中的绝对位置进行插值线性移动。

3.4.5.9.13 MC_MoveLinearRelative

Instruc-tions	LD Performance	ST Performance
MC_MoveLinearRelative		fbMC_MoveLinearRelative_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), TransitionMode:= (* MC_TRANSITION_MODE *),

Instructions	LD Performance	ST Performance
		TransitionParameter:= (* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

引脚类型	名称	类型	描述
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	轴组工艺对象
VAR_INPUT	Execute	BOOL	上升沿触发
	Distance	REAL	需要运动到的相对坐标系下的目标位置
	Velocity	REAL	运动过程中的最大线速度
	Acceleration	REAL	运动过程中的最大线加速度
	Deceleration	REAL	运动过程中的最大线减速度
	Jerk	REAL	运动过程中的最大线加加速度
	Coord System	MC_COORD_SYSTEM	参考适用的坐标系：ACS、MCS、PCS
	Buffer Mode	MC_BUFFER_MODE	参考上面BufferMode介绍
	TransitionMode	MC_TRANSITION_MODE	参考上面TransitionMode介绍
	TransitionParameter	REAL	参考上面TransitionMode介绍

引脚类型	名称	类型	描述
VAR_OUTPUT	Done	BOOL	功能块执行完成
	Busy	BOOL	功能块执行还未结束
	Active	BOOL	表明FB对轴组拥有控制权
	Command Aborted	BOOL	命令被另一个命令中止
	Error	BOOL	轴组故障信号,true为故障,false为无故障
	ErrorID	MC_ERRORCODE	轴组故障代码

功能块使用说明

此功能块命令在轴组上从TCP的实际位置到指定坐标系中的相对位置进行插值线性移动。

MC_GroupContinue
MC_GroupDisable
MC_GroupEnable
MC_GroupHalt
MC_GroupStop
MC_GroupHome
MC_Groupinterrupt
MC_MoveCircularAbsolute
MC_MoveCircularRelative
MC_MoveDirectAbsolute
MC_MoveDirectRelative
MC_MoveLinearAbsolute
MC_MoveLinearRelative

3.5 EtherCAT

3.5.1 Definitions (定义)

该定义文件夹包含EtherCAT库所使用到的定义，包括枚举，结构体。

3.5.1.1 Enumerations (枚举)

ECATDCSTATE		
名称	值	描述
ECAT_DCSTATE_IDLE	0	主时钟与从时钟完全独立。
ECAT_DCSTATE_SYN- CHRONIZING	1	主时钟正在进行初步测量以 与从设备同步
ECAT_DCSTATE_WAIT- ING_START_TIME	2	在主设备与从设备同步之 后,必须等待START_TIME,这 是从设备首次执行 I/O 的 时间
ECAT_DCSTATE_COM- PENSATING	3	如果主时钟与从设备的时钟 同步,并且在START_TIME之 后,进入此状态,这意味着 主机正在不断跟踪从设备的 时钟
ECAT_STATE_NOT_SYN CHRONIZED	4	如果由于任何原因主时钟未 能同步从时钟,则进入此状 态,所有相关的从设备也进 入SAFE_OP状态

ECATERERROR		
名称	值	描述
ECAT_ERROR_KE00000_ ERROR_OK	0	无错误
ECAT_ERROR_KE10000_ FRAME_CORRUPTED	10000	以太网帧损坏
ECAT_ERROR_KE10001_ MASTER_NETWORK_CHECK	10001	主站检测到网络异常
ECAT_ERROR_KE10002_ FRAME_CORRUPTED_WCK	10002	帧损坏,工作计数器WCK 错误
ECAT_ERROR_KE10003_ REGISTER_ACCESS	10003	寄存器访问失败
ECAT_ERROR_KE10004_ SLAVE_NOT_FOUND	10004	变量对应的从站未找到
ECAT_ERROR_KE10005_ CONFIG_FILE_NOT_FOUND	10005	配置文件未找到

ECATERERROR		
名称	值	描述
ECAT_ERROR_KE10006_ CONFIG_FILE_PARSE	10006	配置文件解析错误
ECAT_ERROR_KE10007_ SDO_RESPONSE	10007	未获取到SDO响应
ECAT_ERROR_KE10008_ INTERNAL	10008	内部错误
ECAT_ERROR_KE10009_ SDO_INTERNAL	10009	SDO内部错误
ECAT_ERROR_KE10010_ SDO_WRITE	10010	SDO写入错误
ECAT_ERROR_KE10011_ SDO_WRITE	10011	SDO写入错误
ECAT_ERROR_KE10012_ SDO_RESPONSE	10012	未获取到SDO响应
ECAT_ERROR_KE10013_ PDO_RX_MAPPING	10013	接收PDO映射错误
ECAT_ERROR_KE10014_ PDO_TX_MAPPING	10014	发送PDO映射错误
ECAT_ERROR_KE10015_ PDO_RX_MAPPING_CNT	10015	接收PDO映射对象数量 配置错误
ECAT_ERROR_KE10016_ PDO_TX_MAPPING_CNT	10016	发送PDO映射对象数量 配置错误
ECAT_ERROR_KE10017_ PDO_ASSIGN_SMANAG	10017	同步管理器分配给PDO 失败
ECAT_ERROR_KE10018_ PDO_CNT_SMANAG	10018	同步管理器映射PDO数 量设置失败
ECAT_ERROR_KE10019_ SDO_WRITE	10019	SDO写入错误
ECAT_ERROR_KE10020_ SLAVE_STATE_CHANGE	10020	从站状态切换失败
ECAT_ERROR_KE10021_ EOE_VIRTUAL_MAC	10021	EoE虚拟MAC地址未分配

ECATERORR		
名称	值	描述
ECAT_ERROR_KE10022_ EOE_INITCMD	10022	EoE初始化命令执行失败
ECAT_ERROR_KE10023_ EOE_IP_MAC	10023	EoE设置IP和MAC地址失败
ECAT_ERROR_KE10024_ EOE_PARAMS	10024	EoE参数配置错误
ECAT_ERROR_KE10025_ EOE_MISS_PREVF	10025	EoE缺失上一分片报文
ECAT_ERROR_KE10026_ EOE_FRAGMENT_LEN	10026	EoE分片长度超出最大值
ECAT_ERROR_KE10027_ EOE_UNKNOWN	10027	EoE未知报文类型
ECAT_ERROR_KE10028_ EOE_CHANGE_STATE	10028	EoE状态切换失败
ECAT_ERROR_KE10029_ FOE_TIMEOUT	10029	FoE通信超时
ECAT_ERROR_KE10030_ FOE_TIMEOUT	10030	FoE通信超时
ECAT_ERROR_KE10031_ FOE_SEQUENCE	10031	FoE序列号错误
ECAT_ERROR_KE10032_ FOE_GENERAL	10032	FoE通用错误
ECAT_ERROR_KE10033_ FOE_OPCODE	10033	FoE无效操作码
ECAT_ERROR_KE10034_ FOE_TIMEOUT	10034	FoE通信超时
ECAT_ERROR_KE10035_ INTERNAL	10035	内部错误
ECAT_ERROR_KE10036_ SSC	10036	SSC配置写入错误
ECAT_ERROR_KE10037_ MAILBOX_REPLY	10037	轮询过程中邮箱应答错误

ECATERORR		
名称	值	描述
ECAT_ERROR_KE10038_ MAILBOX_UNKNOWN	10038	邮箱未知协议类型
ECAT_ERROR_KE10039_ MAC	10039	读取网络MAC地址失败
ECAT_ERROR_KE10040_ JOIN_MULTICAST	10040	加入组播地址失败
ECAT_ERROR_KE10041_ NET_IFACE	10041	打开网络接口失败
ECAT_ERROR_KE10042_ NET_GLB_STAT	10042	未在周期帧中检测到网络全局状态
ECAT_ERROR_KE10043_ MASTER_INITCMD	10043	主站初始化命令校验失败
ECAT_ERROR_KE10044_ MASTER_ACYCLIC	10044	主站收到非法非周期帧
ECAT_ERROR_KE10045_ REGISTER_ACCESS	10045	寄存器访问失败
ECAT_ERROR_KE10046_ MASTER_WKC	10046	主站检测到同步单元WKC计数不匹配
ECAT_ERROR_KE10047_ MASTER_PROP_DELAY	10047	主站传播延时计算异常
ECAT_ERROR_KE10048_ REGISTER_ACCESS	10048	寄存器访问失败
ECAT_ERROR_KE10049_ MASTER_DC_DELAY	10049	分布式时钟延时计算失败，网络配置错误
ECAT_ERROR_KE10050_ MASTER_PROP_DELAYS	10050	主站从站传播延时计算错误
ECAT_ERROR_KE10051_ MASTER_PROP_DELAYS	10051	主站从站传播延时计算错误
ECAT_ERROR_KE10052_ MASTER_PROP_DELAYS	10052	主站从站传播延时计算错误
ECAT_ERROR_KE10053_ REGISTER_ACCESS	10053	寄存器访问失败

ECATERORR		
名称	值	描述
ECAT_ERROR_KE10054_REGISTER_ACCESS	10054	寄存器访问失败
ECAT_ERROR_KE10055_REGISTER_ACCESS	10055	寄存器访问失败
ECAT_ERROR_KE10056_REGISTER_ACCESS	10056	寄存器访问失败
ECAT_ERROR_KE10057_REGISTER_ACCESS	10057	寄存器访问失败
ECAT_ERROR_KE10058_NET_SYNC_ACYCLIC	10058	经过指定非周期帧后网络仍未同步
ECAT_ERROR_KE10059_REGISTER_ACCESS	10059	寄存器访问失败
ECAT_ERROR_KE10060_MASTER_CLOCK_64	10060	主站非周期读取64位时钟异常
ECAT_ERROR_KE10061_RESET_COUNT	10061	复位流程执行失败
ECAT_ERROR_KE10062_MASTER_INITCMD	10062	主站初始化命令状态切换失败
ECAT_ERROR_KE10063_MASTER_SLAVES_COUNT	10063	主站检测到从站数量不匹配
ECAT_ERROR_KE10064_SLAVE_PARENT	10064	无法找到从站上级节点
ECAT_ERROR_KE10065_SLAVE_PORT	10065	从站端口未开启，但上级链路需要该端口
ECAT_ERROR_KE10067_MASTER_INITCMD	10067	主站初始化命令校验失败
ECAT_ERROR_KE10068_MASTER_INITCMD	10068	主站初始化命令校验失败
ECAT_ERROR_KE10069_SLAVE_MISSING	10069	从站掉线缺失
ECAT_ERROR_KE10070_FILE_TRANF_ACCESS	10070	文件传输访问错误

ECATERORR		
名称	值	描述
ECAT_ERROR_KE10071_INTERNAL	10071	内部错误
ECAT_ERROR_KE10072_SLAVE_MISSING_ERROR	10072	掉线从站恢复后携带故障
ECAT_ERROR_KE10073_SLAVE_LEN	10073	从站邮箱读取命令长度异常
ECAT_ERROR_KE10074_SLAVE_RESET	10074	从站复位失败
ECAT_ERROR_KE10075_SLAVE_DC	10075	从站配置使用分布式时钟但硬件不支持
ECAT_ERROR_KE10077_INTERNAL	10077	内部错误
ECAT_ERROR_KE10078_INTERNAL	10078	内部错误
ECAT_ERROR_KE10079_INTERNAL	10079	内部错误
ECAT_ERROR_KE10080_INTERNAL	10080	内部错误
ECAT_ERROR_KE10081_ACYCLIC	10081	暂未收到非周期报文
ECAT_ERROR_KE10082_SLAVE_COUNT	10082	从站数量不匹配
ECAT_ERROR_KE10083_EOE_OVER_EOE	10083	通过从站EoE服务发送EoE报文失败
ECAT_ERROR_KE10084_MASTER_DC_SYNCM	10084	主站启用分布式时钟但未找到同步主站
ECAT_ERROR_KE10085_INTERNAL	10085	内部错误
ECAT_ERROR_KE19999_INTERNAL	19999	内部错误

ECATEVENTID		
名称	值	描述
ECAT_EVENT_NONE	0	无事件
ECAT_EVENT_ERROR	1	从站或主站产生错误事件
ECAT_EVENT_EMERGENCY	2	CoE从站产生紧急事件
ECAT_EVENT_NODESTATUSCHANGED	3	从站状态发生改变时触发
ECAT_EVENT_DCSTATUSCHANGED	4	主站在跟踪分布式时钟时状态改变触发
ECAT_EVENT_NETWORKRECOVERY	5	网络重启，用于从严重故障中恢复
ECAT_EVENT_NETWORKRESYNC	6	所有从站准备进入运行状态时触发
ECAT_EVENT_SLAVERESYNC	7	单个从站准备进入运行状态时触发

ECATINITERRORCODE		
名称	值	描述
ECAT_INITERR_OK	0	无错误
ECAT_INITERR_DISABLED	500	EtherCAT功能已禁用
ECAT_INITERR_SYS	501	内部系统错误
ECAT_INITERR_XML	502	XML文件解析阶段出错
ECAT_INITERR_EMPTYCFG	503	配置文件为空
ECAT_INITERR_OUTOFMEMORY	504	内存耗尽
ECAT_INITERR_FILE	505	读取时发生I/O错误
ECAT_INITERR_NO_CONFIG	506	配置节点不存在
ECAT_INITERR_NO_ENINAME	507	无ENI名称
ECAT_INITERR_NO_DBTYPE	508	无数据块类型
ECAT_INITERR_NO_DBNUM	509	无数据块索引
ECAT_INITERR_NO_DBOFF	510	无数据块偏移量
ECAT_INITERR_NO_BITLEN	511	缺少位长度
ECAT_INITERR_NO_DBBYTEOFF	512	缺少字节偏移量
ECAT_INITERR_DBMAPDBTYPEINV	513	无效的I/O映射
ECAT_INITERR_ENIVARINV	514	配置的变量与ENI文件不匹配

ECATINITERRORCODE		
名称	值	描述
ECAT_INITERR_TOOMANYVARIABLES	515	超出最大变量数量限制
ECAT_INITERR_DC_UNSUPPORTED	516	不支持分布式时钟
ECAT_INITERR_WRONGARRAYLEN	517	数组长度错误
ECAT_INITERR_CONFIG_CORRUPTED	518	配置文件损坏
ECAT_INITERR_WRONG_INTF_CHECK	519	以太网接口配置校验失败

ECATSTATE		
名称	值	描述
ECAT_STATE_UNKNOWN	0	未知状态
ECAT_STATE_INIT	1	初始化状态
ECAT_STATE_PREOP	2	预运行状态
ECAT_STATE_BOOT	3	启动状态
ECAT_STATE_SAFEOP	4	安全运行状态
ECAT_STATE_OP	8	运行状态
ECAT_STATE_STOPPED	254	停止状态
ECAT_STATE_FATAL	255	致命故障状态

3.5.1.2 Structures（结构体）

ECATDCSTATUSEVENT		
名称	类型	描述
event_id	SINT	其值固定: 4 对应 DC 状态变化事件
ecat_chn	USINT	主站通道，用于多主站设备
from	USINT	之前的状态
to	USINT	当前状态

ECATEMCEVENT		
名称	类型	描述
event_id	SINT	其值固定: 2，对应紧急事件
ecat_chn	USINT	主站通道，用于多主站设备

ECATEMCYEVENT		
名称	类型	描述
node	UINT	发出事件的从站物理地址
emcyCode	UINT	CoE 紧急事件代码
emcyReg	USINT	CoE 紧急事件寄存器
emcyData	ARRAY[0..4] OF USINT	紧急事件数据

ECATERROREVENT		
名称	类型	描述
event_id	SINT	其值固定：1，对应错误事件
ecat_chn	USINT	主站通道，用于多主站设备
node	UINT	发出事件的从站物理地址。若错误来自主站，该字段值为0
errorCode	ECATERERROR	错误代码
iP1	DINT	错误代码的特定参数
iP2	DINT	错误代码的特定参数
iP3	DINT	错误代码的特定参数

ECATMASTERSTATUS		
名称	类型	描述
error_code	INT	主站栈初始化阶段的最后错误代码（参考 ECATINITERRORCODE 枚举）
ecat_enable	BOOL	EtherCAT 栈已启用
data_exch_enable	BOOL	数据交换已启用：当交换对象与相关 PLC 变量完成有效映射时。该功能为可选，即使未配置对象，网络也可进入运行模式
chn	USINT	通道号，用于包含多个 EtherCAT 网络的系统
eni_ok	BOOL	ENI 文件已成功读取并解析
xml_ok	BOOL	包含附加配置的 XML 文件已成功读取并解析
init_ok	BOOL	主站栈初始化已成功完成
num_slaves	UINT	已定义从站数量

ECATMASTERSTATUS		
名称	类型	描述
dc_sync_master	UINT	分布式时钟同步主站的物理地址
net_stat	ECAT-STATE	当前网络状态
target_net_stat	ECAT-STATE	主站为所有从站设置的目标状态（无论从站是否已达到）
dc_stat	ECAT-DC-STATE	主站与同步主站的同步状态（仅使用 DC 时有效）。主站必须处于 3-COMPENSATING 状态，从站才能进入 SAFEOP/OP 状态
use_dc	BOOL	分布式时钟（DC）已启用并至少配置给一个从站
drives Clock	BOOL	EtherCAT 栈为整个系统（包括 PLC）提供时钟。这对于检测第一个从站故障非常重要，因为在此情况下 EtherCAT 栈可能会丢失数据包并失去网络同步
num_slaves_cfg	UINT	已通过初始化命令正确配置的从站数量
num_pdo_tx	UDINT	发送 PDO 数量
num_pdo_rx	UDINT	接收 PDO 数量
network_ok	BOOL	EtherCAT 网络正常运行无错误：所有从站均已达到 OP 状态。若发生任何错误，该值将变为 false
network_starting	BOOL	EtherCAT 网络正在启动中
num_frame_tx	ULINT	发送的总帧数
num_frame_tx_cyclic	ULINT	发送的周期性帧数
num_frame_tx_acyclic	ULINT	发送的非周期性帧数
num_frame_rx	ULINT	接收的总帧数

ECATMASTERSTATUS		
名称	类型	描述
num_frame_rx_cyclic	ULINT	接收的周期性帧数
num_frame_rx_acyclic	ULINT	接收的非周期性帧数
num_frame_lost	ULINT	丢失的总帧数
num_frame_lost_cyclic	ULINT	丢失的周期性帧数
num_frame_lost_acyclic	ULINT	丢失的非周期性帧数

ECATNETWORKRECOVERYEVENT		
名称	类型	描述
event_id	SINT	其值固定：8，对应网络恢复事件
ecat_chn	USINT	主站通道，用于多主站设备

ECATNODESTATUSEVENT		
名称	类型	描述
event_id	SINT	其值固定：3，对应从站状态变化事件
ecat_chn	USINT	主站通道，用于多主站设备
node	UINT	发出事件的从站物理地址
from	UINT	之前的状态
to	UINT	当前状态

ECATRESYNCINFO		
名称	类型	描述
event_id	SINT	其值可为 6 或 7：分别对应网络/主站或从站准备进入运行状态
ecat_chn	USINT	主站通道，用于多主站设备
node	UINT	发出事件的从站物理地址
is_master	BOOL	TRUE 表示为网络重新同步事件，FALSE 表示为从站重新同步事件
ready_to_start	BOOL	FALSE 表示存在抑制从站进入运行状态的错误

ECATSLAVEMONITOR		
名称	类型	描述
idx_requested	UINT	可写字段，用于选择要监控的从站
idx_actual	UINT	若 idx_requested 的值有效，该字段与其值相同，并监控对应的状态结构体；若 idx_requested 无效，则该字段值为 65535
status	ECATSLAVE-STATUS	idx_actual 对应从站的当前状态

ECATSLAVESTATUS		
名称	类型	描述
cfg	BOOL	已配置：从站已接收所有初始化命令
connected	BOOL	已存在且物理连接正常
address	UINT	物理地址，由主站默认分配或用户配置
state	ECATSTATE	从站当前状态
error_cnt	UDINT	检测到的错误次数
vendor_id	UDINT	厂商ID；仅当启用“检查厂商ID”或“始终读取识别信息”时有效
product_code	UDINT	产品代码；仅当启用“检查产品代码”或“始终读取识别信息”时有效
revision_number	UDINT	修订号；仅当启用“检查修订号”或“始终读取识别信息”时有效
serial_number	UDINT	序列号；仅当启用“检查序列号”或“始终读取识别信息”时有效
lastEmergencyCode	UINT	最后一次紧急事件代码

ECATSLAVESTATUS		
名称	类型	描述
lastEmergencyReg	USINT	最后一次紧急事件寄存器
lastEmergencyData	ARRAY[0..4] OF USINT	紧急事件数据
index	UINT	在网络中的位置
name	STRING[32]	从站名称
lastError Code	UINT	最后一次错误代码
lastError Reg	USINT	最后一次错误寄存器
lastError Data	ARRAY[0..4] OF USINT	错误特定数据
AlStatusCode	UINT	AlStatusCode 寄存器

3.5.2 Master control functions (主站控制功能)

这个文件夹下是主站的控制函数，包含主站的停止，启动，与复位。

3.5.2.1 sysECAT_Recovery

恢复EtherCAT网络。这是一个非阻塞函数，可以在任何任务中调用。但该函数不能循环调用，否则会出现异常现象。

引脚类型	名称	类型	描述
VAR_INPUT	dummy	UINT	占位变量

返回值类型	描述
BOOL	TRUE:成功 FALSE:失败

3.5.2.2 sysECAT_StartMaster

功能描述：

启动一个已停止的EtherCAT主站。这是一个非阻塞函数，可以在任何任务中调用。

返回值类型：BOOL

引脚类型	名称	类型	描述
VAR_INPUT	dummy	UINT	占位变量

返回值类型	描述
BOOL	TRUE:成功 FALSE:失败

3.5.2.3 sysECAT_StopMaster

功能描述：

停止正在运行的EtherCAT主站。这是一个非阻塞函数，可以在任何任务中调用。

引脚类型	名称	类型	描述
VAR_INPUT	dummy	UINT	占位变量

返回值类型	描述
BOOL	TRUE:成功 FALSE:失败

3.5.3 Master diagnostic functions (主站诊断功能)

3.5.3.1 sysECAT_GetSlaveCfgByAddress

功能描述：

此函数返回通过物理地址选择的从设备在ECAT-SLAVESTATUS结构体中的‘cfg’字段。这是一个非阻塞函数，可以在任何任务中调用。

引脚类型	名称	类型	描述
VAR_INPUT	address	UINT	从站物理地址

返回值类型	描述
BOOL	TRUE:已配置，从站已接收所有初始化命令 FALSE:失败

3.5.3.2 sysECAT_GetSlaveCfgByIndex

功能描述：

该函数返回网络中通过位置索引选择的从站的ECAT-SLAVESTATUS结构体中的‘cfg’字段。这是一个非阻塞函数，可以在任何任务中调用。

引脚类型	名称	类型	描述
VAR_INPUT	index	UINT	从站索引

返回值类型	描述
BOOL	TRUE:已配置，从站已接收所有初始化命令 FALSE:失败

3.5.3.3 sysECAT_GetSlaveDiagByAddress

功能描述：

该函数返回通过物理地址选择的从设备对应的ECAT-SLAVESTATUS 结构体的指针。这是一个非阻塞函数，可以在任何任务中调用。注意返回类型为指针需要使用指针

引脚类型	名称	类型	描述
VAR_INPUT	address	UINT	从站物理地址

返回值类型	描述
@ECATSLAVE-STATUS	它返回一个类型为@ECATSLAVESTATUS的值，或在出错时返回NULL

3.5.3.4 sysECAT_GetSlaveDiagByIndex

功能描述：

该函数返回网络中按位置索引选定的从站对应的ECATSLAVESTATUS结构体的指针。这是一个非阻塞函数，可以在任何任务中调用。注意返回类型为指针需要使用指针

引脚类型	名称	类型	描述
VAR_INPUT	index	UINT	从站索引

返回值类型	描述
@ECATSLAVE-STATUS	它返回一个类型为@ECATSLAVESTATUS的值，或在出错时返回NULL

3.5.3.5 sysECAT_GetSlavePdoVarByAddress

功能描述：

该函数通过从站地址返回PDO映射的变量地址

引脚类型	名称	类型	描述
VAR_INPUT	slave_address	UINT	从站物理地址
VAR_INPUT	index	UINT	对象索引
VAR_INPUT	sindex	USINT	对象子索引

返回值类型	描述
PVOID	返回一个任意类型的指针，该指针指向从站中映射好的变量的地址，在出错时返回NULL。

3.5.3.6 sysECAT_GetSlavePdoVarByIndex

功能描述：

该函数通过从站索引返回PDO映射的变量地址

引脚类型	名称	类型	描述
VAR_INPUT	slave_index	UINT	从站索引
VAR_INPUT	index	UINT	对象索引
VAR_INPUT	sindex	USINT	对象子索引

返回值类型	描述
PVOID	返回一个任意类型的指针，该指针指向从站中映射好的变量的地址，在出错时返回NULL。

3.5.3.7 sysECAT_GetSlavePresByAddress

功能描述：

此函数返回通过物理地址选择的从设备在ECAT-SLAVESTATUS结构体中的‘connected’ 字段。这是一个非阻塞函数，可以在任何任务中调用。

引脚类型	名称	类型	描述
VAR_INPUT	address	UINT	从站物理地址

返回值类型	描述
BOOL	TRUE:物理连接成功 FALSE:物理连接失败

3.5.3.8 sysECAT_GetSlavePresByIndex

功能描述：

此函数返回网络中通过位置索引选择的从站所对应的ECATSLAVESTATUS结构体中的‘connected’字段。这是一个非阻塞函数，可以在任何任务中调用。

引脚类型	名称	类型	描述
VAR_INPUT	index	UINT	从站索引

返回值类型	描述
BOOL	TRUE:物理连接成功 FALSE:物理连接失败

3.5.3.9 sysECAT_GetSlaveStateByAddress

功能描述：

此函数返回通过物理地址选择的从设备的ECAT-SLAVESTATUS结构体中的‘state’字段。这是一个非阻塞函数，可以在任何任务中调用。

引脚类型	名称	类型	描述
VAR_INPUT	address	UINT	从站物理地址

返回值类型	描述
ECATSTATE	返回枚举类型的值，表示从站状态

3.5.3.10 sysECAT_GetSlaveStateByIndex

功能描述：

此函数返回网络中通过位置索引选择的从站的ECAT-SLAVESTATUS结构体中的‘state’字段。这是一个非阻塞函数，可以在任何任务中调用。

引脚类型	名称	类型	描述
VAR_INPUT	index	UINT	从站索引

返回值类型	描述
ECATSTATE	返回枚举类型的值，表示从站状态

3.5.4 SDO function blocks

3.5.4.1 sysECAT_Sdo_Read

功能描述：

该功能块可用于在支持CoE协议的从设备中读取对象值，该设备使用SD0命令。它不能从定时任务中调用，但可以从后台任务或事件回调任务中调用。仅支持对象长度小于等于4字节。这是一个阻塞功能块，单次执行后返回结果。

引脚类型	名称	类型	描述
VAR_INPUT	slave_address	UINT	从站物理地址
VAR_INPUT	index	UINT	对象索引
VAR_INPUT	sindex	USINT	对象子索引
VAR_INPUT	length	UDINT	对象长度
VAR_OUTPUT	value	DINT	读取的值
VAR_OUTPUT	errorCode	DINT	错误码
VAR_OUTPUT	result	BOOL	读取成功时为TRUE

3.5.4.2 sysECAT_Sdo_Read2

功能描述：

该功能块可用于在支持CoE协议的从设备中读取对象值，该设备使用SD0命令。它可以被任何任务调用，因为它不是阻塞的，并允许底层并行执行。仅支持对象长度小于等于4字节。允许多个实例。典型用途是实例化多个功能块，并在指定任务的每次迭代中调用它们。使能变量的上升沿决定处理的开始。在每次后续调用中，必须保持enable变量处于激活状态，直到处理结束。在执行过程中更改功能块参数（包括使能）的任何值均无效。

处理完成时：底层引擎将done或error变量设置为true、调用者以enable变量为false调用该功能块

引脚类型	名称	类型	描述
VAR_INPUT	slave_address	UINT	从站物理地址
VAR_INPUT	index	UINT	对象索引
VAR_INPUT	sindex	USINT	对象子索引
VAR_INPUT	length	USINT	对象长度（1-4）
VAR_INPUT	enable	BOOL	上升沿使能
VAR_OUTPUT	active	BOOL	后台 SDO 处理中
VAR_OUTPUT	done	BOOL	功能块处理完成
VAR_OUTPUT	error	BOOL	完成时, 发生错误则为 TRUE
VAR_OUTPUT	value	DINT	读取的值
VAR_OUTPUT	error Code	DINT	错误码 (error=TRUE 时有效, 详情参见功能块说明)

3. 5. 4. 3 sysECAT_Sdo_Write

功能描述：

该功能块可用于在支持CoE协议的从设备中通过SDO命令写入对象值。它不能从定时任务中调用，但从后台任务或事件回调任务中调用。仅支持长度小于等于4字节的对象。这是一个阻塞功能块，单次执行后返回结果。

引脚类型	名称	类型	描述
VAR_INPUT	slave_address	UINT	从站物理地址
VAR_INPUT	index	UINT	对象索引
VAR_INPUT	sindex	USINT	对象子索引
VAR_INPUT	length	UDINT	对象长度
VAR_INPUT	value	DINT	待写入的值
VAR_OUTPUT	error Code	DINT	错误码(详情参见说明)
VAR_OUTPUT	result	BOOL	写入成功时为 TRUE

3. 5. 4. 4 sysECAT_Sdo_Write2

功能描述：

该功能块可用于在支持CoE协议的从设备中通过SDO命令写入对象值。它可以被任何任务调用，因为它不是阻塞的，并允许底层并行执行。仅支持对象长度小于等于4字节。允许多个实例。典型用法是实例化多个功能块，并在指定任务的每次迭代中调用它们。使能变量的上升沿决定处理开始。enable变量必须在每次后续调用中保持激活状态，直到处理结束。在执行过程中更改功能块参数(包括使能)的任何值均无效。处理完成时:底层引擎将done或error变量设置为true、调用者以enable变量为false调用该功能块。

引脚类型	名称	类型	描述
VAR_INPUT	slave_address	UINT	从站物理地址
VAR_INPUT	index	UINT	对象索引
VAR_INPUT	sindex	USINT	对象子索引
VAR_INPUT	value	DINT	待写入的值
VAR_INPUT	length	USINT	对象长度（1-4）
VAR_INPUT	enable	BOOL	上升沿使能
VAR_OUTPUT	active	BOOL	后台 SDO 处理中
VAR_OUTPUT	done	BOOL	功能块处理完成
VAR_OUTPUT	error	BOOL	完成时, 发生错误则为 TRUE
VAR_OUTPUT	error Code	DINT	错误码 (error=TRUE 时有效, 详情参见功能块说明)

错误码	
值	描述
0	执行成功
< 0表示内部错误	
-11	内存不足

错误码	
值	描述
-520	主站未启用或已停止
-521	从站不存在或未配置
-522	内部超时
-523	内部队列异常
-524	参数无效
>0表示从站错误（允许厂商自定义错误码）	
0x05030000	翻转位未变化
0x05040000	SDO 通信超时
0x05040001	未知命令标识符
0x05040005	内存不足
0x06010000	不支持的访问方式
0x06010001	只读对象，仅可写入
0x06010002	只写对象，仅可读取
0x06010003	子索引 0 非 0,无法写入该对象
0x06010004	该对象不支持完整访问模式
0x06020000	对象不存在
0x06040041	对象无法映射到 PDO
0x06040042	映射对象长度超出 PDO 限制
0x06040043	参数不兼容
0x06040047	设备不兼容
0x06060000	硬件故障
0x06070010	参数长度错误
0x06070012	参数数据过长
0x06070013	参数数据过短
0x06090011	子索引（条目）不存在
0x06090030	数值超出范围
0x06090031	数值过大
0x06090032	数值过小
0x06090033	检测模块标识列表（0xF030）与配置模块标识列表（0xF050）不匹配
0x06090036	数值小于最小值限制
0x08000000	通用未知错误
0x08000020	数据无法读取或写入

错误码	
值	描述
0x08000021	本地控制锁定，无法访问数据
0x08000022	当前设备状态下无法读写数据
0x08000023	对象字典中未找到该对象

3.5.5 Vars（EtherCAT库中的全局变量）

以下变量是EtherCAT库中的全局变量，表示在任意任务中可以直接调用这些变量，去检测主站，或者从站的状态。

变量名称	变量类型	中文描述
sysECATDCStatusChangedEvent	ECATDCSTATUSCHANGEVENT	主站状态的最新变化，与分布式时钟同步相关
sysECATEmcyEvent	ECATEMICYEVENT	CoE 从站发送的最新紧急事件
sysECATErrorEvent	ECATERROREVENT	EtherCAT 总线上检测到的最新错误事件
sysECATEventId	ECATEVENTID	最近发生事件的 ID
sysEcatMasterStatus	ECATMASTERSTATUS	EtherCAT 主站和网络状态总览
sysECATNetworkRecoveryEvent	ECATNETWORKRECOVERYEVENT	网络恢复的最新事件
sysECATNodeStatusChangedEvent	ECATNODESTATUSCHANGEVENT	从站状态的最新变化
sysECATResyncInfo	ECATRESYNCCINFO	重新同步信息
sysEcatSlaveMonitor	ECATSLAVEMONITOR	可放入监控表中监视任意从站的特定状态，通过在 idx_requested 字段写入从站索引（网络中从 0 开始）选择监视从站

3.6 ModbusRTU

3.6.1 Definitions

3.6.1.1 Structures (结构体)

MbInpOutCmdStruct		
名称	类型	描述
StartupCmd	MbInpOutCmdMode	启动命令
CurrentCmd	MbInpOutCmdMode	当前命令
FnzRead	INT	用于读取的功能块
FnzWrite	INT	用于写入的功能块
Flags	DWORD	内部标志位
PValue	DWORD	当前值指针
Reserved	DWORD	预留, 供后续使用
ImageRead	LWORD	读取映像
ImageWrite	LWORD	写入映像

MbMRtuNetList		
名称	类型	描述
chn	USINT	Modbus 主站 RTU 通道, 由 sysMbMRtu_OpenChn 函数返回
adr	USINT	Modbus 从站地址
cfg	BOOL	Modbus 从站已添加到网络列表
pres	BOOL	Modbus 从站在线
active	BOOL	Modbus 从站通信中
swapWordsMode	USINT	字交换模式
numParamMsg	UINT	从站参数化消息总数
numQueuedMsg	UINT	配置的从站消息总数
minPollingTime	UINT	最小轮询时间 (毫秒)
status	MbStatus	Modbus 从站状态机, 参考 MbStatus 枚举
error	MbError	Modbus 从站最新错误, 参考 MbError 枚举
exception	MbException	Modbus 从站最新异常, 参考 MbException 枚举

MbMRtuNetList		
名称	类型	描述
slaveExceptionCount	UINT	从站异常错误计数
slaveMessageCount	UINT	从站消息计数
slaveNoResponseCount	UINT	从站无响应计数
slaveNoAckCount	UINT	从站无应答计数
slaveBusyCount	UINT	从站忙计数
slaveChrOvrCount	UINT	从站字符溢出计数
maxRetries	UINT	最大重试次数
disableWrite	BOOL	禁用写入命令时为 TRUE
reserved	USINT	预留

MbMRtuStatus		
名称	类型	描述
cfg	BOOL	Modbus 主站已配置
chn	USINT	Modbus 主站 RTU 通道, 由 sysMbMRtu_OpenChn 函数返回
numSlaves	USINT	网络列表中已配置从站总数
reserved	USINT	预留字段
status	MbStatus	Modbus 主站状态机, 参考 MbStatus 枚举类型
error	MbError	Modbus 主站最近错误, 参考 MbError 枚举类型
numMsg	UINT	已配置广播及单从站消息总数
numBroadcastMsg	UINT	已配置广播消息总数
numBusMsgCount	UINT	总线消息计数
numBusCommErrCount	UINT	总线通信错误计数

MbRtuDiagnostics		
名称	类型	描述
busMsgCount	UINT	总线收发成功消息通用计数器
busCommErrCount	UINT	总线收发失败消息通用计数器
exceptionCount	UINT	通用异常错误计数器
messageCount	UINT	主站：发送消息计数； 从站：接收消息计数
noResponseCount	UINT	无响应消息计数
slaveRspNoAckCount	UINT	仅主站可用：从站 NAK 无应答响应消息计数
slaveRspBusyCount	UINT	仅主站可用：从站 BUSY 忙响应消息计数
chrOvrCount	UINT	字符溢出计数

3.6.1.2 Enumerations (枚举)

MB_RESULT_VALUE		
名称	值	描述
MB_E_OK	0	命令执行成功
MB_E_WRITE_DISABLED	-349	写命令已禁用
MB_E_POLLINGTIME	-348	未遵守轮询时间
MB_E_REQUESTMISMATCH	-347	请求不匹配
MB_E_NOCONNSYSTEMBUSY	-346	系统繁忙无法建立连接
MB_E_NOCONNSLOTAVAILABLE	-345	无可连接插槽
MB_E_CMDPARAMERROR	-344	命令参数无效
MB_E_CMDSAMEBUFFER	-343	指定了相同缓冲区
MB_E_CMDBUFFERTOOSMALL	-342	缓冲区过小
MB_E_CMDBUFFERERROR	-341	命令缓冲区错误
MB_E_CMDBUFFERSIZEERROR	-340	缓冲区大小无效
MB_E_NETWORK	-339	网络 ID 无效
MB_E_CHECKSUM	-338	校验和错误
MB_E_NEGATIVEACKNOWLEDGE	-337	否定应答

MB_RESULT_VALUE		
名称	值	描述
MB_E_NOTALLOWEDINTIMED-TASK	-336	禁止在定时任务调用
MB_E_SWITCHEDOFF	-335	节点已关闭
MB_E_ILLEGALDATAPROTOCOL	-332	协议非法数据值
MB_E_TIMEOUT	-331	请求超时
MB_E_GATEWAYDEVICEFAILED	-330	网关目标设备无响应
MB_E_GATEWAYPATHUNAV	-329	网关路径不可用
MB_E_MEMORYPARITYERROR	-328	存储器奇偶校验错误
MB_E_SLAVEDEVICEBUSY	-327	从站设备繁忙
MB_E_ACKNOWLEDGE	-326	应答错误
MB_E_SLAVEDEVICEFAILURE	-325	从站设备故障
MB_E_ILLEGALDATAVALUE	-324	非法数据值
MB_E_ILLEGALDATAADDR	-323	非法数据地址
MB_E_ILLEGALFUNCTION	-322	非法功能码
MB_E_IDE	-321	IDE 标识不匹配
MB_E_PROTOCOL	-320	协议错误
MB_E_STOPPED	-317	主站未运行
MB_E_DATATYPE	-311	数据类型无效
MB_E_NODENOTEXIST	-303	节点不存在
MB_E_NODE	-301	无效节点

MB_RTU_NETWORK_ID		
名称	值	描述
MB_RTU_NETWORK_0	0	网络 0
MB_RTU_NETWORK_1	1	网络 1
MB_RTU_NETWORK_2	2	网络 2
MB_RTU_NETWORK_3	3	网络 3
MB_RTU_NETWORKS	4	支持的最大网络数量

MbBaudrate		
名称	值	描述
MbBaud_600	600	600 比特率
MbBaud_1200	1200	1200 比特率

MbBaudrate		
名称	值	描述
MbBaud_2400	2400	2400 比特率
MbBaud_4800	4800	4800 比特率
MbBaud_9600	9600	9600 比特率
MbBaud_19200	19200	19200 比特率
MbBaud_38400	38400	38400 比特率
MbBaud_57600	57600	57600 比特率
MbBaud_76800	76800	76800 比特率
MbBaud_115200	115200	115200 比特率
MbBaud_Auto	0	由底层驱动自动选择

MbChn		
名称	值	描述
MbChn_0	0	串口 1 操作句柄
MbChn_1	1	串口 2 操作句柄

MbError		
名称	值	描述
MbError_Ok	0	无错误
MbError_System	1	系统内部错误
MbError_CommChn	8	通信通道错误
MbError_CommHdl	11	通信句柄无效
MbError_CommTx	12	无法发送请求
MbError_CommOpen	19	无法打开通信端口
MbError_SlaveMaxNum	20	从站数量已达最大值
MbError_SlaveAddrress-Invalid	21	从站地址无效
MbError_SlaveNotCfg	22	从站未在网络列表中配置
MbError_SlaveInUse	23	从站地址已被占用
MbError_SlaveLoss	24	从站丢失
MbError_MsgMaxNum	30	Modbus 消息数量已达最大值
MbError_MsgFnzInvalid	31	Modbus 功能码无效

MbError		
名称	值	描述
MbError_MsgFnzSize	32	Modbus 功能码长度无效
MbError_MsgDatablock-Invalid	33	数据块无效
MbError_ParamInvalid	40	参数无效
MbError_ParamException	41	发生异常
MbError_ParamTimeout	42	超时

MbException		
名称	值	描述
MbExc_Ok	0	无异常
MbExc_IllegalFunction	1	非法功能码
MbExc_IllegalDataAddress	2	非法数据地址
MbExc_IllegalDataValue	3	非法数据值
MbExc_SlaveDeviceFailure	4	从站设备故障
MbExc_Acknowledge	5	应答确认
MbExc_SlaveDeviceBusy	6	从站设备繁忙
MbExc_MemoryParityError	7	存储器奇偶校验错误
MbExc_GatewayPathUnavailable	10	网关路径不可用
MbExc_GatewayTargetDevice	11	网关目标设备无响应

MbFunction		
名称	值	描述
MbFunc_ReadCoils	1	读线圈
MbFunc_ReadDiscreteInputs	2	读离散输入
MbFunc_ReadHoldingRegs	3	读保持寄存器
MbFunc_ReadInputRegs	4	读输入寄存器
MbFunc_WriteSingleCoil	5	写单个线圈
MbFunc_WriteSingleReg	6	写单个寄存器
MbFunc_WriteMultipleCoils	15	写多个线圈
MbFunc_WriteMultipleRegs	16	写多个寄存器

MbInpOutCmdMode		
名称	值	描述
MbInpOutCmdMode_Read	0	读取模式
MbInpOutCmdMode_Write	1	写入模式

MbMode		
名称	值	描述
MbMode_8N1	0	8 位数据位, 无校验, 1 位停止位
MbMode_8E1	1	8 位数据位, 偶校验, 1 位停止位
MbMode_8O1	2	8 位数据位, 奇校验, 1 位停止位
MbMode_8N2	3	8 位数据位, 无校验, 2 位停止位
MbMode_8E2	4	8 位数据位, 偶校验, 2 位停止位
MbMode_8O2	5	8 位数据位, 奇校验, 2 位停止位
MbMode_8S1	6	8 位数据位, 空校验, 1 位停止位
MbMode_8M1	7	8 位数据位, 标志校验, 1 位停止位
MbMode_8S2	8	8 位数据位, 空校验, 2 位停止位
MbMode_8M2	9	8 位数据位, 标志校验, 2 位停止位
MbMode_Auto	255	由底层驱动自动选择

MbParam		
名称	值	描述
MbParam_Init	0	初始化
MbParam_Busy	1	忙碌中
MbParam_SendRequest	2	发送请求
MbParam_WaitResponse	3	等待响应
MbParam_Ended	4	执行完成

MbRqErr		
名称	值	描述
MbRqErr_Ok	0	无错误
MbRqErr_IllegalFunction	1	请求消息无效
MbRqErr_IllegalDataAddress	2	非法数据地址
MbRqErr_IllegalDataValue	3	非法数据值
MbRqErr_SlaveDeviceFailure	4	从站设备故障
MbRqErr_Acknowledge	5	应答错误

MbRqErr		
名称	值	描述
MbRqErr_SlaveDeviceBusy	6	从站设备繁忙
MbRqErr_ParityError	7	存储器奇偶校验错误
MbRqErr_CommChn	8	通信通道配置错误
MbRqErr_InvalidSlaveAddress	9	无效从站地址 (地址范围 0-247)
MbRqErr_GatewayPathUnavailable	10	网关路径不可用
MbRqErr_GatewayDeviceFailed	11	网关目标设备无响应
MbRqErr_MsgMaxElements	12	功能元素数量无效
MbRqErr_MsgRequestInvalid	13	无法正确构建请求消息
MbRqErr_MsgResponseInvalid	14	响应消息无效
MbRqErr_MsgSize	15	响应长度错误
MbRqErr_Timeout	16	通信超时
MbRqErr_TaskInvalid	17	无效任务调用功能
MbRqErr_Sys	18	通用 Modbus 错误
MbRqErr_CommOpen	19	通信端口未打开
MbRqErr_CommTx	20	消息发送失败
MbRqErr_SlaveMax	21	已达到最大从站数量
MbRqErr_SlaveNotExists	22	无法获取从站句柄
MbRqErr_SlaveExists	23	从站已存在于同一网络
MbRqErr_SlaveLoss	24	从站丢失
MbRqErr_MSGDB	33	数据块物理地址无效
MbRqErr_PrmInvalid	40	参数化配置无效
MbRqErr_PrmException	41	参数化异常
MbRqErr_PrmTmo	42	参数化超时
MbRqErr_MemoryCfg	43	可用内存不足
MbRqErr_PollingTime	44	未遵守最小轮询时间间隔
MbRqErr_Stopped	45	Modbus 未完全运行时发起请求
MbRqErr_WriteDisabled	46	从站禁止写命令

MbRSMoDe		
名称	值	描述
RSMoDe_Auto	0	自动配置模式
RSMoDe_485	1	RS485 模式
RSMoDe_422	2	RS422 模式
RSMoDe_232	3	RS232 模式

MbSlaveAddressConfigMoDe		
名称	值	描述
MbSlaveAddressConfigMoDe_Static	0	静态地址
MbSlaveAddressConfigMoDe_ByVariableAddress	1	变量指定地址
MbSlaveAddressConfigMoDe_By16BitKeyValue	2	16 位键值指定地址
MbSlaveAddressConfigMoDe_By32BitKeyValue	4	32 位键值指定地址

MbStatus		
名称	值	描述
MbStatus_NotConfigured	0	未配置
MbStatus_Configured	1	已配置，等待启动
MbStatus_Configuring	2	配置中
MbStatus_Stopped	3	已停止
MbStatus_Starting	4	启动中，等待运行
MbStatus_Running	5	运行中

3.6.2 ModbusRTU_ReadCoils2

功能描述：

使用FC 01 Modbus功能从Modbus从站读取线圈。线圈值将写入提供的pDataOut缓冲区。从站无需在Modbus主网络（资源栏）中配置。此功能块，仅可在Background任务调用。

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号
VAR_INPUT	modbusAddr	UINT	Modbus 节点地址
VAR_INPUT	coilAddrStart	UINT	线圈起始地址
VAR_INPUT	coilsNum	UINT	读取线圈的数量
VAR_INPUT	pollingTime	UINT	最小轮询时间(毫秒)； 0 = 连续读取；大于 0 时,仅在上次调用后经过该最小时间才执行命令
VAR_INPUT	timeoutMs	UINT	超时时间（毫秒）
VAR_INPUT	waitBeforeSendMs	UINT	发送前等待时间(毫秒)
VAR_INPUT	pDataOut	PVOID	输出缓冲区地址(@BOOL 或 @WORD,参见 packed-Bits)
VAR_INPUT	sizeIn	UINT	输出缓冲区 pDataOut 的字节大小\
VAR_INPUT	packed-Bits	BOOL	若为 FALSE, pDataOut 应为 @BOOL 类型,接收拆分后的位;若为 TRUE, pDataOut 应为 @WORD 类型,接收原始未拆分值(暂未实现)
VAR_INPUT	_reserved	ARRAY OF BYTE [0..2]	保留字段
VAR_OUTPUT	sizeOut	UINT	写入 pDataOut 缓冲区的数据字节大小\
VAR_OUTPUT	result	MB_RESULT_VALUE	命令执行结果

3.6.3 ModbusRTU_ReadDeviceIdent

功能描述：

使用FC43/14 Modbus功能读取Modbus从站的设备标识。仅实现单个对象访问从站无需在Modbus主站网络中配置。此功能块，仅可在Background任务调用。

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号
VAR_INPUT	modbusAddr	UINT	Modbus 节点地址
VAR_INPUT	readDeviceIDCode	USINT	读取访问类型（暂未实现，仅支持 4 = 单独访问），默认值：4
VAR_INPUT	objectID	USINT	读取对象ID
VAR_INPUT	pDataOut	PVOID	输出缓冲区地址
VAR_INPUT	sizeIn	UINT	输出缓冲区 pDataOut 的字节大小
VAR_INPUT	timeoutMs	UINT	超时时间（毫秒）
VAR_INPUT	waitBeforeSendMs	UINT	发送前等待时间（毫秒）
VAR_OUTPUT	sizeOut	UINT	写入 pDataOut 缓冲区的数据字节大小
VAR_OUTPUT	result	MB_RESULT_VALUE	命令执行结果

3.6.4 ModbusRTU_ReadDiscrInputs2

功能描述：

使用FC02 Modbus功能从Modbus从站读取离散输入。输入值将写入提供的pDataOut缓冲区。

从站无需在Modbus主站网络中进行配置。此功能块，仅可在Background任务调用

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号
VAR_INPUT	modbusAddr	UINT	Modbus 节点地址
VAR_INPUT	discrInputStart	UINT	离散输入起始地址
VAR_INPUT	discrInputsNum	UINT	待读取离散输入数量
VAR_INPUT	pollingTime	UINT	最小轮询时间（毫秒）；0 = 连续读取；大于 0 时，仅在上次调用后经过该最小时间才执行命令
VAR_INPUT	timeoutMs	UINT	超时时间（毫秒）
VAR_INPUT	waitBeforeSendMs	UINT	发送前等待时间（毫秒）
VAR_INPUT	pDataOut	PVOID	输出缓冲区地址（@BOOL 或 @WORD，参见 packedBits）
VAR_INPUT	sizeIn	UINT	输出缓冲区 pDataOut 的字节大小
VAR_INPUT	packedBits	BOOL	若为 FALSE，pDataOut 应为 @BOOL 类型，接收拆分后的位；若为 TRUE，pDataOut 应为 @WORD 类型，接收原始未拆分值（暂未实现）
VAR_INPUT	_reserved	ARRAY [0..2] OF BYTE	保留字段
VAR_OUTPUT	sizeOut	UINT	写入 pDataOut 缓冲区的数据字节大小
VAR_OUTPUT	result	MB_RESULT_VALUE	命令执行结果

3. 6. 5 ModbusRTU_ReadHoldingRegs2

功能描述：

使用FC 03 Modbus功能从Modbus从站读取保持寄存器。寄存器值将写入提供的pDataOut缓冲区。从站无需在Modbus主网络中配置。此功能块，仅可在Background任务调用

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号
VAR_INPUT	modbusAddr	UINT	Modbus 节点地址
VAR_INPUT	regAddrStart	UINT	寄存器起始地址
VAR_INPUT	regNum	UINT	寄存器数量
VAR_INPUT	pollingTime	UINT	最小轮询时间(毫秒)； 0 = 连续读取; 大于 0 时, 仅在上次调用后经过该最小时间才执行命令
VAR_INPUT	timeoutMs	UINT	超时时间（毫秒）
VAR_INPUT	waitBeforeSendMs	UINT	发送前等待时间(毫秒)
VAR_INPUT	pDataOut	PVOID	输出缓冲区地址
VAR_INPUT	sizeIn	UINT	输出缓冲区 pDataOut 的字节大小
VAR_OUTPUT	sizeOut	UINT	写入 pDataOut 缓冲区的数据字节大小
VAR_OUTPUT	result	MB_RESULT_VALUE	命令执行结果

3. 6. 6 ModbusRTU_ReadInputRegs2

功能描述：

使用FC04Modbus函数从Modbus从站读取输入寄存器。寄存器值将写入提供的pDataOut缓冲区。从站无需在Modbus主站网络中进行配置。此功能块，仅可在Background任务调用

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号
VAR_INPUT	modbusAddr	UINT	Modbus 节点地址
VAR_INPUT	regAddrStart	UINT	寄存器起始地址
VAR_INPUT	regNum	UINT	寄存器数量
VAR_INPUT	pollingTime	UINT	最小轮询时间(毫秒)； 0 = 连续读取; 大于 0 时, 仅在上次调用后经过该最小时间才执行命令
VAR_INPUT	timeoutMs	UINT	超时时间（毫秒）
VAR_INPUT	waitBeforeSendMs	UINT	发送前等待时间(毫秒)
VAR_INPUT	pDataOut	PVOID	输出缓冲区地址
VAR_INPUT	sizeIn	UINT	输出缓冲区 pDataOut 的字节大小
VAR_OUTPUT	sizeOut	UINT	写入 pDataOut 缓冲区的数据字节大小
VAR_OUTPUT	result	MB_RESULT_VALUE	命令执行结果

3.6.7 ModbusRTU_WriteMultCoils2

功能描述：

使用FC15 Modbus功能向Modbus从站写入多个线圈。数值将从提供的pDataIn缓冲区中获取。从站无需在Modbus主网络中配置。此功能块，仅可在Background任务调用

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号
VAR_INPUT	modbusAddr	UINT	Modbus 节点地址
VAR_INPUT	coilStart	UINT	首个线圈地址
VAR_INPUT	coilsNum	UINT	线圈数量
VAR_INPUT	sizeIn	UINT	输入缓冲区 pDataIn 的字节大小
VAR_INPUT	pDataIn	PVOID	输入缓冲区地址
VAR_INPUT	polling-Time	UINT	最小轮询时间(毫秒); 0 = 连续写入; 大于 0 时, 仅在上次调用后经过该最小时间才执行命令
VAR_INPUT	timeoutMs	UINT	超时时间 (毫秒)
VAR_INPUT	waitBeforeSendMs	UINT	发送前等待时间(毫秒)
VAR_INPUT	packed Bits	BOOL	若为 FALSE, pDataIn 应为 @BOOL 类型, 包含待写入的拆分位; 若为 TRUE, pDataIn 应为 @WORD 类型, 包含待写入的原始未拆分值 (暂未实现)
VAR_INPUT	_reserved	BYTE	保留字段
VAR_OUTPUT	result	MB_RESULT_VALUE	命令执行结果

3.6.8 ModbusRTU_WriteMultRegs2

功能描述：

使用FC16 Modbus功能将多个寄存器写入Modbus从站。数值将从提供的pDataIn缓冲区中读取。从站无需在Modbus主站网络中配置。此功能块，仅可在Background任务调用

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号
VAR_INPUT	modbusAddr	UINT	Modbus 节点地址
VAR_INPUT	regAddrStart	UINT	首个寄存器地址
VAR_INPUT	regNum	UINT	寄存器数量
VAR_INPUT	sizeIn	UINT	输入缓冲区 pDataIn 的字节大小
VAR_INPUT	pDataIn	PVOID	输入缓冲区地址
VAR_INPUT	polling Time	UINT	最小轮询时间 (毫秒); 0 = 连续写入; 大于 0 时, 仅在上次调用后经过该最小时间才执行命令
VAR_INPUT	timeout Ms	UINT	超时时间 (毫秒)
VAR_INPUT	waitBeforeSendMs	UINT	发送前等待时间(毫秒)
VAR_INPUT	_reserved	WORD	保留字段
VAR_OUTPUT	result	MB_RESULT_VALUE	命令执行结果

3.6.9 ModbusRTU_WriteSingleCoil2

功能描述：

使用FC 05 Modbus功能向Modbus从站写入单个线圈。从站无需在Modbus主站网络中配置。此功能块，仅可在Background任务调用

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号
VAR_INPUT	modbusAddr	UINT	Modbus 节点地址
VAR_INPUT	coilAddr	UINT	线圈地址
VAR_INPUT	coilValue	BOOL	线圈值
VAR_INPUT	_reserved	BYTE	保留字段
VAR_INPUT	polling Time	UINT	最小轮询时间(毫秒); 0 = 连续写入;大于 0 时,仅在上次调用后经过该最小时间才执行命令
VAR_INPUT	timeoutMs	UINT	超时时间 (毫秒)
VAR_INPUT	waitBefore SendMs	UINT	发送前等待时间(毫秒)
VAR_OUTPUT	result	MB_RESULT_VALUE	命令执行结果

3.6.10 ModbusRTU_WriteSingleReg2

功能描述：

使用FC 06 Modbus功能将单个寄存器写入Modbus从站。从站无需在Modbus主站网络中配置。此功能块，仅可在Background任务调用

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号

引脚类型	名称	类型	描述
VAR_INPUT	modbusAddr	UINT	Modbus 节点地址
VAR_INPUT	regAddr	UINT	寄存器地址
VAR_INPUT	regValue	UINT	寄存器值
VAR_INPUT	polling Time	UINT	最小轮询时间(毫秒); 0 = 连续写入;大于 0 时,仅在上次调用后经过该最小时间才执行命令
VAR_INPUT	timeout Ms	UINT	超时时间 (毫秒)
VAR_INPUT	waitBefore SendMs	UINT	发送前等待时间 (毫秒)
VAR_OUTPUT	result	MB_RESULT_VALUE	命令执行结果

3.6.11 sysMbMRtu_ChangePortSettings

功能描述：

在运行时更改串行配置(关闭并重新打开通道以使用新设置)如果更改发生则返回TRUE，否则返回FALSE。

引脚类型	名称	类型	描述
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus 串口编号
VAR_INPUT	new_baudrate	MbBaudrate	新的波特率值
VAR_INPUT	new_frameMode	MbMode	新的帧模式值
VAR_INPUT	new_RSMode	MbRSMode	新的串口模式值

3.6.12 sysMbMRtu_DisableWriteForSlave

功能描述：

启用或禁用从设备的写入命令。当前值可以在相应的sysMbMRtuNetList[]元素中找到。

引脚类型	名称	类型	描述
VAR_INPUT	netId	USINT	Modbus 串口编号 (参考枚举 MB_RTU_NETWORK_ID)
VAR_INPUT	handle disable	USINT	Modbus 从站句柄 (由 sysMbMRtu_GetSlaveHandle 返回)
VAR_INPUT	Write	BOOL	为 TRUE 时禁用写入命令

3. 6. 13 sysMbMRtu_GetMasterStatus

功能描述：

Modbus主站RTU获取主站状态

引脚类型	名称	类型	描述
VAR_INPUT	netId	USINT	Modbus 串口编号 (参考枚举 MB_RTU_NETWORK_ID)
VAR_OUTPUT	cfgOk	BOOL	Modbus 主站已配置
VAR_OUTPUT	status	MbStatus	Modbus 主站状态机 (参考枚举 MbStatus)
VAR_OUTPUT	lastError	MbError	Modbus 主站最后错误 (参考枚举 MbError)
VAR_OUTPUT	nSlaves	USINT	网络列表中已配置的从站总数
VAR_OUTPUT	nQueuedMsg	UINT	已配置的广播和单从站消息总数
VAR_OUTPUT	nQueuedBroadcastMsg	UINT	已配置的广播消息总数
VAR_OUTPUT	nBusMsgCount	UINT	总线消息计数
VAR_OUTPUT	nBusCommErrCount	UINT	总线通信错误计数

3. 6. 14 sysMbMRtu_GetSlaveHandle

功能描述：

在对应从站的Netlist中获取Modbus主站RTU偏移量

引脚类型	名称	类型	描述
VAR_INPUT	netId	USINT	Modbus 串口编号 (参考枚举 MB_RTU_NETWORK_ID)
VAR_INPUT	adr	USINT	Modbus 从站地址

3. 6. 15 sysMbMRtu_GetSlaveStatus

功能描述：

Modbus主站RTU获取对应从站的状态

引脚类型	名称	类型	描述
VAR_INPUT	netId	USINT	Modbus 串口编号 (参考枚举 MB_RTU_NETWORK_ID)
VAR_INPUT	hdl	USINT	Modbus 从站句柄 (由 sysMbMRtu_GetSlaveHandle 返回)
VAR_OUTPUT	adr	USINT	Modbus 从站地址
VAR_OUTPUT	cfg	BOOL	Modbus 从站已添加到网络列表
VAR_OUTPUT	pres	BOOL	Modbus 从站存在于网络中
VAR_OUTPUT	active	BOOL	Modbus 从站通信正常
VAR_OUTPUT	n_prm	UINT	从站参数化消息总数
VAR_OUTPUT	n_qmsg	UINT	已配置的从站消息总数
VAR_OUTPUT	min_poll_time	UINT	最小轮询时间 (毫秒)
VAR_OUTPUT	status	MbStatus	Modbus 从站状态机 (参考枚举 MbStatus)
VAR_OUTPUT	error	MbError	Modbus 从站最后错误 (参考枚举 MbError)
VAR_OUTPUT	exception	MbException	Modbus 从站最后异常 (参考枚举 MbException)
VAR_OUTPUT	n_SlaveExceptionCount	UINT	从站异常错误计数
VAR_OUTPUT	n_SlaveMessageCount	UINT	从站消息计数

引脚类型	名称	类型	描述
VAR_OUTPUT	n_SlaveNoResp-	UINT	从站无响应计数
VAR_OUTPUT	onsCountn_SlaveNAKCount	UINT	从站否定应答计数
VAR_OUTPUT	n_SlaveBusyCo-	UINT	从站忙计数
VAR_OUTPUT	untChrOvr-	UINT	从站字符溢出计数
VAR_OUTPUT	Count		

3. 6. 16 sysMbMRtu_SetMaxRetriesForSlave

功能描述：
设置从属设备的最大重试次数:超过此值后，该从属设备将被宣布为非活动并丢失。当前值可在对应的sysMbMRtuNetList[]元素中找到。

引脚类型	名称	类型	描述
VAR_INPUT	netId	USINT	Modbus 串口编号（参考枚举 MB_RTU_NETWORK_ID）
VAR_INPUT	handle	USINT	Modbus 从站句柄（由 sysMbMRtu_GetSlaveHandle 返回）
VAR_INPUT	maxRe-	UINT	最大重试次数；16#FFFF表示无限重试
VAR_INPUT	tries		

3. 6. 17 sysMbMRtu_SlaveActivation

功能描述：
Modbus主站RTU激活/停用从站通信

引脚类型	名称	类型	描述
VAR_INPUT	netId	USINT	Modbus 串口编号（参考枚举 MB_RTU_NETWORK_ID）

引脚类型	名称	类型	描述
VAR_INPUT	handle	USINT	Modbus 从站句柄（由 sysMbMRtu_GetSlaveHandle 返回）
VAR_INPUT	activate	BOOL	为 TRUE 时,从站参与通信

3. 6. 18 sysMbMRtu_SlaveChangeAddress

功能描述：
Modbus主站RTU更改从站地址。地址只能在Init任务中更改。

引脚类型	名称	类型	描述
VAR_INPUT	netId	USINT	Modbus 串口编号（参考枚举 MB_RTU_NETWORK_ID）
VAR_INPUT	handle	USINT	Modbus 从站句柄（由 sysMbMRtu_GetSlaveHandle 返回）
VAR_INPUT	new_ad-	USINT	新的从站地址
VAR_INPUT	dress		

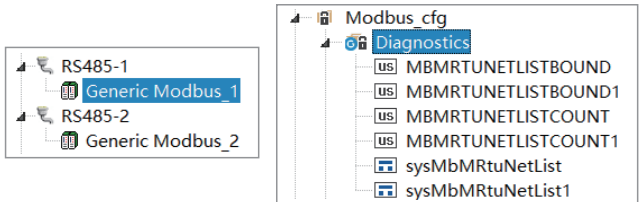
3. 6. 19 ModbusVars

变量说明：
该变量是ModbusRTU库中的全局变量，作用是简单查看主站状态，以及简单诊断主站状态。更多主站控制以及状态查询，需要在sysMbMRtuNetList[]元素中找到。

名称	类型	描述
sysMbMRtuStatus	MbMRtuStatus	Modbus主站RTU状态(网络0)
sysMbRtuDiagnostics	MbRtuDiagnostics	Modbus RTU标准诊断

3.6.20 sysMbMRtuNetList

说明：该变量是一个系统全局变量，由资源栏中的配置页面自动生成。路径为“工程栏窗口->Modbus_cfg->Diagnostics”当在资源栏中，添加了从站节点，点击保存编译后，会在工程栏目中出现该变量。由于X3000系列支持双主站，所以，如果在资源栏中还配置了RS485-2模块为主站且添加了从站节点，那么在工程栏中还会额外出现名称为sysMbMRtuNetList1的全局变量。如下图所示：



名称	类型	描述
sysMbMRtuNetList	MbMRtuNetList	Modbus主站RTU状态列表，对应资源栏中的RS485-1，该变量可以直接在
sysMbMRtuNetList1	MbMRtuNetList	Modbus主站RTU状态列表，对应资源栏中的RS485-2
MBMRTUNETLISTBOUND	USINT	该变量是一个常量，由资源栏配置，保存编译后生成，表示sysMbMRtuNetList数组能访问下标范围。对应RS485-1
MBMRTUNETLISTCOUNT	USINT	该变量是一个常量，表示在RS485-1 主站网络节点下，从站的个数
MBMRTUNETLISTBOUND1	USINT	该变量是一个常量，由资源栏配置，保存编译后生成，表示sysMbMRtuNetList1数组能访问下标范围。对应RS485-2
MBMRTUNETLISTCOUNT1	USINT	该变量是一个常量，表示在RS485-2 主站网络节点下，从站的个数

3.7 ModbusTCP

3.7.1 Enumerations

sysMbMTcpEsi		
名称	值	描述
MB_TCP_E_OK	0	命令执行成功
MB_TCP_E_NETWORK	-339	指定的网络 ID 无效
MB_TCP_E_CHECKSUM	-338	校验和错误
MB_TCP_E_NEGATIVEACKNOWLEDGE	-337	否定应答
MB_TCP_E_NOTALLOWEDINTIMEDTASK	-336	禁止在定时任务中调用该功能
MB_TCP_E_SWITCHEDOFF	-335	节点已关闭
MB_TCP_E_ILLEGALDATAPROTOCOL	-332	协议中存在非法数据值
MB_TCP_E_TIMEOUT	-331	请求超时
MB_TCP_E_GATEWAYDEVICEFAILED	-330	网关目标设备无响应
MB_TCP_E_GATEWAYPATHUNAV	-329	网关路径不可用
MB_TCP_E_MEMORYPARITYERROR	-328	存储器奇偶校验错误
MB_TCP_E_SLAVEDEVICEBUSY	-327	从站设备繁忙
MB_TCP_E_ACKNOWLEDGE	-326	应答异常
MB_TCP_E_SLAVEDEVICEFAILURE	-325	从站设备故障
MB_TCP_E_ILLEGALDATAVALUE	-324	非法数据值
MB_TCP_E_ILLEGALDATAADDR	-323	非法数据地址
MB_TCP_E_ILLEGALFUNCTION	-322	非法功能码
MB_TCP_E_IDE	-321	IDE 标识不匹配
MB_TCP_E_PROTOCOL	-320	协议错误
MB_TCP_E_DATATYPE	-311	数据类型参数无效
MB_TCP_E_NODENOTEXIST	-303	节点未配置 / 不存在
MB_TCP_E_NODE	-301	无效节点

sysMbMTcpSlaveAddressConfigMode		
名称	值	描述
MB_TCP_CONFIG_MODE_STATIC	0	静态地址
MB_TCP_CONFIG_MODE_BY_VAR_ADDRESS	1	变量指定地址
MB_TCP_CONFIG_MODE_BY_16BIT_KEY	2	16 位键值指定地址
MB_TCP_CONFIG_MODE_BY_32BIT_KEY	4	32 位键值指定地址

sysMbMTcpTypeD		
名称	值	描述
MB_TCP_DATA_T_SINT	0	有符号短整型
MB_TCP_DATA_T_USINT	1	无符号短整型
MB_TCP_DATA_T_BYTE	2	字节
MB_TCP_DATA_T_INT	3	整型
MB_TCP_DATA_T_UINT	4	无符号整型
MB_TCP_DATA_T_WORD	5	字
MB_TCP_DATA_T_DINT	6	双整型
MB_TCP_DATA_T_UDINT	7	无符号双整型
MB_TCP_DATA_T_DWORD	8	双字
MB_TCP_DATA_T_REAL	9	浮点型
MB_TCP_DATA_T_BOOL	10	布尔型
MB_TCP_DATA_T_NULL	255	空类型

3.7.2 Structures

MbTcpSlaveState		
名称	类型	描述
cfg	BOOL	已配置
pres	BOOL	设备在线
active	BOOL	节点激活
miss	BOOL	延时周期
IP	STRING[19]	IP 地址
address	UINT	子网 Modbus 地址

MbTcpSlaveState		
名称	类型	描述
missCnt	UINT	延时周期计数
state	INT	从站最近 Modbus 错误
netStat	UINT	从站网络状态

MbTcpState		
名称	类型	描述
netConfigOk	BOOL	网络配置正常
clientOK	BOOL	驱动客户端已初始化
clientRun	BOOL	驱动客户端运行中
taskRun	BOOL	通信任务运行中
sysMemoryError	BOOL	内存配置错误（该字段用于调试）

3.7.3 sysMbMTcp_ReadReg_R0

功能描述：

使用ReedInputRegister Modbus命令读取R0 Modbus寄存器

引脚类型	名称	类型	描述
VAR_INPUT	node	UINT	网络中节点的索引
VAR_INPUT	reg_addr	UINT	待读取寄存器的地址
VAR_INPUT	data_type	sysMbMTcpTypeD	待读取数据的类型
VAR_INPUT	tmo	UDINT	超时时间（毫秒）
VAR_OUTPUT	esi	sysMbMTcpEsi	操作结果
VAR_OUTPUT	res_dword	DWORD	读取结果统一存储为双字类型

3.7.4 sysMbMTcp_ReadReg_RW

功能描述：

使用ReadHoldingRegister Modbus命令读取RW Modbus寄存器

引脚类型	名称	类型	描述
VAR_INPUT	node	UINT	网络中节点的索引
VAR_INPUT	reg_addr	UINT	待读取寄存器的地址
VAR_INPUT	data_type	sysMbMTcp TypeD	待读取数据的类型
VAR_INPUT	tmo	UDINT	超时时间（毫秒）
VAR_OUTPUT	esi	sysMbMTcp Esi	操作结果
VAR_OUTPUT	res_dword	DWORD	读取结果统一存储为 双字类型

3.7.5 sysMbMTcp_WriteReg_M

功能描述：

使用WriteMultipleRegisters Modbus命令写入Modbus寄存器

引脚类型	名称	类型	描述
VAR_INPUT	node	UINT	网络中节点的索引
VAR_INPUT	reg_addr	UINT	待读取寄存器的地址
VAR_INPUT	data_type	sysMbMTcp TypeD	待读取数据的类型
VAR_INPUT	data_raw	DWORD	待写入的数据
VAR_INPUT	tmo	UDINT	超时时间（毫秒）
VAR_OUTPUT	esi	sysMbMTc- pEsi	操作结果

3.7.6 sysMbMTcp_WriteReg_S

功能描述：

使用WriteSingleRegister Modbus命令写入Modbus寄存器

引脚类型	名称	类型	描述
VAR_INPUT	node	UINT	网络中节点的索引
VAR_INPUT	reg_addr	UINT	待写入寄存器的地址
VAR_INPUT	data_type	sysMbMTcp TypeD	待写入数据的类型

引脚类型	名称	类型	描述
VAR_INPUT	data_raw	DWORD	待写入的数据
VAR_INPUT	tmo	UDINT	超时时间（毫秒）
VAR_OUTPUT	esi	sysMbMTc- pEsi	操作结果

3.7.7 Diagnostics

说明：

该变量是ModbusTCP库中的全局变量,可以在程序里面直接使用以下表格中的变量。去获取主从站中的状态。

名称	类型	描述
sysMbTcpMState	MbTcpState	系统变量，可直接查询ModbusTCP主站的相关状态
sysMbTcpSlaveState	MbTcpSlaveState	系统变量，可直接查询ModbusTCP从站的相关状态

4 其他说明及注意事项

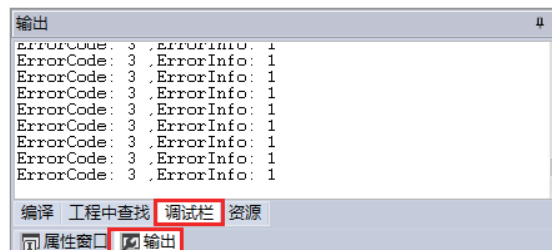
4.1 故障说明

当CPU模块 ERR指示灯红灯亮起时表示出现错误，引起错误的原因有两种：扩展模块组态错误（多组态模块、模块顺序错误）或者扩展模块本身错误。

输出窗口调试栏能够查看出错误的类型和从站号：

- 1) ErrorCode: 3表示扩展组态错误, ErrorCode: 1表示扩展模块自身错误；
- 2) ErrorInfo表示第几个扩展模块组态错误。

下图表示第一个扩展模块组态错误：



注意：X3000-CI-01-1拓展模块不参与组态，错误的扩展模块号不包含此模块，即当此模块安装在CPU模块右侧第一个位置时，ErrorInfo: 1实际组态错误模块为X3000-CI-01-1右侧的第一个模块。

错误排除后，重新下载程序并重新上电，控制器错误复位。

4.2 注意事项

- 1) 在梯形图 (LD) 编辑器中编辑程序时，将变量或功能块拖入程序中时，需要注意鼠标变为“”才能将其放在程序中；
- 2) 以ST语言式新建变量，在退出时需要点击保存按钮，否则会丢失更改的数据；
- 3) 控制器做Modbus TCP从站时参数中参数类型有32bit时,直接编译会导致地址冲突,需要对32bit参数及以下参数进行重新计算。按键盘shift或Ctrl选择需要重新计算的参数,点击重新计算。
- 4) 在使用BonengControlLib库中功能块时，有部分管脚必须要有初始值，否则功能块或者驱动器会报故障；驱动器内参数也需要设置，相关参数需要正确才能保证运行正常；
- 5) 如果变量已经在程序中调用,在变量表中删除变量,程序中此变量并不会被删除,会导致编译报错;
- 6) 在程序中删除对象时需要注意变量表中是否有变量为被选择状态（变量表中为黄色部分），否则将会误删；
- 7) 开始实际编译之前，确保至少已将程序分配给一个任务；
- 8) 监视窗口可以监视添加变量的当前值，也可以更改变量的值以用来检查应用程序的行为是否符合预期。
- 9) 使用变量表规定变量初始值时，如果变量初始值需要更改，编译程序后，需要将设备冷启动，才能将变量的初始值对应改变。

售后服务

售后服务请按以下步骤提交售后流程。

- 登录 “www.boneng.com”
- 点击 “服务” 和 “售后服务”



→ 登录系统

邮箱登录

新用户注册

请输入合法邮箱

请输入密码

请输入验证码



登录

其他地区

控制器/驱动器 : 0512-66182005
马达/齿轮马达/齿轮箱 : 0512-66189918

博能传动（沈阳）有限公司

控制器/驱动器 : 024-31271571
马达/齿轮马达/齿轮箱 : 024-31292571

博能传动（天津）有限公司

控制器/驱动器 : 022-86928559
马达/齿轮马达/齿轮箱 : 022-26929558

博能传动（开封）有限公司

控制器/驱动器 : 0371-23335230
马达/齿轮马达/齿轮箱 : 0371-23277771

博能传动（潍坊）有限公司

控制器/驱动器 : 0536-4699687
马达/齿轮马达/齿轮箱 : 0536-4699667

博能传动（长沙）有限公司

控制器/驱动器 : 0731-88386958
马达/齿轮马达/齿轮箱 : 0731-88380725

博能传动设备（成都）有限公司

控制器/驱动器 : 028-87740066
马达/齿轮马达/齿轮箱 : 028-87740066

博能传动（肇庆）有限公司

控制器/驱动器 : 0757-86719757
马达/齿轮马达/齿轮箱 : 0758-2699830

博能传动（苏州）有限公司

控制器/驱动器 苏南区: 0512-66182005
马达/齿轮马达/齿轮箱 苏南区: 0512-66189918
控制器/驱动器 浙沪区: 0512-66182005
马达/齿轮马达/齿轮箱 浙沪区: 0512-66189918
控制器/驱动器 苏皖区: 0512-66182005
马达/齿轮马达/齿轮箱 苏皖区: 025-52171612

博能传动（美国）有限公司

技术支持/调试/售后服务:
1250 E 222nd Euclid, OH 44117, United Staes
Email: America@boneng.com
Tel: 1-216-618-3099/1-216-618-0138

博能传动（印度）有限公司

技术支持/调试/售后服务:
Plot No. E-10/3, MIDC sinnar (Malegaon) Industrial Area,
Nashik, 422123, Maharashtra, India.
Email: india@boneng.com
Tel: +91-11-4507 6293/+91-22-2781 3385

Contents

1 Controller technical data and installation	
wiring	333
1.1 CPU Module	333
1.2 Technical data	333
1.2.1 Terminal Introduction	336
1.2.2 CPU Module wiring diagram	336
1.3 Expansion Module	337
1.3.1 Expansion Module General Data	337
1.3.2 Digital Input Module	337
1.3.3 Digital Output Module	338
1.3.4 Analog input module	339
1.3.5 Analog output module	339
1.3.6 Communication Interface Module	340
1.3.7 Wiring Diagram	341
1.4 Installation dimensions	341
2 Controller software installation and programming	
debugging	342
2.1 Quick Start	342
2.1.1 BonengLogicLab Software installation	342
2.1.2 Software programming	344
2.1.3 Communication settings	345
2.1.4 Typical steps of user programming	345
2.1.5 Basic framework for running BonengLogicLab program	346
2.1.6 Programming and Debugging	346
2.1.7 Switching the PLC operating mode	355
2.1.8 Online monitoring and debugging	356
2.1.9 Oscilloscope applications	359
2.1.10 Program scan cycle setting	362
2.1.11 EtherCAT Task Cycle Setting	363
2.1.12 Importing and exporting objects	363
2.2 Programming Basics	364
2.2.1 General	364
2.2.2 Variables	364
2.2.3 Resources - Parameters (Power Down Maintenance)	375
2.2.4 Resources - State Variables (Power Down Not Maintained)	378
2.2.5 Library Management Wizard	378
2.2.6 Cross referencing	381
2.2.7 Refactoring	382
2.2.8 Search	383
2.2.9 Simulation	384
2.2.10 Upload	388
2.2.11 Firmware Upgrade	390
2.3 Programming Language	393
2.3.1 Programming Language Types	393
2.3.2 Ladder Diagram (LD)	393
2.3.3 Structured Text Programming (ST)	401
2.3.4 Functional Block Diagram (FBD)	403
2.3.5 Sequential Function Diagram (SFC)	403
2.3.6 Instruction List (IL)	403
2.4 Expansion Module	403
2.4.1 Digital Input DI	406
2.4.2 Digital Output DQ	410
2.4.3 Analog Input AI	411
2.4.4 Analog output AQ	412
2.4.5 Communication module CI	412
2.5 Serial Communication	412
2.5.1 Overview	412
2.5.2 Modbus RTU Master Station Configuration	413
2.5.3 Modbus RTU Slave Configuration	417
2.5.4 Modbus Fault Code Explanation	418
2.6 Ethernet Communication	420
2.6.1 Overview	420
2.6.2 Modbus TCP Host Configuration	420
2.6.3 Modbus TCP Slave Configuration	421
2.6.4 Modbus Fault Code Explanation	421
2.7 EtherCAT Communication	422
2.7.1 Overview	422
2.7.2 EtherCAT Master Station Configuration	422
2.7.3 EtherCAT Slave Configuration	427

Contents

2.8 Motion Control	435	3.5.2 Master control functions	661
2.8.1 Introduction to Motion Control Axis	435	3.5.3 Master diagnostic functions	662
2.8.2 General steps for creating axis objects-	437	3.5.4 SDO function blocks	665
2.8.3 Axis Page Configuration	440	3.5.5 Vars(EtherCAT)	670
2.8.4 Overview of Axis Configuration/Status-	457	3.6 ModbusRTU	671
2.8.5 Fault Categories	457	3.6.1 Definitions	671
2.9 Electronic cam	458	3.6.2 ModbusRTU_ReadCoils2	680
2.9.1 Introduction to Electronic Cams	458	3.6.3 ModbusRTU_ReadDeviceIdent	681
2.9.2 Software Configuration	458	3.6.4 ModbusRTU_ReadDiscrInputs2	682
2.9.3 Use of electronic cam	464	3.6.5 ModbusRTU_ReadHoldingRegs2	683
2.10 Electronic Gear	468	3.6.6 ModbusRTU_ReadInputRegs2	684
2.10.1 Gear Action	468	3.6.7 ModbusRTU_WriteMultCoils2	685
2.10.2 Use of electronic gears	468	3.6.8 ModbusRTU_WriteMultRegs2	686
2.11 Axes Group	471	3.6.9 ModbusRTU_WriteSingleCoil2	687
2.11.1 Introduction to Axes group	471	3.6.10 ModbusRTU_WriteSingleReg2	688
2.11.2 Software Configuration	471	3.6.11 sysMbMRtu_ChangePortSettings	689
3 Instruction for Function Blocks	478	3.6.12 sysMbMRtu_DisableWriteForSlave	689
3.1 BonengControlLib	478	3.6.13 sysMbMRtu_GetMasterStatus	689
3.1.1 BN_EtherCAT	478	3.6.14 sysMbMRtu_GetSlaveHandle	690
3.1.2 BN_Basic	505	3.6.15 sysMbMRtu_GetSlaveStatus	691
3.1.3 BN_Modbus	507	3.6.16 sysMbMRtu_SetMaxRetriesForSlave...	692
3.2 BonengEmbedLogicLib	509	3.6.17 sysMbMRtu_SlaveActivation	692
3.2.1 ArithmeticOperationClassExtLib	509	3.6.18 sysMbMRtu_SlaveChangeAddress	693
3.2.2 ClockClassExtLib	510	3.6.19 ModbusVars	693
3.2.3 DataConversionClassExtLib	511	3.6.20 sysMbMRtuNetList	693
3.2.4 ConfigClassExtLib	512	3.7 ModbusTCP	695
3.3 BonengMachiningProcessLib	516	3.7.1 Enumerations	695
3.3.1 BN_TensionControl	516	3.7.2 Structures	696
3.4 MotionControl	523	3.7.3 sysMbMTcp_ReadReg_RO	697
3.4.1 MotionControl Lib Introduce	523	3.7.4 sysMbMTcp_ReadReg_RW	697
3.4.2 Basic	535	3.7.5 sysMbMTcp_WriteReg_M	698
3.4.3 Cam and Gear	584	3.7.6 sysMbMTcp_WriteReg_S	698
3.4.4 Encoder	600	3.7.7 Diagnostics	698
3.4.5 Axes Group Control	602	4 Other Notes and Precautions	699
3.5 EtherCAT	647	4.1 Fault description	699
3.5.1 Definitions	647	4.2 Precautions	700
		After-sale service	701

1 Controller technical data and installation wiring

1.1 CPU Module

1.2 Technical data

The CPU modules of the Boneng X3000 series controller are Medium-sized PLC that supports motion control. The specific order numbers are as follows

Order number	Description
X3050-EC-00-S	Basic PLC support motion control

The X3050 function are as follows

	X3050
Basic functions of PLC	✓
EtherCAT	✓
Modbus	✓
Ethernet	✓
Maximum number of EtherCAT axis	32
Maximum number of local expansion IO moduler	16
Speed control	✓
Torque control	✓
Position control	✓
PID controler	✓
Tension Control	✓
MC Standard Function Block	✓
Encoder axis	✓
Virtual axis	✓
Electronic gear synchronization	✓
Electronic cam synchronization	✓
Group of Axes(PTP)	✓
Group of Axes(Spatial linear interpolation)	✓
Group of Axes(Spatial arc interpolation)	✓
Group of Axes(Longmen master-slave synchronization)	✓
Group of Axes(attitude control)	✓

The following table shows the application of controller functions in specific industries.

X3050	
Speed	Synchronous/cam
Crane	Concave-printing machine
Door crane	Laminator
Tower crane	Coater
Drawbench	Rewinder
Hoist	Slitter
Conveyor	Embossing machine
Roller press	Printing machine
Veneer reeling machine	Tensioning machine
Mixer	Slasher
	Spinning machine
	Twister
	Coloring machine
	Multi-wire saw
	Filling machine
	Capper
	Labeller
	Case packer
	Bending machine
	Clipper
	Forming machine
	Fly saw
	Fly shear
	Gluing machine
	Truss robot
	Gantry machine tool
	Intelligent bridge crane

The main parameters are shown in the table below

Order Number	X3050-EC-00-S	Standard Controllability
CPU Module	X3050	ARM-A8 800MHz
Storage	User program space	1MB
	User data space	8MB
	Power off storage space	4kB
Working power supply	External terminal power supply	Voltage range: DC 19.2 ... 28.8V Maximum current: 2.6A
Communication interface	EtherCAT	1* RJ45
	Ethernet	1* RJ45
	Modbus	No built-in terminating resistor
Indicator light	3 monochrome indicator lights	PWR indicator light:green RUN Indicator light:green ERR indicator light:red
Daily clock	Last for 1 week after power outage	
Programming	LD	✓
Language	ST	✓
	IL	✓
	FBD	✓
	SFC	✓
Expanded quantity	15 DI/DQ/AI/AQ Module + 1 CI Module	
Working temperature	-20°C ... +50°C	
Storage temperature	-40°C ... +70°C	
Relative air humidity	<95%	No condensation is allowed
Pollution Level	Conform IEC 61800-5-1	Suitable for environment with pollution level 2

Electromagnetic compatibility	Conform IEC 61800-3
IP Grade	IP20
Weight	230g

1.2.1 Terminal Introduction

Ethernet Interface

Type	RJ45 female socket with metal shielding
Function	Connect Ethernet protocol devices
	Upgrade module firmware
	Download user program



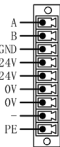
EtherCAT Interface

Type	RJ45 female socket with metal shielding
Function	Connect EtherCAT protocol devices



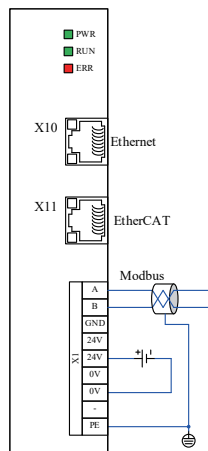
X1 terminal

Power supply	24V DC(±20%), max.2.6A
Communication	Modbus-RTU



1.2.2 CPU Module wiring diagram

When the green PWR indicator light is on, it indicates that the power supply has been connected. When the green RUN indicator light flashes, it indicates that the CPU module is running normally. When RUN is not flashing, it indicates that the CPU module is running abnormally. When the red ERR indicator light is on, it indicates that there is an error in the CPU module, and the power needs to be restored after troubleshooting. Check if the green RUN indicator light is flashing and running normally. If it is still not normal, it is necessary to contact the relevant technical department for maintenance.



1.3 Expansion Module

1.3.1 Expansion Module General Data

General	Monochrome	RDY indicator light: Green indicator light
Working temperature	-20°C ... +50°C	
Storage temperature	-40°C ... +70°C	
Relative air humidity	<95%, No condensation allowed	
Pollution level	Compliant with IEC 61800-5-1, suitable for environments with pollution	
Electromagnetic	Compliant with IEC 61800-3	
IP Grade	IP20	

1.3.2 Digital Input Module

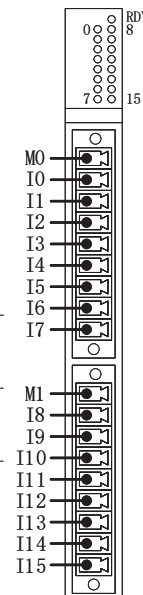
Technical data

X3000-DI-16-1	No external power supply required
	Electrical isolation
	16 * DI terminal
	Every 8 share a common terminal, M0, M1

Voltage: DC24V
 24V Rated current: 4mA
 Support source and sink mode
 "1" Signal Voltage: >11V
 "0" Signal Voltage: <5V

Indicator light status

- RDY Green light indicates the connection status of the module
 Flashing indicates waiting for configuration
 Slow flashing indicates normal operation
 0~15 Green light indicates input status
 Constant light indicates input 1



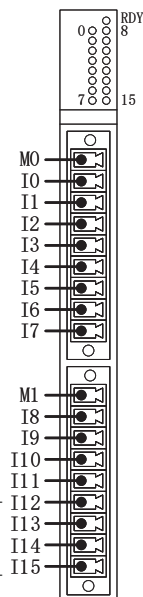
1.3.3 Digital Output Module

Technical data

X3000-DQ-16-3	No external power supply required
	Relay type
	16 * DO terminal
	Every 8 share a common terminal, M0, M1
	Voltage:DC24V、 AC36V、 AC48V
	Continuous current: max 2A
	Switching current: max 2A
	Contact type:1 normally open

Indicator light status

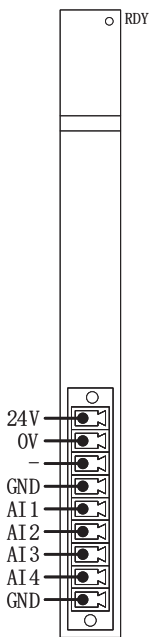
- RDY Green light indicates the connection status of the module
 Flashing indicates waiting for configuration
 Slow flashing indicates normal operation
 0~15 Green light indicates input status
 Constant light indicates input 1



1.3.4 Analog input module

Technical data

X3000-AI-04-3	External supply DC24V ± 20%/max.150mA	
	4 channels	
	Single ended input, shared GND	
	Voltage/current mode programmable	
	Voltage type supports 0V-10V	
	Current type supports 0mA-20mA, 4mA-20mA	
	Current type internal resistance 250 Ω	
	Voltage type: greater than 50k Ω	
	Accuracy: ± 1%	



Indicator light status

RDY Green light indicates the connection status of the module
Flashing indicates waiting for configuration
Slow flashing indicates normal operation

1.3.5 Analog output module

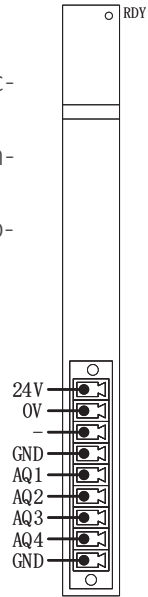
Terminal connection

X3000-AQ-04-1	External supply DC24V ± 20%/max.150mA
	4 channels
	Single ended input, shared GND
	Voltage type supports 0V-10V
	Voltage type load resistance should be greater than 5k Ω
	Accuracy: ± 1%

Indicator light status

RDY Green light indicates the connection status of the module
Flashing indicates waiting for configuration
Slow flashing indicates normal operation

PS: The switching of voltage and current input detection for analog signals is implemented by the custom library of Boneng. Please refer to section 3.2.4 for specific calls



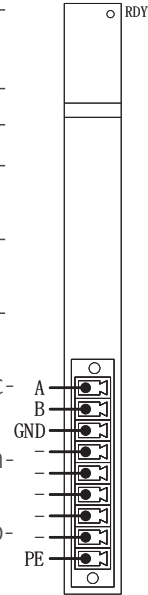
1.3.6 Communication Interface Module

Technical data

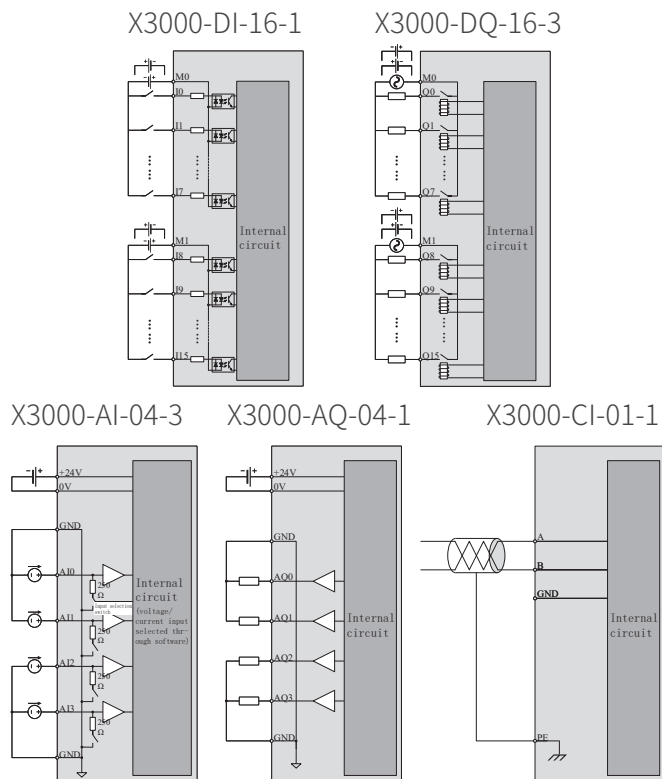
X3000-CI-01-1	No external power supply required
	1 channel
	Supports Modbus
	Baud rate: maximum 115200bps
	No onboard terminal resistor

Indicator light status

RDY Green light indicates the connection status of the module
Flashing indicates waiting for configuration
Slow flashing indicates normal operation

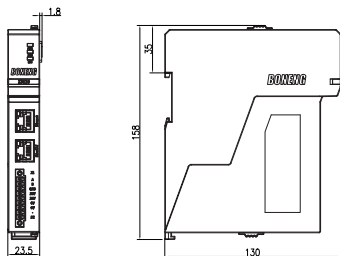


1.3.7 Wiring Diagram



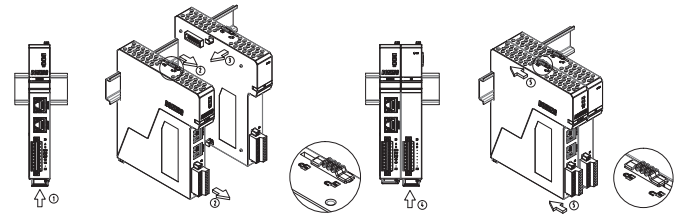
1.4 Installation dimensions

The shape and installation dimensions of the module are shown in the figure below. During the actual installation, there should be 60mm of space above and below the module to facilitate heat dissipation.



Using a standard 35mm wide DIN rail mount, mount the CUP to the correct position on the rail and hold it in place with the snap-in slider at the

bottom pushed up. After the docking of the expansion module is completed, push up the snap slider to fix it, and dial the fixed latches above and below the module body on the left to the locked state.



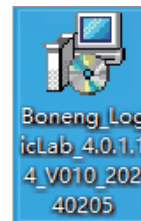
2 Controller software installation and programming debugging

2.1 Quick Start

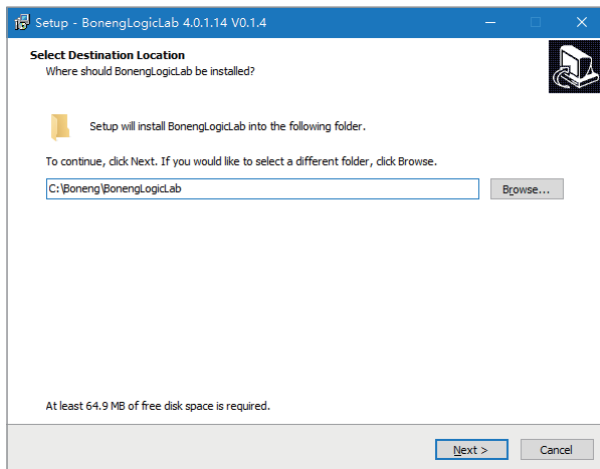
2.1.1 BonengLogicLab Software installation

Please download the latest version of the installation package from the www.boneng.com website of Bonang Transmission (Suzhou) Co. The installation steps are as follows:

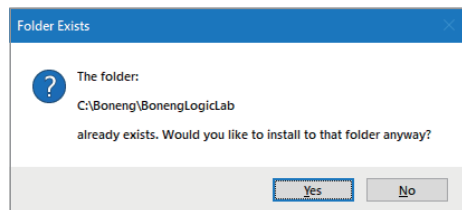
1) Run the installation package



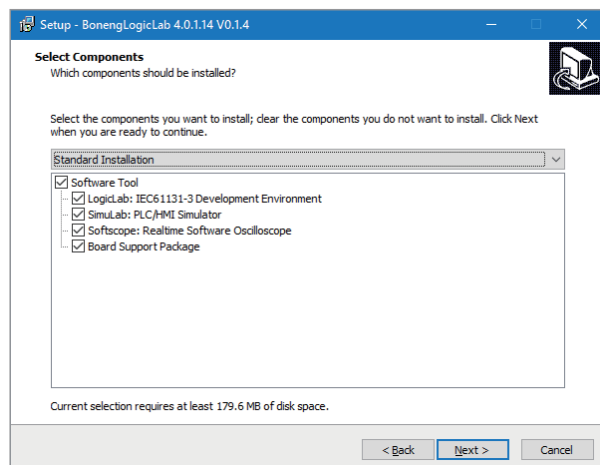
- 2) Language selection: Simplified Chinese and English are available for selection;
- 3) Agree to the agreement;
- 4) Select the installation path of the software, click Browse to change the location of the installation path, the installation path can not have Chinese;



If you click next to encounter the folder already exists, this time you need to go to this check whether you uninstalled the previous programming software, if so, click Yes to continue the installation.



5) Select standard installation;



6) According to the system prompts to complete the installation of the software, installed after the desktop icon as shown;



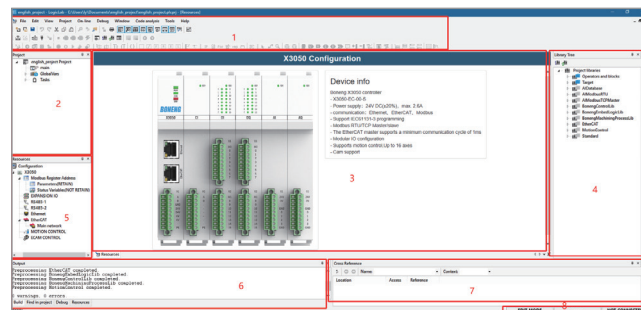
2.1.2 Software programming

2.1.2.1 Introduction to the software

The programming software for the small and medium-sized X3000 series is BonengLogiclab, which has a user-friendly interface that makes it easy for users to configure, program, debug, download, and monitor the PLC according to their application needs.

Programming Interface

The main interface of the programming software is shown below:



1. Main menu and shortcut tools: Programming software operation menu, including programming, debugging, communication and other related settings, file management and programming tool shortcuts.

2. Program engineering area, creating task programs, global variable lists, structures, function blocks, enumerations.

3. The main display area, which mainly shows the resource configuration, and programming areas.

4. Other auxiliary display areas, mainly containing the library management wizard, properties window, monitoring window, oscilloscope.
5. Resource allocation area, mainly for configuration of peripheral equipment.
6. Output display area, mainly displaying compilation results, compilation information, version information and so on.
7. Cross-reference area, mainly to facilitate and quickly locate the global search variables, the use of function blocks and other information;
8. Status display area, including communication connection status, etc.

2.1.3 Communication settings

2.1.3.1 Communication connections

Only supports communication connection between Ethernet and BonengLogicLab software, enabling program upload, download, monitoring, and debugging.

2.1.3.2 Ethernet Connection

The connection of BonengLogicLab is very simple, there is no need to manually set the IP address. Just click on the "Execute Network Scan" button in the toolbar, select the user's device, and click "OK". The software will automatically switch the IP address of the project. Easy to connect later.

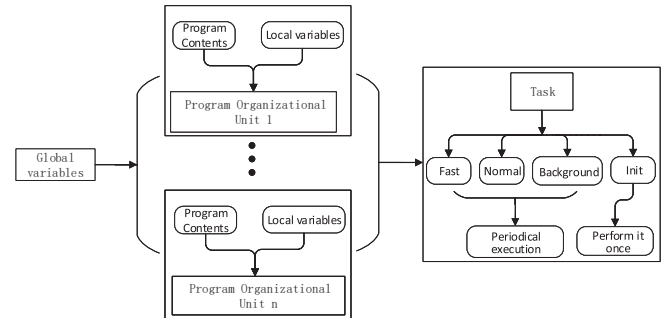
2.1.4 Typical steps of user programming

Users who are using the Boneng PLC for the first time need to pay attention to the following steps when writing and debugging a program. Taking the X3000 series PLC as an example, this will be explained.

1. New project for X3000 series.
2. Configure the hardware system according to the required peripherals.
3. Establish task programs and variables based on one's own task requirements.

4. Scan network selection control, connect to PLC, compile, download, debug.

2.1.5 Basic framework for running BonengLogiclab program



The difference between global variables and local variables:

Global variables: can be used in any program organization unit in this project.

Local variables: can only be used within the organizational unit of the program.

Global variables include custom global variable groups, GlobalVars (originally included in the project), I-O-mappings (generated when configuring extensions), Global shared (generated when configuring parameters and state variables), Modbus_Vars (generated when configuring Modbus communication), and variables in the library management wizard.

2.1.6 Programming and Debugging

2.1.6.1 Example Engineering Requirements

Complete programming and debugging of a flashing light program in the BonengLogiclab environment; Using a 16 point digital output module, PLC programming achieves conduction and disconnection every 1 second. Due to the fact that the digital output module is of the relay type, a beep sound can be heard when it lights up, and the sound of the electromagnet in the relay engaging can be heard. The indicator light will also light up to indicate engagement

2.1.6.2 New project

To create a new X3000 series project in BonengLogicLab programming software, proceed as follows:

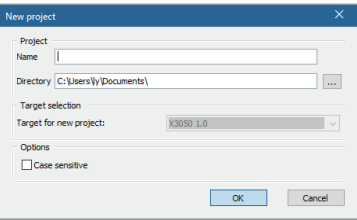
There are two ways to create a project, the first is to create a project with a controller and the other is to create a project without a controller. The specific use of that method depends on the user's own site conditions. These two methods are described below:

Type 1:

1. Open the BonengLogicLab programming software and go to the welcome interface. Clicking on 'Scan Local Network...' will display the actual connected devices scanned. It is worth noting that the computer's IP address needs to be in the same network segment as the controller's IP address. Otherwise, it may be impossible to directly connect to the device after scanning and establishing the project.



2. Clicking on ⊕ will bring up the following dialog box. Enter the user's project name, select the path of the project that the user wants to store, and click OK.



Type 2:

1. In the welcome page, click to select the user's desired control device, then click to create a basic project and then, enter the project name, the project storage path and click OK, then a basic project will be created according to the user's selected device.

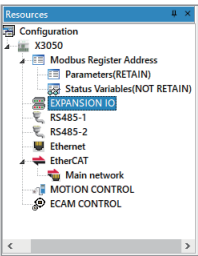


It is worth noting that there is also a list of recently used projects below the welcome page, which records and saves the information of the most recently opened projects, so that it is convenient to open the project directly by double-clicking it. In the control selection, click on the controller below the list of recently used projects will automatically help users filter out other controller types of projects, in order to quickly lock the project.

2.1.6.3 Configuring the hardware system

The procedure for hardware configuration is as follows:

After creating a new project, you can see the extension module inside the resource list: as shown below:



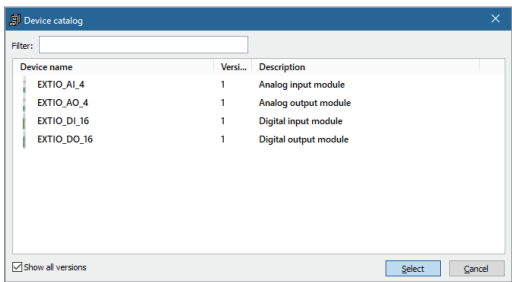
Double click to open the extension module, you can see the following operation page:



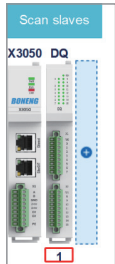
At this point, you can manually or automatically add 5 types of extension modules, depending on the configuration of the actual extension module. The specific use of that kind, need to be used in accordance with the specific circumstances.

Way 1, manually add: Click the plus sign in the above picture, the following page will pop up.

Choose an extension module that corresponds to the actual installed slot number. Otherwise, you need to modify the position of the slot number and modify the extension slot number, as explained below.



Click on the selection to successfully add, as shown in the figure below:

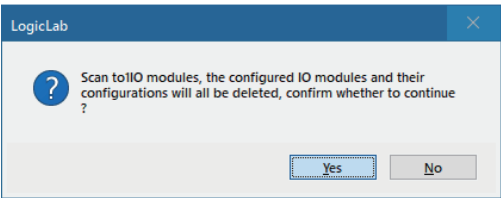


It can be seen that there is a label 1 under the DQ module, indicating that the manually added DQ module is in slot one.

Way 2: Auto Add. Click on Scan Slave in the following image,



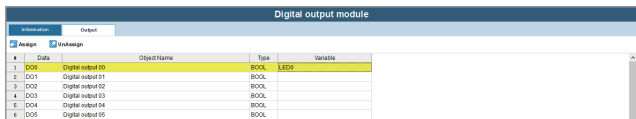
After clicking, it will pop up that no extension RS485 interface was found. After clicking OK, the actual number of extension modules scanned will appear. As shown in the following figure:



Click Yes (Y), and the slave display interface in the software will be arranged according to the actual placement order of the backplane slot numbers, corresponding to them. Added 1 extension module here.

After configuring the expansion module according to the above requirements, assign the corresponding PLC variable to each output point (any variable name can be directly entered, and the corresponding variable will be generated directly after saving).

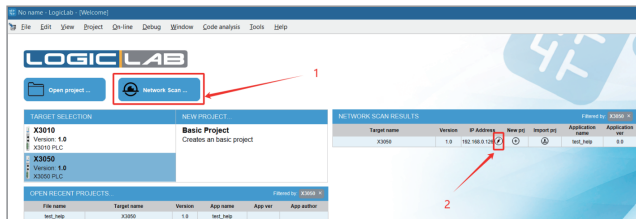
Click on the added module, click on output, and enter the PLC variable name corresponding to the point in the variable column to bind it, as shown in the following figure:



After clicking 'save', a new global variable group named 'I_O-mappings' will appear in the project bar, where the variable names entered above are stored. Convenient for future use in program development.

2.1.6.4 IP Address Modification

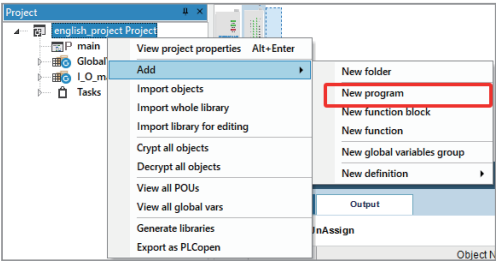
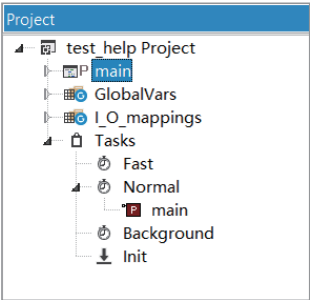
When this controller uses Ethernet to establish communication with HMI or other upper computers, it is necessary to configure the IP address of the PLC.



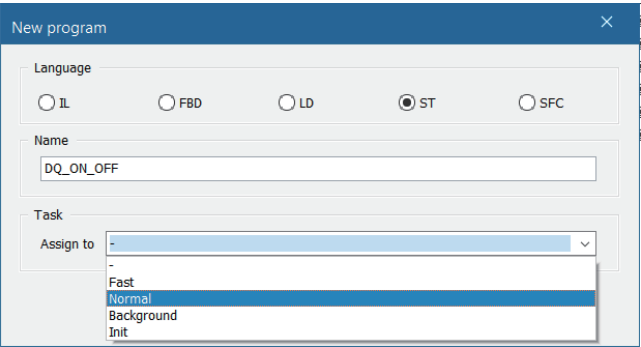
In the welcome interface, first click to scan the local network. In the scanned results, after clicking, a dialog box will pop up to modify the IP address of the controller. The default port number for the controller is 502 and cannot be modified at this time.

2.1.6.5 Programming and Compilation

After creating a new project, it will come with a program organization unit named 'main'. The program defaults to using ST language. Here, for the convenience of everyone to understand how to create a program organization unit and create a new one.

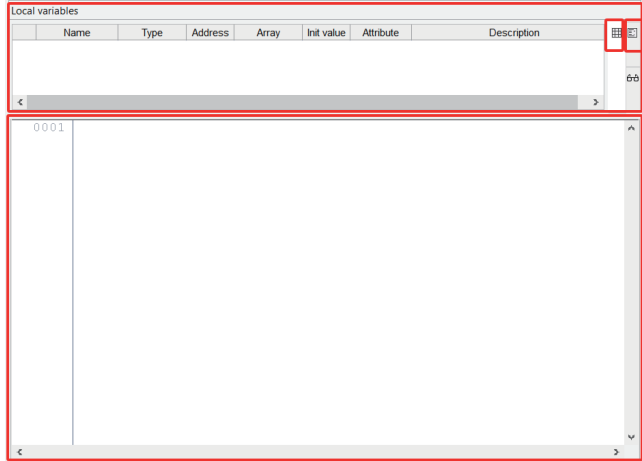


Click to create a new program as shown in the following figure:



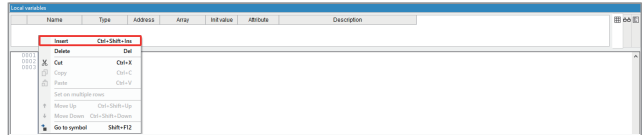
Here you can choose five programming languages, select ST here (the operation methods of other programming languages are basically the same as ST), and then enter the name of the program organization unit, which is named DQ-ON_OFF here. Here, you can assign the program to the tasks you want to put into it, and select Normal task here. Then click OK.

Click on the newly created POU in the project, as shown in the following figure:



Local variables can be created in two modes: table format and text format.

It is recommended to use a table format here. Right click on the blank area of the local variable and insert it to add a variable.



After adding a default variable, users can modify the relevant configuration of this variable. The settings that a variable should have include the variable name, variable type, variable address (default is Auto if not set), whether the variable is an array, the initialization value of the variable, whether the variable is a constant or a power-off hold variable (to be modified in the variable properties), and the description of the variable. It is worth noting that the power-off hold variable can only be a global variable. As shown in the following figure:

Local variables							
	Name	Type	Address	Array	Init value	Attribute	Description
1	NONAME	BOOL	Auto	No			

When modifying the type of a variable, you can manually input the variable type. You can also click on the position shown in the figure below to bring up the object browser and select the desired variable type.

Local variables							
	Name	Type	Address	Array	Init value	Attribute	Description
1	NONAME	BOOL	Auto	No			

Now, using the method of creating PLC variables above, create the variables shown in the following figure one after another, and edit the program in DQ-ON_OFF. The edited program is as follows:

Local variables							
	Name	Type	Address	Array	Init value	Attribute	Description
1	TIME2S_TON	TON	Auto	No			
2	TIME_IN	BOOL	Auto	No			
3	TIME_2S	TIME	Auto	No	T#2S		
4	TIME_Q	BOOL	Auto	No			
5	TIME_ET	TIME	Auto	No			

```

0001 IF NOT TIME_Q THEN
0002   TIME_IN:=TRUE;
0003 END_IF;
0004
0005 TIME2S_TON(IN := TIME_IN(* BOOL *),
0006            PT := TIME_2S(* TIME *),
0007            Q => TIME_Q(* BOOL *),
0008            ET => TIME_ET(* TIME *));
0009
0010 IF TIME2S_TON.ET < T#1S THEN
0011   LED0:=TRUE;
0012 ELSE
0013   LED0:=FALSE;
0014 END_IF;
0015
0016 IF TIME_Q THEN
0017   TIME_IN := FALSE;
0018 END_IF;
0019

```

Click the  button in the shortcut menu bar, or click on the project in the menu bar, click compile, or compile all. As shown in the following figure:

Local variables							
	Name	Type	Address	Array	Init value	Attribute	Description
1	TIME2S_TON	TON	Auto	No			
2	TIME_IN	BOOL	Auto	No			
3	TIME_2S	TIME	Auto	No	T#2S		
4	TIME_Q	BOOL	Auto	No			
5	TIME_ET	TIME	Auto	No			

```

0001 IF NOT TIME_Q THEN
0002   TIME_IN:=TRUE;
0003 END_IF;
0004
0005 TIME2S_TON(IN := TIME_IN(* BOOL *),
0006            PT := TIME_2S(* TIME *),
0007            Q => TIME_Q(* BOOL *),
0008            ET => TIME_ET(* TIME *));
0009
0010 IF TIME2S_TON.ET < T#1S THEN
0011   LED0:=TRUE;
0012 ELSE
0013   LED0:=FALSE;
0014 END_IF;
0015
0016 IF TIME_Q THEN
0017   TIME_IN := FALSE;
0018 END_IF;
0019

```

After clicking, a lot of information will be output in the compilation bar of the output box. After there are no errors, you can proceed with the download operation.

Local variables							
	Name	Type	Address	Array	Init value	Attribute	Description
1	NONAME	BOOL	Auto	No			

2.1.6.6 Communication Settings

This controller is connected to PLC via Ethernet.

1 Click the  button in the menu bar to connect. If it prompts a connection failure, click Tools and then click Execute Network Scan. If the network cable connection is normal, the scanned device will pop up. Select the device that the user needs to connect to and click OK.

Local variables							
	Name	Type	Address	Array	Init value	Attribute	Description
1	TIME2S_TON	TON	Auto	No			
2	TIME_IN	BOOL	Auto	No			
3	TIME_2S	TIME	Auto	No	T#2S		
4	TIME_Q	BOOL	Auto	No			
5	TIME_ET	TIME	Auto	No			

```

0001 IF NOT TIME_Q THEN
0002   TIME_IN:=TRUE;
0003 END_IF;
0004
0005 TIME2S_TON(IN := TIME_IN(* BOOL *),
0006            PT := TIME_2S(* TIME *),
0007            Q => TIME_Q(* BOOL *),
0008            ET => TIME_ET(* TIME *));
0009
0010 IF TIME2S_TON.ET < T#1S THEN
0011   LED0:=TRUE;
0012 ELSE
0013   LED0:=FALSE;
0014 END_IF;
0015
0016 IF TIME_Q THEN
0017   TIME_IN := FALSE;
0018 END_IF;
0019

```

As shown in the following figure:

Local variables							
	Name	Type	Address	Array	Init value	Attribute	Description
1	TIME2S_TON	TON	Auto	No			
2	TIME_IN	BOOL	Auto	No			
3	TIME_2S	TIME	Auto	No	T#2S		
4	TIME_Q	BOOL	Auto	No			
5	TIME_ET	TIME	Auto	No			

```

0001 IF NOT TIME_Q THEN
0002   TIME_IN:=TRUE;
0003 END_IF;
0004
0005 TIME2S_TON(IN := TIME_IN(* BOOL *),
0006            PT := TIME_2S(* TIME *),
0007            Q => TIME_Q(* BOOL *),
0008            ET => TIME_ET(* TIME *));
0009
0010 IF TIME2S_TON.ET < T#1S THEN
0011   LED0:=TRUE;
0012 ELSE
0013   LED0:=FALSE;
0014 END_IF;
0015
0016 IF TIME_Q THEN
0017   TIME_IN := FALSE;
0018 END_IF;
0019

```

At this point, click on the connection button to successfully connect, and the connection status will be clearly visible in the status bar. If the connection is still not successful, check if the network cable is plugged in properly.

EDIT MODE	SOURCE OK	CONNECTED
-----------	-----------	-----------

2.1.6.7 Program Download

The procedure for downloading a program from the

BonengLogicLab programming software is as follows:
In the shortcut menu bar, just click on the button shown below

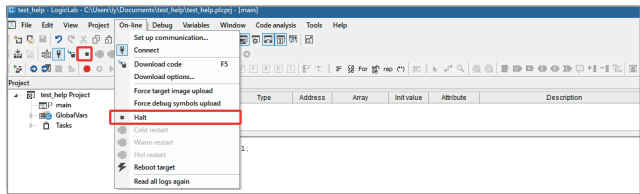


2.1.6.8 Display result run

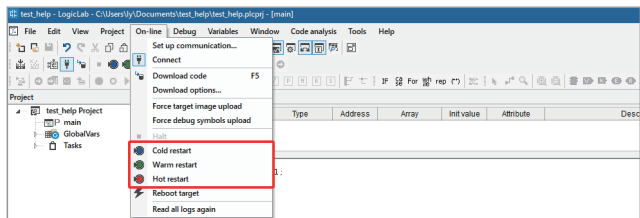
When the download is successful the PLC will run up automatically and the program will run successfully when you hear it ringing every 1s interval.

2.1.7 Switching the PLC operating mode

The PLC has two operating modes, run and stop.
Stopping the PLC can only be done through the BonengLogicLab programming software by clicking on button in the shortcut menu bar or by clicking on the online click stop as shown in the figure below:



There are three modes of running the PLC program, click button in the shortcut menu, in order of color, cold start, warm start, hot start, or click "online" in the menu bar and then select the mode that the user wants to start. As shown in the figure below:



Differences between the three types of startup.
Cold restart: The PLC application will be restarted and both holdable and non-holdable variables will be reset;

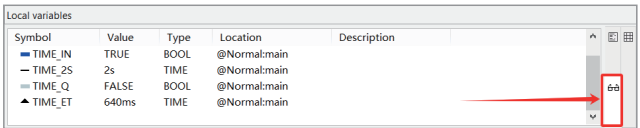
Warm restart: The PLC application will be restarted and the non-power-down held variables will be reset;
Hot restart: The PLC application will be restarted without resetting any type of variable;

2.1.8 Online monitoring and debugging

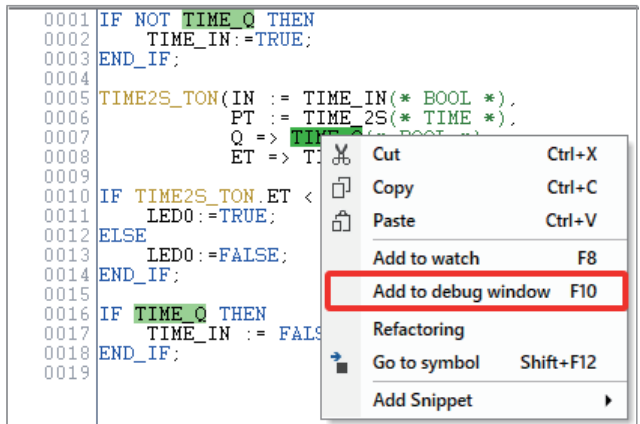
2.1.8.1 Online monitoring

BonengLogicLab supports the online modification of the real-time values of variables, which makes it easier for the user to debug faster. Online debugging is divided into variable list monitoring (for quick and direct monitoring of the values of variables in this variable list), and separate monitoring windows (for consolidating and monitoring variables in one place). And monitoring variable values directly in the program.

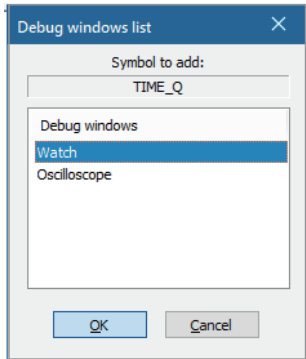
- Monitor the value of a variable directly in the variable list. Select a list of variables and click on to view them, as shown below:



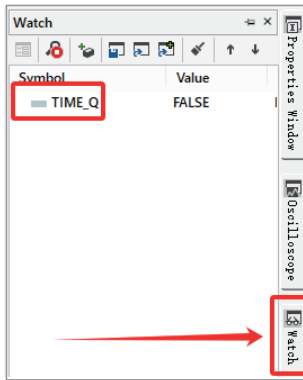
- Use a separate monitoring window
Right-click on the TIME_Q variable in the selection and click Add to Monitor Window.



Then just click on Select Monitor Window.

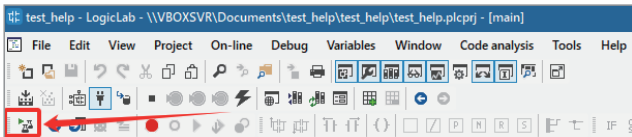


Put the mouse on the right monitoring window, the monitoring serial port will pop up to monitor the variables you just added.



- Monitoring directly in the program (also a required step for online debugging)

Click the button shown below in the shortcut menu bar (online debugging mode) to enter the online debugging page.



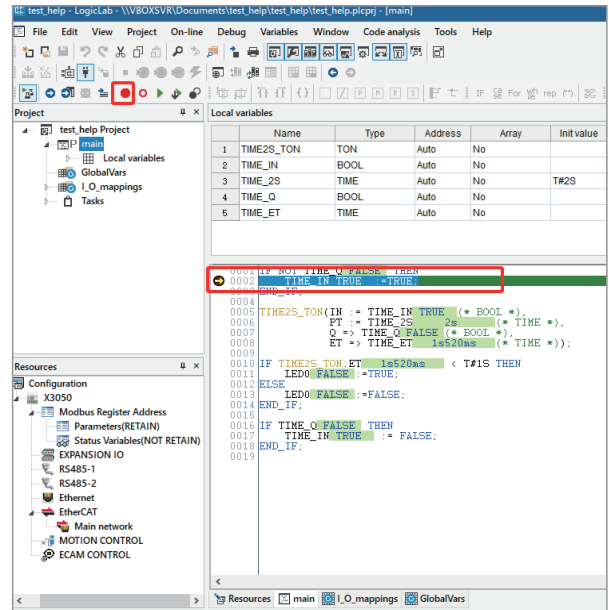
Once clicked, the program edit page will no longer run to be edited, and you can directly view the values of the variables contained in this program page, as shown below:

```

0001
0002 (*cnt := cnt + 1;*)
0003
0004 IF NOT TIME_Q FALSE THEN
0005 TIME_IN TRUE := TRUE;
0006 END_IF;
0007
0008 TIME2S_ON(IN := TIME_IN TRUE (* BOOL *),
0009 PT := TIME_2S 2s (* TIME *),
0010 Q => TIME_Q FALSE (* BOOL *),
0011 ET => TIME_ET 1s940ms (* TIME *));
0012
0013 IF TIME2S_ON.ET 1s940ms < T#1s THEN
0014 LED0 FALSE := TRUE;
0015 ELSE
0016 LED0 FALSE := FALSE;
0017 END_IF;
0018
0019 IF TIME_Q FALSE THEN
0020 TIME_IN TRUE := FALSE;
0021 END_IF;

```

2.1.8.2 Online debugging



Click online debugging button , select a line of program in the program editing area, click ● in the shortcut menu bar (or click F12), you can add a breakpoint for the line of the program, if the program conditions are met, when the program runs up to this point, it will stop, and a yellow indicator arrow will appear to indicate that it runs up to this point. In the shortcut menu bar ● ○ ► ▼ ◻ these buttons are, from left to right, Add or Remove a Breakpoint, Remove All Breakpoints, Run All, Run in Single Step, and Breakpoint List.

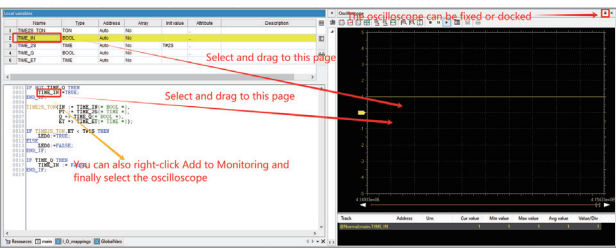
2.1.9 Oscilloscope applications

2.1.9.1 Introduction to Oscilloscope Functions

The oscilloscope analyzes the maximum value of the variable. Minimum values, differences between signals are monitored, etc. Oscilloscopes are often needed to analyze the relationship between multiple data when the data is complex. Graphically view changes in the value of a variable to help users quickly analyze the operation of a program.

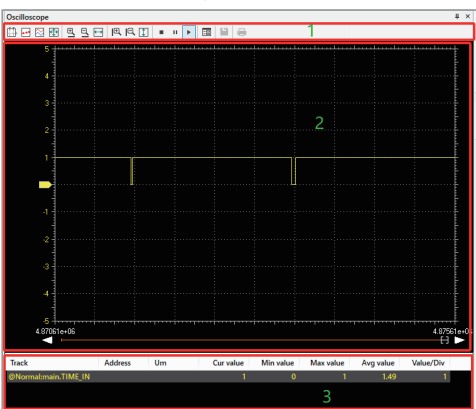
2.1.9.2 Adding Variables to the Oscilloscope

There are two ways to add variables to the oscilloscope, the first is to directly drag and drop the variables in the variable window area, or the variables in the program serial port, and then drag and drop them to the oscilloscope window. The second is the right-click form, right-click to add to the monitor window, and then select the oscilloscope.



2.1.9.3 Introduction to Oscilloscope Functions

The oscilloscope page is mainly divided into three areas 1, the oscilloscope control and debugging area. 2, the main waveform display area 3, the monitored variable list storage area. As shown in the figure below:



1. The Oscilloscope Control and Debug area contains the following commonly used debug buttons, as shown in the table below:

icon	Icon name	functionality
	Display Measurement Column	When turned on, two horizontal cursors (one blue, and one red), and two vertical cursors (one blue, and one red) appear. They are used to measure the horizontal and vertical values respectively. When turned on, the mouse can be used to adjust the cursor position in the display area of the oscilloscope to obtain the current horizontal or vertical height value.
	Show Sample	Display the sample points sampled by the oscilloscope.
	vertical segmentation	Used when the page has multiple variables sampled at the same time, you can split them with one click for easier observation
	Show all values	Display all sampled horizontal and vertical values on the current page.
	Horizontal magnification	You can zoom in horizontally when a variable is selected.
	Horizontal reduction	You can shrink a variable horizontally by selecting it.
	Horizontal display of all values	Selecting a variable displays all of its horizontal values.
	Vertical magnification	Selecting a variable allows you to zoom in vertically.
	Vertical zoom	Selecting a variable allows you to zoom in vertically.
	Vertical display of all values	Selecting a variable displays all of its vertical values.
	Stop sampling	Stop sampling, restarting after stopping will lose buffer data

2.1.11 EtherCAT Task Cycle Setting

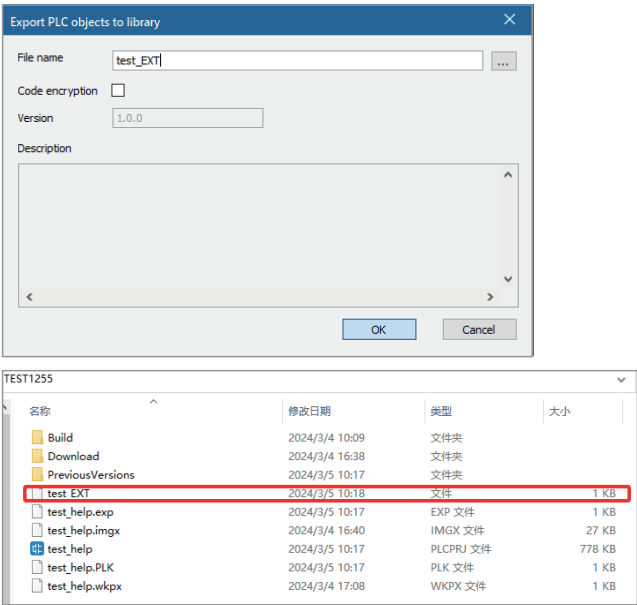
The EtherCAT task program runs in the background and is bound to the Fast task. When the cycle value of the Fast task is changed, the EtherCAT cycle is changed as well.

2.1.12 Importing and exporting objects

In BonengLogiclab, you can export the POU in the project window as a separate file, and when you use it in other projects, you can directly import this POU. the operation is described below:

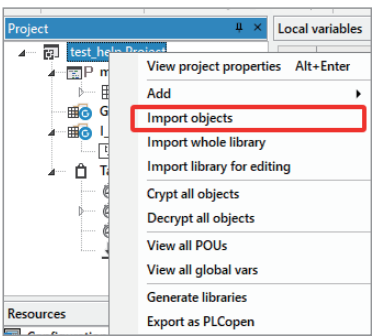
2.1.12.1 Exporting the object generation library

Select a POU and right-click, select Export to Library, enter a file name, select whether to encrypt this library or not, and then just click OK, and a file with that name will be automatically generated in that project directory.



2.1.12.2 Importing objects

Right-click on the name of the project, click on the import object, find the file that the user needs to import, click on it, selectively import the object that the user needs. Convenient to quickly port the program.

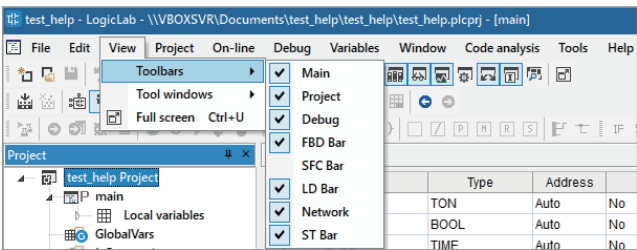


2.2 Programming Basics

2.2.1 General

BonengLogiclab's variable assignment is relatively flexible, and can be divided into two categories: global variables and local variables. In the Quick Start, we have already introduced the parameterization of the variables, so we will not repeat the explanation here. Figuring out the types of variables will get you up to speed with the software quickly. Simply put, on the left side, the variables in the project window, are all global variables; of course there are a few global variables in the library.

In the process, did not find some windows, you can click on the menu bar in the view, the toolbar, tool windows, check the window they want to open. As shown in the figure below:



2.2.2 Variables

2.2.2.1 Creating simple variables

There are many ways to create a variable in BonengLogiclab, as mentioned in the Quick Start, you can add a variable by right-clicking in a table editing format, or you can switch to a text format for text ed-

iting. The rest of the quick ways to create a variable are described below.

In the ST program, you can output the name of the new variable that the user wants to create directly in the text editor area, as shown in the following figure:

```
0001 cnt := cnt + 1;
0002
0003
0004 Test_Create
0005
0006
0007
```

Then enter to end the input, the following dialog box will pop up:

New variable

ScopeLocal

AddressAuto

NameTest_Creat

TypeBOOL

Class / Group

ArrayNo

AttributeAUTO

Init values

Description

OKCancel

In this page, you can set the scope of the variable, the address of the variable, the address of the variable is the default Auto without any special usage, and you can also change the name of the variable and the type of the variable. In the edit mode of ST, you can select function blocks and function types here, but when this dialog box appears in LD, you can only select the base variable type. If you are selecting a global variable in this place, you need to assign a group to the variable. Place the global variable in this group.

2.2.2.2 Object Browser

When creating a variable, it is often necessary to modify its variable type, which is divided into, IEC basic type, and user type. Or assign a function type, function type. At this point, the name of the type is too large to be entered correctly, so you need a search engine to help you find the type of variable you want. In the table of variables, click on the type of small white blocks, as shown in the following figure:

	Name	Type	Address	Array	Init value	Attribute	Description
1	MCNAME	BOOL	Auto	No			

When you click on it, the Object Browser will appear, as shown below:

Object browser

☐ Programs

☐ Function Blocks

☐ Functions

☐ Variables

☐ User types

☐ Operators

☐ Standard functions

☐ Local variables

☒ Basic types

Check all

Check none

Name*

OK

LocationAll

LibraryAll

Vars typeAll

Vars group

Name

Type

BOOL

Basic types

BYTE

Basic types

DATE

Basic types

DATE_AND_TIME

Basic types

DINT

Basic types

DWORD

Basic types

INT

Basic types

LDATE

Basic types

LDATE_AND_TIME

Basic types

LINT

Basic types

LREAL

Basic types

LTIME

Basic types

LTIME_OF_DAY

Basic types

LWORD

Basic types

PROVID

Basic types

REAL

Basic types

SINT

Basic types

STRING

Basic types

TIME

Basic types

In this Object Browser, it can be divided into 3 parts, which contain two filters (left side) and one filter result box (right side). Usually you can use the above filter, the other filters are used when the project is already very large and there are still many results after passing the above object filter. And the use of other filters, will be affected by the above filter, when the above object filter is not checked, the following other filters can not filter out the results.

In other filters, the input rule for the Name field is to use * as a wildcard and is not case sensitive. This is shown in the figure below:

Object browser

☐ Programs

☒ Function Blocks

☐ Functions

☐ Variables

☒ User types

☐ Operators

☐ Standard functions

☐ Local variables

☒ Basic types

Check all

Check none

Name*mC_m*

OK

LocationAll

LibraryAll

Vars typeAll

Vars group

Name

Type

MC_MoveAbsolute

Function blocks

MC_MoveAdditive

Function blocks

MC_MoveContinuousAbsolute

Function blocks

MC_MoveContinuousRelative

Function blocks

MC_MoveRelative

Function blocks

MC_MoveSuperimposed

Function blocks

MC_MoveVelocity

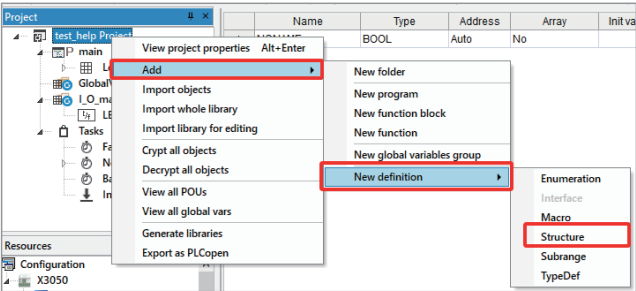
Function blocks

2.2.2.3 Creating complex variable types

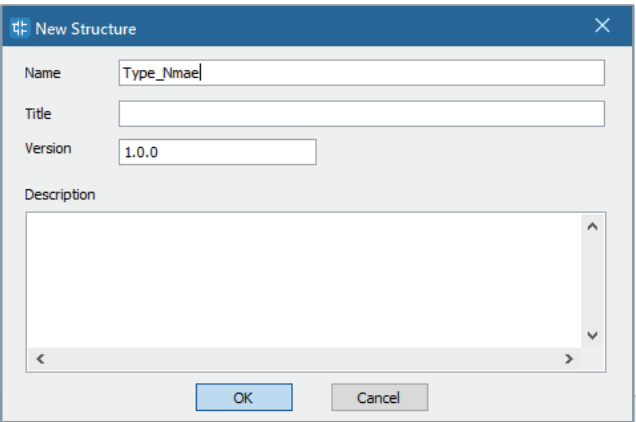
In BonengLogiclab you can create complex variable types in the project window for your own convenience. For example, structures, enumerations, etc.

2.2.2.3.1 Creation of structure types

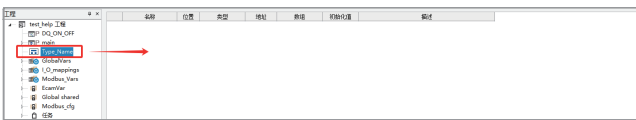
In the Project window, right-click the project name, Add, click New Definition, and click Structures. As shown in the following figure:



After clicking on the structure, such as will pop up the window shown below:

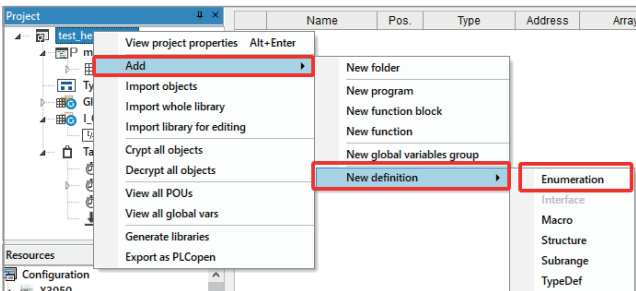


Enter the name of the structure type and click OK. The title and description are used to explain the customized structure type. Just leave them blank, and the version number is to better distinguish the version when importing or exporting. After creating the structure type, the next step is to create the members of the structure type. In the project window, click on the structure type you just created. In the main window on the right, right-click to add the members of the structure type.

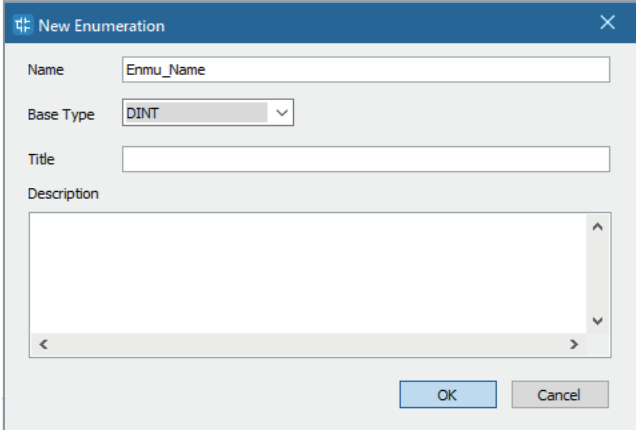


2.2.2.3.2 Creation of enumerated types

In the project window, right-click on the project name, add, click on new definition, and click on enumeration. As shown in the following figure:



After clicking, as shown in the following figure, users can select the basic types of enumeration they need. The title and description can still be left blank.

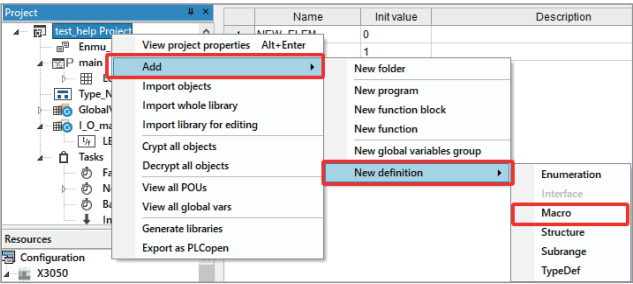


By clicking on the added enumeration type, you can insert the members of the enumeration by right-clicking on them in the main window on the right. The members of the enumeration can have their names changed, as well as their initial values. This is shown in the following figure:

	Name	Init value	Description
1	NEW_ELEM	0	
2	NEW_ELEM1	1	

2.2.2.3.3 Macro Creation

In the project window, right-click on the project name, add, click on new definition, and click on macro. As shown in the following figure:

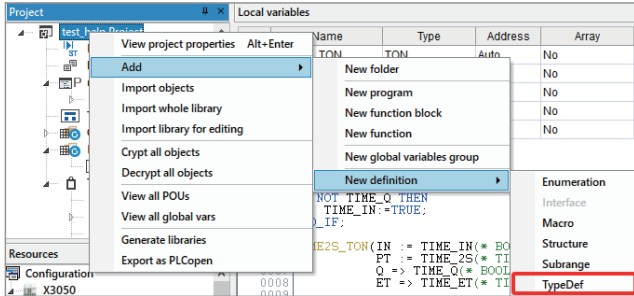


After clicking on the macro, the following dialog box will pop up. In the name, output the name of the macro, select the language, and only support the use of macros in ST and IL when using macros. Choose the commonly used ST here.

After clicking OK, an editing window will pop up where you can edit the program, and then use this macro in ST to replace the large program.

2.2.2.3.4 Type Definition Creation

In the project window, right-click on the project name, add, click on new definition, and click on type definition. As shown in the following figure:

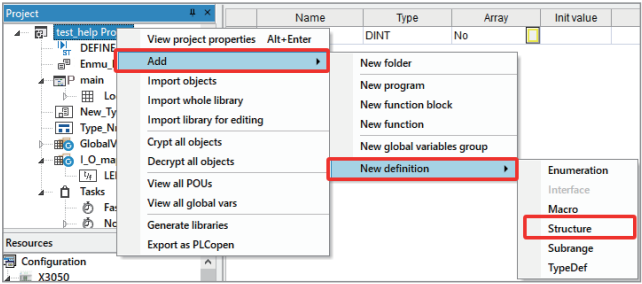


After clicking, as shown in the following figure:

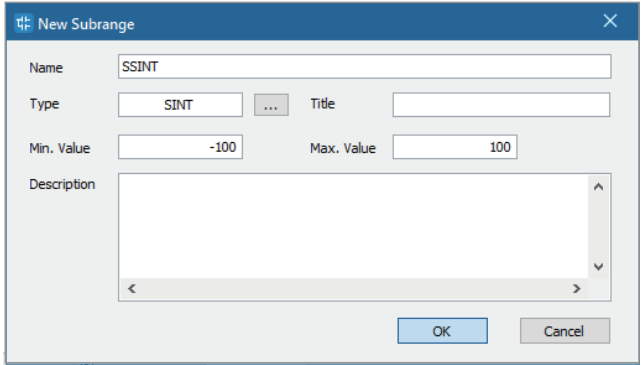
In the new type definition, it is necessary to set the name, type, initial value, array, etc. of the new type. In summary, the new type definition is essentially renaming the variable types that come with software, including structures, functional blocks, functions, and other types, and then allowing users to call this custom variable type name to create a new variable name.

2.2.2.3.5 Subscope Types

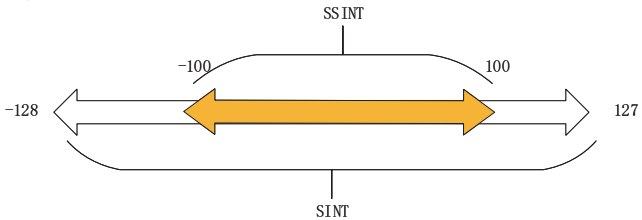
In the project window, right-click on the project name, add, click on new definition, and then click on sub scope. As shown in the following figure:



After clicking, the following dialog box appears, where you need to fill in the name of the sub range, select or enter the type, and enter the maximum and minimum values. As shown in the following figure:

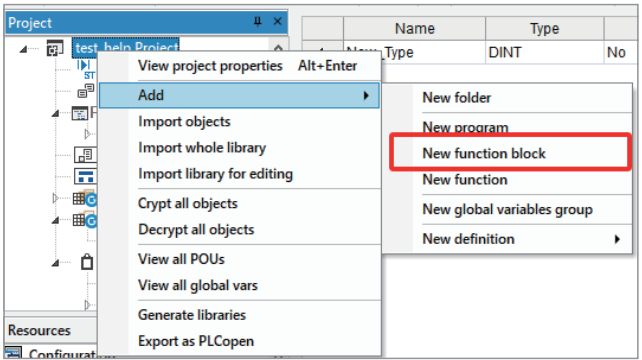


According to the above settings, the schematic diagram of the sub range is shown in the following figure:

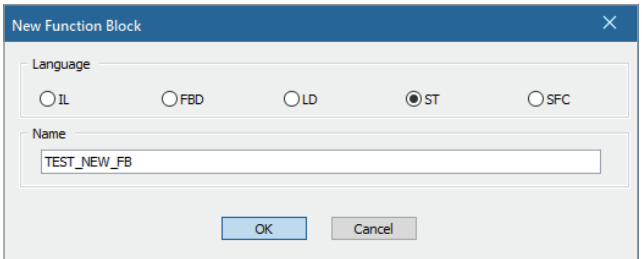


2.2.2.4 Creating Function Blocks

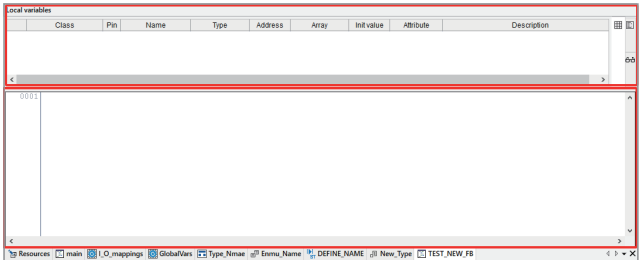
Right click on the project name, click on "Add", and then click on the "New" function block, as shown in the following figure:



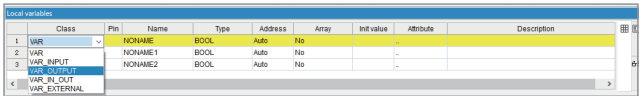
After clicking, the following dialog box will pop up, where you need to select the language used to create the function block and the name of the function block.



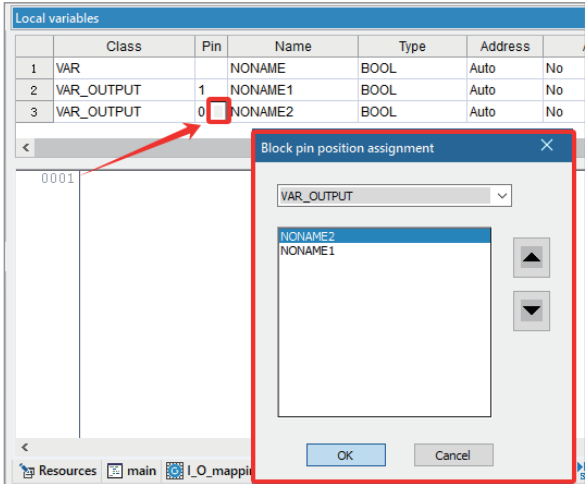
Clicking on the created function block above will bring up a window for editing the function block,



This window is similar to the program editing window, but for local variable editing in the function block, the variable needs to be classified into ordinary internal variables, input variables, output variables, input-output variables, and external variables. As shown in the following figure,



You can also rearrange the order of the pins according to your own habits. The purpose of sorting is to change the order of the pins in the function block. Clicking on the white square of the pin will pop up the module pin position allocation dialog box, as shown in the following figure:



In this dialog box, you can select the pin types that need to be sorted, and click the up and down buttons with the mouse to sort them.

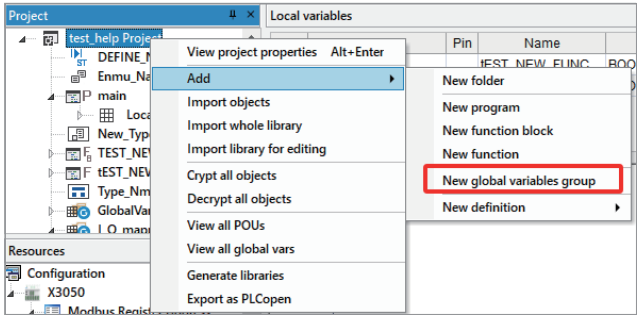
2.2.2.5 Creating Functions

Creating a function is similar to creating a function block, but due to the nature of the function itself, only one output pin is allowed. Therefore, when adding, the output pin has already been specified as 'Result' in the local variables. And in the selection of classes, it is not possible to choose output pins.

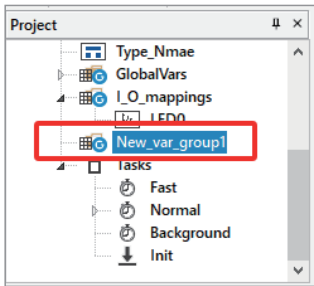
Class	Pin	Name	Type	Array	Init value	Attribute	Description
1	RESULT	TEST_NEW_FUNC	BOOL	No			Function result
2	VAR	NONAME	BOOL	No			
	VAR	VAR_INPUT					
	VAR	VAR_EXTERNAL					
	VAR	VAR_OUT					

2.2.2.6 Creating Global Variable Groups

The purpose of creating a global variable group is to better manage your own global variables. Right click on the project, click on "Add", and then click on "New Global Variable Group", as shown in the following figure:



After adding it, you can implement your own variable grouping management, as shown in the following figure:



2.2.2.7 Pointer Variable Types

Pointer type variables are relatively easy to use, but at the same time, they are also quite dangerous operations. So it is not recommended to use pointer type variables.

Pointer type variables need to insert an A symbol before the type to indicate that the variable is a pointer variable.

Name	Type	Address	Array	Init value	Attribute	Description
1	cnt	@CNT	Auto	No		
2	cnt1	CNT	Auto	No		

Address taking operations and pointer variables usually occur in pairs. To get an address in ST, you can use "?" to get the address of a variable, for example: cnt:=? cnt1; Add before the variable? Indicate the address of the variable to be retrieved, or use the instruction 'ADR()' in the library to perform the address retrieval operation.

2.2.3 Resources - Parameters (Power Down Maintenance)

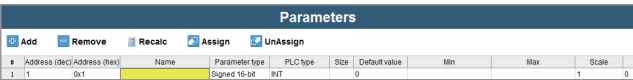
In the resource window, expand the common objects and you can see the node for the parameter (power down hold). The page for this parameter is mainly used to design the register address when the controller acts as a "slave" in Modbus communication (TCP/RTU).

2.2.3.1 Adding a Register Address

As shown in the following figure: Click Add to add a register address



After adding, you need to enter a name for the register address, and the name will automatically change to the address

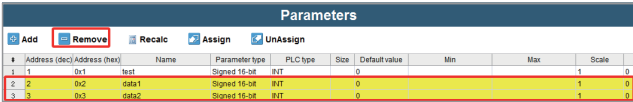


Address (dec)	Display register addresses in decimal format
Address (hex)	Display register addresses in hexadecimal format
name	The variable name can be entered by oneself or assigned, and then automatically bound to that address.
Type	The parameter types for communication can be selected
PLC type	The parameter types displayed in PLC can be selected
Default value	Default initial value
maximum	The maximum value that can be received, communication failure beyond the range
minimum	The minimum value that can be received, communication failure beyond the range

proportion	The amplification or reduction ratio of the received data (only for the PLC end, unchanged for the main station end) is set to 1 by default without special requirements.
shifting	The size of the offset of the received data (only for the PLC end, unchanged for the main station end) is set to 0 by default without special requirements.
describe	Add a description to this variable

2.2.3.2 Deleting a Register Address

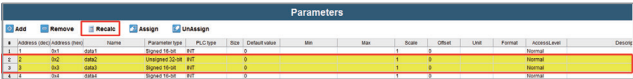
Select the row to be deleted and click the delete button, as shown in the following figure:



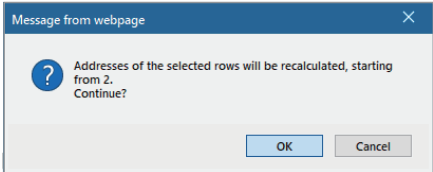
Hold down the Shift key or Ctrl key to select multiple items, and the selected objects will turn yellow.

2.2.3.3 Recalculation

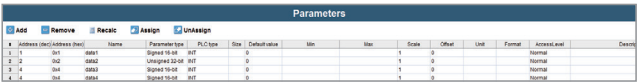
The function of recalculation is to recalculate the PLC address when the data type appears as 32 bits, as the communication data is transmitted according to 16 bits. If the address at this time is only incremented by 1, problems will occur, and the PLC address needs to be recalculated. The operation method for recalculation is shown in the following figure:



Select the 32-bit data in the middle and the data in the bottom row, and click on recalculate. After clicking, as shown in the following figure:

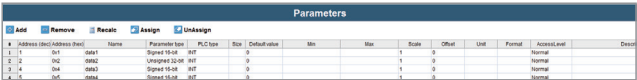


click ok



#	Address (hex)	Address (dec)	Name	Parameter type	PLC type	Size	Default value	Min	Max	Scale	Offset	Unit	Format	Access level	Delete
1	1	1	data1	Signed 16-bit	INT	2	0			1	0		Normal		
2	2	2	data2	Unsigned 32-bit	UINT	4	0			1	0		Normal		
3	4	4	data3	Signed 16-bit	INT	2	0			1	0		Normal		
4	4	4	data4	Signed 16-bit	INT	2	0			1	0		Normal		

It can be seen that the address of data3 no longer starts from 3, but from 4. However, it was found that the address of data4 is already 4. Therefore, when recalculating, it is usually necessary to select all addresses and click on "recalculate" to ensure the uniqueness and rationality of each variable's address. After selecting all, recalculate as shown in the following figure:

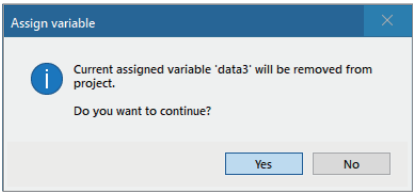


#	Address (hex)	Address (dec)	Name	Parameter type	PLC type	Size	Default value	Min	Max	Scale	Offset	Unit	Format	Access level	Delete
1	1	1	data1	Signed 16-bit	INT	2	0			1	0		Normal		
2	2	2	data2	Unsigned 32-bit	UINT	4	0			1	0		Normal		
3	4	4	data3	Signed 16-bit	INT	2	0			1	0		Normal		
4	8	8	data4	Signed 16-bit	INT	2	0			1	0		Normal		

Another situation that requires recalculation is when multiple variables use the same address (or the address order is out of order) during copy and paste, which is not allowed. Therefore, recalculation can be performed to facilitate the reassignment of addresses.

2.2.3.4 Allocation

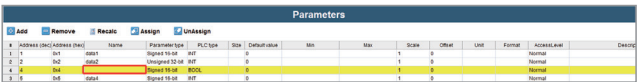
Select an address (line), click Assign, and the software will prompt to remove the original variable and delete it from the project. Do you want to continue? As shown in the following figure:



Click Yes, select the variables (only global variables), and then you can proceed.

2.2.3.5 Cancel Allocation

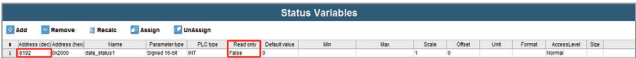
Select an address (line) and click 'Unallocate' to remove the variable from the parameter group in the project



#	Address (hex)	Address (dec)	Name	Parameter type	PLC type	Size	Default value	Min	Max	Scale	Offset	Unit	Format	Access level	Delete
1	1	1	data1	Signed 16-bit	INT	2	0			1	0		Normal		
2	2	2	data2	Unsigned 32-bit	UINT	4	0			1	0		Normal		
3	4	4	data3	Signed 16-bit	INT	2	0			1	0		Normal		
4	8	8	data4	Signed 16-bit	INT	2	0			1	0		Normal		

2.2.4 Resources - State Variables (Power Down Not Maintained)

The state variables and parameters are basically the same and are also used for Modbus communication. However, the register address of the state variable starts from 8192 and is not maintained during power failure. It is worth noting that if no special modifications are made, after adding an address. This address is read-only and cannot be modified. As shown in the following figure:



#	Address (hex)	Address (dec)	Name	Parameter type	PLC type	Size	Default value	Min	Max	Scale	Offset	Unit	Format	Access level	Delete
1	8192	8192	data1	Signed 16-bit	INT	2	0			1	0		Normal		

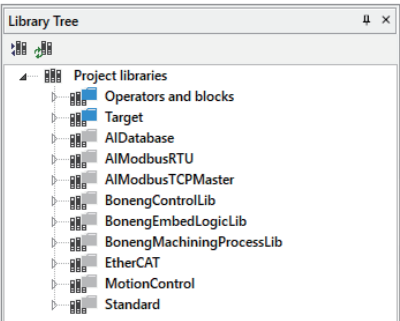
If data needs to be written, then it is necessary to change read-only to false.

2.2.5 Library Management Wizard

This only introduces the functions of the library management wizard, including importing libraries, exporting libraries, and other operations, and will not explain each functional block and function. Please refer to Chapter 3 for detailed instructions on functional blocks. In BonengLogiclab, library files play a crucial role in the entire project. Manage your library files well and learn to call library files to write your own programs. A library file is a file that ends with '.plclib'.

2.2.5.1 Introduction to Library Files

BonengLogiclab provides many libraries, which can be seen when creating a new project. Taking X3050 project as an example, as shown in the following figure:

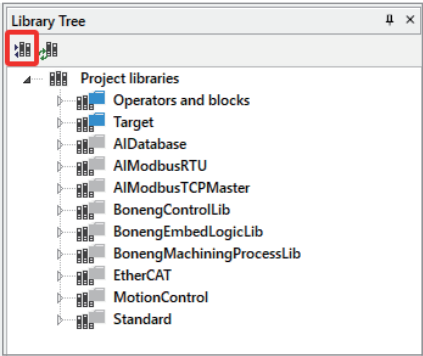


According to the above figure, the table below will provide a brief explanation of the figure:

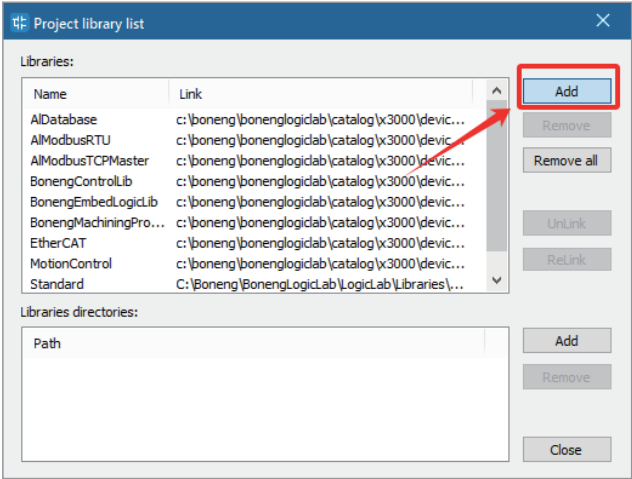
Library Name	Illustrate
Basic instructions	Contains some basic instructions such as conversion, arithmetic, logic, comparison, etc
controller	Contains some system variables, system clock, etc.
AI Database	Contains storage for data regions
AI Modbus RTU	Contains some functional blocks and functions related to Modbus RTU, which are usually completed through page configuration rather than using this library
AI Modbus TCPMaster	Contains some functional blocks and functions related to Modbus TCP, which are usually completed using page configuration instead of this library
Bon eng - Controllib	A library that includes simple control of servos and frequency converters
BonengEm-bedLogiclib	Contains some embedded libraries that require special usage
Bon eng - Machining-ProcessLib	A library that includes functions related to unwinding and unwinding rollers
EtherCAT	Contains a functional library related to ECAT communication, which is generally not used
Mot i o n - Control	Contains complex libraries related to motion control
Standard	A library that includes functions such as timer, set, reset, edge trigger, etc

2.2.5.2 Importing Library Files

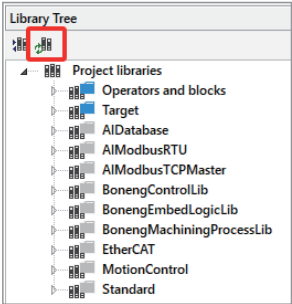
Click on the Library Management Wizard window page and click on the Library Manager button, as shown in the following figure:



After clicking, a window as shown in the following figure will pop up:

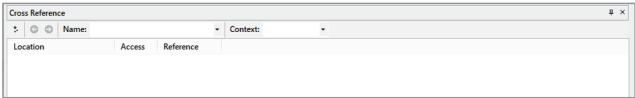


Click Add, find the file ending with ". plcLib", and click OK to proceed. If the added files are still not displayed in the library, you can try clicking to refresh the library files as shown in the following figure:



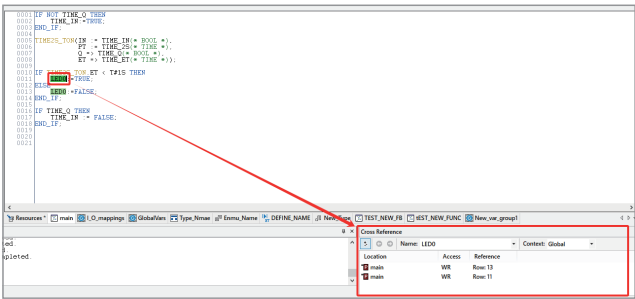
2.2.6 Cross referencing

The function of cross referencing is to quickly locate the location and usage of a global variable throughout the entire project. Cross referencing can only retrieve programs that have been added to the task. If the program has not been added to the task, it cannot be retrieved. The cross reference window is shown in the following figure:



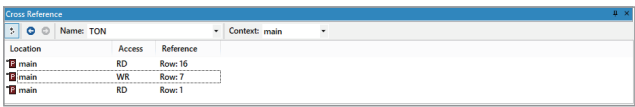
2.2.6.1 General steps for cross referencing

In cross referencing, it is necessary to click on button to ensure automatic display of the object. After clicking, simply use the mouse to click on the variable in the program, and the usage of the variable will appear in the cross reference window. Clicking on the results retrieved from the cross reference will automatically jump to the row where the variable is located. The LD language is also applicable.



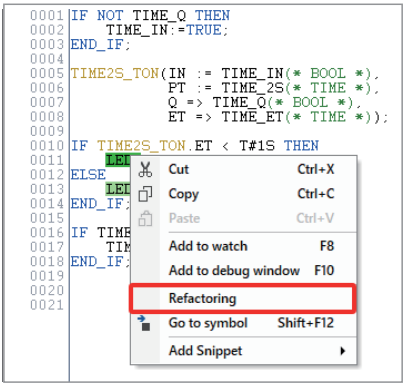
2.2.6.2 Directly select or input variable names

In cross referencing, you can also enter or select the name of the variable directly in the name column. After entering it, the search results will also be displayed directly below. You can also retrieve the usage of a certain functional block, such as retrieving the usage of TON timers throughout the entire project, as shown in the following figure:

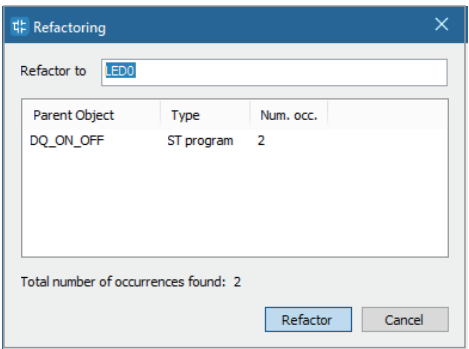


2.2.7 Refactoring

The function of refactoring is to rename a variable and modify its name throughout the entire project. The name of the variable used will also be changed. The specific operation method is as follows: In the program, select a variable, right-click with the mouse, and select Refactor: as shown in the following figure:



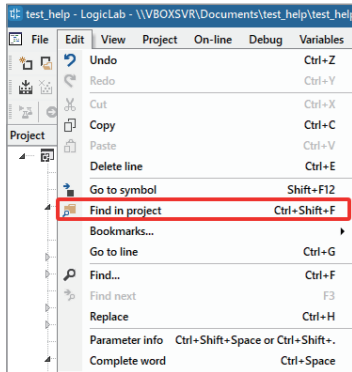
After clicking on refactoring, the following window will pop up:



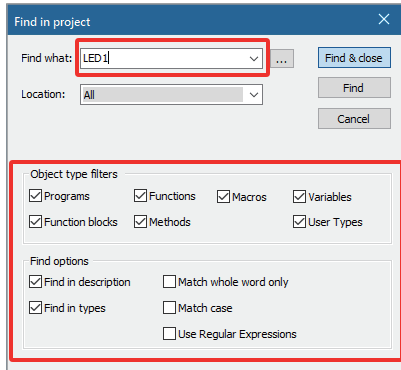
In this window, the places where this variable is used will be listed. You also need to enter the name of the refactoring here, click on refactoring, and it will be completed.

2.2.8 Search

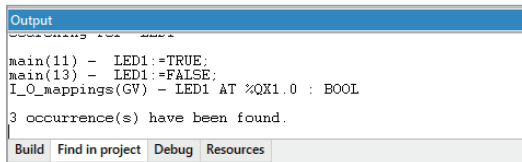
BonengLogicl原因 provides tools for searching variables, similar to cross referencing, which can be divided into local search and global search within the project. Click "Edit" in the menu bar (you need to go to the program editing page), and then click "Find" in the project as shown in the following figure:



After clicking, a filter window will pop up, as shown below:



Enter the missing content, then configure the filter below, click search, and the search result is shown in the following figure:



Click on the search results to directly jump to the location.

2.2.9 Simulation

For BonengLogicl原因 simulations, you can only do some simple logical simulations, if in the program, other resources are configured. For example, Modbus, EtherCAT, and the use of some "Boneng" related libraries and other resources will cause the compilation to fail. Simulation is not available.

2.2.9.1 Simulating structures

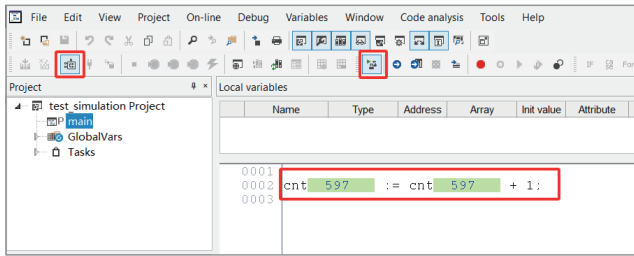
In the shortcut menu bar  of the BonengLogicl原因 software, click the button, and after that, the software will pop up the following display window.



There are 4 buttons in this window, and the specific functions of the buttons are shown in the following table

icon	function
	Stop the PLC program from running in simulation mode
	Start the PLC program to run in simulation mode
	Enable all I/O control panels
	Expand/shrink the IO simulation window

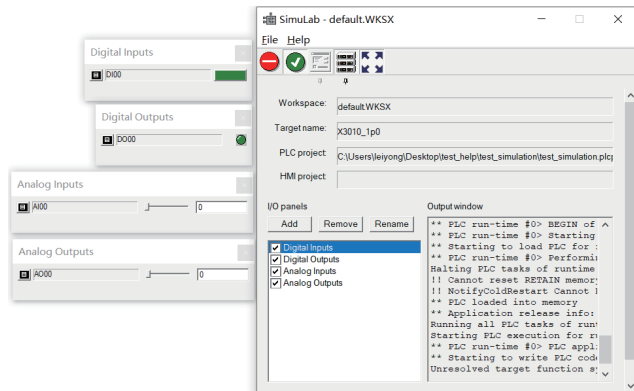
After clicking the simulation button, you need to compile and download the PLC program, and when the project source code matches in the lower right corner, you can start using BonengLogicl原因 normally. The software monitors the values online. This is shown in the figure below:



If you encounter a failure to compile the PLC program and an error is reported, you need to download the latest version of the software from the official website before you can use it.

2.2.9.2 Extended IO simulation

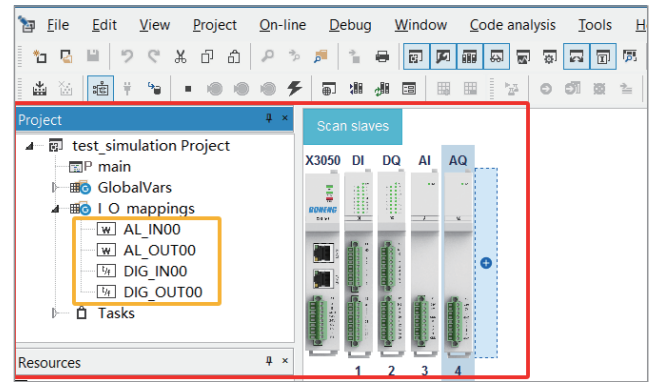
Click to expand the I/O simulation window and enable the I/O control panel, as shown in the following figure:



In this simulation page, 4 IO control panels have been preset, and the default four IO control panels are displayed, if you don't want to display a control panel, you can uncheck it. DI00, DO00, AI00, AO00 are just a display example added, which has no practical meaning in the user's project, and can be deleted directly. You can also create your own control panel in this window.

- Design test procedures

In PLC engineering, four expansion modules need to be added, and variables are created in the extension modules. This is shown in the figure below:



Using the extended IO variable created above, add the following program in the main() task:

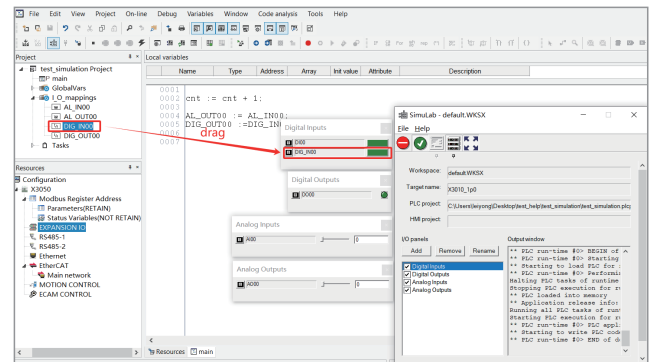
```

0001 cnt := cnt + 1;
0002
0003
0004 AL_OUT00 := AL_IN00;
0005 DIG_OUT00 := DIG_IN00;

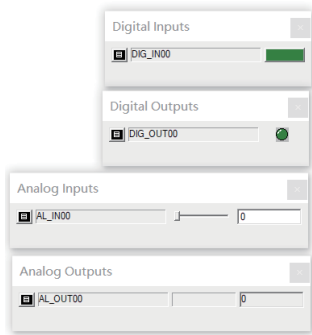
```

This test procedure makes it easy to review the results of the experiment later.

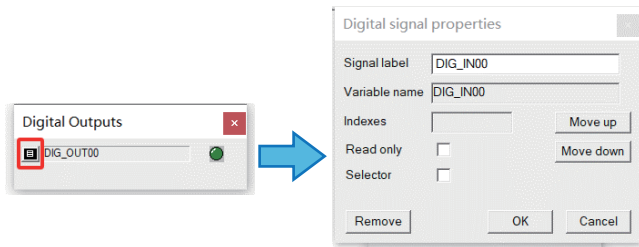
- Add extended I/O variables to the I/O Control Panel
- After compiling, saving, and downloading, drag and drop the extended I/O variables in the project to the I/O simulation control panel



And delete the original initial built-in point control DI00, DO00, AI00, AO00. The specific deletion operations are shown below:



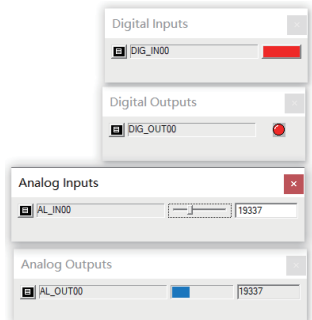
- Adjust the settings of the extended IO panel



In this control panel, you can set the point of the control panel to be read-only, delete the point in the panel, or move the point up and down in the panel. The variable name and tag name can be the same, and generally do not change it. Here, you need to set the DIG_OUT00 and AL_OUT00 to read-only in the point configuration properties. It is convenient to observe the results of the demonstration.

- Final Operational Demo

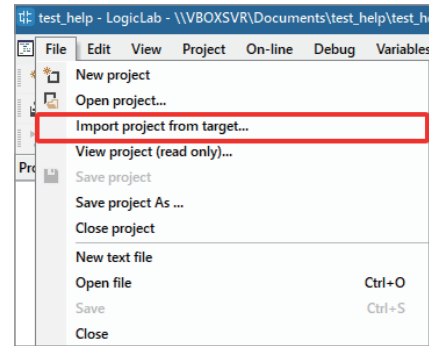
When you click the green square button, the light at the output point will also turn red, and when you slide the slider of the analog input, the size of the analog output will also change.



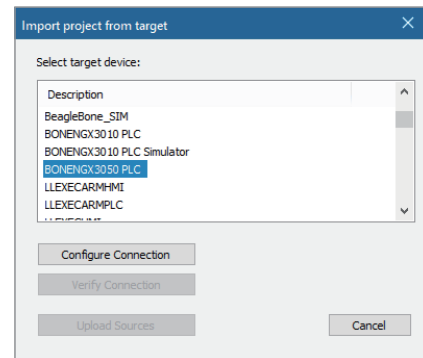
387

2.2.10 Upload

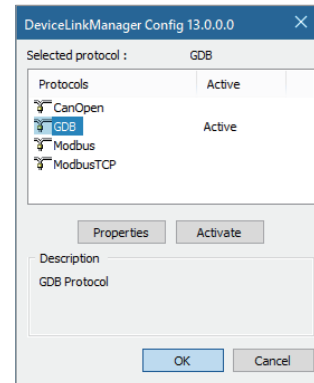
BonengLogiclab supports uploading programs to a computer. Click on "File" in the menu bar, and then click on "Import Project from Device". The operation is shown in the following figure:



After clicking, find the BONENGX3050 PLC and select it.

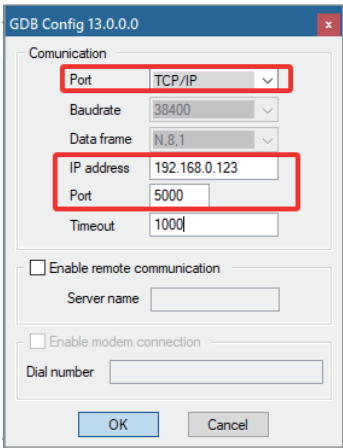


Click on the connection configuration and the following window will pop up.

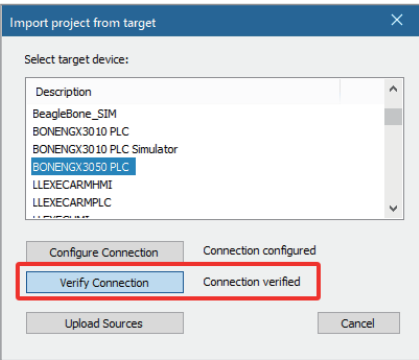


388

Select GDB, click Activate, and then click Properties.



Select TCP/IP protocol, enter the IP address of the controller, port 5000, timeout 1000, and then click OK to close the previously opened windows. Return to the window shown in the following image. Click to verify the connection, and a prompt will pop up indicating that the verification is valid.



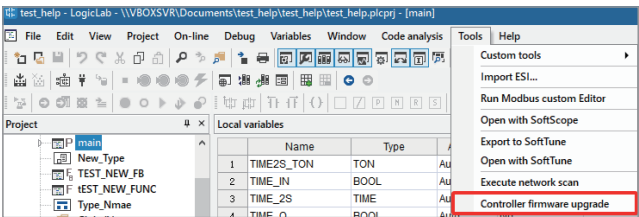
After verifying the validity, click on upload source code and a window will pop up. You need to select a folder to store the source code. After selecting the folder, click OK, and the project will automatically close and open the following projects.

2.2.11 Firmware Upgrade

This firmware upgrade method is only available in BonengLogiclab Version: V0.1.4. You can click on 'About' on the BonengLogiclab help page to view the current software version.

2.2.11.1 Open the upgrade tool

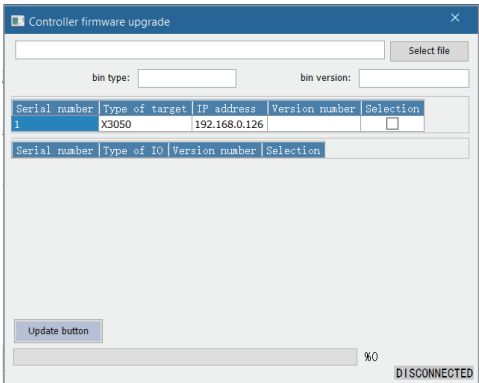
When using a PLC, there may be situations where the functionality does not meet the requirements and firmware upgrades are needed. Taking X3050 firmware upgrade as an example, the operation steps are as follows:



Click on "Tools" in the menu bar and select "Controller Firmware Upgrade".

2.2.11.2 Enter the upgrade page

After entering the upgrade page, the page will automatically scan the device status of the controller actually connected to the computer network cable, as shown in the following figure:

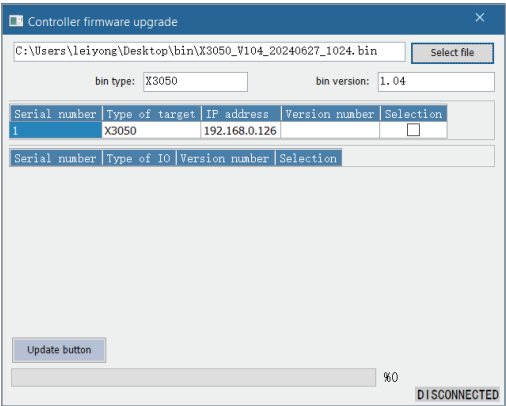


Display the actual connection control type and IP address information. If multiple controllers are actually connected through a switch, they will all be

displayed in the table. If the network cable is unplugged or unplugged midway, the page needs to be closed again and accessed again to display the latest controller device information

2.2.11.3 Select upgrade file

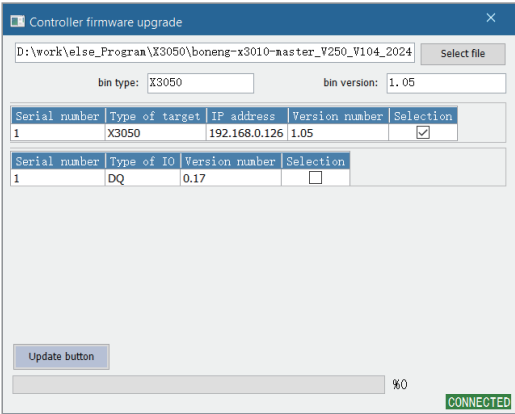
Click to select the file, and a file dialog box will pop up. Select the controller burning file published by Boneng, and after selecting the burning file, it will be as shown in the following figure:



After selecting the firmware that needs to be upgraded, the BIN file will be automatically parsed, and the controller type and version number of the BIN file will be displayed in the corresponding dialog box.

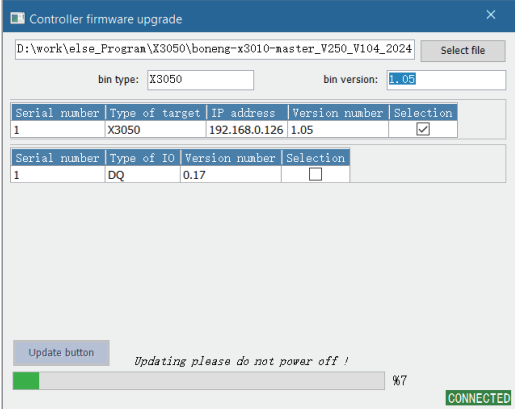
2.2.11.4 Connecting the controller

After selecting a controller and checking it, the version number in the controller will be automatically read. If the controller is also equipped with extended IO, the configuration information of the extended IO will also be displayed in the dialog box below. After selecting the controller, the lower right corner will display 'Connected'. Indicates that the controller has been connected.



2.2.11.5 Controller body upgrade

After selecting the controller, there is no need to choose extended IO. Choosing extended IO indicates that the upgraded object is a certain extended IO under the controller. Click the upgrade button to upgrade the controller. Please do not power off during the upgrade process! Otherwise, it will cause the controller to permanently crash.

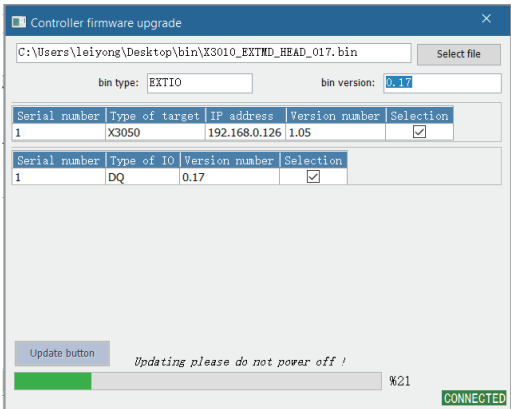


When the progress bar reaches 100%, it indicates that the upgrade is complete and will automatically disconnect from the controller.

2.2.11.6 Expansion module upgrade

The way to upgrade the expansion module is similar to upgrading the control body. First, select the expansion module file that needs to be burned,

then check the controller, select the expansion IO that the user needs to upgrade from the pop-up expansion IO list, and click upgrade. As shown in the following figure:



2.3 Programming Language

2.3.1 Programming Language Types

The programming languages supported by BonengLogicLab are:

- Ladder diagram (LD)
- Structured Text Programming (ST)
- Functional Block Diagram (FBD)
- Sequential Function Diagram (SFC)
- Instruction List (IL)

2.3.2 Ladder Diagram (LD)

We have already introduced how to add (create) a program in the Quick Start section, so we will not go into detail here. After adding the ladder diagram program, you can edit the ladder diagram in the network. As shown in the following figure;



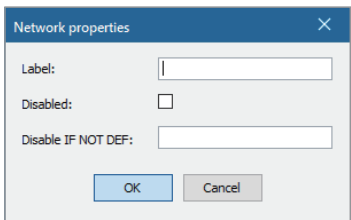
Among them(?), Indicates that the contact or coil has not yet been assigned a variable.

2.3.2.1 Network

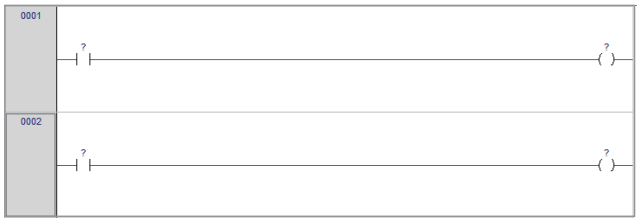
In the ladder diagram, a new network needs to be created in order to edit the LD program in that network. Right click on the network number to create a new network before, after, at the top, or at the bottom of the current network. As shown in the following figure:



As shown in the above figure, clicking on the properties can edit the label of the network (jump in the program, not comments), as well as whether to disable the network.



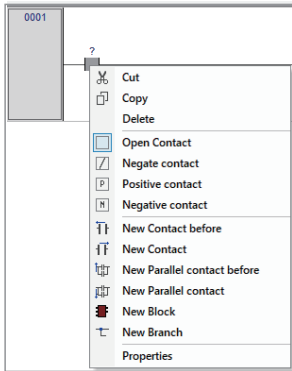
The shortcut menu button  also has the function of creating a new network. After creating a new network, as shown in the following figure:



2.3.2.2 Contact Points

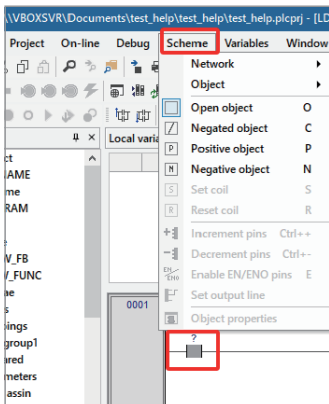
2.3.2.2.1 Adding Contacts

Right clicking on a touchpoint with the mouse will bring up a right-click menu bar, as shown in the following figure:



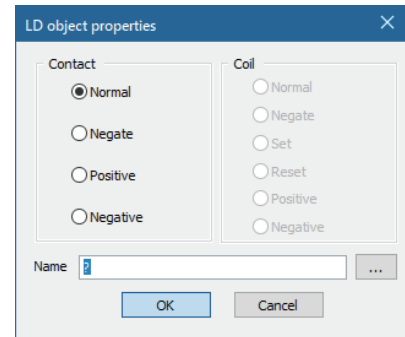
In this menu, you can see that the operations related to this contact are from top to bottom, including cutting, copying, deleting the contact, setting it as a normally open contact, normally closed contact, setting it as a rising or falling edge, creating a new contact on the left side of the contact, creating a new contact on the right side of the contact, creating a parallel contact above the contact, and creating a parallel contact below the contact. Create a new block to the right of the contact and add a new branch to the right of the contact. And set the properties of the contact point.

After selecting a touch point, the shortcut menu button related to LD above will be highlighted, indicating that it can be clicked. As shown in the following diagram: . Similarly, after selecting a contact point, you can click on "Design" in the menu bar above to operate that contact point. As shown in the following figure:

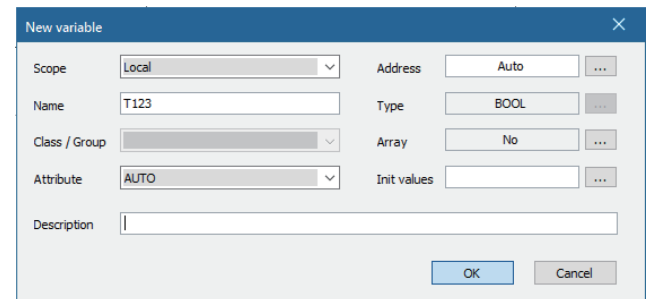


2.3.2.2.2 Attribute Setting of Contact Points

Double clicking on this touchpoint will bring up its property settings (or following the method above, right-click and select properties), as shown in the following figure:

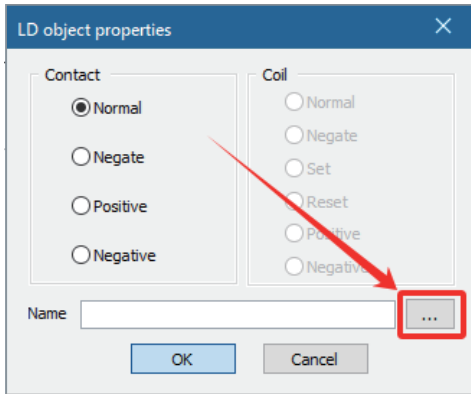


For the selection of contact types, there are normally open, normally closed, rising edge, and falling edge. Here, a name needs to be entered for the contact. After entering the name, if the input name is not found in the global variable or in the local variable, a dialog window for creating a new variable will pop up as shown in the following figure:



Since this is adding variables to the touchpoint, this type can only be BOOL type.

Alternatively, after opening the object properties, click the button to the right of the name to enter the object browser and select the variable. As shown in the following figure:



2.3.2.3 Coil

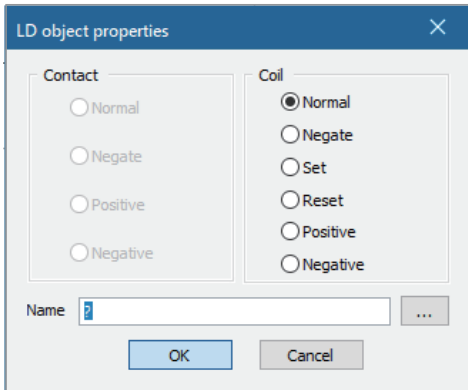
2.3.2.3.1 Adding coils

In a network, right-click on a coil to cut, copy, delete, add another coil (dual coil output), and set its properties, as shown in the following figure:



2.3.2.3.2 Coil attribute settings

Double click on the coil to enter its property settings, which can be set to normally open, normally closed, set, reset, rising edge, falling edge, etc.

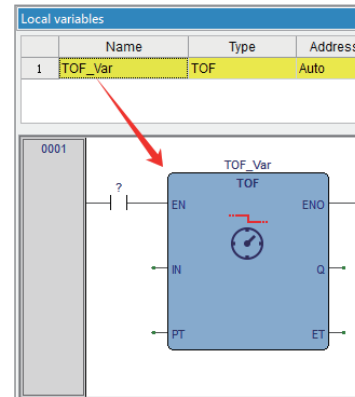


In terms of name setting, it is consistent with the method of setting the name of the contact point.

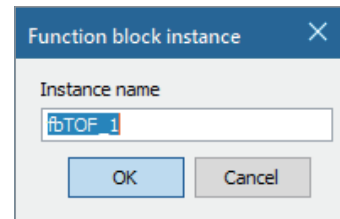
2.3.2.4 Function Blocks and Functions

2.3.2.4.1 Adding Function Blocks and Functions

BonengLogicLab adds functional blocks and functions to the program, which is very simple. There are two specific methods for adding. The first method is to design the required variables in the variable design list, and then use the mouse to drag and drop them onto the touchpoint to add the function block or function to the right of that touchpoint. As shown in the following figure.



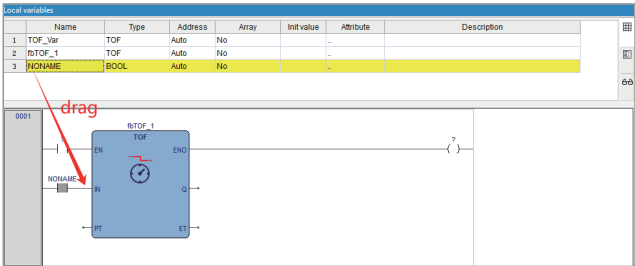
The second method is to drag and drop directly from the library. After dragging, the software will generate a dialog box, which is an instance of the function block.



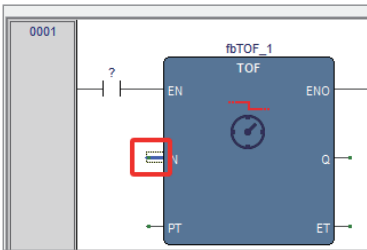
If you don't want to change the name, simply click OK.

2.3.2.4.2 Setting function block pins

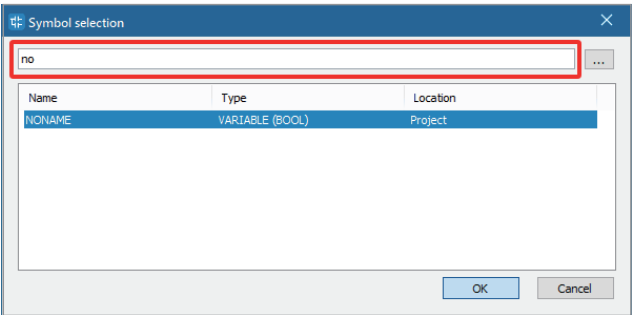
When using a function block, it is necessary to set the pins of the function block, that is, to assign variables to the pins of the function block. To assign variables to a pin, simply drag the variables from the variable table onto that pin. As shown in the following figure:



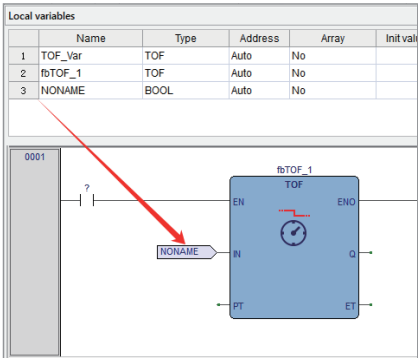
For dragged variables, they will be automatically connected to the left bus. So here we introduce another method of adding variables to pins. The pins of the mouse click function block are shown in the following figure:



After clicking, an object selection window will pop up.



Enter the name in this window to filter, double-click to select the desired variables, and click OK.

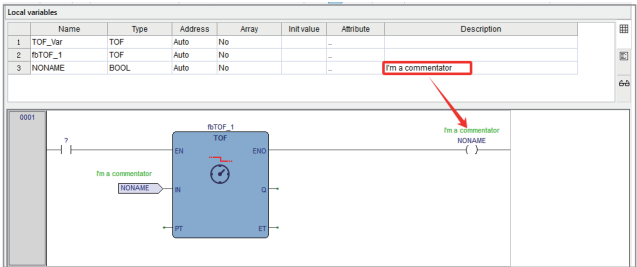


You can also click  the button on the right side of the object selection dialogue window to enter the object browser and directly select the variable name.

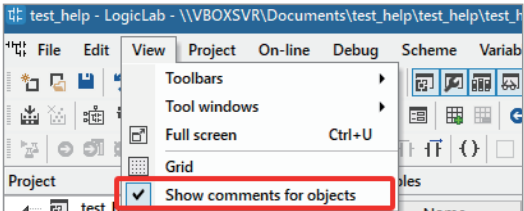
2.3.2.5 Adding Comments

2.3.2.5.1 Variable Annotations

Annotations can only be displayed above the contacts or coils in LD, as shown in the following figure:

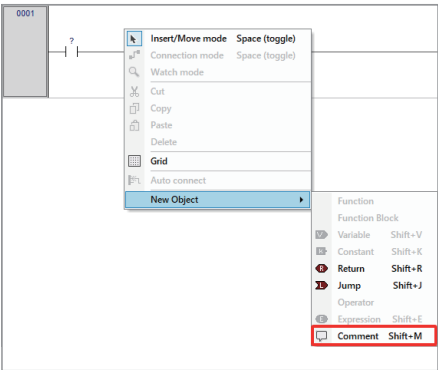


If no comments are displayed, you need to go to the menu bar, click View, and check the comments for the displayed object.



2.3.2.5.2 Network Annotations

Each network can have its own annotation. Right click on the blank space in the network, create a new object, and click on the annotation to add annotations to the network. As shown in the following figure:

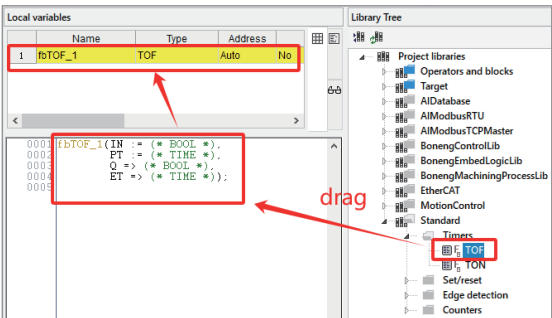


2.3.3 Structured Text Programming (ST)

2.3.3.1 Edit Code

There are two options to add a reference to the user's existing PLC object in the ST code:

- Directly input the name of the PLC object;
- Drag it to a suitable position. Global variables can be obtained from the workspace window, standard operators and feature/function blocks can be dragged from the library window to the target code location, and local variables can be selected from the local variable editor. As shown in the following figure: Drag and drop the function block directly from the library to the code editing area to generate a template and automatically create an instance object within the local variable.



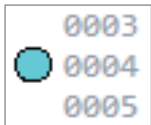
2.3.3.2 Editing Function

The ST editor has features shared by most compilers running on the Windows platform, where selecting text allows for copying, pasting, and other operations.

2.3.3.3 Bookmarks

Users can set bookmarks to indicate frequently accessed lines in the source file. Set the bookmark icon, which can be moved to it using keyboard commands. When the user no longer needs it, they can delete the bookmark.

Set bookmark: Move the insertion point to the line where you want to bookmark, then press Ctrl+F2. The left side of the line is marked by a light blue circle;

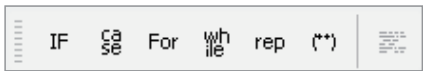


Jump to bookmark: Press F2 repeatedly until the target line is reached;

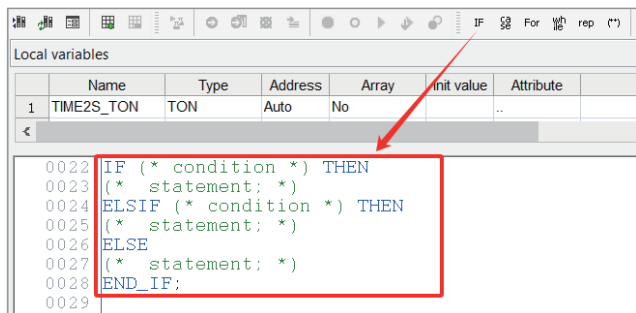
Delete bookmark: Move the cursor to any position on the line containing the bookmark, and then press Ctrl+F2.

2.3.3.4 ST Common Grammar Format Generation

In ST editing, if you forget the ST syntax format, you can directly click on the shortcut button in the menu bar to generate it. When editing ST, as shown in the following figure: it will be highlighted, indicating that it can be clicked. From left to right, it is IF, case, For while rep comments, etc.



Click the IF button to directly generate the syntax shown in the following figure:



Just need to supplement your own conditions and content. The other buttons will not be explained again.

2.3.4 Functional Block Diagram (FBD)

The function of the functional module diagram will not be explained in detail here, and most of its operation methods are similar to LD.

2.3.5 Sequential Function Diagram (SFC)

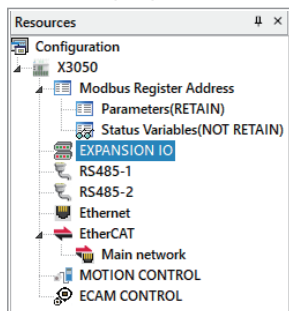
The function of the functional module diagram will not be explained in detail here, and most of its operation methods are similar to LD.

2.3.6 Instruction List (IL)

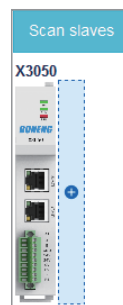
The function of the functional module diagram will not be explained in detail here, and most of its operation methods are similar to LD.

2.4 Expansion Module

After creating a new project, you can see the extension module in the resource list, as shown in the following figure:



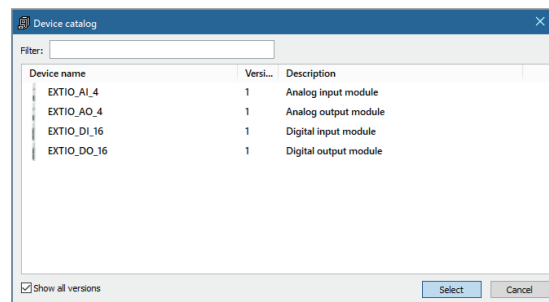
Double click to open the extension module, and you will see the following operation page:



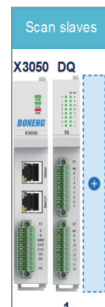
At this point, you can manually or automatically add 5 types of expansion modules based on the actual configuration of the expansion module.

1. Manual addition: Click the plus sign in the above picture, and the following page will pop up.

Select an expansion module that corresponds to the actual installed slot number, otherwise the slot number position needs to be modified and the extension slot number needs to be modified. The following explanation will be provided.



Click on the selection to successfully add, as shown in the figure below:

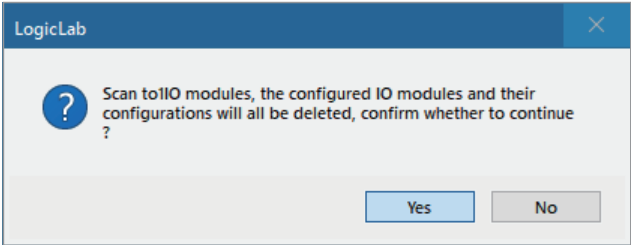


It can be seen that there is a label 1 under the DI module, indicating that the manually added DI module is in slot one.

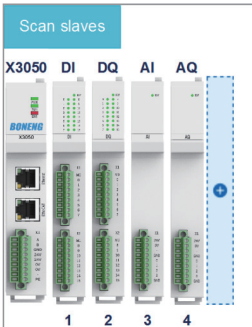
2. Automatically add, click on Scan Slave in the following image.



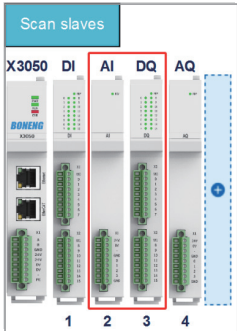
After clicking, it will pop up that no extension RS485 interface was found. After clicking OK, the actual number of extension modules scanned will appear. As shown in the following figure:



Click Yes (Y), and the slave display interface in the software will be arranged according to the actual placement order of the backplane slot numbers, corresponding to them. Four extension modules have been added here, as shown in the following figure:



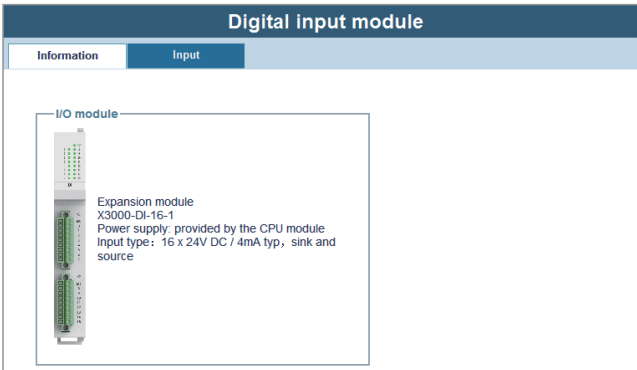
When manually adding the order of arrangement is out of order (compared to the actual situation), it can be seen that the positions of slot 2 and slot 3 are out of order as shown in the following figure:



Simply hover the mouse over slot 2 and right-click to select , or right-click on slot 3 and select . You can match the configuration with the actual installed expansion slot number.

2.4.1 Digital Input DI

Clicking on the DI module in slot 1 will display the following basic information:



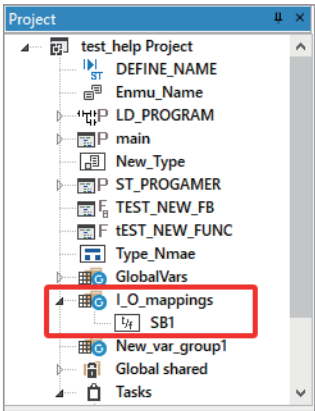
Click on the input to see the information shown in the following figure. This extension module has 16 switch inputs, and the input variable type is BOOL. You can set the variable name of the indirect access address in the variable list.

Information				
Input				
Assign UnAssign				
#	Data	Object Name	Type	Variable
6	DI5	Digital input 05	BOOL	
7	DI6	Digital input 06	BOOL	
8	DI7	Digital input 07	BOOL	
9	DI8	Digital input 08	BOOL	
10	DI9	Digital input 09	BOOL	
11	DI10	Digital input 10	BOOL	
12	DI11	Digital input 11	BOOL	
13	DI12	Digital input 12	BOOL	
14	DI13	Digital input 13	BOOL	
15	DI14	Digital input 14	BOOL	
16	DI15	Digital input 15	BOOL	

Here, after setting the variable name SB1 for DIO, which is the switch input 00, as shown in the following figure:

#	Data	Object Name	Type	Variable
6	DI5	Digital input 05	BOOL	SB1

At this point, click save and you will see that the I2 mappings appear in the project column, with the SB1 variable that was just set under it.

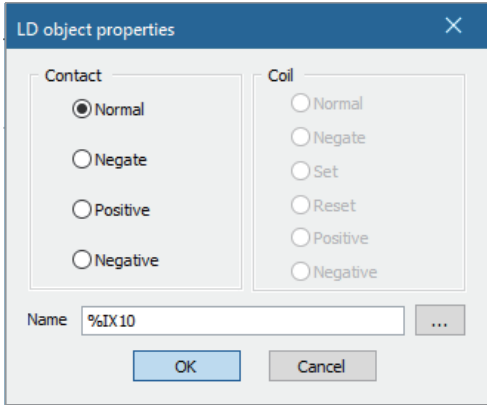


Clicking on I_O-mappings will bring up the following page, where you can see that the address of SB1 is% IX1.0. (The address is automatically assigned and can also be modified proactively, and the format needs to be correct)

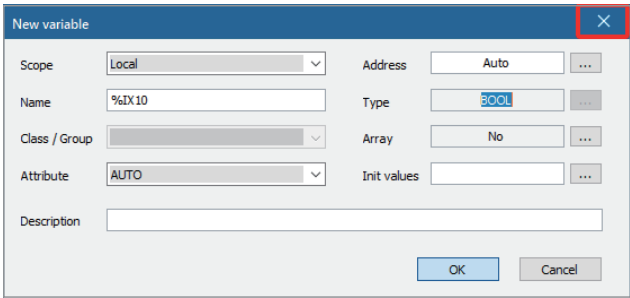
	Name	Type	Address	Array	Init value	Attribute	Description
1	SB1	BOOL	%IX1.5	No			

Here, extension terminals can be represented through indirect and direct access methods.

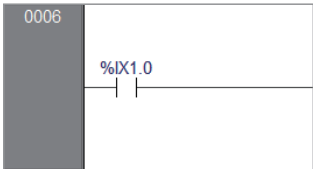
1. To access directly as shown, simply change the name to address in the trapezoidal contacts, select a contact to change the name to% IX1.0, and click OK.



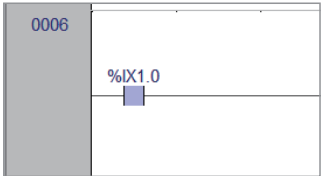
After clicking "OK", the following dialog box will pop up. When accessing directly, simply click "X". (Clicking on it will result in an error, this is the dialog box used when creating variables)



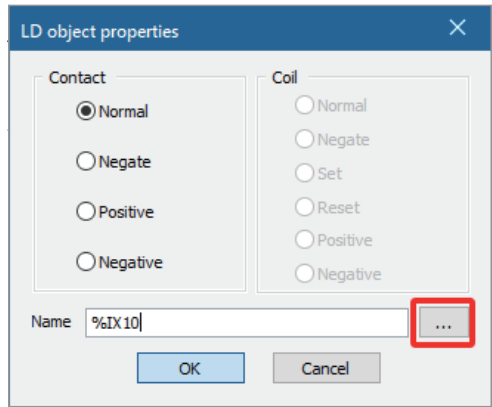
Successfully established as shown in the following figure: The address corresponds to the input variable, entering debugging mode.



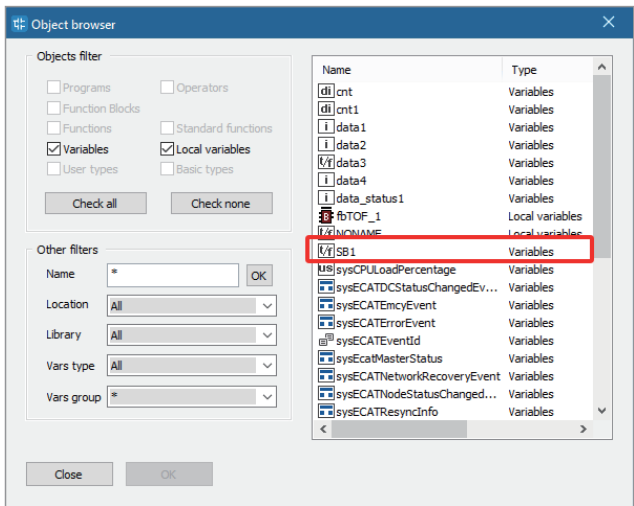
When turned on, the contact will conduct when the switch is turned on, as shown in the following figure:



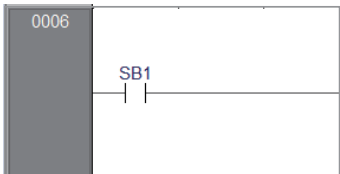
2. Indirect access is represented by directly calling the established variable name. Clicking on the touchpoint will bring up the following dialog box (LD object properties). Click on the red box "..." in the following image.



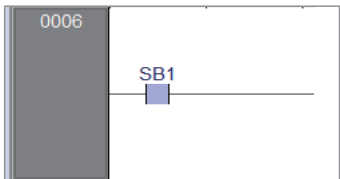
After clicking, the following dialog box will appear. (Object Browser)



In the object browser, you can see the variable name that was just set. The type of SB1 variable is Variables. Click SB1 and then click OK. Then the object properties will be returned. Click OK again to display the following image.



Recompile, download, debug.



2.4.2 Digital Output DQ

The use of DQ module is similar to DI. In the expansion module, select the DQ module, click the output button, set the variable name of the switch output 00 as OUT1, and click save.

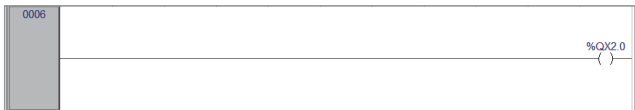
Digital output module				
Information				
Output				
Assign UnAssign				
#	Data	Object Name	Type	Variable
1	D00	Digital output 00	BOOL	OUT1
2	D01	Digital output 01	BOOL	
3	D02	Digital output 02	BOOL	
4	D03	Digital output 03	BOOL	
5	D04	Digital output 04	BOOL	
6	D05	Digital output 05	BOOL	
7	D06	Digital output 06	BOOL	
8	D07	Digital output 07	BOOL	
9	D08	Digital output 08	BOOL	
10	D09	Digital output 09	BOOL	

In the I-O-mappings in the project, the following figure shows that the address of OUT1 is% IX2.0.

	Name	Type	Address	Array	Init value	Attribute	Description
1	SB1	BOOL	%IX1.5	No			
2	OUT1	BOOL	%QX2.0	No			

The usage methods are also divided into direct call and indirect call

3. Directly call as shown in the following figure, and refer to the usage of the DI module for specific calling methods.



Open debugging mode and obtain the result shown in the following figure.



The indirect call is shown in the following figure, and its calling method can be found in the DI calling method



2.4.3 Analog Input AI

Click on the analog input module, and then click on the input to set the variable for analog input 00 to AI_in1. You can see that the data type for analog input is WORD. Click to save.

Analog input module				
Information		Input		
Assign		UnAssign		
#	Data	Object Name	Type	Variable
1	AI0	Analog input 00	WORD	AI_in1
4	AI3	Analog input 03	WORD	
3	AI2	Analog input 02	WORD	
2	AI1	Analog input 01	WORD	

After clicking save, the following image can be seen in the I-O-mappings section of the project, where the address of AI_in1 is %IW3.0.

	Name	Type	Address	Array	Init value	Attribute	Description
1	SB1	BOOL	%X1.5	No			
2	OUT1	BOOL	%QX2.0	No			
3	AI_in1	WORD	%IW3.0	No			

When this variable is used in the program, the value will change when the corresponding channel of the extension module has voltage or current input.

The switching between voltage and current modes requires the use of Boneng's library functions. The specific implementation method can be found in section 3.2.4.1.1. If voltage detection or current detection is not set, voltage detection will be defaulted.

The voltage range of the input voltage is 0-10V, corresponding to a value of 0-10000;

The current input range is 0-20mA, corresponding to a value of 0-20000;

2.4.4 Analog output AQ

The use of AQ extension module is similar to AI extension. Create variables, AQ_out1, and click save.

Analog output module				
Information		Output		
Assign		UnAssign		
#	Data	Object Name	Type	Variable
2	AQ1	Analog output 01	WORD	AQ_Out1
3	AQ2	Analog output 02	WORD	
4	AQ3	Analog output 03	WORD	
1	AQ0	Analog output 00	WORD	

After clicking save, the following image can be seen in the I-O-mappings section of the project, where the address of AQ_out1 is %QW4.0.

	Name	Type	Address	Array	Init value	Attribute	Description
1	SB1	BOOL	%X1.5	No			
2	OUT1	BOOL	%QX2.0	No			
3	AI_in1	WORD	%IW3.0	No			
4	AQ_Out1	WORD	%QW4.1	No			

By changing the value of variable AQ_out1 in the program, the output voltage of its corresponding channel can be altered.

The voltage output by AQ ranges from 0-10V, corresponding to a value of 0-10000;

Note: The AQ module only has electrical output and no current mode.

2.4.5 Communication module CI

The CI module does not have actual software configuration information. It is worth noting that only one CI module can be mounted in a PLC device, and it must be closely attached to the CPU module body. The X3000 series supports Modbus RTU dual master stations, one of which is this CI module. Specific configuration information can be found in chapter 2.5 Serial Communication.

2.5 Serial Communication

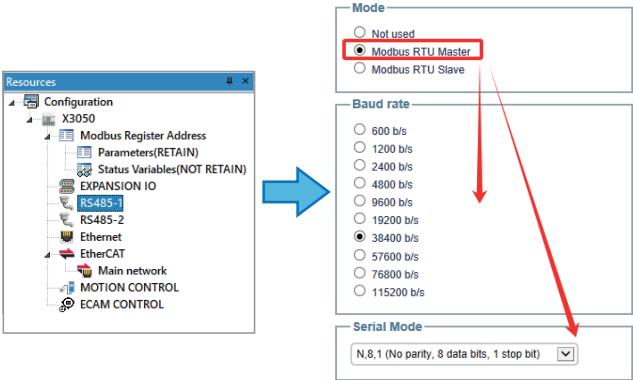
2.5.1 Overview

BonnegLogicLab supports dual master communication, with one master on the CPU module and the other on the expansion module. But the configuration methods and usage steps of the two main stations are the same. Configure resources.

2.5.2 Modbus RTU Master Station Configuration

2.5.2.1 Open Modbus RTU main station

Click on RS485-1 in the resource and select Modbus RTU master in the opened window page. Two configuration pages for baud rate and serial mode will automatically pop up, as shown in the following figure:

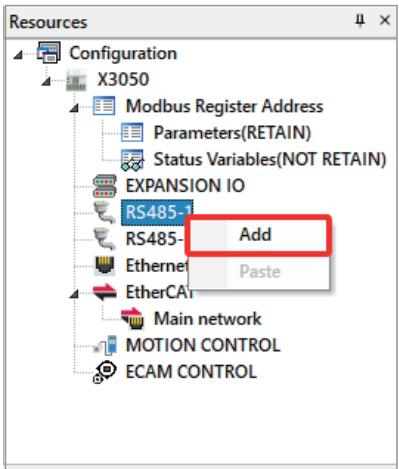


In the configuration page above, you can select the baud rate and serial mode. The instructions for selecting serial mode are shown in the following table:

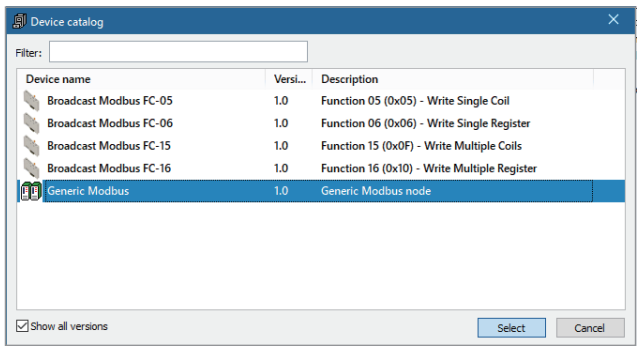
Mode selection	illustrate
N,8,1 (No parity,8 data bits, 1 stop bit)	No verification, 8-bit data bit, 1-bit stop bit
E,8,1 (Even parity, 8 data bits, 1 stop bit)	Even check, 8-bit data bit, 1-bit stop bit
O,8,1 (Odd parity,8 data bits, 1 stop bit)	Odd check, 8-bit data bit, 1-bit stop bit
N,8,2 (No parity, 8 data bits, 2 stop bits)	No verification, 8-bit data bit, 2-digit stop bit
E,8,2 (Even parity,8 data bits, 2 stop bits)	Even check, 8-bit data bit, 2-digit stop bit
O,8,2 (Odd parity,8 data bits, 2 stop bits)	Odd check, 8-bit data bit, 2-digit stop bit

2.5.2.2 Adding Communication Nodes

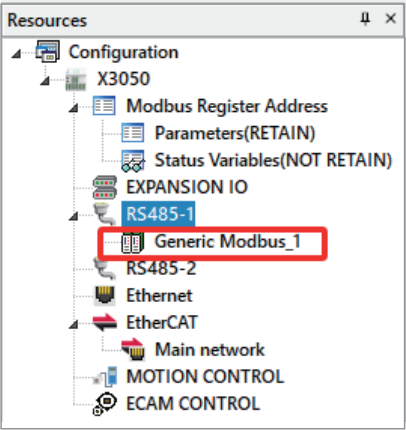
After opening the main station, right-click on RS485-1 and click add as shown in the following figure:



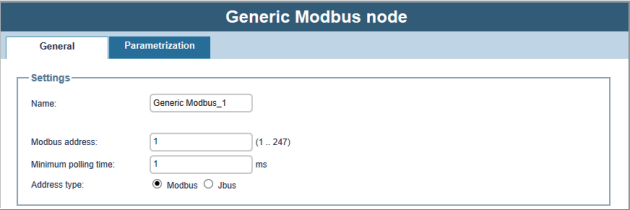
After clicking, a device directory window will pop up, as shown in the following figure:



Select the universal Modbus node in the device directory window (other devices have broadcast function), click select, and this universal node will appear below RS485-1. As shown in the following figure:



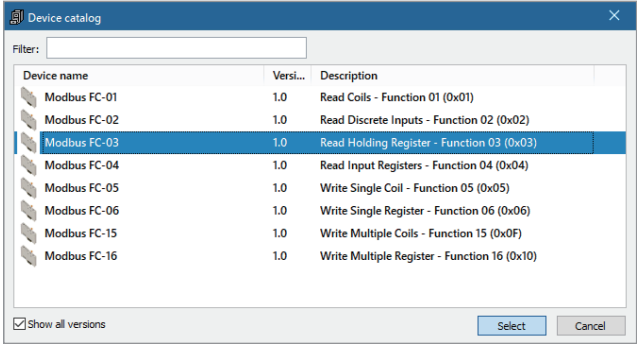
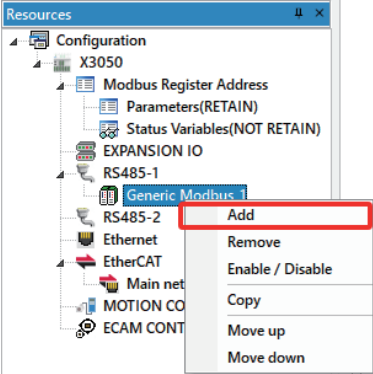
Click on the node with the mouse to configure it, as shown in the following figure:



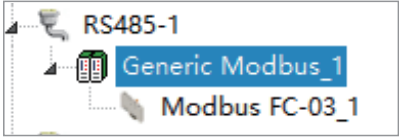
You can change the name of the node, the address of the slave station (1-247), the minimum polling time, etc. Modifying the minimum polling time can constrain the polling time of functional nodes such as FC03 and FC06 under this node.

2.5.2.3 Adding Function Nodes

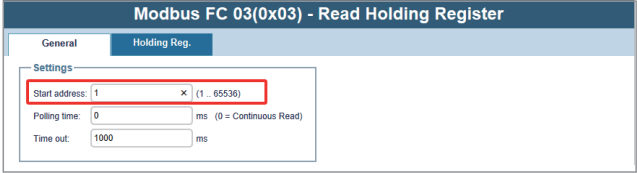
Right click on the created generic node and click Add as shown in the following figure:



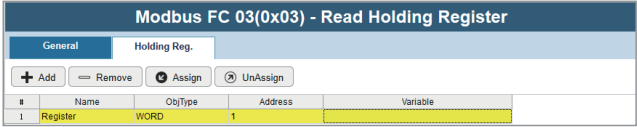
Taking the example of adding a read hold register - function FC-03, select FC-03 and click select. After adding it, this function node will appear under the general node. As shown in the following figure:



Click on the Modbus FC-03 function node, and in the General tab, you can configure register addresses, polling times, timeout times, and more. As shown in the following figure, the starting address needs to start from 1.

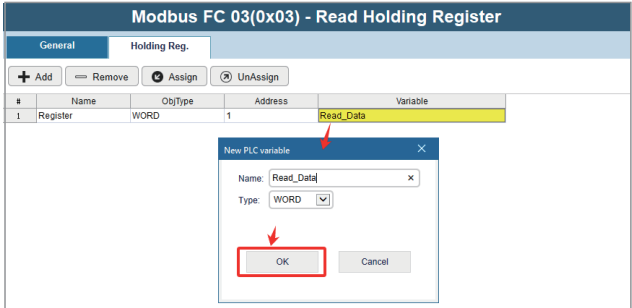


Click on the Keep Register tab in the above image to add, delete, and list variables in one row. You can assign and unassign a variable.

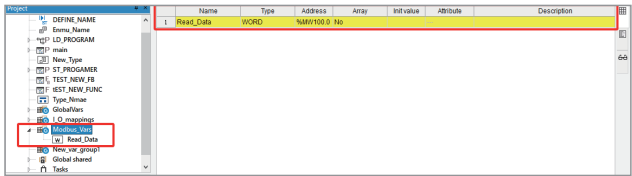


The main function of this page is to set a variable. Under the function node FC-03 mentioned above, it will access the register address configured for this function and assign the accessed data to this variable. This variable can be directly entered in

the variable box as shown in the following figure: enter the name Read_Sata, and a dialog box for creating a new PLC variable will pop up. Select the type of the PLC variable, click OK, and then it will be ready.



Afterwards, in the engineering window, a global variable group for Modbus_Vars will appear, in which the PLC variables created above will appear. As shown in the following figure:



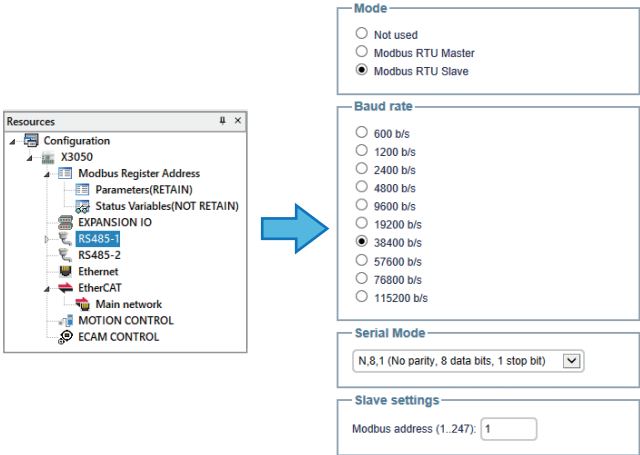
2.5.2.4 Precautions

When setting the register address to be accessed as a Modbus RTU master station, it is generally necessary to increment the address by 1.

2.5.3 Modbus RTU Slave Configuration

2.5.3.1 Open Modbus RTU Slave

When configuring Modbus RTU slave, you need to click on RS485 in the resources, select Modbus RTU slave on the configuration page, configure the slave baud rate, serial mode, and slave station number, as shown in the following figure:

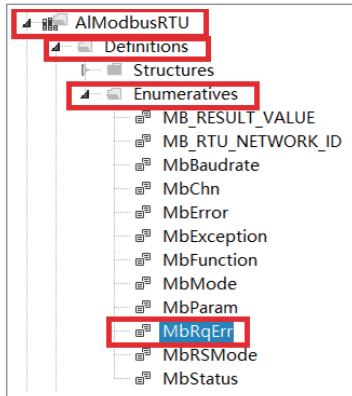


2.5.3.2 Creating Register Address

After creating a Modbus RTU node, it is necessary to design some register address variables for the master station to access. The variable design of register address needs to be designed in the parameter (power down hold) in the resource column. Detailed design methods can be found in sections 2.2.3 and 2.2.4.

2.5.4 Modbus Fault Code Explanation

The fault code can be accessed in the library management wizard for enumeration variable comparison



You can also refer to this place

Name	Value	Description
MbRqErr_Ok	0	No error occurred
MbRqErr_Illegal-Function	1	Invalid request message
MbRqErr_Illegal-DataAddress	2	Illegal data address
MbRqErr_IllegalDataValue	3	Illegal data value
MbRqErr_SlaveDeviceFailure	4	Slave device failure
MbRqErr_Acknowledge	5	Acknowledge error
MbRqErr_SlaveDeviceBusy	6	Slave device busy
MbRqErr_ParityError	7	Memory parity error
MbRqErr_CommChn	8	Communication channel configuration error
MbRqErr_InvalidSlaveAddress	9	Invalid slave address (address range 0 - 247)
MbRqErr_GatewayPathUnavailable	10	Gateway path unavailable
MbRqErr_GatewayDeviceFailed	11	Gateway target device failed to respond
MbRqErr_MsgMaxElements	12	Invalid number of element for this function
MbRqErr_MsgRequestInvalid	13	Cannot correctly build the request message
MbRqErr_MsgResponseInvalid	14	Invalid response message
MbRqErr_MsgSize	15	Response size error
MbRqErr_Timeout	16	Timeout reached
MbRqErr_TaskInvalid	17	Function invoked by an invalid task

2.6 Ethernet communication

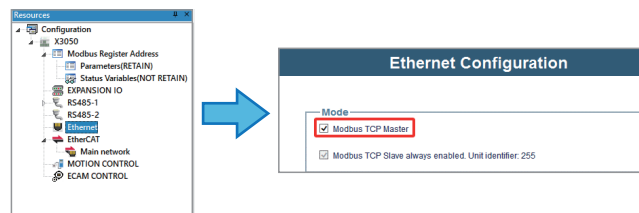
2.6.1 Overview

The Ethernet communication part of X3000 series PLC mainly integrates Modbus TCP protocol, including server and client. It can easily communicate and exchange data with devices that support Modbus TCP.

2.6.2 Modbus TCP Host Configuration

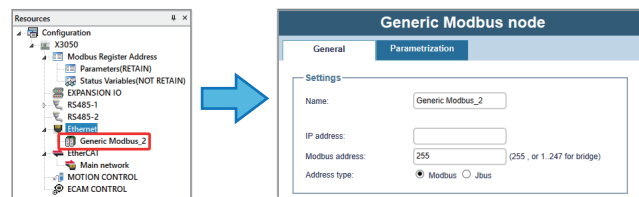
2.6.2.1 Activate Modbus TCP Master Station

Find Ethernet in the resource window, click on Ethernet, and select Modbus TCP master in Ethernet configuration. As shown in the following figure:

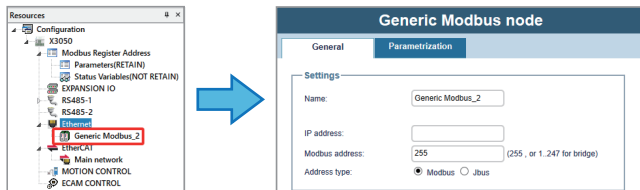


2.6.2.2 Adding Modbus TCP Universal Node

After selecting the Modbus TCP master, you can add a universal node by right clicking on Ethernet and selecting the universal Modbus node from the device directory. As shown in the following figure:



Click on the universal Modbus TCP node that was just added. Configure the node as shown in the following figure:



In this node, the node name, IP address of the slave, and Modbus slave address can be set.

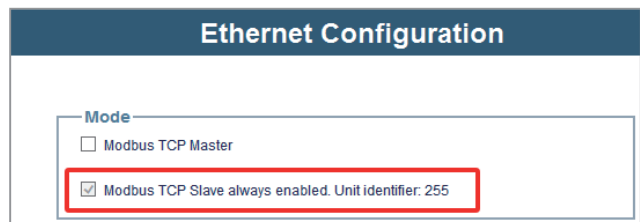
2.6.2.3 Adding Function Nodes

Under the Modbus TCP universal node, adding functional nodes is the same as adding Modbus RTU, which will not be elaborated here. For details, please refer to section 2.5.2.3.

2.6.3 Modbus TCP Slave Configuration

2.6.3.1 Open Modbus TCP Slave

Modbus TCP slave is always enabled by default, as shown in the following figure: simply configure the register address.



The IP address of the slave station is the IP address of the PLC, and the port number is 502.

2.6.3.2 Creating Register Address

The register addresses of Modbus TCP and Modbus RTU slave stations are shared, and the address for creating register variables can be found in sections 2.2.3 and 2.2.4.

2.6.4 Modbus Fault Code Explanation

Please refer to 2.5.4 Modbus fault code description

2.7 EtherCAT Communication

2.7.1 Overview

EtherCAT is an open industrial field technology based on Ethernet, characterized by short communication refresh cycles, low synchronization jitter, and low hardware costs. To learn about the principles and related technologies of EtherCAT, please refer to the book "Design and Application of EtherCAT Driver for Industrial Ethernet Fieldbus", or visit the official website of the EtherCAT Technical Committee at: <https://www.EtherCAT.org.cn>.

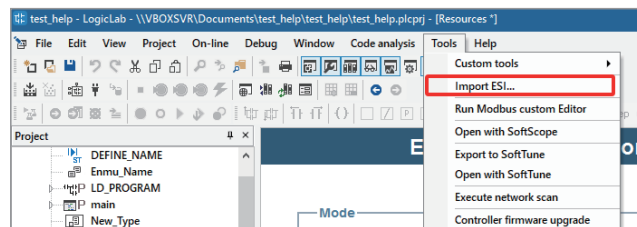
The X3000 series supports standard EtherCAT interfaces (1 RJ45 interface) and can support up to 32 slave stations using a linear topology structure. The minimum EtherCAT bus cycle can be set to 1ms.

2.7.2 EtherCAT Master Station Configuration

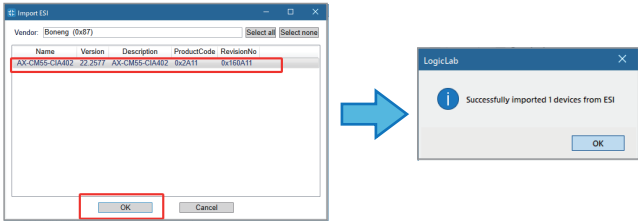
2.7.2.1 Importing Device XML

Before conducting EtherCAT communication, it is necessary to import the slave ESI file so that the master can recognize the slave information. This operation only needs to be imported once, and the slave information will be saved in Logiclab, so that the slave information can be recognized every time the slave is scanned.

Select Logiclab Tools>Import EtherCAT ESI file.

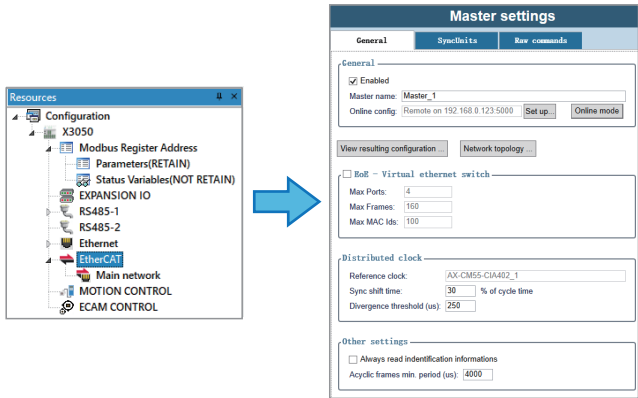


Then select the XML file location in the opened file window and choose the device where the file is saved. As shown in the following figure:



After adding the XML device, all new projects can be used without the need for repeated addition.

2.7.2.2 Basic Settings

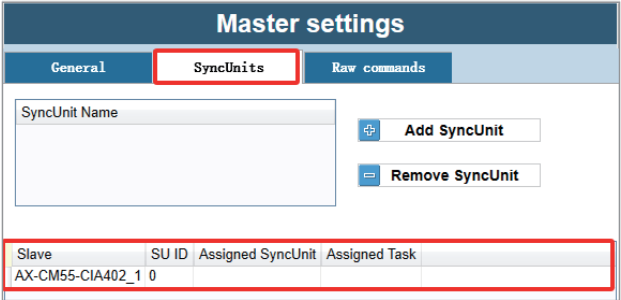


name	function
Enabled	Configure Enable Master Station
Master name	Set the name of the main station
Online config	The IP address of the PLC generally does not need to be modified by default
Online mode	The online mode button, when the logiclab software is connected to the controller, can be clicked to enter the online function
View resulting configuration	Browse configuration results, including slave related information, as well as startup configuration commands, etc.
Network topology	Graphic display configuration network connection
EoE- Virtual ethernet switch	Virtual network conversion related configurations are generally not involved
Distributed clock	Set the distributed clock to its default value

name	function
Other settings	Include the minimum period for synchronizing data frames and whether identity information is continuously read.

2.7.2.3 Synchronization Unit

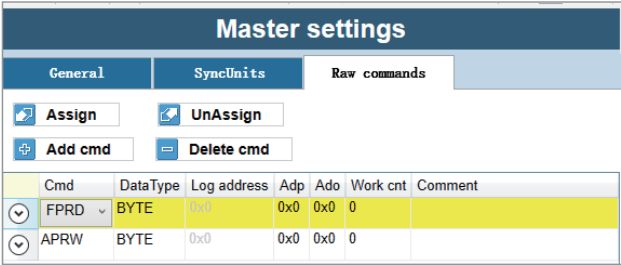
To avoid abnormal interference from some slave stations with other devices or to distinguish the source of faults, synchronous unit data grouping function can be used. As shown in the following figure



In a synchronization unit, a new synchronization unit can be created, a synchronization unit can be deleted, and a slave station can be assigned to a synchronization unit. If there are no special requirements, you don't need to worry, just default.

2.7.2.4 Custom Addressing Method

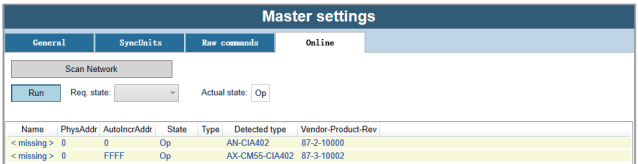
The custom addressing method associated with PLC variables can read the relevant registers or memory of the slave station. As shown in the following figure:



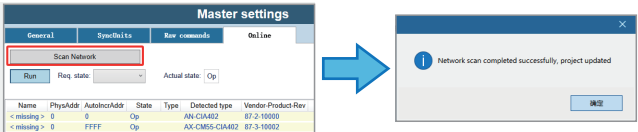
When creating an addressing method, it is achieved by associating PLC variables, so the content read and written is also based on changes in PLC variables. So here we need to input the PLC variable name, and after saving it, an additional global variable group will be added in the project, EtherCAT_mappings. It is necessary to understand the function of the command and the format of the message in order to use this function effectively. For detailed information, please refer to EtherCAT Addressing Method Analysis. If there are no specific requirements, there is no need to worry about this feature.

2.7.2.5 Scan and Add EtherCAT Slave Devices (Online)

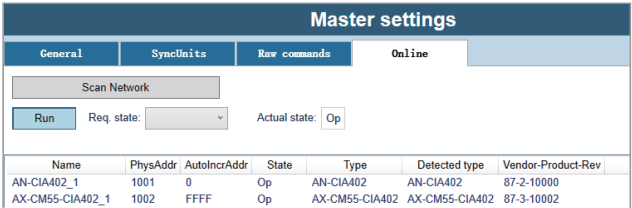
First connect the PLC and computer network cables, and then connect them to the EtherCAT network cable. On the EtherCAT main station basic settings page, click the Online button, and you will see an Online tab. As shown in the following figure:



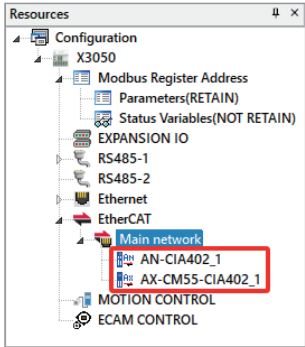
In this box, the connected slave devices will be pre scanned (displaying the device name and device ID, indicating that the XML device has been imported), but the device has not been configured into the project yet, so you need to click Scan Network to scan.



After successful scanning, as shown in the figure below, the font changes from blue to black, indicating successful scanning.



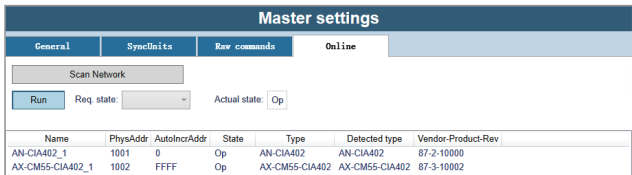
And two additional device nodes can be found in the main network of EtherCAT.



It is worth noting that if the XML file for this device has not been added in the BonengLogiclab software before, it will not be able to scan the device properly. When the main station is already running, if additional or reduced slave devices are added midway, the main station needs to be stopped in order to rescan for the added device information.

2.7.2.6 Online Monitoring

In the online monitoring of the main station, users can view the status of the main station and its sub stations, as well as some relevant information about the sub stations.

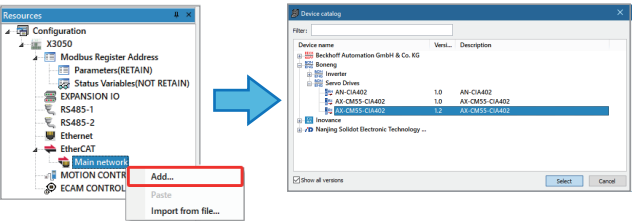


Actual state: represents the status of the main station. Click 'Run' to start or stop the main station.

2.7.3 EtherCAT Slave Configuration

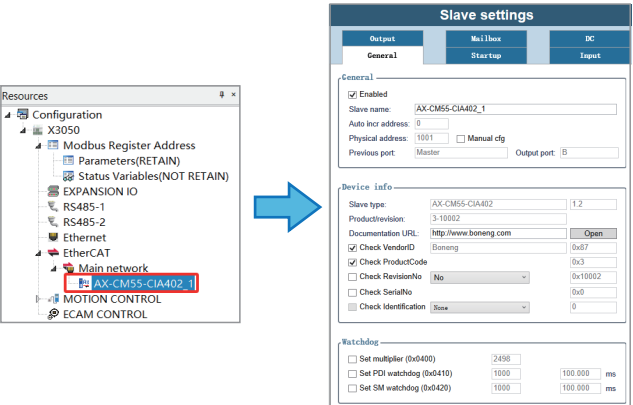
2.7.3.1 Manually Adding EtherCAT Slave

Right click on the main network, click Add, and a device directory dialog box will pop up. Find Boneng and expand it, then find Servo Drives and expand it, and select a servo drive device. Then click to select. As shown in the following figure:



2.7.3.2 General Settings

Click on the added EtherCAT slave device, and the slave configuration page will appear, as shown in the following figure:



name	describe
Enabled	Enable options from the station
Slave name	Station name
Auto incr address	Automatically added address (no special requirements need to be changed)
Physical address	Physical address (no special requirements, no need to change)
Previous port	Previous port

name	describe								
Output port	output port								
Device info	Device information container, in which the type of slave station, XML version number, product version number, and product supplier ID can be seen. Unless there are other special requirements, the content in this box generally does not need to be changed.								
Watch dog (container)	The watchdog setting box generally does not need to be set.								
Timeouts for state machine (ms) (container)	The timeout for state machine jump is generally the default value and does not need to be modified. The abbreviations are as follows: <table><tr><td>I->P</td><td>Initialize to PreOP</td></tr><tr><td>P->S,S->O</td><td>PreOP to SafeOP, SafeOP to OP</td></tr><tr><td>Back to P,I</td><td>Return to Init or PreOP</td></tr><tr><td>O->S</td><td>Op to SafeOP</td></tr></table>	I->P	Initialize to PreOP	P->S,S->O	PreOP to SafeOP, SafeOP to OP	Back to P,I	Return to Init or PreOP	O->S	Op to SafeOP
I->P	Initialize to PreOP								
P->S,S->O	PreOP to SafeOP, SafeOP to OP								
Back to P,I	Return to Init or PreOP								
O->S	Op to SafeOP								
Use LRD/LWR instead of LRW	Replacing LRW with LRD/LWR generally does not require any changes								

It is worth noting that. When using the X3000 series controller with a slave, it generally does not directly use addresses to manage the slave, but uses indexes for management. The index directly connected to the control is designated as index 0, and when linearly connected, the index is sequentially incremented by 1.

2.7.3.3 Process Data

The Output tab and Input tab are two parts that are often required to be configured in the slave configuration project. Click on the Output tab on the Slave settings page as shown in the following figure:

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x1600	Outputs	SM 2	☐	☐	0
0x1701	Outputs	---	☐	☐	0
0x1702	Outputs	---	☐	☐	0
0x1703	Outputs	---	☐	☐	0
0x1704	Outputs	---	☐	☐	0

The column headings in the table above are explained as follows:

name	describe
Index	Index of PDO
PDO Name	The name of PDO
Assigned SM	The PDO group selected varies among different devices. Some devices support selecting multiple PDO groups, while others only allow selecting one PDO group (such as the servo device AX-CM55-CIA402 from Boneng). Therefore, when selecting only one PDO group, it is necessary to first cancel the previously selected group and then reselect it.
Fixed	Fixed PDO group, fixed PDO group does not allow modification of the content in this group. If there is no fixed group, users are allowed to add object directories, delete object directories, map PLC variables, and other operations under this group.
Mandatory	Is it mandatory
SU ID	The synchronization unit ID can be assigned in the "SyncUnit" tab of the main station

2.7.3.3.1 Selecting PDO Group

In different slave devices, the group and quantity of PDOs are different. Usually, different PDOs need to be selected for configuration according to different usage situations, as shown in the following figure: the group where SM2 is located indicates that it has been selected.

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x1600	Outputs	SM 2	☐	☐	0
0x1701	Outputs	---	☐	☐	0
0x1702	Outputs	---	☐	☐	0
0x1703	Outputs	---	☐	☐	0
0x1704	Outputs	---	☐	☐	0

2.7.3.3.2 Adding PDO Group

Find Add PDO in the Output tab to add PDO groups, as shown in the following figure:

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x0	---	---	☐	☐	0
0x1600	Outputs	---	☐	☐	0
0x1701	Outputs	SM 2	☐	☐	0
0x1702	Outputs	---	☐	☐	0
0x1703	Outputs	---	☐	☐	0
0x1704	Outputs	---	☐	☐	0

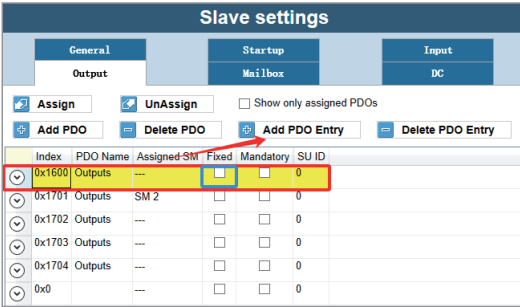
2.7.3.3.3 Delete PDO group

After selecting a PDO group (the line where the cursor is located), click Delete PDO to delete the line, that is, delete the PDO group. As shown in the following figure:

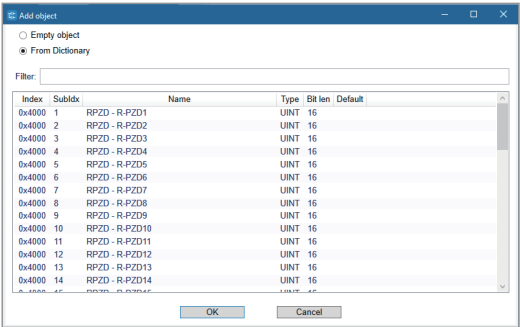
Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x0	---	---	☐	☐	0
0x1600	Outputs	---	☐	☐	0
0x1701	Outputs	SM 2	☐	☐	0
0x1702	Outputs	---	☐	☐	0
0x1703	Outputs	---	☐	☐	0
0x1704	Outputs	---	☐	☐	0

2.7.3.3.4 Adding Objects within PDO Group (Object Dictionary)

In the OutPut tab, select an unsecured PDO object group and click Add PDO Entry.

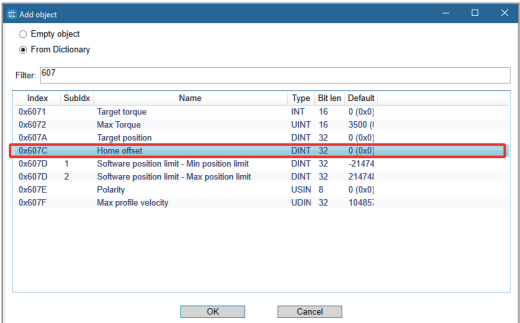


After clicking, a dialog box as shown in the following figure will appear.

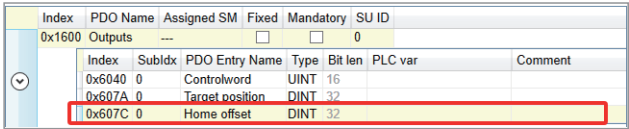


In this dialog box, one board selects from the object dictionary (From Dictionary). In Filter, you can input name or index related information to filter related results.

For example, to add an object with an origin offset, enter 607 to filter out all objects related to 607. As shown in the following figure:

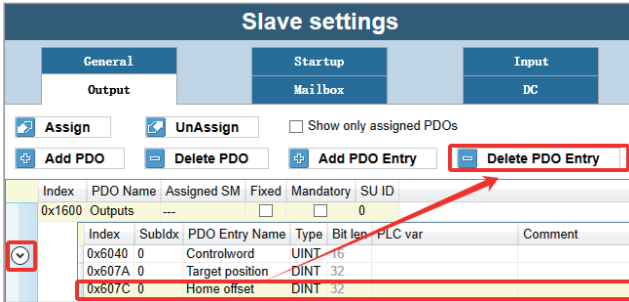


Select the origin offset 0x607c and click OK. Expand the PDO group you just selected, and you can see the newly added origin offset object below, as shown in the following figure:

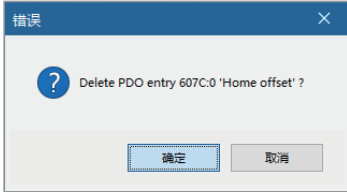


2.7.3.3.5 Delete Objects within PDO Group (Object Dictionary)

Select a PDO group, click the expand button on the left to expand the PDO group, select an object, and click Delete PDO Entry above to delete the object, as shown in the following figure.



After clicking, it will ask if the object needs to be deleted. Click Yes to delete the object, as shown in the following figure:



2.7.3.3.6 Map PLC variables to group objects

In PLC programs, an object within the PDO group can be directly accessed through PLC variables.

After expanding the PDO group, you can see the intra group object table. In the intra group object table, the column where PLC var is located is the mapped PLC variable.

Slave settings

General Startup Input

Output Mailbox DC

Assign UnAssign Show only assigned PDOs

Add PDO Delete PDO Add PDO Entry Delete PDO Entry

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x1600	Outputs	---	☐	☐	0

Index	Subidx	PDO Entry Name	Type	Bit len	PLC var	Comment
0x607A	0	Target position	DINT	32		
0x6040	0	Controlword	UINT	16		

In PLC var, users can directly input the name of the PLC variable they want to generate. After entering the name, the type of the PLC variable will be selected, which should generally be consistent with the object type. Relevant comments can be entered in the Comment. As shown in the following figure:

Index	PDO Name	Assigned SM	Fixed	Mandatory	SU ID
0x1600	Outputs	---	☐	☐	0

Index	Subidx	PDO Entry Name	Type	Bit len	PLC var	Comment
0x6040	0	Controlword	UINT	16	Control_word	
0x607A	0	Target position	DINT	32		

After completing the input, the PLC variable will appear in the EtherCAT_mappings global variable group.

If the PLC variable has already been defined before, you can use the Assign button to assign the selected object to a PLC variable. Similarly, Unassign can also be used to cancel the assignment.

2.7.3.4 Online Monitoring

After enabling Online mode on the main page, an Online tab will appear on the sub page tab. Click on the Online tab as shown in the following figure:

Slave settings

General Startup Input Output

Mailbox DC Online

ESC Registers access ... EEPROM access ... COE SDO access ... Diagnostics ...

Req. state: Actual state: Op

Direction	Index	Subidx	Name	Type	Bit len	Forced	Value	PLC variable
Input	0x603F	0	Errorcode	UINT	16	☐	0 (0x0)	
Input	0x6041	0	Statusword	UINT	16	☐	0 (0x0)	
Input	0x6061	0	Modes of operation display	SINT	8	☐	0 (0x0)	
Input	0x6064	0	Position actual value	DINT	32	☐	-168139581 (0xF5FA6)	
Input	0x606C	0	Velocity actual value	DINT	32	☐	0 (0x0)	
Input	0x6077	0	Torque actual value	INT	16	☐	0 (0x0)	
Input	0x60B9	0	Touch probe status	UINT	16	☐	0 (0x0)	
Input	0x60BA	0	Touch probe pos1 pos value	DINT	32	☐	0 (0x0)	
Input	0x60BC	0	Touch probe pos2 pos value	DINT	32	☐	0 (0x0)	
Input	0x60FD	0	Digital inputs	UDINT	32	☐	0 (0x0)	
Output	0x6040	0	Controlword	UINT	16	☐	0 (0x0)	
Output	0x6060	0	Modes of operation	SINT	8	☐	0 (0x0)	
Output	0x6071	0	Target torque	INT	16	☐	0 (0x0)	
Output	0x607A	0	Target position	DINT	32	☐	-168139491 (0xF5FA61D)	
Output	0x60FF	0	Target velocity	DINT	32	☐	0 (0x0)	
Output	0x6081	0	Profile velocity	UDINT	32	☐	0 (0x0)	
Output	0x60B8	0	Touch probe function	UINT	16	☐	0 (0x0)	

Four buttons appear on the above page, with specific functions as shown in the table below:

Button Name	illustrate
ESC Registers access ...	ESC register access
EEPROM access ...	EEPROM access
COE SDO access ...	COE SDO Access
Diagnostics ...	Diagnostic information viewing

In the table below, you can double-click or right-click to select and modify the value of an object.

Direction	Index	Subidx	Name	Type	Bit len	Forced	Value	PLC variable
Input	0x603F	0	Errorcode	UINT	16	☐	0 (0x0)	
Input	0x6041	0	Statusword	UINT	16	☐	0 (0x0)	
Input	0x6061	0	Modes of operation display	SINT	8	☐	0 (0x0)	
Input	0x6064	0	Position actual value	DINT	32	☐	-168139581 (0xF5FA6)	
Input	0x606C	0	Velocity actual value	DINT	32	☐	0 (0x0)	
Input	0x6077	0	Torque actual value	INT	16	☐	0 (0x0)	
Input	0x60B9	0	Touch probe status	UINT	16	☐	0 (0x0)	
Input	0x60BA	0	Touch probe pos1 pos value	DINT	32	☐	0 (0x0)	
Input	0x60BC	0	Touch probe pos2 pos value	DINT	32	☐	0 (0x0)	
Input	0x60FD	0	Digital inputs	UDINT	32	☐	0 (0x0)	
Output	0x6040	0	Controlword	UINT	16	☐	0 (0x0)	
Output	0x6060	0	Modes of operation	SINT	8	☐	0 (0x0)	
Output	0x6071	0	Target torque	INT	16	☐	0 (0x0)	
Output	0x607A	0	Target position	DINT	32	☐	-168139491 (0xF5FA61D)	
Output	0x60FF	0	Target velocity	DINT	32	☐	0 (0x0)	
Output	0x6081	0	Profile velocity	UDINT	32	☐	0 (0x0)	
Output	0x60B8	0	Touch probe function	UINT	16	☐	0 (0x0)	

After double clicking an object, the following window will appear.

Write once value

Bit size

1 8 16 32 64 Raw

Signed: 0

Unsigned: 0

Hex: 0

Float:

Boolean: TRUE FALSE

String:

Raw bytes: 00 00

Enum:

OK Cancel

You can input data types such as signed, unsigned, hexadecimal, etc., and click OK to modify them. If the modified value is changed by the program, you can check 'Forced' to force the value.

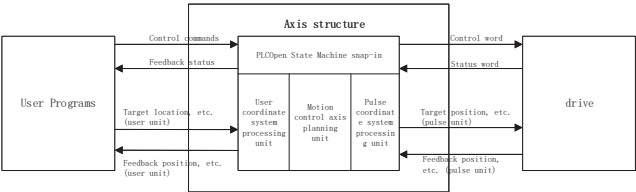
2.8 Motion Control

2.8.1 Introduction to Motion Control Axis

2.8.1.1 Overview

◆ Basic composition and logical control

In motion control systems, the object of motion control is referred to as an axis. Axis is the bridge connecting the driver and PLC instructions. The motion control axis is used to control EtherCAT bus drivers that comply with the 402 protocol, and can also control encoder axes, virtual axes, and so on. Inside the PLC, the basic structure and processing logic of the axis are as follows:



◆ Axis type

The supported axis types include bus servo axis, bus encoder axis, and virtual axis.

Axis Type	content
Bus servo axis	Axis controlled by EtherCAT slave servo drive. This axis is used to allocate to actual servo drives for use. The bus servo axis supports control of several basic modes such as position, speed, and origin regression.
Bus encoder shaft	Axis remote IO control using EtherCAT slave station This axis is used to allocate to actual EtherCAT remote IO counters for use.
virtual axis	Virtual bus servo axis, no need to associate servo drivers.

To describe the properties of the axis and control its motion, divide the axis into two parts:

Axis structure	function
Axis configuration parameters	Used to configure various parameters of the shaft, such as gear ratio, origin regression type, encoder mode, etc.
Axis control command	In the user program, execute axis motion control using MC motion control instructions. Axis control commands are divided into management classes (such as MC_Sower), motion classes (such as MC_SoveAbsolute), and status classes (MC_SeadStatus).

2.8.1.2 Axis Units

Two units are used in the axis structure, namely the user unit and the pulse unit.

• User unit

The measurement units used on the instruction side, such as millimeters, centimeters, angles, etc., are called user units and are usually represented by Unit. The user coordinate system is divided into linear coordinate system and rotational coordinate system according to different operating conditions. A linear coordinate system typically contains a zero point, with forward motion when the target position increases and backward motion when the target position decreases. The linear coordinate system can be set with positive and negative soft limits. The rotating coordinate system consists of a zero point and a rotation period. Within one cycle period, when the target position increases, it moves clockwise, and when the target position decreases, it moves counterclockwise.

• Pulse unit

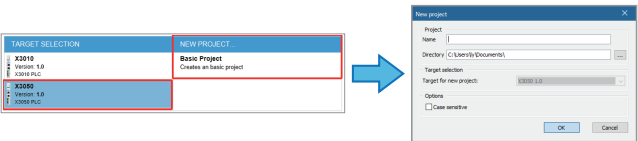
The unit of measurement used on the driver side, usually denoted as plus, is based on the number of pulses. Drivers typically include two parameters: the pulse zero point and the number of pulses for the motor to rotate one revolution encoder.

2.8.2 General steps for creating axis objects

2.8.2.1 New construction project

To control the axis correctly, first create a configuration according to the project needs, then set relevant parameters based on the working conditions, download the project and perform simple operations through online debugging to determine whether the parameter settings are reasonable and the hardware connections are reliable. Finally, write a PLC program to complete the overall control logic function. Here is an example.

When creating a project for X3050, it is necessary to select X3050. As shown in the following figure:



2.8.2.2 Creating Configuration

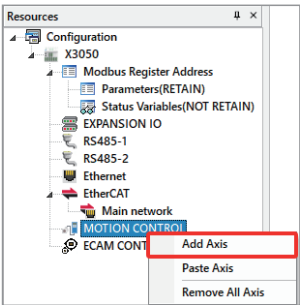
To control the motion of AX-CM55, it is necessary to configure a servo driver and a bus servo axis, and link them together.

2.8.2.2.1 Adding servo drivers

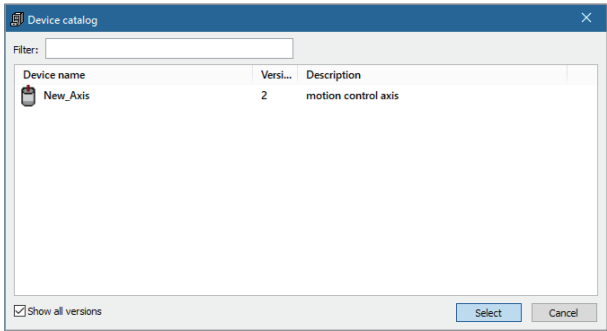
Add or scan a servo drive in EtherCAT. (See section 2.7 EtherCAT Communication for details)

2.8.2.2.2 Adding Process Axis Objects

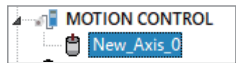
On the BonengLogislab page, find the resource window, right-click on the motion control process object, and click on add axis object, as shown in the following figure:



After clicking on the add axis object, a device directory window will pop up. Select the motion control axis, as shown in the following figure:

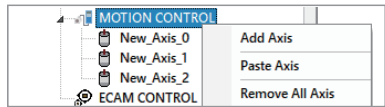


After the addition is complete, a New_Six-0 will appear under the motion control process object as shown in the following figure:

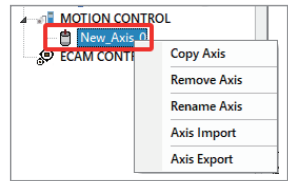


2.8.2.3 Operating Axis Objects

Right click on the motion control process object to bring up a menu where you can add, paste, and remove all axis objects, as shown in the following figure:

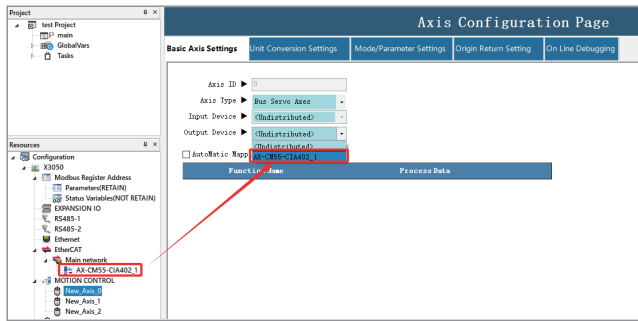


Right click on an axis object to copy, remove, rename, import, and export axis objects.



2.8.2.4 Associated servo drives

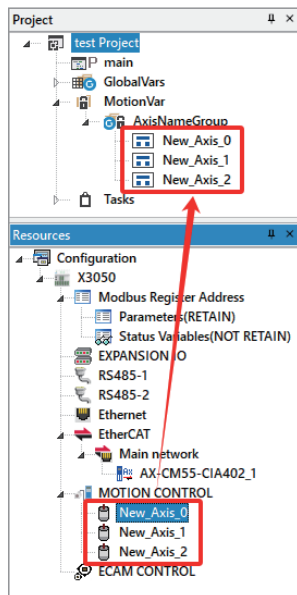
After adding the servo driver in the EtherCAT node, it can be found in the output device, as shown in the following figure:



By selecting this servo drive, you can associate it with the axis.

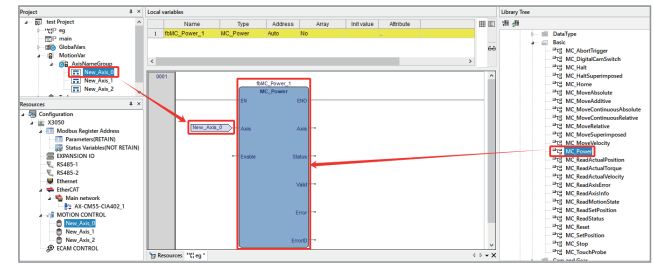
2.8.2.5 Save or Compile Generate

On the axis configuration page, in the unit conversion, mode/parameter settings, and origin return settings, set the parameters that need to be set, and then save and compile them. In the project window, a variable group will be generated. In this group, you can find that the variable name with the same name as that of the previous axis creation will be generated. In the PLC program, when calling the MC_ block, you need to call this variable to identify which axis it is. As shown in the following figure:



2.8.2.6 Writing Programs

In a ladder diagram program, if you want to enable the axis, you can view the library management wizard on the right and click to expand the MotionControl folder, click to expand the Basic folder, find the MC-Power function block, use the mouse to drag it onto the ladder diagram network, and then drag the variable that was just generated in the left project to the Axis pin to complete the axis enable program writing. Next, you can control the enable state of the axis by controlling the Enable pin. As shown in the following figure:

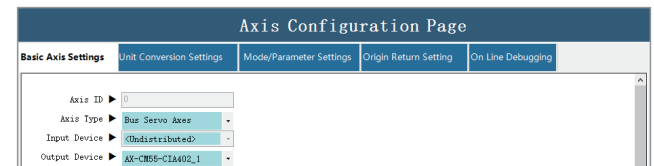


2.8.3 Axis Page Configuration

2.8.3.1 Basic Axis Settings

2.8.3.1.1 Bus servo axis

Click on the axis you just created and select the desired axis type from the axis types. Supports bus servo axis, bus encoder axis, and virtual axis. Choosing to keep is invalid. And compile an error.



When the selected axis type is bus servo axis, the input device cannot select it. When the shaft type is selected as encoder shaft, the output device cannot be selected. When selecting a bus servo axis, at least the control word, status word, and target position need to be included in the PDO of

the bound servo axis, otherwise there will be three error prompt icons.

Function Name	Process Data
⊖ Output (controller to device)	
Control Word*	!
Modes Of Operation	
Target Position*	!
Profile Velocity	
Profile Acceleration	
Profile Deceleration	
Target Velocity	
⊖ Input (device to controller)	
Error Code	
Status Word*	!
Modes of operation display	
Position actual value	
Velocity actual value	
Torque actual value	
Current actual value	
Digital inputs	

If not, you need to go to the ECAT configuration page to configure PDO.

2.8.3.1.2 Bus encoder shaft

The settings and servo type in the bus encoder. If you select the bus encoder axis, you need to configure the four data in the drop-down box as shown in the following figure: enable, direction, preset value, and current count value. If you do not configure it, the compilation will report an error.

Axis Configuration Page	
Basic Axis Settings	Unit Conversion Settings
Mode/Parameter Settings	
Axis ID ▶ 0	
Axis Type ▶ Bus Encoder Axis	
Input Device ▶ K4-200-1	
Output Device ▶ <Undistributed>	
☐ Automatic Mapping	
Function Name	Process Data
⊖ Output (controller to device)	
Enable*	<Undistributed> ❌
Direction*	<Undistributed> ❌
Preset value*	<Undistributed> ❌
⊖ Input (device to controller)	
Current count value*	<Undistributed> ❌

2.8.3.2 Unit Conversion

In the direction setting, when reverse is selected, the actual direction of motion of the axis will change.

Direction Setting

☐ Reverse

In the unit conversion page, it is necessary to design the number of command pulses for the motor or encoder to rotate one revolution. This value is related to the electronic gear ratio on the driver. When the electronic gear ratio on the driver is set to 1:1, the encoder resolution (encoder on the motor) should be filled in here. When the electronic gear ratio on the drive is not 1:1, the encoder resolution divided by the electronic gear ratio should be filled in here. As shown in the following figure:

The number of command pulses for a motor or encoder to rotate once= $\frac{\text{Encoder resolution}}{\text{electronic gear ratio}}$

Taking the 23 bit encoder of Boneng as an example, when the electronic gear ratio set on the driver is 1:1, 8388608 should be filled in here.

Travel Distance

Axis Type is Linear Mode

Number of motor revolutions n1: 1

Number of command pulses for one rotation of the motor/encoder: 8388608

PS: If the electronic gear ratio set on the drive is set to 1:1, then this is equivalent to the encoder resolution

Number of load revolutions n2: 1

The load travels one lead corresponds: 1000 Unit

So there is
$$\text{Pulse number} = \frac{\text{Encoder resolution}}{\text{Electronic gear ratio}} * \frac{\text{Motor speed } n1}{\text{Motor speed } n2} * \frac{\text{Distance traveled per unit lead}}{\text{Moving distance}}$$

Note: The unit lead corresponds to one rotation of the load.

As an example of the configuration shown above: When the target displacement given by the relative position command is 10, the actual number of pulses sent by the motion control axis is 83886080, and the motor rotates 10 times at this time. When the axis type is in rotation mode, the same applies to linear mode.

2.8.3.3 Parameter Mode Setting

The interface for setting parameter modes is as follows:

Encoder Mode Selection ▶	☒ Absolute mode ☐ Incremental mode	
Module Setting ▶	☒ Finite axis ☐ Modulo axis	
Software Limit ▶	☐ Negative limit enabled ☐ Positive limit enabled: Negative limit value: 0 Unit Forward limit value: 1000.0 Unit	
Software Error Response ▶	Limit deceleration: 1000 Unit/s ² Axis failure decelerates: 1000 Unit/s ²	
Threshold setting ▶	Follow the error threshold: 300 Unit Speed reaches threshold: 0.1 Unit	
Axis Speed Setting ▶	Maximum speed: 100 Unit/s Maximum acceleration: 1000 Unit/s ² Jog maximum speed: 100 Unit/s	
Torque Setting ▶	Positive torque maximum: 3000 0.1% Negative torque maximum: 3000 0.1%	
Options ▶	☒ The ErrorStop state is not entered after the collision limit	

2.8.3.3.1 Encoder Mode

- Incremental mode

The PLC side does not consider the increase in the number of turns caused by the overflow of the 32-bit counter of the servo drive encoder. When the PLC is powered off and restarted, it does not save the current position of the encoder. After powering on again, the current position of the shaft is calculated only based on the feedback of the servo driver for a single turn.

- Absolute mode

Consider the increase in the number of turns caused by the overflow of the 32-bit counter in the servo drive encoder on the PLC side. When the PLC is powered off and restarted, it saves the current position feedback from the encoder. After powering on again, during the initialization process, the encoder position and servo drive feedback position of the shaft stored inside the PLC are read to calculate the current absolute position of the shaft.

- Mode setting

According to the actual operating conditions, the motion control axis can be set as a linear axis and a modular axis.

- Limited axis

Linear mode is commonly used in devices with mechanical motion ranges in the X-Y linear coor-

dinate system.

In linear mode, it usually contains a zero point.

When the feedback position increases during motion, it indicates forward motion, otherwise it indicates reverse motion.

Allow the setting of forward and reverse software limits. When the software limit is enabled, the rear axle can only move within the limit range.

Absolute positioning method: When the target position is greater than the starting position, the distance between the target position and the starting position is moved forward; When the target position is smaller than the starting position, move the distance between the starting position and the target position in reverse.

Relative positioning method: When the target displacement is greater than 0, move the distance of the target displacement in the forward direction, and when the target displacement is less than 0, move the distance of the "target displacement" in the reverse direction.

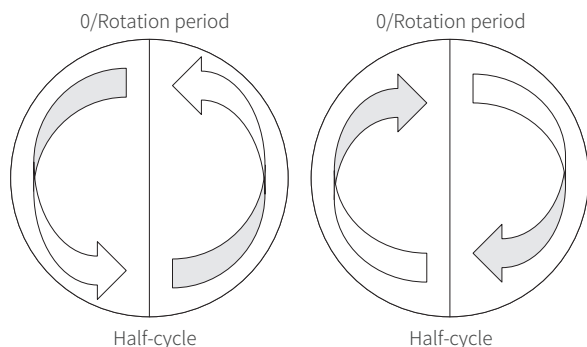
The processing method of speed commands in linear mode: if the target speed is greater than 0, it will move forward, and if the target speed is less than 0, it will move backward.

- Analog axis

Circular mode is a counter changing mode that repeats infinite counting within a set range. Usually used in turntables or scrolls.

The circular mode usually includes a zero point and a rotation period. The feedback position range of the circular counter is $0 \leq \text{feedback position} < \text{rotation period}$.

In circular mode, if the feedback position increases, it is considered clockwise motion, and if the feedback position decreases, it is considered counterclockwise motion.

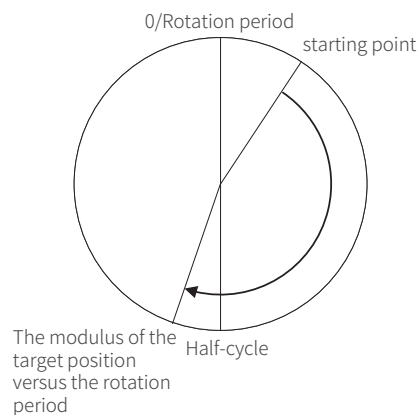


There is no soft limit under the modulus axis.

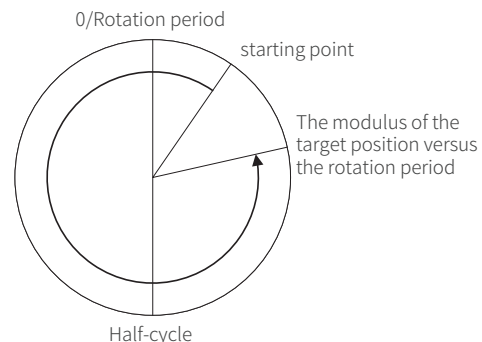
Relative positioning processing method: If the target displacement is greater than 0, move the distance of the target displacement clockwise; if the target displacement is less than 0, move the distance of the target displacement counterclockwise.

Absolute positioning processing method:

Forward: First, take the modulus of the target position to the rotation period, and then move the axis clockwise from the starting position to the target position.



Reverse: First, take the modulus of the rotation period for the target position, and then move the axis counterclockwise from the starting position to the target position.



Shortest distance: First, take the modulus of the rotation period of the target position to obtain the target position. Then calculate the displacement from the starting point clockwise to the target position. If the displacement is less than or equal to half a period, move clockwise. Otherwise, move counterclockwise to the target position.

Current direction: Move to the target position according to the direction of the axis's most recent motion. If it is the first power on, move forward to the target position. The speed command processing method in modular axis mode: if the target speed is greater than 0, it will move clockwise, and if the target speed is less than 0, it will move counterclockwise.

2.8.3.3.2 Software limit

Allow setting soft limit in linear mode. If the soft limit is effective, during the operation of the shaft, it will constantly detect the absolute position of the shaft when it undergoes a T-shaped deceleration from the current speed according to the set limit deceleration to 0. If the absolute position of the shaft exceeds the limit range, the shaft will execute the soft limit deceleration algorithm and interrupt the currently executing positioning or speed command. The soft limit is invalid under the origin regression and torque mode.

2.8.3.3.3 Axis Failure Deceleration

If the logic of the motion command itself fails during the operation of the shaft, causing the shaft

to have to switch to the erroneous stop state, the shaft will perform a T-shaped deceleration according to the deceleration set by the shaft fault deceleration, until it decelerates to 0 before entering the erroneous stop state.

2.8.3.3.4 Follow error

During the execution of positioning and speed commands, the servo drive actually operates in CSP (Periodic Synchronous Position) mode, and the planning of the position curve is completed on the PLC side. The PLC sends the target position to the servo driver through 0x607A, which drives the servo motor to move. The position of the motor encoder is fed back to the PLC through 0x6064. Due to the reasons of the servo driver and the motor itself, there is a difference between 0x607A and 0x6064. After converting the difference into user units, it is called tracking error. Set the maximum value of tracking error. If the absolute value of the tracking error of the axis exceeds the maximum value of the tracking error, the axis will report a fault of excessive tracking error and enter the error stop state.

2.8.3.3.5 Axis speed setting

Allow setting three parameters: maximum speed, maximum acceleration, and maximum jog speed. When the target speed, acceleration, deceleration and other parameters in the positioning command or speed command exceed the speed limit value, the relevant command reports a fault and the axis enters the Error Stop state. In the bus servo axis, the maximum speed will be converted into pulse units through unit conversion, and written into the object dictionary 0x607f of the servo drive through startup parameters.

2.8.3.3.6 Torque Setting

The torque setting is only used for the bus servo axis. If the target torque in the torque command exceeds the maximum torque, the command will

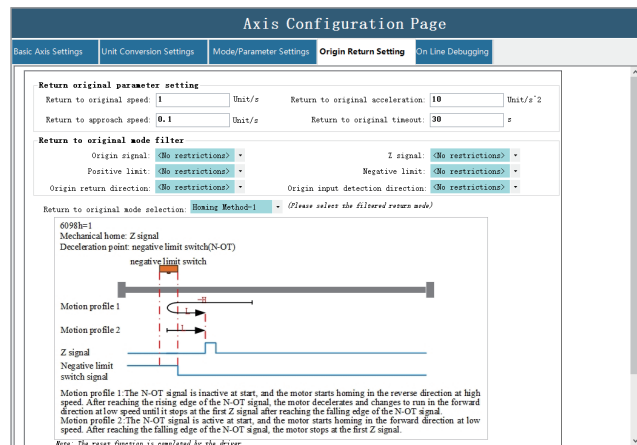
report a fault and the shaft will enter the error stop state. The forward torque limit value will be written into the object dictionary 0x60e0 of the servo drive through the startup parameter, and the reverse torque limit value will be written into the object dictionary 0x60e1 of the servo drive through the startup parameter.

2.8.3.3.7 Do not enter the Error Stop state after touching the limit position

When this option is selected, only the axis control command will report a fault when the motion control axis encounters the limit, and the motion control axis will no longer enter the Error Stop state. When this option is not selected, except for MC_Jog, if the limit is reached due to other motion control commands, the motion control axis enters the Error Stop state and the command reports an error.

2.8.3.4 Origin Return Setting

Supports the return to original mode supported by the Cia402 protocol from the 1st to the 35th. The origin return setting interface is as follows:



The parameters in the configuration interface are as follows:

Set pa-rameters	Description
Origin signal	Used to select whether to use the origin signal. When selecting unassigned, it is not used as a mandatory filtering condition. When not in use, the return to origin method that requires the use of the origin signal will be removed. When selecting to use, the return to origin method that does not support the origin signal will be removed.
Negative limit po-sition	Used to select whether to use the hardware left limit signal. When selecting unassigned, it is not used as a mandatory filtering condition. When not in use, the return to original mode that requires the use of negative limit signals will be removed. When selecting to use, the return to original mode that does not support negative limit signals will be removed.
Positive limit	Used to select whether to use hardware right limit signal. When selecting unassigned, it is not used as a mandatory filtering condition. When not in use, the return to original mode that requires the use of positive limit signals will be removed. When selecting to use, the return to original mode that does not support negative limit signals will be removed.
Z signal	Used to select whether to use motor z signal. When selecting unassigned, it is not used as a mandatory filtering condition. When choosing not to use, the return to original mode that requires the use of the z signal will be removed.

Set pa-rameters	Description
Origin regression direction	When selecting to use, the return to original mode that does not support z signals will be removed. Used to set the direction of motion at the beginning of the origin regression. Positive direction: When the limit (origin) signal input is invalid, the direction of motion is positive, otherwise it is reverse Negative direction: When the limit (origin) signal input is invalid, the direction of motion is negative, otherwise it is reversed
Origin regression detection direction	The direction of motion when reaching the origin signal. Forward: Stop when encountering the edge of the origin signal during forward motion. Negative direction: Stop at the edge of the origin signal during negative motion.
Origin Return List	The origin return method is set within the range of 1-35, and is written into the object dictionary 0x6098 in the form of a startup parameter.
Return to zero method	Set relative or absolute mode in zero return mode on the 35th, and write the object dictionary 0x60e6 in the form of startup parameters.
Origin return speed	Return speed to the origin, convert user units to pulse units, and write them into sub index 1 of object dictionary 0x6099 in the form of startup parameters.
Return to the origin and approach the velocity	The origin returns the approach velocity, converts the user unit to pulse unit, and writes it into the second sub index of the object dictionary 0x6099 in the form of a startup parameter.

Set parameters	Description
Origin Return Acceleration	Return the acceleration from the origin, convert the user unit to pulse unit, and write it into the object dictionary 0x609A in the form of startup parameters.
Origin return timeout	Unique to Boneng drives, writing from other manufacturers is invalid.

In practical use, the origin regression method is limited by several parameters including the origin signal, positive limit, negative limit, Z signal, origin regression direction, and origin regression detection direction. Then, the desired mode is selected through the origin return list options.

It should be noted that there are still multiple ways to return to the original after setting the return to original filtering conditions. At this time, it is necessary to select the appropriate return to original method through the origin return list. For example, when setting parameters as shown in the table below, two ways of returning to the original can be filtered out.

signal	price
Origin signal	use
Negative limit position	Not used
Positive limit	use
Z signal	use
Origin return direction	forward direction
Origin return detection direction	forward direction

Based on the above settings, two ways of returning to the original position can be selected: 8th and 10th. The difference between the two is whether the original input signal is valid when the return is completed.

2.8.3.5 Online Debugging

2.8.3.5.1 Online Monitoring

Obtain the status of the axis through PLC instructions and the online debugging interface in the background.

1. Obtain axis status through instruction.

Obtain the status of the axis through state class instructions such as MC_SeadStatus and MC_SeadPosition. Please refer to Chapter 3 for specific usage instructions.

2. Obtain axis status through online debugging interface.

Through the online debugging interface, the real-time movement status of each axis can be viewed, and the online monitoring interface of the axis is shown in the following figure.

Variables	Setting Values	Actual Values	State: disabled		
Position	55.07612228	55.07612228	Communication:Op		
Speed	0	0	Axis Error:0/16#0		
Acceleration	0	0	Servo Error:0/16#0		
Torque	0	0			
/	Hardware Positive Limit	Hardware Negative Limit	Origin Switch	Software Positive Limit	Software Negative Limit
State	False	False	False	False	False

The data available for monitoring is shown in the following table:

object	function
Set position	Target position (user unit) when PLC executes path planning
Set speed	Target speed (in user units) when PLC executes path planning
Set acceleration	Target acceleration (in user units) when PLC executes path planning
Set torque	Target torque (%) when PLC executes torque planning
Actual position	Current position of driver feedback (in user units)
Actual acceleration	Acceleration calculated based on actual speed (in user units)
Actual moment	Actual torque feedback from the driver (%)

object	function
status	Status of PLCOpen state machine: 0: PowerOff, go enable it 1: Error Stop, fault shutdown 2: Stopping, stop 3: StandStill, Enable 4: DiscreteMotion, Point Motion 5: Continuous Motion 7: Homing, Origin Regression 8: Synchronized Motion
Communicate	
Axis error	Internal error in motion control axis
Servo error	If it is a fault code for the local pulse axis, it corresponds to the value of 0x603F for the bus servo axis. Please refer to the manual of the driver used.
Hardware positive limit	Indicates whether the hardware positive limit input of the motion control axis is valid.
Hardware negative limit	Is the hardware negative limit input of the motion control axis valid.
Origin switch	Is the input of the origin switch for the motion control axis valid.
Software positive limit	Indicate whether the motion control axis has reached the software positive limit.
Software negative limit	Indicate whether the motion control axis has reached the software negative limit.

2.8.3.5.2 Online Control

Basic servo control functions such as enable, stop, jog, and position control can be achieved through online debugging. After confirming that the basic actions are normal, complex logic control can be achieved through motion control instructions.

Online debugging and PLC instruction control cannot be used simultaneously, with the following limitations:

It is worth noting that the PLC program will stop when the axis enters online control.

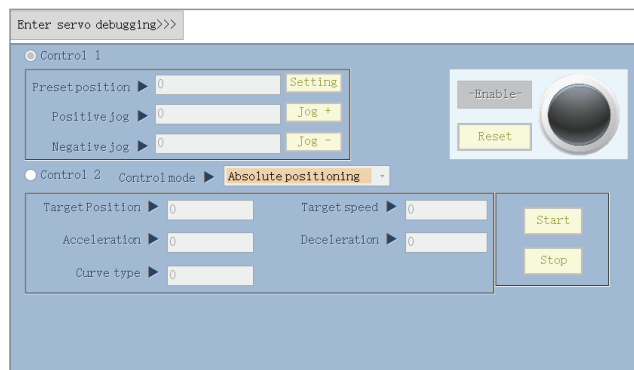
The functions of online control are shown in the table below:

Function points	describe
Enter online debugging mode	Put the axis into online debugging mode. After entering online debugging mode, PLC operation control commands will no longer be able to continue executing that 's ok.
enable	Similar to calling the MC-Power instruction to enable or disable the axis
reset	Used to reset shaft faults after shaft failure
cease	Stop the movement of the axis
Set current location	Set the current position of the axis
jog	Implement jog function
Motion type	For selecting which sports mode to use, the available options are: 1: Absolute positioning 2: Relative positioning 3: Continuous motion 5: Reciprocating motion
Relative positioning	Implement relative positioning function
Absolute positioning	Realize absolute positioning function
Continuous motion	Continuously moving at a certain speed
reciprocating motion	Realize reciprocating motion between two absolute positions
torque control	Implement torque control function
Origin regression	Perform origin regression action

Online debugging steps

1. Enter online debugging mode

As shown in the figure below, clicking the "Enter Online Debugging" button can put the axis into online debugging mode.



After receiving the command to enter online debugging, the backend will automatically stop the PLC program.

2. Basic operations

Enable: After entering online debugging mode, click the "Enable" button to enable the axis, and the execution effect is equivalent to the following command.

Jog: When the axis is in the Standstill state, clicking the "Jog+" button will move the axis in the positive direction according to the target speed set in the "Positive Jog" box, and release it to stop; Clicking on "Jog -" will cause the rear axle to move in the opposite direction according to the target speed set in the "Negative Jog" box, and release it to stop.

Reset: When the axis is in the error stop state, clicking the "Reset" button will attempt to reset the axis's fault. If the servo drive's fault is a non resettable fault, there may be a situation where the reset fails.

Stop: There are two stop buttons in the interface with the same function, both used to stop the movement of the axis. The deceleration used is the limit deceleration in the axis mode/parameter settings.

3. Sports mode

It needs to be enabled first, and clicking on 'run' will be invalid if not enabled.

- Absolute positioning

After selecting absolute positioning in the control mode, five parameters can be set: target position, target speed, acceleration, deceleration, and curve type. Clicking the start button will cause the rear axle to perform absolute positioning according to the above parameters.

- Relative positioning

After selecting the relative positioning control mode, five parameters can be set: target position, target speed, acceleration, deceleration, and curve type. Clicking the start button will cause the rear axle to perform relative positioning according to the above parameters.

- Continuous motion

After selecting the continuous motion control mode, four parameters can be set: target speed, acceleration, deceleration, and curve type. Clicking the start button will cause the rear axle to perform absolute positioning according to the above parameters.

- Reciprocating motion

After selecting the reciprocating motion control mode, two sets of target positions, two sets of target velocities, two sets of accelerations, two sets of decelerations, and two sets of curve types can be set. Clicking the start button will first position the rear axle to target position 1, and then position it to target position 2 according to the above parameters, and repeat the process in sequence.

- Origin regression

After setting the origin offset, origin return speed, origin return acceleration, and origin return approach speed, click the start button and the rear axle will perform origin regression according to the above parameters.

2.8.4 Overview of Axis Configuration/Status

Click on the motion control process node to view partial configuration information for all axes, including axis name, axis ID, axis type, bound servo name, axis communication status, and axis debugging status. As shown in the following figure:

Axis configuration/status overview					
Axis Name	Axis ID	Axis Type	Bound Servo	Axis Status	Debug Status
New_Axis_0	0	Servo	AP-C1440L1	Op	True
New_Axis_1	1	Servo	<Redistributed>	Unknown	False
New_Axis_2	2	Servo	<Redistributed>	Unknown	False

After entering the debugging state, the PCL program will stop running and can be started through three startup modes: cold, hot, and warm. When any startup mode is clicked, all axes will exit the servo debugging mode.

2.8.5 Fault Categories

Axis faults are divided into three categories: command faults, axis faults, and driver faults.

- Command failure is a fault caused by the MC axis control command itself, such as unreasonable command parameters, changes in the PLCOpen state machine of the axis during operation, resulting in command errors, etc. The fault code can be obtained by viewing the Error ID of the fault command. Axis fault is a fault reported by the axis itself, such as excessive tracking error. There are four ways to view axle fault codes. The first method can be obtained through the "Axis Fault" column in the "Online Debugging" interface of the backend. The second method is obtained through the AxiseError ID in the MC_SeadAxisError instruction.
- Driver failure refers to a malfunction of the EtherCAT bus driver or local pulse output shaft. To obtain the fault of EtherCAT bus driver, 0x603F must be configured in PDO mapping and associated with the axis. There are three ways to view drive faults. The first method can be obtained through the "Servo Error" column in the "Online Debugging" interface of the backend. The second method is obtained through the ServoError ID in the MC_SeadAxisError instruction.

2.9 Electronic cam

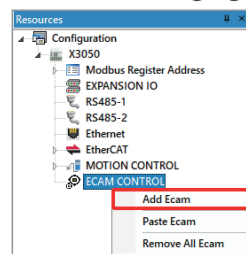
2.9.1 Introduction to Electronic Cams

- The electronic cam follows the movement of the main shaft from the shaft, and the position relationship between the main shaft and the slave shaft can be set through a cam table.
- Cam tables can be generated through software page configuration, and electronic cam tables can be used to establish up to 32 key point data.
- During the execution of the cam, it is not allowed to add, delete, or modify key points in the cam table. After modifying the key point data, the code needs to be downloaded again to take effect.

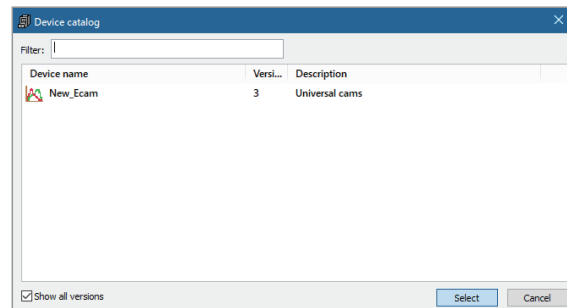
2.9.2 Software Configuration

2.9.2.1 Adding Cam

Find the node of the cam process object in the resource bar, right-click to add the cam as shown in the following figure:



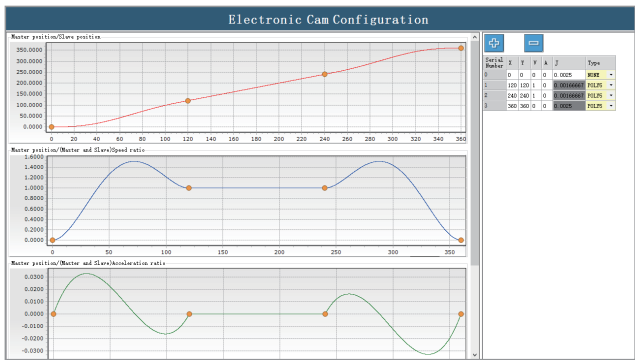
After clicking on 'Add Cam', select 'Universal Cam' from the pop-up device directory box. As shown in the following figure:



After adding, a new node will appear under the cam process object, usually the first one is (New_Cam-0). Click on this node to enter the design of the cam table data.

2.9.2.2 Design Cam

Click on the added universal cam to go to the electronic cam configuration page, which is mainly divided into two parts: the left side is the cam curve editing section, and the right side is the cam table section. The cam table is mainly used to store key point data. The cam curve can provide a more intuitive display of the cam's motion trajectory.



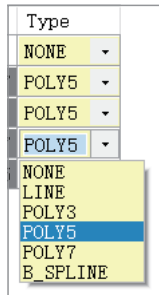
2.9.2.3 Edit Cam Table

Users can add, delete, and modify key data in the table according to their own needs.

Serial Number	X	Y	V	A	J	Type
0	0	0	0	0	0.0025	NONE
1	120	120	1	0	0.00166667	POLY5
2	240	240	1	0	0.00166667	POLY5
3	360	360	0	0	0.0025	POLY5

Click  to add a row of data above the row where the triangle is located (the first row cannot be added), click  to delete (the first row cannot be deleted, and the last row cannot be deleted), the row where the triangle symbol is located. The modifiable data in the table consists of 5 columns, namely X (spindle position), Y (slave position), V (master-slave velocity ratio), A (master-slave acceleration ratio), and J (master-slave plus acceleration ratio).

tion), V (master-slave velocity ratio), A (master-slave acceleration ratio), J (master-slave plus acceleration ratio), and Type (curve type). In the column of curve types, you can select the type of curve between every two keypoints, supporting straight, cubic, cubic, and cubic, as shown in the following figure:

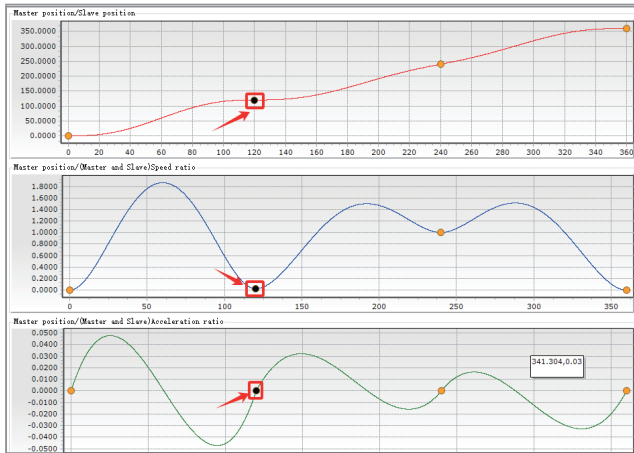


2.9.2.4 Editing Cam Shapes

The operation of cam curves allows dragging of key points. For the master-slave axis position image curve, it is allowed to drag the upper points up and down, left and right, while the speed, acceleration, and acceleration curves only allow dragging up and down. The specific rules are shown in the table below:

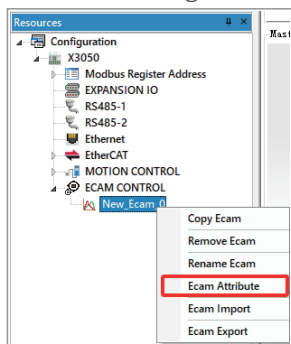
curve type graph	LINE	POLY3	POLY5	POLY7
Master slave position	contin- uous	contin- uous	contin- uous	contin- uous
Master-slave speed ratio	nonse- quence	contin- uous	contin- uous	contin- uous
Master-slave acceleration ratio	nonse- quence	nonse- quence	contin- uous	contin- uous
Master-slave acceleration ratio	nonse- quence	nonse- quence	nonse- quence	contin- uous

When the selected curve type is discontinuous, the key points of the curve cannot be dragged to move. When the mouse hovers over a key point on a cam curve, it will turn black and highlight the data in the cam table. And the points of the remaining curves at the same spindle position will also turn black.

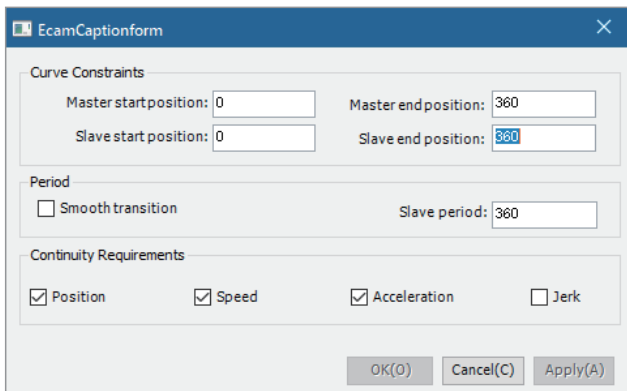


2.9.2.5 Cam Attribute Configuration

Right click on the added cam and select the cam attribute configuration



After clicking, as shown in the following figure:



Here, you can set some related constraints for key points.

The starting position of the spindle will change the spindle position of the first key point after setting.

The end position of the spindle will change the spindle position of the last key point after setting.

Starting from the axis position, it is not allowed to be less than this value after setting from the axis position.

From the end position of the axis, it is not allowed to exceed this value after setting from the axis position.

Smooth transition, once checked, the main axis position of the first key point will be equal to the main axis position of the last key point.

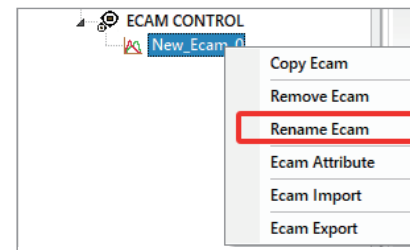
2.9.2.6 Import and Export

Right click on the cam device and select Export Cam File. The exported cam file is saved in CSV format.

Right click on the cam device and select Import Cam File. The imported cam file must be in CSV format and meet the import requirements. Otherwise, the import will fail.

2.9.2.7 Heavy Duty Cam Object

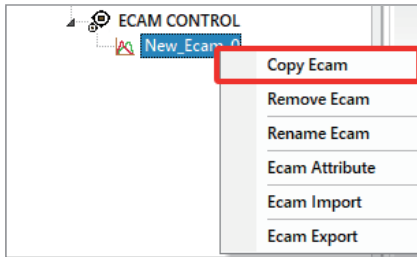
Right click on the added cam device and select Rename Cam to open the editing mode and edit the name of the cam.



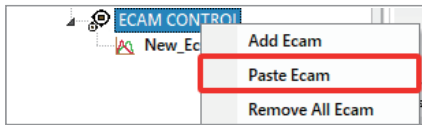
2.9.2.8 Copy and paste cam

Right click on the edited cam with the mouse, and when the user needs multiple identical cams, they can use the copy and paste function

Right click on the added and edited cam and select 'Copy Cam' as shown in the following figure:

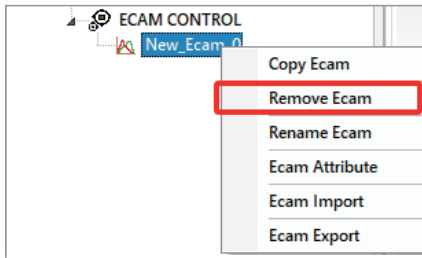


After copying the cam, right-click on the cam process object and paste the cam to obtain two cams with the same data.



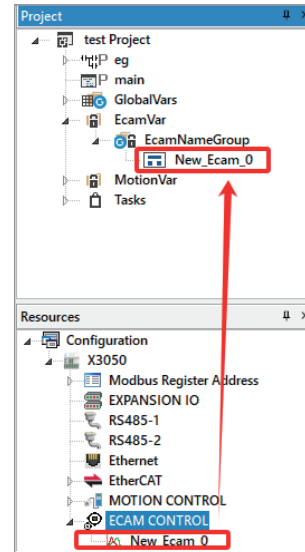
2.9.2.9 Removing the cam

Right click on the unwanted cam and select 'Remove Cam'.



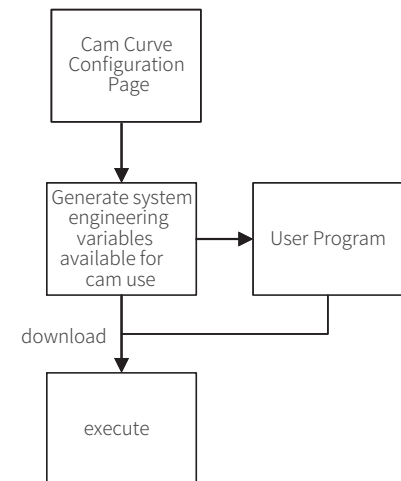
2.9.2.10 Generating Cam Variables

After editing the cam table data, click save. BonengLogicLab will directly generate a variable with the same name in the engineering window based on the name of the added cam device. This variable is a global variable that is easy to call directly in the function block. As shown in the following figure:



2.9.3 Use of electronic cam

- After the user creates the cam table according to their needs, they can call the corresponding function block for programming.

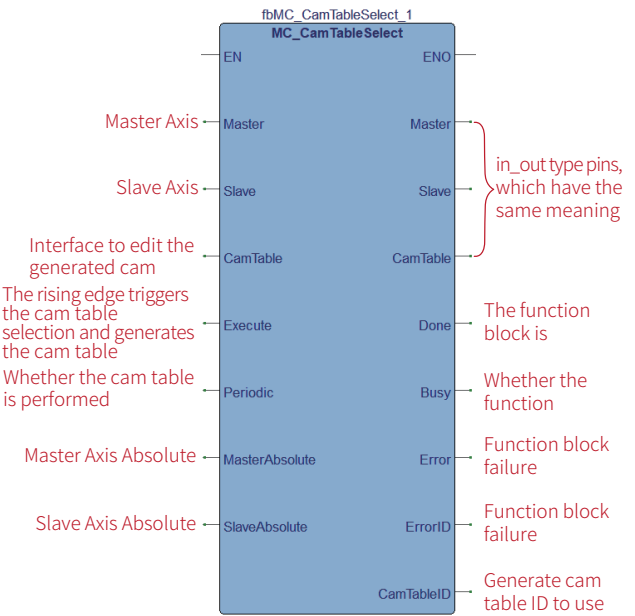


- The cam related functional blocks include MC_CamTableSelect, MC_CamIn, MC_CamOut.
- MC_CamTableSelect is used to select the cam table (MC_Cam-REF) and generate cam IDs (MC_CamID) that can be called by MC_CamIn.

- After MC_CamIn is triggered, it transitions from the shaft to the main shaft and enters cam synchronization, while MC_CamOut is used to contact the passive relationship between the shaft and the main shaft, and exits cam synchronization from the shaft after triggering.

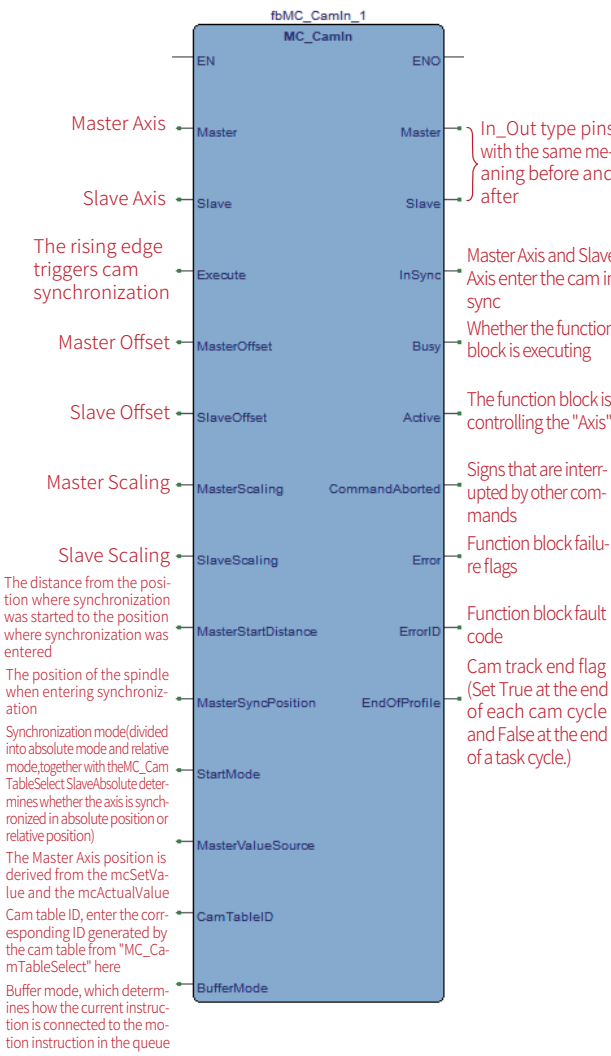
2.9.3.1 MC_CamTableSelect

The MC_CamTableSelect function block converts the CamTable generated by editing the cam table interface into a CamTableID that can be used by MC_CamIn.



2.9.3.2 MC_CamIn

MC_CamIn engages the master and slave shafts, controlling the movement of the slave shaft according to the corresponding relationship between the main and slave shaft positions on the cam table.



2.9.3.2.1 Master-slave axis scaling and offset

Users can adjust the input of the MC_CamIn function block (MasterOffset, SlaveOffset, MasterScaling, SlaveScaling) according to their needs to achieve the scaling and offset of the master and slave axes of the cam curve.

2.9.3.2.2 Transition interval

- To enter synchronization from the axis following the main axis, a transition interval is required,

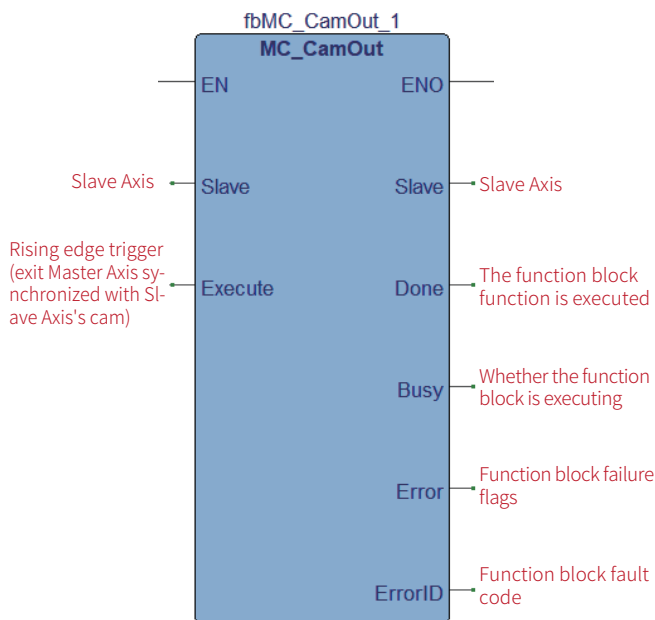
and users can customize the position parameters of the transition interval (MasterStartDistance, MasterSyncPosition).

- MasterSyncPosition specifies the position of the main axis at the moment of synchronization, and the position relationship between the master and slave axes on the cam table is determined. Therefore, the synchronization position of the slave axis is also determined.
- MasterStartDistance specifies the length of the transition interval before entering synchronization.

2.9.3.3 MC_CamOut

The MC_CamOut function block cancels the engagement relationship between the slave axis and the spindle.

After MC_CamOut is triggered, the axis will continue to run at the speed at which it was triggered, and the axis state machine will transition from synchronous motion to continuous motion.



2.10 Electronic Gear

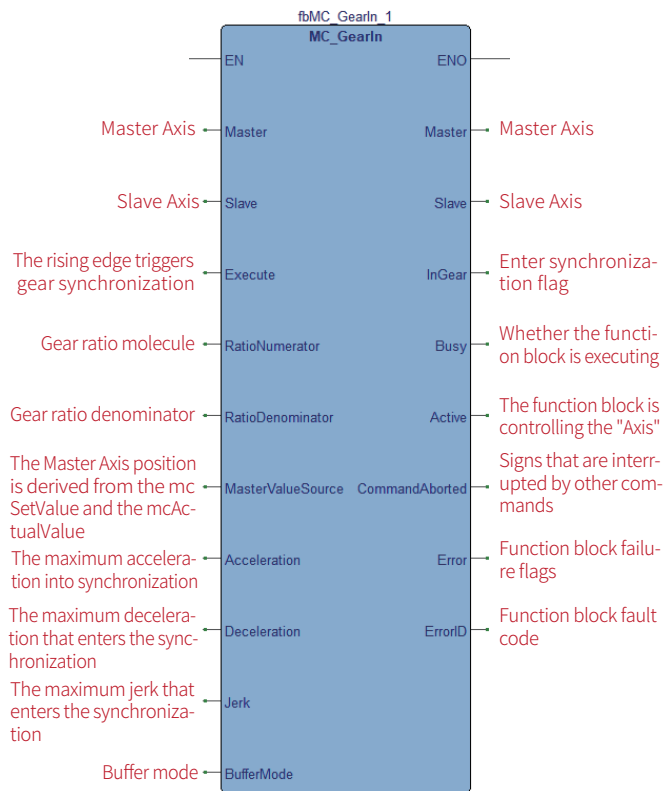
2.10.1 Gear Action

- The main and secondary shaft types required for gear action are as follows.
- The spindle can be a bus servo axis, a virtual axis, or a bus encoder axis.
- From axis: bus servo axis and virtual axis.
- The gear action is synchronized through the MC_GearIn command, and the synchronization between the slave shaft and the main shaft is released through the MC_GearOut command or MC_Stop command.
- MC_GearIn gear synchronization is divided into two stages, the stage before synchronization is called the transition stage, and the stage after synchronization is called the synchronization stage.
- During the gear transition phase, the target speed is obtained by multiplying the main shaft speed by the gear ratio based on the acceleration and deceleration given by the user on the function block, and catching up with the main shaft to enter synchronization.
- In the gear synchronization stage, the target speed is obtained by multiplying the spindle speed by the gear ratio from the shaft speed, and the synchronization action is performed.
- The gear ratio is positive and moves in the same direction as the main axis from the shaft.

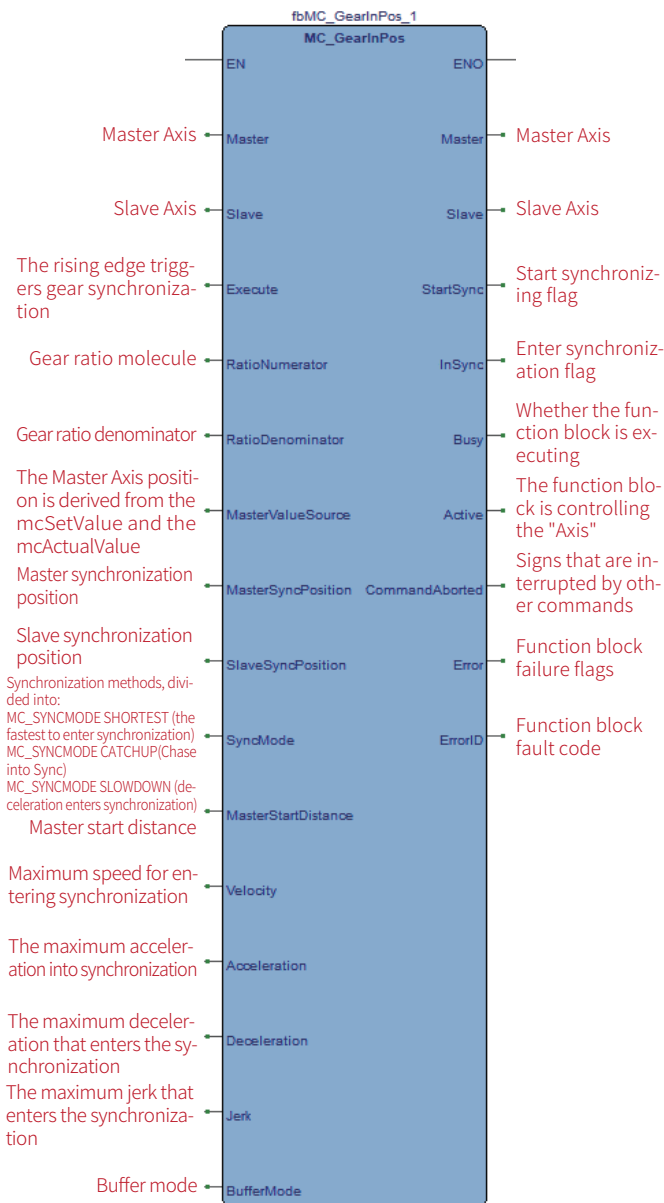
2.10.2 Use of electronic gears

2.10.2.1 MC_GearIn

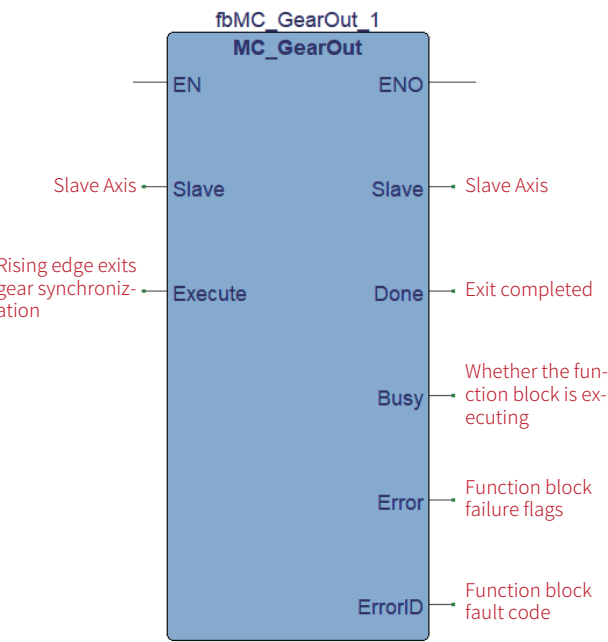
The MC_GearIn function block controls the ratio of the slave axis speed to the spindle speed that matches the set value.



2.10.2.2 MC_GearInPos



2.10.2.3 MC_GearOut



2.11 Axes Group

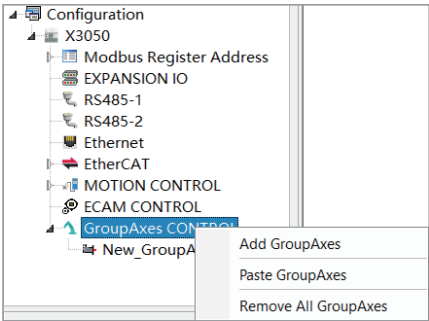
2.11.1 Introduction to Axes group

- An axis group allows a robot mechanism composed of multiple axes to move, capable of performing point-to-point, linear, circular, and linear-circular transitional movements.
- The robot configuration composed of an axis group can be generated through software page configuration. Currently, robots such as three-coordinate robots can be configured, and axes related to the robot can also be configured.
- Axes associated with the robot can only be controlled by the axis group and cannot be controlled individually.

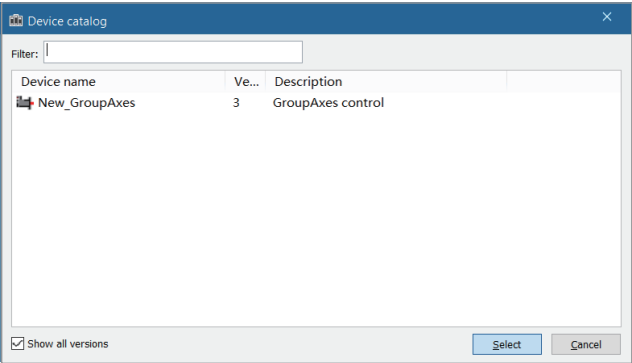
2.11.2 Software Configuration

2.11.2.1 Adding an Axis group

In the resource panel, locate the node for the shaft assembly process object, right-click to add a shaft assembly as shown in the figure below:



After clicking Add Axis Group, select Axis Group Control in the pop-up device directory box. As shown in the figure below:



After adding it, a new node will appear under the axis group process object, usually the first one is (New_GroupAxes_0). Click on this node to enter the design of the cam table data.

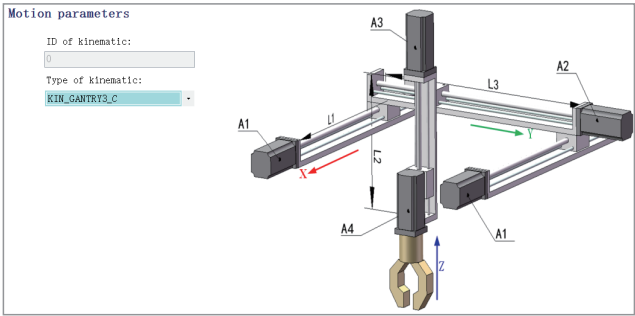
2.11.2.2 Axes Group Configuration

Click on the added axis group to go to the axis group configuration page. This page is mainly divided into six sections: basic parameters, interconnections, geometric structure, dynamics, object coordinate system, and tool coordinate system.

Axes Group Configuration					
Basic parameters	Interconnection	Geometry	Kinematics	Tool Coordinate System	Workpiece Coordinate System

1. Basic Parameter Configuration

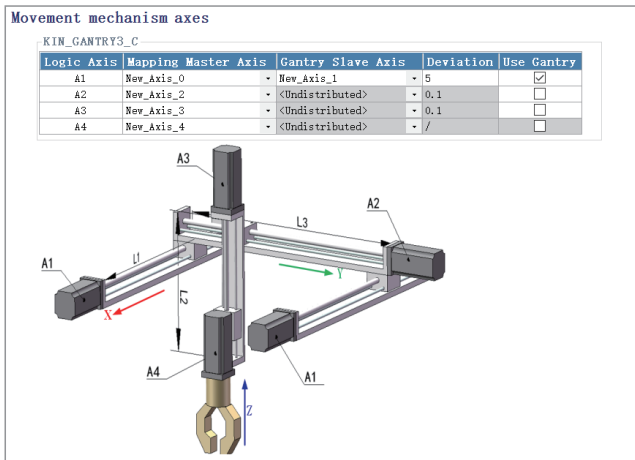
The configuration interface is as follows:



The motion mechanism ID is automatically generated. You can choose the type of motion mechanism, and after selecting it, the corresponding configuration diagram will appear on the right.

2. Interconnection Configuration

The configuration interface is as follows:



This interface mainly configures which servo axes correspond to the joint motors in the mechanism type. The servo axes need to have their axis process objects configured in advance. Additionally, gantry slave axes can be configured.

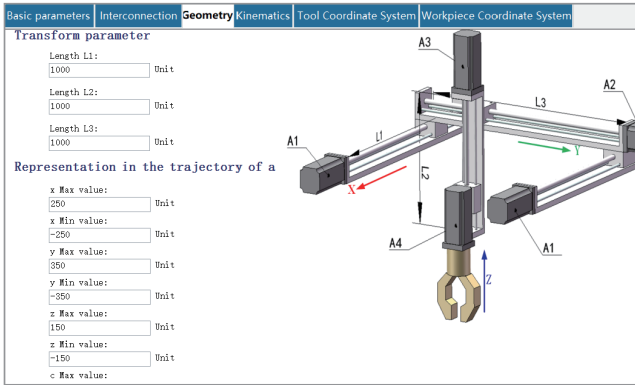
In the logical axes, A1 to A3 correspond to the positions of the joint motors in the configuration diagram. The links under interconnected spindles correspond to the relevant servo axes. To configure the servo axes under the gantry slave axis, you need to click to enable the gantry. Checking the

gantry slave axis will also change the corresponding configuration diagram.

Deviation represents the allowable positional deviation between the gantry slave axis and the main axis. If the deviation exceeds this value, an error will be reported.

3. Geometric Structure Configuration

This section mainly configures the geometric parameters of the mechanism. The configuration interface is shown below, with the length unit in millimeters (mm).

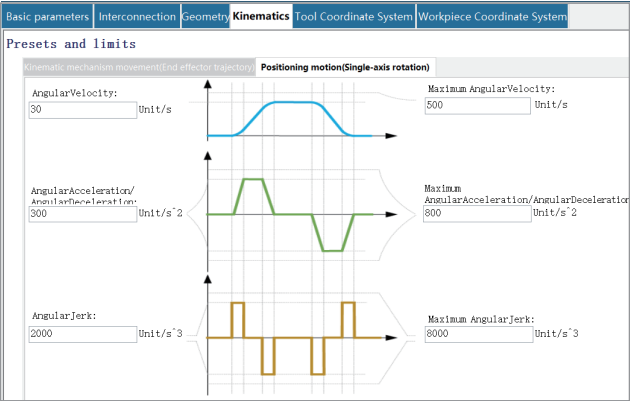


In the above figure, L1, L2, and L3 are the lengths corresponding to the configurations shown. The maximum values of X, Y, Z, and C are the farthest positions the configuration can reach; if input parameters exceed these positions, the system will report an error.

4. Kinematic Configuration

This section mainly configures the motion speed, acceleration, and other parameters related to the axis group. There are two configuration options: motion mechanism configuration and positioning motion configuration.

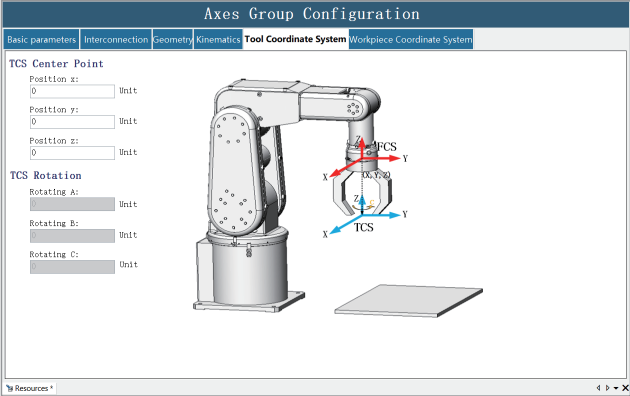
The motion mechanism configuration interface is shown below:



The angular velocity, angular acceleration/deceleration, and angular jerk during positioning motion represent the maximum kinematic limits for point-to-point motion, as well as the kinematic limits of the fourth rotational axis. The maximum angular velocity, maximum angular acceleration/deceleration, and maximum angular jerk represent the limits during the actual motion process. The angular velocity, angular acceleration, and angular jerk during actual motion may experience some fluctuations.

5. Tool Coordinate System Configuration

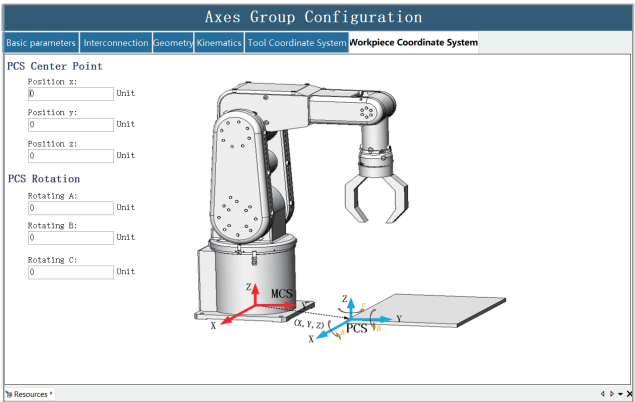
This section mainly configures the positional transformation relationship of the tool coordinate system relative to the end-effector coordinate system. The configuration interface is shown below:



The three parameters under the TCS center point represent the offset of the tool coordinate system relative to the end-effector coordinate system in the X, Y, and Z directions. The three parameters under TCS rotation represent the rotation of the tool coordinate system around the axes of the end-effector coordinate system.

6. Workpiece Coordinate System Configuration

the main focus is on configuring the position transformation relationship of the workpiece coordinate system relative to the base coordinate system. The configuration interface is shown below:



The three parameters under the PCS center point represent the offset of the workpiece coordinate system relative to the base coordinate system in the X, Y, and Z directions. The three parameters under PCS rotation represent the rotation of the workpiece coordinate system around the axes of the base coordinate system.

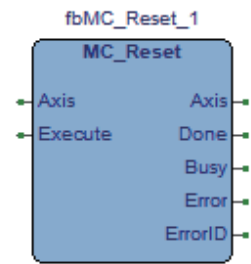
2.11.2.3 Using Axis Groups

After the user creates an axis group robot according to their needs, they can call the corresponding function blocks for programming. The specific usage of axis groups is as follows:

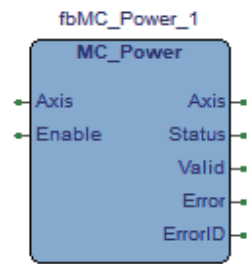
First, configure the relevant parameters of the axis group in the configuration interface. Once completed, an axis group process object can be obtained.

GroupAxes CONTROL
New_GroupAxes_0

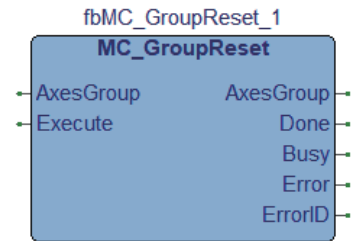
Subsequently, we call functional blocks within the program framework to control the axis group devices. The specific implementation logic is as follows: First, it is necessary to call the single-axis functional block to perform a reset operation on the single axis.



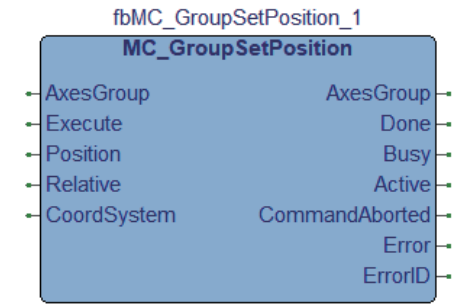
Then call the single-axis enable function block to enable power to the single axis.



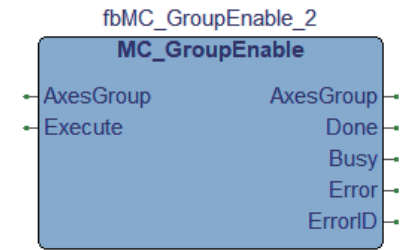
When calling the function blocks related to the axis group again, first perform a reset operation on the axis group.



Then call the command to set the current position to determine the position of each axis:



Then call the axis group enable module to enable the axis group:



After completing the above work, you can call the relevant planning commands for straight lines, arcs, and point-to-point movements to control the motion of the axis group.

3 Instruction for Function Blocks

3.1 BonengControlLib

3.1.1 BN_EtherCAT

3.1.1.1 Precautions for use

- 1) The function block of the BN_EtherCAT library is used to control EtherCAT slaves. When using the X3050 controller and establishing an axis, if the axis selects the corresponding EtherCAT slave as the output device, the slave cannot be used by the BN_EtherCAT library function block anymore. Otherwise, an error message FBError Code: 32 (16 # 0020) will be reported
- 2) The BN_EtherCAT function block is divided into two categories, which control servo drives and variable frequency drives respectively, as shown in the table.

They are corresponding to the servo (AX, AN) and frequency converter (AX, AD) in Boneng Drive products. Among them, the AX series product A04.14 can switch between servo mode and frequency conversion mode.

A04.14	Drive mode selection	1: Servo mode
A04.16	Servo motor number	0: Frequency conversion mode 1: Servo mode

If choosing the inverter mode, C06.04 needs to be modified to the BoNeng Freq function block, while users can use the inverter function block in Boneng-Logiclab.

If choosing the servo mode, C06.04 needs to be modified to none, and the servo function block can be used at this time

C06.04	Boneng Controller function block	0: None
C06.05	Software version number	0: None 1: BoNeng Freq function

	Function Block	Function	AN/AX PDO Configuration	BonengDrive-Soft100 Configuration
Inverter Function Block	BN_EtherCAT_INV_Speed	Inverter Speed Control Mode	Output: 1704 Input: 1B04 (1702 and 1B02 if AD products are selected)	A04.14 Select: Inverter mode C06.04 Select: Inverter mode function block
	BN_EtherCAT_INV_Torque	Inverter Torque Control Mode		
Servo Function Block	BN_EtherCAT_PP_PV	Multiple functions in servo mode: 1:Relative position(PP) 2:Absolute position(PP) 3:Speed mode(PV) 4:Active return to zero(HM) 5:Set the current position to zero(HM) 6:**Not yet 7:Speed jog (PP) 8:Position jog (PP)	Output: 1703 Input: 1B03	A04.14 Select: Servo mode C06.04 Select: None
	BN_EtherCAT_PV_Speed	Servo mode, Speed control (PV)	Output: 1702 Input: 1B03	
	BN_EtherCAT_PT_Torque	Servo Mode, Torque Control (PT)		

PS:(1) The INV (Inverter) label can be used to confirm that the function block is an inverter function block

(2) PP, PV, and PT belong to servo modes, and the function block can be confirmed as a servo function block

3.1.1.2 BN_EtherCAT_INV_Speed_V02

3.1.1.2.1 RPDO Configuration

Open the output page, cancel the synchronization management output configuration SM2 for other PDOs, and add 1704 to the SM2. When using Boneng EtherCAT driver, you do not need to add parameters to it, as shown in the following figure by default:

Index	SubIdx	PDO Entry Name	Type	Bit len	PLC var	Comment
0x4000 1		R-PZD1	UINT	16		
0x4000 2		R-PZD2	UINT	16		
0x4000 3		R-PZD3	UINT	16		
0x4000 4		R-PZD4	UINT	16		
0x4000 5		R-PZD5	UINT	16		
0x4000 6		R-PZD6	UINT	16		
0x4000 7		R-PZD7	UINT	16		
0x4000 8		R-PZD8	UINT	16		
0x4000 9		R-PZD9	UINT	16		
0x4000 10		R-PZD10	UINT	16		
0x4000 11		R-PZD11	UINT	16		
0x4000 12		R-PZD12	UINT	16		
0x4000 13		R-PZD13	UINT	16		
0x4000 14		R-PZD14	UINT	16		
0x4000 15		R-PZD15	UINT	16		

Open the input page, select 1B04 to configure SM3 for the synchronous management input. When using Boneng EtherCAT servo drive, you do not need to add parameters to it, the default is as shown in the following figure:

Index	SubIdx	PDO Entry Name	Type	Bit len	PLC var	Comment
0x1000 1		T-PZD1	UINT	16		
0x1000 2		T-PZD2	UINT	16		
0x1000 3		T-PZD3	UINT	16		
0x1000 4		T-PZD4	UINT	16		
0x1000 5		T-PZD5	UINT	16		
0x1000 6		T-PZD6	UINT	16		
0x1000 7		T-PZD7	UINT	16		
0x1000 8		T-PZD8	UINT	16		
0x1000 9		T-PZD9	UINT	16		
0x1000 10		T-PZD10	UINT	16		
0x1000 11		T-PZD11	UINT	16		
0x1000 12		T-PZD12	UINT	16		
0x1000 13		T-PZD13	UINT	16		
0x1000 14		T-PZD14	UINT	16		

3.1.1.2.2 Pin Description

BN EtherCAT_INV_SPEED_V02 function block pins are described in the following table:

Name	Data type	Description	Note
AxisIndex	UINT	EtherCat Axis ID; 0–15	\
bEnable	BOOL	Enable operation; 0: The Axis stops, 1: The Axis runs	\
bStopMode	BOOL	Stop mode; 0: Deceleration shutdown, 1: Free shutdown	\
bQuickStop	BOOL	Quick stop; 0:Quick stop is valid, 1:Quick stop is invalid	\
bMotorRotation	BOOL	Motor rotation direction; 0: Forward, 1: Reversed	\
bFaultReset	BOOL	Fault reset; 0: The reset is invalid, and 1: The reset is valid	Rising edge triggering
TargetSpeed	INT	Target Speed, The speed benchmark value is Ref-Speed	\
RefSpeed	UINT	Speed reference value	The initial value is required, otherwise an error is reported
TorqueUpLimit	INT	The upper torque limit of 1000 corresponds to 100.0%, and 100.0% corresponds to the rated torque of the motor	\
TorqueLowLimit	INT	The lower torque limit of 1000 corresponds to 100.0%, and 100.0% corresponds to the rated torque of the motor	\
PZD3	WORD	User-defined PZD3	\
bReadyOK	BOOL	Axis ready status; 0: The Axis is not ready, 1: The Axis is ready	\

Name	Data type	Description	Note
bRunState	BOOL	Operational status; 0: Down, 1: Running	\
StateWord	WORD	Status word	\
bFault	BOOL	Fault status; 0: No fault, 1: Faulty	\
FBErrorCode	WORD	Function block fault code	\
AxisErrorCode	WORD	Drive fault code	\
ActSpeed	INT	Speed of feedback; The benchmark value is Ref-Speed	\
ActCurrent	UINT	Output current; 4096 corresponds to the rated current of the motor	\
ActTorque	INT	Output torque; 4096 corresponds to the rated torque of the motor	\
PZD6	WORD	User-defined PZD6	\

Module Introduction

- 1) The name of the axisIndex axis is 0-15 according to the topology, that is, the 0 axis directly connected to the Boneng controller EtherCat, the 1 axis connected to the 0 axis, and so on.
- 2) QuickStop should be set to 1, otherwise the drive will be blocked for boot;
- 3) RefSpeed, which cannot be 0; When this value is 5000, the TargetSpeed value is 0-5000, which corresponds to the motor 0-50HZ.

3.1.1.2.3 Description of function block Errorcode

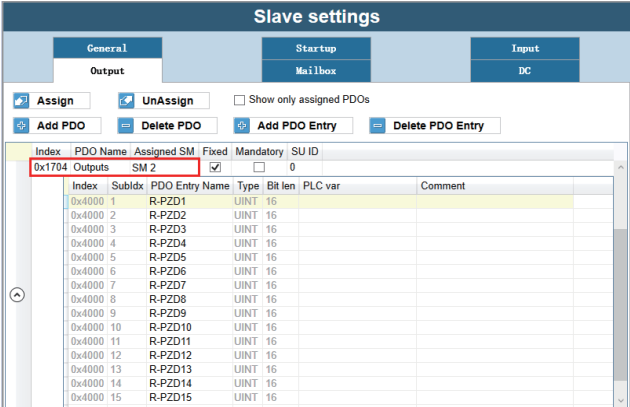
FBErrorCode	Description
16#0000	No Error;
16#0001	Not entering the OP;
16#0002	The AxisIndex is out of range, or does not enter the OP;
16#0100	The reference speed is not set;

FBErrorCode	Description
16#000A	The PDO configuration is incorrect or the address of the StatusWord4100_01 / ControlWord4000_01 cannot be obtained;
16#000B	The PDO configuration is incorrect or the address of the ActVelocity4100_02 / TargetVelocity4000_02 cannot be obtained;
16#000C	The PDO is not configured correctly or cannot obtain the address of the ActCurrent4100_03 / RPZD3_4000_03;
16#000D	The PDO is not configured correctly or cannot obtain the address of the ActTorque4100_04 / TorqueUpLimit_4000_04;
16#000E	The PDO configuration is incorrect or the address of the DriveError4100_05 / TorqueLowLimit_4000_05 cannot be obtained;
16#000F	The PDO is incorrectly configured or cannot obtain the address of the TPZD6_4100_06;
16#0012	The PDO configuration is incorrect or the address of the OperationMode_4000_09 cannot be obtained;
16#0020	The current axis is already in use

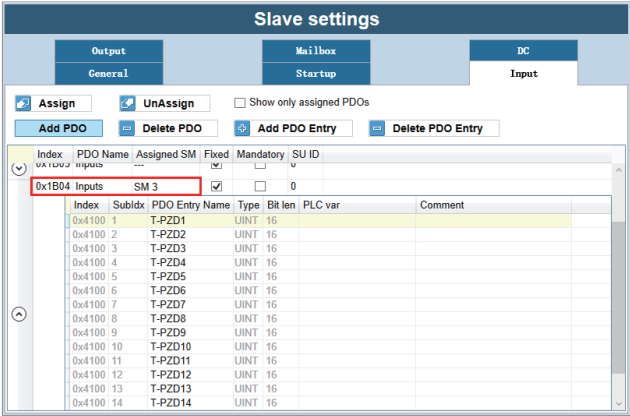
3.1.1.3 BN_EtherCAT_INV_Torque_V02

3.1.1.3.1 RPDO Configuration

Open the output page, cancel the synchronization management output configuration SM2 for other PDOs, and add 1704 to the synchronization snap-in SM2. When using the Boneng EtherCAT driver, you do not need to add parameters to it, as shown in the following figure by default:



Open the input page, select 1B04 to configure SM3 for the synchronous management input. When using the Boneng EtherCAT servo drive, you do not need to add parameters to it, the default is as shown in the following figure:



3.1.1.3.2 Pin Description

The pins of BN_EtherCAT_INV_Torque_V02 function blocks are described in the following table:

Name	Range/Me- aning	Data type	Unit	Description
AxisIn- dex	EtherCat Axis Index	UINT	\	0--15
bEnable	1:Enable 0:Not en- abled	BOOL	\	Enable operation

Name	Range/Me- aning	Data type	Unit	Description
bMotor- Rotation	1: Reverse 0: Forward	BOOL	\	Motor rotation direc- tion, 0: forward, 1: reverse
bFault Reset	1: Fault re- set 0: No Error			
Target Torque	-32768 --- 32767	INT	\	The target torque, 1000 corresponds to 100.0%, and 100.0% corresponds to the rated torque of the motor
Speed UpLimit	-32768 --- 32767	INT	\	The upper speed limit, 1000 corresponds to 100.0%, and 100.0% corresponds to the rated speed of the motor
Speed LowLimit	-32768 --- 32767	INT	\	The lower speed limit, 1000 corresponds to 100.0%, and 100.0% corresponds to the rated speed of the motor
PZD3	User-defi- ned PZD3	WORD	\	\
bReady OK	1: The boot is ready 0: The de- vice is not re- ady to boot	BOOL	\	Boot ready status, 0: boot not ready, 1: boot ready
bRunSt- ate	1: Run 0: Stop			
State Word		WORD	\	Status word
bFault	1: Faulty 0: No error	BOOL	\	Fault status, 0: No Fault, 1: Faulty

Name	Range/Me- aning	Data type	Unit	Description
FBError Code	Function block error code	WORD	\	Function block fault code
AxisError Code	Axis error code	WORD	\	Drive failure code
ActSpe- ed	Actual spe- ed	INT	Number of com- mand pulses	Feed back speed, based on RefSpeed
ActCurr- ent	Actual Curr- ent	UINT	The nu- mer of comm- and pulses	The output current, 4096 corresponds to the rated current of the motor
ActTorq- ue	Actual torq- ue	INT	The per- centage of the mo- tor's rated torque	Output torque, 4096 corresponds to the rated torque of the motor, with symbols
PZD6	User-defin- ed PZD6	WORD	\	\

The name of the axisIndex axis is 0-15 according to the topology, that is, the 0 axis directly connected to the Boneng controller EtherCat, the 1 axis connected to the 0 axis, and so on.

3.1.1.3.3 Description of function block errorcode

FBErrorCode	Description
16#0000	Function block no error ;
16#0001	Not entering the OP;
16#0002	The AxisIndex is out of range, or does not enter the OP;
16#000A	The PDO configuration is incorrect or the address of the Sta- tusWord4100_01 / Control- Word4000_01 cannot be obtained;

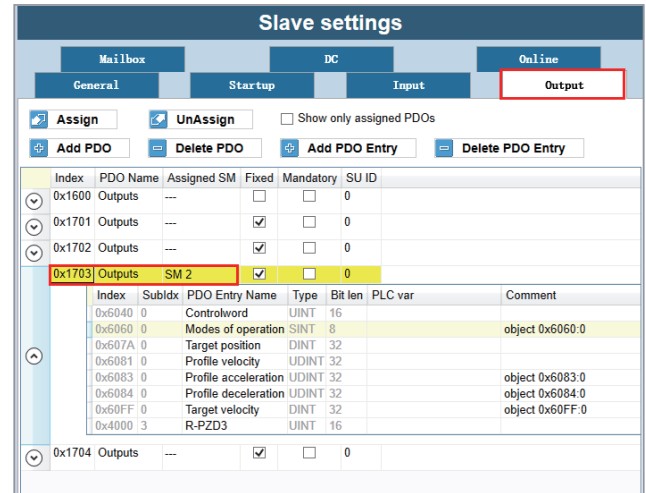
FBErrorCode	Description
16#000B	The PDO configuration is incorrect or the address of the ActVelocity4100_02 / TargetVelocity4000_02 cannot be obtained;
16#000C	The PDO is not configured correctly or cannot obtain the address of the ActCurrent4100_03 / RPZD3_4000_03;
16#000D	The PDO is not configured correctly or cannot obtain the address of the ActTorque4100_04 / TorqueUpLimit_4000_04;
16#000E	The PDO configuration is incorrect or the address of the DriveError4100_05 / TorqueLowLimit_4000_05 cannot be obtained;
16#000F	The PDO is incorrectly configured or cannot obtain the address of the TPZD6_4100_06;
16#0010	The PDO configuration is incorrect or the address of the SpeedUpLimit_4000_07 cannot be obtained;
16#0011	The PDO configuration is incorrect or the address of the SpeedLowLimit_4000_08 cannot be obtained;
16#0012	The PDO configuration is incorrect or the address of the OperationMode_4000_09 cannot be obtained;
16#0020	The current axis is already in use

3.1.1.4 BN_EtherCAT_PP_PV

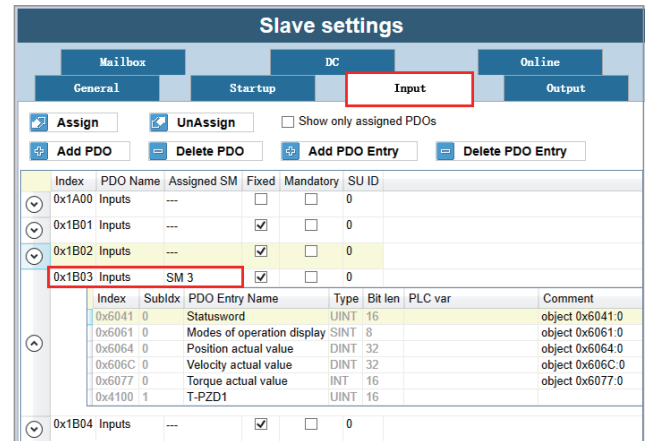
3.1.1.4.1 RPDO Configuration

Open the output page, cancel the synchronization management output configuration SM2 of other PDOs, and add 1703 to the synchronization

snap-in SM2. When using the Boneng EtherCAT servo drive, you do not need to add parameters to it, as shown in the following figure by default:



Open the input page, select 1B03 to configure SM3 for the synchronous management input. When using the Boneng EtherCAT servo drive, you do not need to add parameters to it, as shown in the following figure by default:



3.1.1.4.2 Pin Description

The pin description of the functional block is shown in the following table:

Name	Range/ Meaning	Data type	Unit	Description
AxisIndex	EtherCat 轴ID	UINT	\	0--15
OperationMode	1: Relative position	INT	\	1: PP mode
	2: Absolute position			2: PP mode
	3: Speed mode			3: PV mode
	4: Active return to zero			4: Hm mode
	5: Set the current position to zero			5: Hm mode
	6: **Not yet			
	7: Speed jog			7: PP mode
	8: Position jog			8: PP mode
bAxis Enable	1: Enable 0: Not enabled	BOOL	\	
bTask-Cancel	1: No cancellation order 0: Cancel the task	BOOL	\	Cancel task when OperationMode = 1、2
bAxis Halt	1: No halt order 0: Halt the task	BOOL	\	
bPositive	1: 0:	BOOL	\	The two BOOL pins are used together
bNegative	1: 0:	BOOL	\	

Name	Range/ Meaning	Data type	Unit	Description
				to 1 and bNegative is set to 0, the active return to zero takes the positive direction, and the velocity mode moves in the positive direction; When bNegative is set to 1 and bPositive is set to 0, the active return to zero takes the bNegative direction, and the velocity mode moves in the positive direction; All other cases are in the positive direction
bJog1	1: Triggers jog when set to 1 0: Cancel jog when set to 0	BOOL	\	Jog1
bJog2	1: Triggers jog when set to 1 0: Cancel jog when set to 0	BOOL	\	Jog2
bAxis-Reset	1: Fault reset 0:	BOOL		Rising edge triggering When function block occurs no FBError, user could use this pin to reset the AxisError
bAxis Execute	1: Trigger the command 0:	BOOL		Rising edge triggering Takes effect when OperationMode = 1, 2, 4, and 5
TargetPosition	-2147483647 ----- 2147483647	DINT		The number of command pulses When OperationMode = 1, 2, it is used as the target position

Name	Range/ Meaning	Data type	Unit	Description
Target-Vel	-2147483647 --- 2147483647	DITN	Number of command pulses/s	Speed command when OperationMode = 1, 2, 3
Profile-Acc	0--- 4294967295	UDINT	Number of command pulses/s ²	Acceleration command when OperationMode = 1, 2, 3, 7, 8
Profile-Dec	0--- 4294967295	UDINT	Number of command pulses/s ²	Deceleration command when OperationMode = 1, 2, 3, 7, 8
PZD3	用户自定义PZD3	WORD	\	\
bAxis Enable OK	1: The axis is enabled 0: The axis is not enabled	BOOL	\	\
bTarget Reach	1: Target position reached 0: Not arrived	BOOL	\	\
bAxis Error	1: There is an error in the axis 0: There is no error in the axis	BOOL	\	Shaft fault flags
AxisError Code	Axis error code	INT	\	\
bFBEr- ror	1: There is an error in the function block 0: There is no error in the function block	BOOL	\	Function block error flags

Name	Range/ Meaning	Data type	Unit	Description
FBEr- rorCode	Function block error code	UINT	\	\
ActVe- locity	Actual speed	DINT	Number of command pulses/s	\
ActPo- sition	Actual location	DINT	The number of command pulses	\
Act - Torque	Actual torque	INT	The percentage of the motor's rated torque	\
ActCur- rent	Actual current	INT	Please refer to the drive for the unit	\
ActDi- Status	Actual DI status	UDINT	\	BIT0: negative limit(0: not reached, 1: reached); BIT1: forward limit(0: not reached, 1: arrived); BIT2: ZERO SWITCH (0: NOT REACHED, 1: REACHED); BIT3-15: NA; BIT16-31: Reflects the actual DI terminal status of the drive
Act - Mode	Current mode	INT	\	\
PZD1	User-defined PZD1	WORD	\	\
PZD2	User-defined PZD2	WORD	\	\

Module Introduction:

- 1) The name of the axisIndex axis is 0-15 according to the topology, that is, the 0 axis directly connected to the Boneng controller EtherCat, the 1 axis connected to the 0 axis, and so on.
- 2) Acceleration, measured in command pulses/s2. If the electronic gear has 10000 pulses corresponding to one revolution of the motor, then the acceleration of 10000 command pulses/s2=1r/s2=60rpm/s, which is equivalent to accelerating 60 revolutions per second. This acceleration is relatively slow. The no-load acceleration capability of the Boneng low-power servo is approximately 2500r/s, and some servos can achieve 5000r/s;
- 3) Deceleration, same as above;
- 4) The trajectory planning speed, the unit is the command pulse/s, in the same way, if the electronic gear is still calculated according to the above, then 10000 command pulse/s=1r/s=60rpm, this value is relatively small;

3.1.1.4.3 Description of function block errorcode

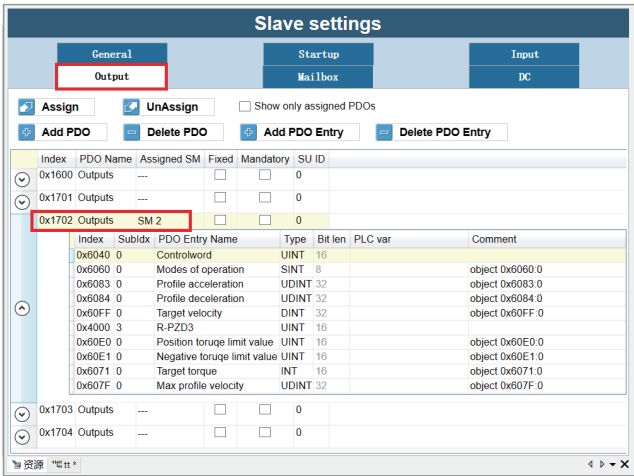
FBErrorCode	Description
0	No error;
1	Not entering the OP;
2	The AxisIndex is out of range, or does not enter the OP;
3	Wrong input mode of operation;
7	sysECAT_Sdo_Write2 function block failure (which can be caused by the following two reasons: (1): During the stop or recovery of the EtherCAT master, the sdo_write2 stack is called with an error (2): This error is reported when the read register is used in aperiodic communication and in online functions

FBErrorCode	Description
9	PDO configuration incorrect or unable to obtain the address of AxisErrorCode603F;
10	PDO configuration incorrect or unable to obtain the address of Status Word 6041/Control Word 6040;
11	PDO configuration incorrect or unable to obtain the addresses of the DisModeOfOP6061 / ModeOfOP6060;
12	PDO configuration incorrect or unable to obtain the addresses of ActPosition6064/targetPosition607A;
13	PDO configuration incorrect or unable to obtain the address of ActVelocity606C/ProfileVelocity6081;
14	PDO configuration incorrect or unable to obtain the address of ActTorque6077/ProfileAcceleration6083;
15	PDO configuration incorrect or unable to obtain the address of ActCurrent6078/ProfileDeceleration6084;
16	PDO configuration incorrect or unable to obtain the address of ActDiStatus 60FD/Target Velocity 60FF;
32	The current axis is already in use

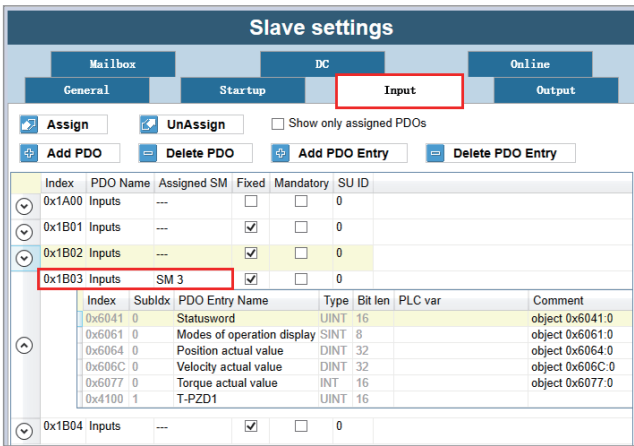
3.1.1.5 BN_EtherCAT_PV_Speed

3.1.1.5.1 RPDO Configuration

Open the output page, cancel the synchronous management output configuration SM2 of other PDOs, and add 1702 to the synchronization snap-in SM2. When using the Boneng EtherCAT servo drive, you do not need to add parameters to it, the default is as shown in the following figure:



Open the input page, select 1B03 to configure SM3 for the synchronous management input. When using the Boneng EtherCAT servo drive, you do not need to add parameters to it, as shown in the following figure by default:



3.1.1.5.2 Pin Description

The pins of BN_EtherCat_PV_Speed function blocks are described in the following table:

Name	Range/ Meaning	Data type	Unit	Description
AxisIndex	EtherCat Axis ID	UINT	\	0--15
bAxis Enable	1: Enable 0: Not enable	BOOL	\	
bQuick- Stop	1: Quick stop is valid 0: The Quick stop is invalid	BOOL	\	Quick stop
bMotor- Rotation	1: Reverse 0: Forward	BOOL	\	The direction of rotation of the motor
bAxis- Reset	1: Fault reset 0:	BOOL	Rising edge triggering	When there is no FError, user could use this pin to reset the AxisError
Target-Speed	-32768 ---32767	INT	r/min	Target speed
R e f - Speed	0---65535	UINT	Target speed	Speed limit and acceleration and deceleration time reference value: The default is the rated speed of the motor
AccTime	\	REAL	s	Acceleration time
DecTime	\	REAL	s	Deceleration time
Torque UpLimit	-32768 ---32767	INT	\	The upper torque limit of 1000 corresponds to 100.0%, and 100.0% corresponds to the rated torque of the motor

Name	Range/ Meaning	D a t a type	Unit	Description
Torque- L o w - Limit	-32768 ---32767	INT	\	The lower torque limit of 1000 corresponds to 100.0%, and 100.0% corresponds to the rated torque of the motor
Encoder Resolu- tion	0--- 4294967295	UDINT	\	Encoder Resolu- tion, enter the resolution of the corresponding motor encoder
PZD3	User-defined PZD3	WORD	\	\
bAxis Enable OK	1: The axis is enabled 0: The axis is not enabled	BOOL	\	\
bFault	1: There is an error in the fu- nction block 0: There is no error in the fu- nction block	BOOL	\	Function block error flags
F B E r- rorCode	\	UINT	\	Function block error code
AxisEr- rorCode	\	WORD	\	Axis error code
ActVe- locity	\	DINT	r/min	Actual speed
A c t - Torque	\	INT	The per- centage of the motor's rated torque	Actual torque
ActCur- rent	\	INT	Please refer to the drive for the unit	Actual current

Name	Range/ Meaning	D a t a type	Unit	Description
ActPosi- tion	\	DINT	The number of command pulses, con- verted by electronic gear ratio	Physical location
ActDi- Status	\	UDINT	\	Actual DI status BIT0: negative limit(0:not reach- ed, 1: reached); BIT1: forward limit (0: not reached, 1: arrived); BIT2: ZERO SWITCH (0: NOT REACHED, 1: REACHED); BIT3-15:NA; BIT16-31: Reflects the actual DI terminal status
PZD1	User-de- fined PZD1	WORD	\	\
PZD2	User-de- fined PZD2	WORD	\	\

Module Introduction:

- 1) The name of the axisIndex axis is 0-15 according to the topology, that is, the 0 axis directly connected to the Boneng controller EtherCat, the 1 axis connected to the 0 axis, and so on.
- 2) The user needs to input the servo encoder resolution, the default 8388608, need to input the reference speed (motor rated speed), the default is 3000, if the input error will cause the input acceleration time and target speed can not play a role, the output actual speed is inconsistent with the actual speed.
- 3) The PV block runs in PV mode, in which the block enable pin input high runs according to

the user-given limit and target value, and the enable pin enters low, then it stops running.

3.1.1.5.3 Description of function block errorcode

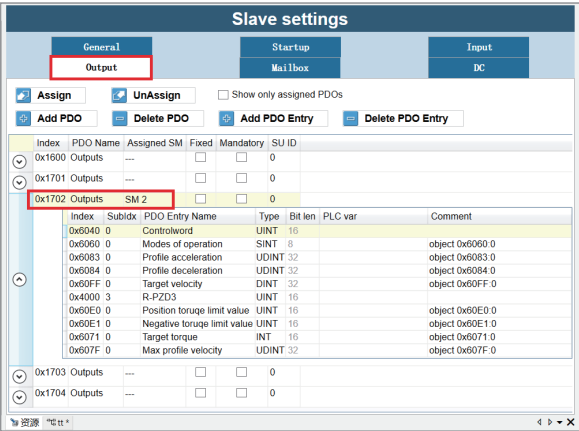
FBErrorCode	Description
0	No error;
1	Not entering the OP;
2	The AxisIndex is out of range, or does not enter the OP;
256	The reference speed is not set;
7	sysECAT_Sdo_Write2 function block failure (which can be caused by the following two reasons:(1): During the stop or recovery of the EtherCAT master, the sdo_write2 stack is called with an error ;(2): This error is reported when the read register is used in aperiodic communication and in online functions;
9	The PDO configuration is incorrect or the address of AxisErrorcode603F cannot be obtained;
10	The PDO configuration is incorrect or the address of StatusWord6041 / ControlWord6040 cannot be obtained;
11	The PDO configuration is incorrect or the address of DisModeOfOP6061 / ModeOfOP6060 cannot be obtained;
12	The PDO configuration is incorrect or the address of ActPosition6064 / TargetPosition607A cannot be obtained;
13	The PDO configuration is incorrect or the address of ActVelocity606C / ProfileVelocity6081 cannot be obtained;
14	The PDO configuration is incorrect or the address of ActTorque6077 / ProfileAcceleration6083 cannot be obtained;

FBErrorCode	Description
15	The PDO configuration is incorrect or the address of ActCurrent6078 / ProfileDeceleration6084 cannot be obtained;
16	The PDO configuration is incorrect or the address of ActDiStatus60FD / TargetVelocity60FF cannot be obtained;
17	The PDO configuration is incorrect or the address of PositiveTorqueLimit60E0 cannot be obtained;
18	The PDO configuration is incorrect or the address of NegativeTorqueLimit60E1 cannot be obtained;
19	The PDO configuration is incorrect or the address of TargetTorque6071 cannot be obtained;
20	The PDO is incorrectly configured or the address of MaxProfileVelocity607F cannot be obtained;
32	The current axis is already in use

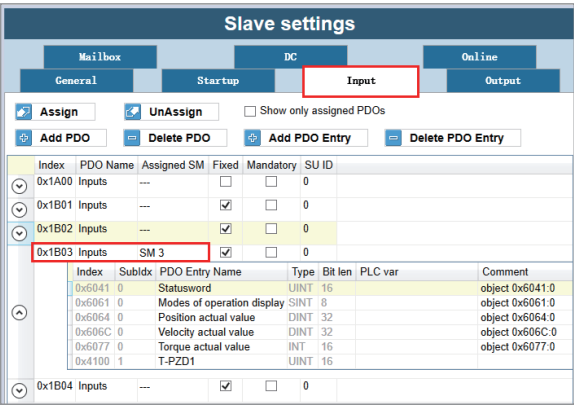
3.1.1.6 BN_EtherCAT_PT_Torque

3.1.1.6.1 RPDO Configuration

Open the output page, cancel the synchronous management output configuration SM2 of other PDOs, and add 1702 to the synchronization snap-in SM2. When using the Boneng EtherCAT servo drive, you do not need to add parameters to it, the default is as shown in the following figure:



Open the input page, select 1B03 to configure SM3 for the synchronous management input. When using the Boneng EtherCAT servo drive, you do not need to add parameters to it, as shown in the following figure by default:



3.1.1.6.2 Pin Description

The pin of BN_EtherCat_PT_Torque function blocks are described in the following table:

Name	Range/ Meaning	D a t a type	Unit	Description
Axis Index	EtherCat Axis ID	UINT	\	0--15
bAxis Enable	1: Enable 0: Not enabled	BOOL	\	

Name	Range/ Meaning	D a t a type	Unit	Description
bMotor- Rotation	1: Reverse 0: Forward	BOOL	\	
bAxis- Reset	1: Fault reset 0: No fault	BOOL	R i s i n g edge trig- gering	When there is no FError, user could use this pin to reset the AxisError
Target- Torque	-32768 ---32767	INT	\	The target torque, 1000 corresponds to 100.0%, and 100.0% corre- sponds to the rated torque of the motor
M a x - Torque	-32768 ---32767	INT	\	Maximum torque, target torque is limited by this parameter, 1000 means 100.0%, maximum 3000
Speed- Limit	0---65535	UINT	r/min	Speed Limit **Unit: r/min
Encoder- Resolu- tion	0--- 4294967295	UDINT	\	Encoder Resolu- tion, enter the resolution of the corresponding motor encoder
PZD3	User-defi- ned PZD3	WORD	\	\
bAxis Enable OK	1: The axis is enabled 0: The axis is not enabled	BOOL	\	\
bFault	1: There is an error in the fu- nction block 0: There is no error in the fu- nction block	BOOL	\	Function block error flags

Name	Range/ Meaning	Data type	Unit	Description
FBErrorCode	\	UINT	\	Function block error code
AxisErrorCode	\	WORD	\	Axis error code
ActVelocity	\	DINT	r/min	Actual speed
ActTorque	\	INT	The percentage of the motor's rated torque	Actual torque
ActCurrent	\	INT	Please refer to the drive for the unit	Actual current
ActPosition	\		The number of command pulses, converted by electronic gear ratio	Physical location
ActDI-Status	\	UDINT	\	Actual DI status BIT0:negative limit (0:not reached, 1: reached); BIT1: forward limit (0: not reached, 1: arrived); BIT2:ZERO SWITCH (0: NOT REACHED, 1: REACHED); BIT3-15:NA; BIT16-31: Reflects the actual DI terminal status
PZD1	User-defined PZD1	WORD	\	\
PZD2	User-defined PZD2	WORD	\	\

Module Introduction:

- 1) The name of the axisIndex axis is 0-15 according to the topology, that is, the 0 axis directly connected to the Boneng controller EtherCat, the 1 axis connected to the 0 axis, and so on.
- 2) The user needs to input the servo encoder resolution, and 8388608 by default, this pin input error will cause the input speed limit and the output actual speed to not match the actual one.
- 3) The PT block runs in PT mode, in which the block enable pin input high runs according to the user-given limit and target value, and the enable pin input low stops.

3.1.1.5.3 Description of function block errorcode

FBErrorCode	Description
0	No error;
1	Not entering the OP;
2	The AxisIndex is out of range, or does not enter the OP;
7	sysECAT_Sdo_Write2 function block failure (which can be caused by the following two reasons:(1): During the stop or recovery of the EtherCAT master, the sdo_write2 stack is called with an error ;(2): This error is reported when the read register is used in aperiodic communication and in online functions;
9	The PDO configuration is incorrect or the address of the AxisErrorCode603F cannot be obtained;
10	The PDO configuration is incorrect or the address of the StatusWord6041 / ControlWord6040 cannot be obtained;
11	The PDO configuration is incorrect or the address of the DisModeOfOP6061 / ModeOfOP6060 cannot be obtained;

FBErrorCode	Description
12	The PDO configuration is incorrect or the address of ActPosition6064 / TargetPosition607A cannot be obtained;
13	The PDO configuration is incorrect or the address of ActVelocity606C / ProfileVelocity6081 cannot be obtained;
14	The PDO configuration is incorrect or the address of ActTorque6077 / ProfileAcceleration6083 cannot be obtained;
15	The PDO configuration is incorrect or the address of ActCurrent6078 / ProfileDeceleration6084 cannot be obtained;
16	The PDO configuration is incorrect or the address of ActDiStatus60FD / TargetVelocity60FF cannot be obtained;
17	The PDO configuration is incorrect or the address of PositiveToruqeLimit60E0 cannot be obtained;
18	The PDO configuration is incorrect or the address of NegativeToruqeLimit60E1 cannot be obtained;
19	The PDO configuration is incorrect or the address of TargetTorque6071 cannot be obtained;
20	The PDO is incorrectly configured or the address of MaxProfileVelocity607F cannot be obtained;
32	The current axis is already in use

3.1.2 BN_Basic

3.1.2.1 BN_PID

PID controller: The PID controller can realize the closed-loop adjustment function of temperature,

pressure, flow, frequency, etc., which generally needs to be used with AI and AQ modules, AI modules and linear correction modules calibrate the feedback value, and the output value is output through the AQ module. The default PID parameters are small, please adjust them according to the actual situation.

3.1.2.1.1 Pin Description

Name	DataType	Explain
set_point	REAL	Controller setpoint
feedback	REAL	Signal actual value (feedback)
Kp	REAL	Proportional gain
Ki	REAL	Integral gain
Kd	REAL	Derivative gain
rst	BOOL	Controller reset
OutMax	REAL	Output signal upper limit
OutMin	REAL	Output signal lower limit
Magnification	BOOL	Adjusting the magnification
out	REAL	Output signal
limitFlag	BOOL	In-limit output status

3.1.2.2 BN_Ramp

BN_Ramp ramp function, users can achieve the required ramp output according to their needs.

3.1.2.2.1 Pin Description

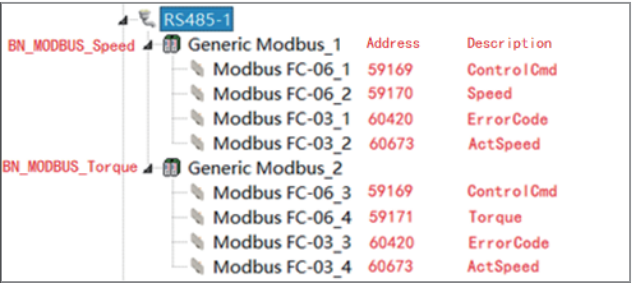
Name	DataType	Explain
Input	REAL	Enter a given value
RefValue	REAL	Reference values
RiseTime	UDINT	Rise time **Unit: ms
FallTime	UDINT	Fall Time **Unit: ms
InputMaxLimit	REAL	Upper limit
InputMinLimit	REAL	Lower limit
ResetValue	REAL	The value of the RAMP after the reset
Reset	BOOL	Reset: TRUE: Reset the output to 0; FALSE: No reset;

Name	Data Type	Explain
bFBError	BOOL	The function block is entered incorrectly
FBErrorCode	WORD	Function block fault codes FBErrorCode : 16#0000 : No error in function block; 16#0001 : Rise or Fall Time is set incorrectly; 16#0002 : Reference value set incorrectly;
Output	REAL	Output value

3.1.3 BN_Modbus

The BN_Modbus library contains two function blocks, BN_MODBUS_Speed and BN_MODBUS_Torque, for controlling the inverter slaves

Modbus RTU communication needs to be configured, the X3000 series controller is the master station, and the inverter is the slave station, and the configuration can be referred to the following figure



3.1.3.1 BN_MODBUS_Speed

Name	Data Type	Explain
bEnable	BOOL	Enable operation, 0: not enable, 1: axis enable
bStopMode	BOOL	Stop mode, 0: deceleration shutdown, 1: free shutdown
bJog	BOOL	Jog control
bMotorRotation	BOOL	Motor rotation direction, 0: forward, 1: reverse

Name	Data Type	Explain
bFaultReset	BOOL	Fault reset, 0: the reset is invalid, 1: the reset is valid
SlaveError Code	WORD	Axis error code
SlaveSpeed-ActPer	WORD	Axis feedback velocity scale value
SpeedSet	INT	Target speed, based on SpeedRef
SpeedRef	UINT	A reference datum for the velocity value
bFault	BOOL	Fault status, 0: No Fault, 1: Faulty
FBErrorCode	WORD	Function block error code
DriveError Code	WORD	Drive error code
SlaveSpeed-Act	INT	The velocity value feedback from the axis, with reference to SpeedRef
SpeedSetTo-Slave	WORD	Speed command given to the slave Axis
CtrlWordTo-Slave	WORD	To the control command of the slave Axis, 1: Run, 2: Jog, 3: Free Stop, 4: Deceleration Stop, 5: Fault Reset

When using it, you need to enter the 60420 errorcode into the SlaveErrorCode, and the 60673 actual speed into the SlaveSpeedActPer

Output SpeedSetToSlave to 59170 speed, output CtrlWordToSlave to 59169 controlword

3.1.3.2 BN_MODEBUS_Torque

Name	Data Type	Explain
bEnable	BOOL	Enable operation, 0: not enable, 1: axis enable
bMotorRotation	BOOL	Stop mode, 0: deceleration shutdown, 1: free shutdown
bFaultReset	BOOL	Jog control
SlaveError Code	WORD	Motor rotation direction, 0: forward, 1: reverse

Name	DataType	Explain
SlaveSpeed-ActPer	WORD	Fault reset, 0: the reset is invalid, 1: the reset is valid
SpeedRef	UINT	A reference datum for the velocity value
TorqueSet	INT	The target torque, 1000 corresponds to 100.0%, and 100.0% corresponds to the rated torque of the motor
bFault	BOOL	Fault status, 0: No Fault, 1: Faulty
FBErrorCode	WORD	Function block error code
DriveError Code	WORD	Drive error code
SlaveSpeed-Act	INT	The velocity value feedback from the axis, with reference to SpeedRef
TorqueSetTo-Slave	WORD	Torque command given to the slave slave
CtrlWordTo-Slave	WORD	To the control command of the slave Axis, 1: Run, 2: Jog, 3: Free Stop, 4: Deceleration Stop, 5: Fault Reset

When using it, you need to enter the 60420 errorcode into the SlaveErrorCode, and the 60673 actual speed into the SlaveSpeedActPer
Output TorqueSetToSlave to 59171 torque, and output CtrlWordToSlave to 59169 control word

3.2 BonengEmbedLogicLib

The Boneng Embedded Library contains some of the functions and function blocks that are commonly used by engineers.

3.2.1 ArithmeticOperationClassExtLib

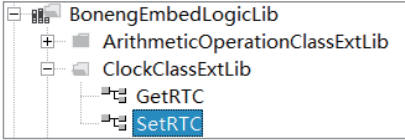
3.2.1.1 GetFraction

The GetFraction function is used to obtain the decimal part of REAL data.
Function input section:

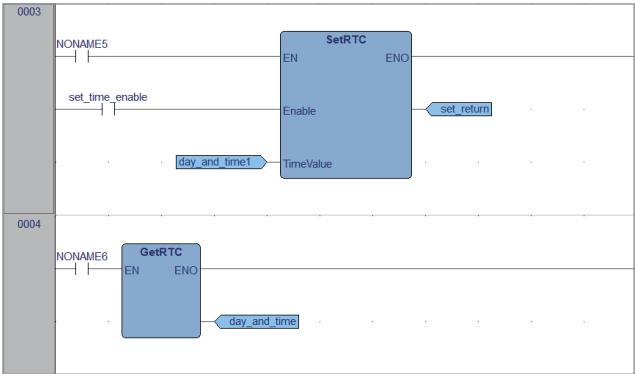
Name	Type	Description
DATA_REAL	REAL	Real type input data

3.2.2 ClockClassExtLib

3.2.2.1 Set the real-time time and get the real-time time
Firstly, find BonengEmbeddLogicLib in the library management wizard, and expand the ClockClassExtLib folder. Now we can see SetRTC and GetRTC



Add these two functions to the program, as the following figure is shown:



In SetRTC, the EN pin is used to enable the function block, and Enable pin is the trigger pin of the function to write data, emphasizing: (the pin needs to be triggered by the rising edge), TimeValue is the time setting pin of the function, the variable type of the pin is TIME_AND_DAY, the pin can be set to the time in format such as →1920-01-01-00:00:00, the time setting is generally the current Beijing time, and the return value of the function does not need to be concerned.
In GetRTC, the EN pin is used to enable the function block, and the return value of the function is a TIME_AND_DAY type of data, and when the function is enabled, you can see that the time changes on the output pin.

3.2.3 DataConversionClassExtLib

3.2.3.1 DATA16_SWAP

DATA16_SWAP function is used to exchange high and low bytes of 16-bit input data.

Name	Type	Description
DATA_IN	UINT	16-bit input data to be exchanged between high and low bytes

3.2.3.2 DATA2BIT

The DATA2BIT function is used to select the corresponding bit of the 16-bit input data.

Name	Type	Description
DATA_IN	UINT	16-bit input data
BIT_NUM	UINT	Selected BIT bits, ranging from 0-15

3.2.3.3 DATA32_SWAP

DATA32_SWAP function is used to exchange high and low bytes of 32-bit input data.

Name	Type	Description
DATA_IN	UINT	32-bit input data to be exchanged between high and low bytes

3.2.3.4 REAL_ENDIAN_SWAP

The REAL_ENDIAN_SWAP function is used to exchange the size of 32-bit REAL data, and is used to convert the REAL data transmitted by Siemens PLC.

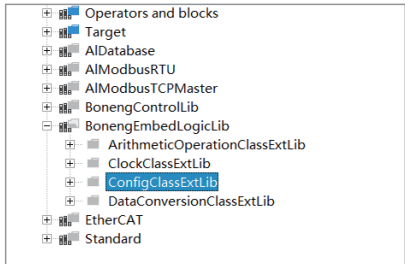
Name	Type	Description
REAL_IN	REAL	The input data of the REAL type to be exchanged between the large and small ends

3.2.4 ConfigClassExtLib

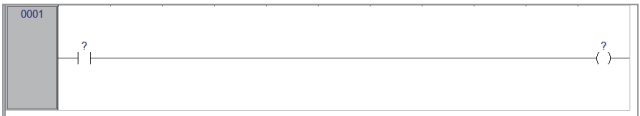
3.2.4.1 Analog settings

3.2.4.1.1 AI_ModeConfig

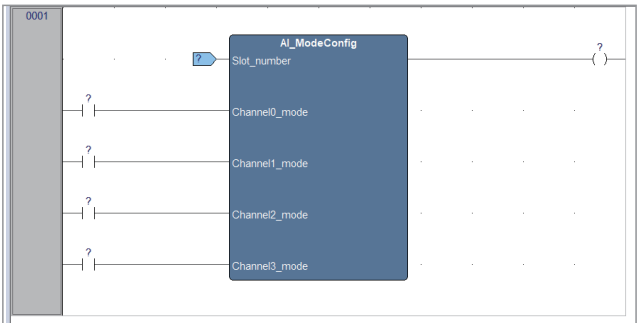
Switching of analog input voltage and current
Firstly, locate BonengEmbeddLogicLib, click on the plus sign to expand the folder, find Configurat-ClassExtLib, click on the plus sign to continue expanding the folder



Find the function shown in the image on the right AI_ModeConfig  add this function to the program, it can be called in 5 languages, here it is only demonstrated to call in LD language;
First, create a blank LD language program, as shown in the figure;



Add the blue cursor AI_ModeConfig above by dragging it onto the contacts of the LD program. After the addition is successful, the following figure shows;



The function has a total of 5 input parameters, you need to set the corresponding slot number, and the mode of the channel can switch the voltage and current between the input detection conversion, if you do not call the function, the default voltage mode input.

Slot_number : UINT; (* Slot number of AI04 extension module, query in backplane configuration diagram, range 1-15*)

Channel0_mode : BOOL; (*AI0 mode selection, 0: voltage mode, 1: current mode*)

Channel1_mode : BOOL; (*AI1 mode selection, 0: voltage mode, 1: current mode*)

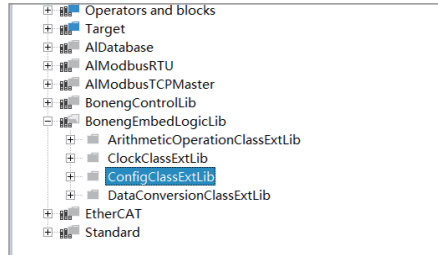
Channel2_mode : BOOL; (*AI2 mode selection, 0: voltage mode, 1: current mode*)

Channel3_mode : BOOL; (*AI3 mode selection, 0: voltage mode, 1: current mode*)

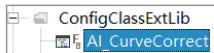
3.2.4.1.2 AI_CurveCorrect

User-defined analog input curves

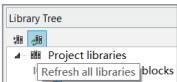
Firstly, locate BonengEmbeddLogicLib, click on the plus sign to expand the folder, and then find ConfiguratClassExtLib. Click on the plus sign to continue expanding the folder



Find the function block AI_CurveCorrect



If the function block is not displayed, please reinstall the latest library file, click the refresh button

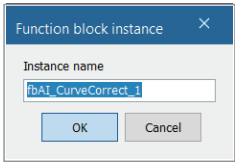


the library file, and refresh the library

file.

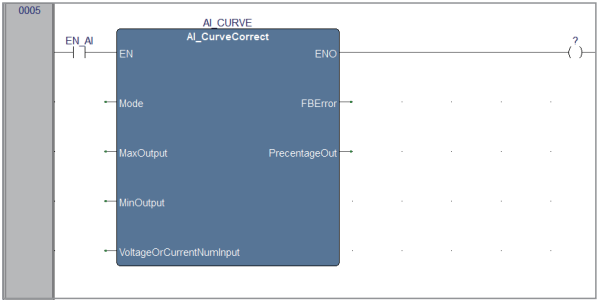
Drag the AI_CurveCorrect into the program, see 2.10.1.1 for details. Unlike adding a function, when

adding a function block to the program, you need to set an instance name, as shown in the figure:



after the addition is complete,

the function block can be called. The addition is completed as shown in the following figure:



The function block has four input parameters:

Mode: BOOL; (* Mode Selection, 0: Voltage Type, 1: Current Type*)

MaxOutput : INT; (* Maximum value of output *)
(The maximum value should be greater than the minimum value, otherwise an error will be reported)

MinOutput : INT;(*Minimum value of output*)

VoltageOrCurrentNumInput : UINT; (* Analog input value of voltage or current *)

The input value of voltage should not exceed 10,000, and the input value of current should not exceed 20,000.

The function block has two output parameters:

FError : UINT; (* Function block error code *) An error is reported when the entered value does not meet the requirements.

PercentageOut: DINT; (* Characteristic Curve Value Output *)

3.2.4.2 Digital settings

3.2.4.2.1 Control_IO_By_Word

Control_IO_By_Word function block can select the corresponding slot number, and control the switching output by entering Word.

Name	Type	Description
Execute	BOOL	Execute the function of control by word
Slot_Index	USINT	The slots of the extend DQ, range: 1-15
Word_Data	WORD	The input word
ERROR	USINT	Error code bit0=1: The extension slot is out of range, and bit1=1: The input control word is out of range

3.2.4.2.2 Reset_IO_By_Index

Reset_IO_By_Index function block can select the corresponding slot number, and select the start and end points to perform the reset of the corresponding switching quantity.

Name	Type	Description
Execute	BOOL	Perform the reset function block
Slot_Index	USINT	The slots of the extend DQ, range: 1-15
Start_IO	USINT	Start point, range 0-15
End_IO	USINT	End point, range 0-15
ERROR	USINT	Error code bit0=1 The slot number is out of range, the start point of bit1=1 is out of range, the end point of bit2=1 is out of range, and the start point of bit3=1 is greater than the end point

3.2.4.2.3 Set_IO_By_Index

Set_IO_By_Index function block can select the corresponding slot number, and select the start and end points to execute the setting of the corresponding switching quantity.

Name	Type	Description
Execute	BOOL	Execute the set function block
Slot_Index	USINT	The slots of the extend DQ, range: 1-15
Start_IO	USINT	Start point, range 0-15
End_IO	USINT	End point, range 0-15
ERROR	USINT	Error code bit0=1 The slot number is out of range, the start point of bit1=1 is out of range, the end point of bit2=1 is out of range, and the start point of bit3=1 is greater than the end point

3.3 BonengMachiningProcessLib

3.3.1 BN_TensionControl

BN_TensionControl library is a curling process library.

3.3.1.1 BN_TensionTaper

BN_TensionTaper function block, the input is set tension, the output is the tension value processed by taper, the taper mode supports parabolic and linear modes, and the proportion of tension attenuation is set by the taper attenuation coefficient.

Name	Type	Description
TensionSet	REAL	Rewinding Tension, Unit(N)
ActDiameterOfRoller	UINT	The actual diameter of the winding roller, 50mm minimum and 2000mm maximum
TaperCoff	REAL	Taper attenuation coefficient, the larger this value, the greater the taper, the default is 0, the range is 0.0~1.0
TaperMode	INT	Taper mode, 0: parabolic taper, 1: linear taper, default 0
TaperDiameterStart	UINT	Taper starting diameter, in (mm)

Name	Type	Description
TaperDiameterMax	UINT	Taper maximum diameter in (mm)
bError	BOOL	Function Block error flags
FBErrorCode	UINT	Function Block error codes
TensionByTaper	REAL	Taper-treated tension value in (Nm)

3.3.1.2 BN_TractionSpeed

BN_TractionSpeed traction axis speed control function block, the speed of the traction motor is calculated according to the set linear speed, the diameter of the traction roller, and the reduction ratio. The traction axis can be either a moudbus speed axis or an EtherCAT speed axis.

Name	Type	Description
LineVelocitySet	REAL	Line speed setting value (m/min)
TractionMotorSpeedAct	INT	Traction motor speed (rpm)
DiameterOfTractionRoller	UINT	Traction roller diameter, minimum 10mm, maximum 1000mm
GearRatio	REAL	Gear ratio, default is 1.0
bError	BOOL	Function Block error flags
FBErrorCode	UINT	Function Block error codes
LineVelocityAct	REAL	Actual value of traction roller line speed, m/min
TractionMotorSpeedSet	INT	Traction roller motor speed (rpm)

3.3.1.3 BN_Unwinder_CTorque

The difference between the closed-loop tension torque control function block of the BN_Unwinder_CTorque unwinding roll and the open-loop tension torque control is that the tension closed-loop

control can be carried out through the PID, which requires the equipment to support tension detection and feedback. This function block is recommended to be executed in a fast task, otherwise the PID is difficult to debug.

Name	Type	Description
LineVelocityAct	REAL	Unwinding line speed(m/min)
UnwinderMotorSpeedAct	INT	Unwinding motor speed (rpm)
DiameterOfFullRoller	UINT	The full roll diameter of the unwinding roller, 10mm minimum and 1000mm maximum
GearRatio	REAL	Gear ratio, default is 1.0
UnwinderMotorRatedTorque	REAL	Rated torque of unwinding roller motor: 9550 * P/n, (Nm)
FrictionCompensation	INT	Unwinding roll friction compensation, base value is motor rated torque, (0.1%)
TensionSet	REAL	The unwinding setting tension (N) is also the PID reference value
TensionAct	REAL	Tension actual feedback value, PID feedback value
Kp	REAL	PID regulator scale coefficient
Ki	REAL	PID regulator integration coefficient
Kd	REAL	PID regulator differential coefficients
PidOutLimit	REAL	PID regulator output limit, rpm
PidReset	BOOL	PID regulator results are reset
bError	BOOL	Function Block error flags
FBErrorCode	UINT	Function Block error codes
DiameterOfRollerAct	REAL	Real-time roll diameter, diameter, (mm)

Name	Type	Description
PidOut	INT	Pendulum tension PID adjustment output, motor rated torque percentage (0.1%)
UnwinderMotorSpeedSet	INT	Unwinding roller motor speed (rpm)
UnwinderMotorTorqueSet	INT	Unwinding roller motor torque, 1000 means 100.0%

3.3.1.4 BN_Unwinder_OTorque

BN_Unwinder_OTorque open-loop tension torque control function block of unwinding roller is used for occasions where there is no tension feedback of the equipment, and when the general tension requirements are not high, there is no PID closed-loop tension regulator.

Name	Type	Description
TensionSet	REAL	Unwinding tension (N)
LineVelocityAct	REAL	Unwinding line speed(m/min)
UnwinderMotorSpeedAct	INT	Unwinding motor speed (rpm)
DiameterOfFullRoller	UINT	The full roll diameter of the unwinding roller, 10mm minimum and 1000mm maximum
GearRatio	REAL	Gear ratio, default is 1.0
UnwinderMotorRatedTorque	REAL	Rated torque of unwinding roller motor: $9550 * P/n$, (Nm)
FrictionCompensation	INT	Unwinding roll friction compensation, base value is motor rated torque, (0.1%)
MassOfFullRoller	REAL	The weight of the unwinding roll when it is full (kg) is used to compensate for acceleration and deceleration torque
DensityRatioOfMaterial	REAL	The density ratio of the square roll material and the roller ranges from 0.0~1.0

Name	Type	Description
bError	BOOL	Function Block error flags
FBErrorCode	UINT	Function Block error codes
DiameterOfRollerAct	REAL	Real-time roll diameter, (mm)
UnwinderMotorSpeedSet	INT	Unwinding roller motor speed (rpm)
UnwinderMotorTorqueSet	INT	Unwinding roller motor torque, 1000 means 100.0%

3.3.1.5 BN_Winder_CTorque

The difference between the closed-loop tension torque control function block of the BN_Winder_CTorque winding roll and the open-loop tension torque control is that the closed-loop tension control can be carried out through the PID, and the equipment is required to support tension detection and feedback. This function block is recommended to be executed in a fast task, otherwise the PID is difficult to debug.

Name	Type	Description
LineVelocityAct	REAL	Winding line speed(m/min)
WinderMotorSpeedAct	INT	Winding motor speed (rpm)
DiameterOfEmptyRoller	UINT	The initial diameter of the winding roller is 10mm minimum and 1000mm maximum
GearRatio	REAL	Gear ratio, default is 1.0
WinderMotorRatedTorque	REAL	Rated torque of winding roller motor: $9550 * P/n$, (Nm)
FrictionCompensation	INT	Winding roll friction compensation, base value is motor rated torque, (0.1%)
TensionSet	REAL	The winding set tension (N) is also the PID reference value
TensionAct	REAL	Tension actual feedback value, PID feedback value
Kp	REAL	PID regulator scale coefficient

Name	Type	Description
Ki	REAL	PID regulator integration coefficient
Kd	REAL	PID regulator differential coefficients
PidOutLimit	REAL	PID regulator output limit, rpm
PidReset	BOOL	PID regulator results are reset
bError	BOOL	Function Block error flags
FBErrorCode	UINT	Function Block error codes
DiameterOfRollerAct	REAL	Real-time roll diameter, diameter, (mm)
PidOut	INT	Pendulum tension PID adjustment output, motor rated torque percentage (0.1%)
WinderMotorSpeedSet	INT	Winding roller motor speed (rpm)
WinderMotorTorqueSet	INT	Winding roller motor torque, 1000 means 100.0%

3.3.1.6 BN_Winder_OTorque

BN_Winder_OTorque open-loop tension torque control function block of winding roller is used for occasions where there is no tension feedback of the equipment, and when the general tension requirements are not high, there is no PID closed-loop tension adjustment.

Name	Type	Description
TensionSet	REAL	Winding Tension (N)
LineVelocityAct	REAL	Winding line speed(m/min)
WinderMotorSpeedAct	INT	Winding motor speed (rpm)
DiameterOfEmptyRoller	UINT	The initial diameter of the winding roller, 10mm minimum and 1000mm maximum

Name	Type	Description
GearRatio	REAL	Gear ratio, default is 1.0
WinderMotorRatedTorque	REAL	Rated torque of winding roller motor:9550 * P/n, (Nm)
FrictionCompensation	INT	Winding roll friction compensation, base value is motor rated torque, (0.1%)
MassOfEmptyRoller	REAL	The weight of the winding roll when it is empty (kg) is used to compensate for acceleration and deceleration torque
DensityRatioOfMaterial	REAL	The density ratio of the winding material and the roller ranges from 0.0~1.0
bError	BOOL	Function Block error flags
FBErrorCode	UINT	Function Block error codes
DiameterOfRollerAct	REAL	Real-time roll diameter, (mm)
WinderMotorSpeedSet	INT	Unwinding roller motor speed (rpm)
WinderMotorTorqueSet	INT	Winding roller motor torque, 1000 means 100.0%

3.3.1.7 BN_WinderUnwinderSpeed

BN_WinderUnwinderSpeed, the closed-loop tension speed control function block of the unwinding roller requires the equipment to have tension detection and feedback. By PID closed-loop tension adjustment, the tension is controlled to be constant by maintaining a constant linear speed of the winding roller. This module is recommended to be executed in the fast task, otherwise PID debugging will be difficult.

Name	Type	Description
LineVelocityAct	REAL	Winding line speed(m/min)
WinderMotorSpeedAct	INT	Winding motor speed (rpm)

Name	Type	Description
DiameterOfEmptyRoller	UINT	The initial diameter of the winding roller is 10mm minimum and 1000mm maximum
GearRatio	REAL	Gear ratio, default is 1.0
TensionSet	REAL	Tension setpoint, PID reference value
TensionAct	REAL	Tension feedback value, PID feedback value
Kp	REAL	PID regulator scale coefficient
Ki	REAL	PID regulator integration coefficient
Kd	REAL	PID regulator differential coefficients
PidOutLimit	REAL	PID regulator output limit, rpm
PidReset	BOOL	PID regulator results are reset
bError	BOOL	Function Block error flags
FBErrorCode	UINT	Function Block error codes
DiameterOfRollerAct	REAL	Real-time roll diameter, diameter, (mm)
PidOut	INT	Pendulum tension PID regulator output value (rpm)
WinderMotorSpeedSet	INT	Winding roller motor speed (rpm)

3.4 MotionControl

3.4.1 MotionControl Lib Introduce

The MotionControl library contains the MC function blocks independently developed by Boneng, which conforms to the PLCOpen specification and implements the basic instructions and master-slave instructions

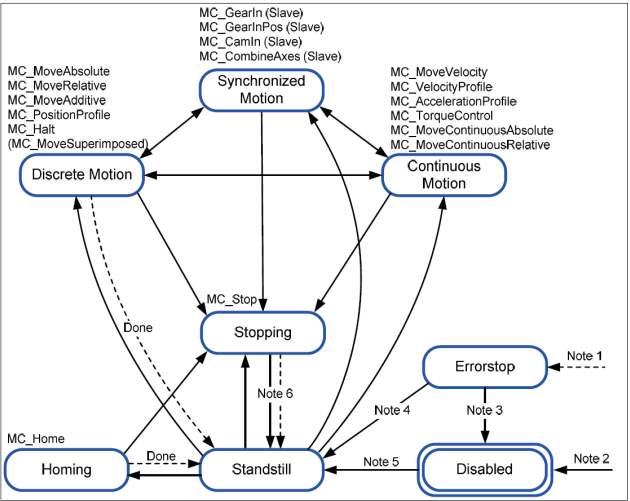
3.4.1.1 Instruction list

The basic motion control instructions are as follows:

instructions		Function
Basic	MC_Power	Enable control
	MC_Home	Homing
	MC_Stop	Stop
	MC_Halt	Halt
	MC_MoveAbsolute	Absolute positioning
	MC_MoveRelative	Relative positioning
	MC_MoveAddtive	Addtive positioning
	MC_MoveSuperimposed	Move superimposed
	MC_HaltSuperimposed	Halt moving superimposed
	MC_MoveVelocity	Speed reference
	MC_MoveContinuousAbsolute	Countinuous moving after absolute positioning
	MC_MoveContinuousRelative	Countinuous moving after relative positioning
	MC_SetPosition	Current position setting
	MC_ReadActualPosition	Current position read
	MC_ReadActualVelocity	Current velocity read
	MC_ReadActualTorque	Current torque read
	MC_ReadStatus	Current PLCOpen status read
	MC_ReadMotionState	Current motion state read
	MC_ReadAxisInfo	Current axis info read
	MC_ReadAxisError	Current axis error read
	MC_Reset	Fault reset
	MC_DigitalCamSwitch	Sets the switching semaphore at the specified position
	MC_TouchProbe	Probe commands
	MC_AbortTrigger	Interrupt the command
	MC_CamTableSelect	Select the CamTable and output the CamID which is available for MC_CamIn application

instructions		Function
Cam & Gear	MC_CamIn	Cam synchronization commands
	MC_CamOut	Exit the cam synchronization command
	MC_GearIn	Gear synchronization commands
	MC_GearOut	Exit the gear synchronization command
	MC_GearInPos	Gear Synchronization Command (Added Parameters to Limit the Transition Process Before Entering Synchronization)

3.4.1.2 PLCopen状态机



The state machine is described in detail as follows:

State	Description of the function
Disabled	Axis not enable
ErrorStop	Axis fault shutdown
Standstill	Axis enable and no motion commands active
Homing	Axis is homing

State	Description of the function
Stopping	Axis in stopping
Discrete Motion	Axis is in discrete motion
Continuous Motion	Axis is in Continuous motion
Synchronized Motion	Axis is in Synchronized motion

PS:

- Note 1: From any state. An error in the axis occurred.
- Note 2: From any state. MC_Power.Enable = FALSE and there is no error in the axis.
- Note 3: MC_Reset AND MC_Power.Status = FALSE
- Note 4: MC_Reset AND MC_Power.Status = TRUE AND MC_Power.Enable = TRUE
- Note 5: MC_Power.Enable = TRUE AND MC_Power.Status = TRUE
- Note 6: MC_Stop.Done = TRUE AND MC_Stop.Execute = FALSE

3.4.1.3 BufferMode

Some motion function blocks have a pin of Buffer-Mode, which means buffer mode, which specifies the connection mode between the two motion function blocks.

Name	Value	Description
MC_BUFFERMODE_ABORTING	0	Start FB immediately (default mode)
MC_BUFFERMODE_BUFFERED	1	Start FB after current motion has finished
MC_BUFFERMODE_BLENDINGLOW	2	The velocity is blended with the lowest velocity of both FBs
MC_BUFFERMODE_BLENDINGPREVIOUS	3	The velocity is blended with the velocity of the first FB
MC_BUFFERMODE_BLENDINGNEXT	4	The velocity is blended with velocity of the second FB
MC_BUFFERMODE_BLENDINGHIGH	5	The velocity is blended with highest velocity of both FBs

3.4.1.4 Description of the fault code

Users can find the description of the fault code through the enumeration type in the data types of MotionControl library, and open it through the property window or double-click to view it

MC_ERRORCODE

View properties

MC_BUFFER_MODE

MC_COMBINE_MODE

MC_DIRECTION

MC_ERRORCODE

MC_SOURCE

MC_START_M

MC_SYNC_MO

Structures

MC_CAM_ID

MC_CAMSWIT

0008

0009

0010

0011

0012

0013

0014

0015

A

F

E

E

D

I

T

I

O

N

Enumeration: MC_ERRORCODE (ver.1.0.0, UINT)

Location: Project

Folder: MotionControlDataType\Numeratives

Name	Value	Description
NO_ERROR	0	No error
AXIS_NOT_EXIST	1	Axis ID does not exist
AXIS_TYPE_NOT_MATCH	2	Axis type does not match
AXIS_QUEUE_FULL	3	Axis queue is full
AXIS_ENCODER_OVERFLOW	4	Axis encoder overflow
AXIS_POWER_OFF	5	Axis not enabled
AXIS_POWER_ON	6	Axis already enabled
FREQUENCY_ILLEGAL	7	Frequency is illegal
AXIS_BUSY	8	Axis is busy, another function block is controlling the axis movement
FAILED_TO_BUFFER	9	Buffer mode not supported
BUFFER_MODE_ILLEGAL	16	BufferMode value is illegal
DIRECTION_ILLEGAL	17	Direction value is illegal
EXCUTION_MODE_ILLEGAL	18	ExecutionMode value is illegal
SOURCE_ILLEGAL	19	Source value is illegal
OVERRIDE_ILLEGAL	20	Override value is illegal
SHIFTING_MODE_ILLEGAL	21	Shifting mode is illegal
CONTROL_MODE_ILLEGAL	22	Control mode setting error
POS_ILLEGAL	256	Position is illegal
ACC_DEC_ILLEGAL	257	Acceleration/Deceleration is illegal
VEL_ILLEGAL	258	Velocity is illegal
CMD_P_POS_OVER_LIMIT	259	Command position exceeds positive limit in configuration file
CMD_N_POS_OVER_LIMIT	260	Command position exceeds negative limit in configuration file

You can also consult the table below

MC_ERRORCODE	Value	Meaning
NO_ERROR	0	No error
AXIS_NOT_EXIST	1	The axis ID number does not exist
AXIS_TYPE_NOT_MATCH	2	The axis type does not match

MC_ERRORCODE	Value	Meaning
AXIS_QUEUE_FULL	3	The axis queue is full
AXIS_ENCODER_OVERFLOW	4	Axis encoder overflow
AXIS_POWER_OFF	5	The axis is not enabled
AXIS_POWER_ON	6	The axis is enabled
FREQUENCY_ILLEGAL	7	The frequency is not legal
AXIS_BUSY	8	The axis is busy, and there is a function block controlling the axis movement
FAILED_TO_BUFFER	9	It cannot be added as a buffer
BUFFER_MODE_ILLEGAL	16	The BufferMode value is invalid
DIRECTION_ILLEGAL	17	The Direction value is invalid
EXCUTION_MODE_ILLEGAL	18	ExcutionMode value is illegal
SOURCE_ILLEGAL	19	The source value is invalid
OVERRIDE_ILLEGAL	20	Override value is illegal
SHIFTING_MODE_ILLEGAL	21	Shifting mode is illegal
CONTROL_MODE_ILLEGAL	22	The control mode is set incorrectly
POS_ILLEGAL	256	The location is illegal
ACC_DEC_ILLEGAL	257	The acceleration/deceleration is illegal
VEL_ILLEGAL	258	The velocity is illegal
CMD_P_POS_OVER_LIMIT	259	The command position exceeds the forward limit of the profile
CMD_N_POS_OVER_LIMIT	260	The command position exceeds the negative limit of the profile

MC_ERRORCODE	Value	Meaning
FORBIDDEN_P_POS_MOVE	261	Forward movement is prohibited
FORBIDDEN_N_POS_MOVE	262	Negative movement is prohibited
POS_LAG_OVER_LIMIT	263	The axis position following error is out of bound
VEL_LAG_OVER_LIMIT	264	The shaft speed following error is out of bound
CMD_VEL_OVER_LIMIT	265	The axis commands the speed to exceed the limit
CMD_ACC_OVER_LIMIT	266	The axis commands that the acceleration exceeds the limit
POS_INFINITY	267	The axis setting position is not legal
SOFTWARE_EMGS	494	The user is at an emergency stop
SYSTEME_MGS	495	The system is at an emergency stop
COMMUNICATION_EXCEPTION	496	Communication is abnormal
HOMING_VEL_ILLEGAL	608	The return to zero speed is illegal
HOMING_ACC_ILLEGAL	609	Acceleration back to zero is illegal
HOMING_APPR_ILLEGAL	610	The return to zero approach speed is illegal
HOMING_METHOD_ILLEGAL	611	The zero-back method is illegal
HOME_POSITION_ILLEGAL	612	The home location is illegal
HOMING_INTERRUPT_OR_NSTART	613	Homing Interrupted/Not Started
HOMING_TARGET_NOT_REACH	614	The Homing target was not reached
HOMING_ERR	615	Homing error
HOMING_ERR_V0	616	The Homing velocity is 0

MC_ERRORCODE	Value	Meaning
HOMING_TIMEOUT	617	Homing timeout
HOMING_MODE_CFG_FAILED	618	Failed to set the Homing mode
HOMING_VEL_CFG_FAILED	619	Failed to set the Homing speed
HOMING_ACC_CFG_FAILED	620	Failed to set the Homing acceleration
HOMING_APPR_CFG_FAILED	621	Failed to set Homing approach speed
HOMING_OP_MODE_CFG_FAILED	622	Failed to set the Homing mode
ENCODER_PRESET_OVERFLOW	650	Encoder preset position overflow
PLCOPEN_DISABLED	1280	PLCOpen state machine:DISABLED
PLCOPEN_STANDSTILL	1281	PLCOpen state machine:STANDSTILL
PLCOPEN_AXISHOMING	1282	PLCOpen state machine:HOMING
PLCOPEN_DISCRETEMOTION	1283	PLCOpen state machine:DISCRETEMOTION
PLCOPEN_CONTINUOUSMOTION	1284	PLCOpen state machine:CONTINUOUSMOTION
PLCOPEN_SYNCHRONIZEDMOTION	1285	PLCOpen state machine:SYNCHRONIZEDMOTION
PLCOPEN_STOPPING	1286	PLCOpen state machine:STOPPING
PLCOPEN_ERRORSTOP	1287	PLCOpen state machine:ERRORSTOP
PARAMETER_ILLEGAL	1536	The parameter is invalid
CURRENT_NOT_SUPPORTED	1537	Not at this time
CAMSWITCH_MODE_ILLEGAL	1616	CamSwitchMode illegal

MC_ERRORCODE	Value	Meaning
CAMSWITCH_DIRECTION_ILLEGAL	1617	CamSwitchDirection illegal
TRIGGERINPUT_USED	1621	Trigger input has been used
TRIGGEREVENT_FULL	1622	Trigger event is full
RIGGEREVENT_BUSY	1623	Trigger event has been used
USED_UPLICATE_AXIS	1632	The master and slave axis are the same axis
DENOMINATOR_0	1633	The gear denominator is 0
MASTER_VALUE_SOURCE_ILLEGAL	1634	MasterValueSource illegal
NO_CAMSELECTED	1637	NO CamTableID
NOT_CAMREFTYPE	1638	No cam table data
CAMTABLE_EMPTY_MASTER_RANGE	1639	The cam table master axis data is abnormal
CAM_NO_POINTER	1640	The cam table data is empty
MASTER_START_DISTANCE_ILLEGAL	1641	Cam MasterStartDistance is 0, illegal
MASTERNOTMATCH	1642	Phasing function block master axis does not match the actual master axis
ECT_SDO_READ_ERR	2048	SDO read error
ECT_SDO_WRITE_ERR	2049	SDO writing error
AXIS_UNASSOCIATED_DRIVER	4096	The axis is not associated with a device
DRIVER_NOT_READY	4097	The drive device is not ready
DRIVER_ERROR	4098	An error is reported on the driving device
DRIVER_ARRAY_FULL	4099	The number of drive devices exceeds the limit

MC_ERRORCODE	Value	Meaning
DRIVER_INDEX_OUTRANGE	4100	The driver device index is out of range
DRIVER_INDEX_USED	4101	The driver device index is already in use
DRIVER_CONTROL_MODE_UNSUPPORTED	4102	The drive device control mode is not supported
FSM_NOT_READY_TO_SWITCH_ON	4368	402 State Machine: NOT_READY_TO_SWITCH_ON
FSM_SWITCH_ON_DISABLE	4369	402 state machine: SWITCH_ON_DISABLE
FSM_READY_TO_SWITCH_ON	4370	402 state machine: READY_TO_SWITCH_ON
FSM_SWITCH_ON	4371	402 State Machine: SWITCH_ON
FSM_OPERATION_ENABLE	4372	402 State Machine: OPERATION_ENABLE
FSM_QUICK_STOP_ACTIVE	4373	402 State Machine: QUICK_STOP_ACTIVE
FSM_FAULT_REACTION_ACTIVE	4374	402 State Machine: FAULT_REACTION_ACTIVE
FSM_FAULT	4375	402 State Machine: FAULT
FSM_UNKNOW	4376	402 State Machine: UNKNOW
PERMISSION_DENIED	5000	Permission denied
GROUP_NOT_EXIST	5001	Axis group does not exist
GROUP_AXIS_ARRAY_FULL	5002	Axis group space is full
AXIS_ALREADY_IN_GROUP	5003	Axis already exists in the group
AXIS_NOT_IN_GROUP	5004	Axis group does not contain this axis
GROUP_QUEUE_FULL	5005	Axis group queue is full
GROUP_DISABLED	5006	Axis group is not enabled

MC_ERRORCODE	Value	Meaning
GROUP_AXIS_DISABLED	5007	Axis member of the group is not enabled
GROUP_AXIS_ERR	5008	Axis member error in axis group
GROUP_AXIS_0_ERR	5009	Axis 0 member error in axis group
GROUP_AXIS_1_ERR	5010	Axis 1 member error in axis group
GROUP_AXIS_2_ERR	5011	Axis 2 member error in axis group
GROUP_AXIS_3_ERR	5012	Axis 3 member error in axis group
GROUP_SYNC_AXIS_0_ERR	5013	Synchronous axis 0 member error in axis group
GROUP_SYNC_AXIS_1_ERR	5014	Synchronous axis 1 member error in axis group
GROUP_SYNC_AXIS_2_ERR	5015	Synchronous axis 2 member error in axis group
GROUP_AXIS_0_HOME_FAILED	5016	Axis 0 homing failed in axis group
GROUP_AXIS_1_HOME_FAILED	5017	Axis 1 homing failed in axis group
GROUP_AXIS_2_HOME_FAILED	5018	Axis 2 homing failed in axis group
GROUP_AXIS_3_HOME_FAILED	5019	Axis 3 homing failed in axis group
GROUP_SYNC_AXIS_0_HOME_FAILED	5020	Synchronous axis 0 homing failed in axis group
GROUP_SYNC_AXIS_1_HOME_FAILED	5021	Synchronous axis 1 homing failed in axis group
GROUP_SYNC_AXIS_2_HOME_FAILED	5022	Synchronous axis 2 homing failed in axis group
GROUP_AXIS_MODULO_NOT_ALLOWED	5023	Axis group member cannot be a modal axis
GROUP_FSM_DISABLED	5100	PLCOpen state machine: DISABLED

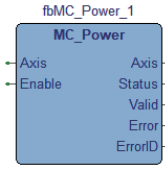
MC_ERRORCODE	Value	Meaning
GROUP_FSM_STANDBY	5101	PLCOpen state machine: STANDBY
GROUP_FSM_HOMING	5102	PLCOpen state machine: HOMING
GROUP_FSM_MOVING	5103	PLCOpen state machine: MOVING
GROUP_FSM_STOPPING	5104	PLCOpen state machine: STOPPING
GROUP_FSM_ERRORSTOP	5105	PLCOpen state machine: ERRORSTOP
KINEMATIC_DEFINED	5140	Kinematic transformation defined
KINEMATIC_NOT_ALLOWED	5141	Motion mechanism type not allowed
KIN_TRANSFORM_NOT_CONSTANT	5142	Motion mechanism transformation is not unique
GROUP_AIMPOUT_LIMIT	5150	Target position exceeds limit
GROUP_KIN_PARAMERROR	5151	Kinematic parameter limit is unreasonable
GROUP_TRANS_MODEERROR	5152	Transition mode error
GROUP_CIRCLEPARAERROR	5153	Arc mode parameter error
GROUP_PLANFAIL	5154	S-curve planning failed
GROUP_PARAM_ERROR	5155	Parameter error
GROUP_ACC_DEC_ILLEGAL	5156	Acceleration/deceleration invalid
GROUP_JERK_ILLEGAL	5157	Jerk invalid
GROUP_AIMEQUAL	5158	Target position overlaps
GROUP_BLENDING_ERROR	5159	Trajectory blending error
GROUP_MOTION_LIMIT_AVOUT	5160	Joint angle and angular acceleration exceed limits during motion

MC_ERRORCODE	Value	Meaning
G R O U P _ M O - TION_LIMIT_POSOUT	5161	Joint position exceeds limit during motion
GROUP_PLAN_ER- ROR	5162	Axis group planning error
GROUP_INTERPO- LATE_ERROR	5180	Axis group interpolator error
GROUP_SYNC_PO- SLAGOVERLIMIT	5181	Synchronous axis follow- ing error exceeds limit
GROUP_SYNC_C- MDPSLAGOVER- LIMIT	5182	Unsupported parameter combination
G R O U P _ U N A - V A L I D _ C O M - B I N E _ P A R A M	5183	Axis group interrupted motion does not exist
GROUP_INTER- RUPT_NODE_NOT_ EXIST	5184	Synchronous axis fol- lowing error exceeds limit

3.4.2 Basic

3.4.2.1 MC_Power

3.4.2.1.1 Description of the function block interface

Instruc- tions	LD Performance	ST Performance
MC_ Power		<pre>fbMC_Power_1(Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Status=>(* BOOL *), Valid=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));</pre>

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Enable	BOOL	As long as 'Enable' is true, power is being enabled.

Pins Type	Name	Type	Description
VAR_OUTPUT	Status	BOOL	Effective state of the power stage
	Valid	BOOL	If true, a valid set of out- puts is available at the FB
	Error	BOOL	Signals that an error has occurred within the Function Block
	Error ID	WORD	ErrorID. See MotionCon- trol/DataType/Numerati- ves/MC_ERRORCODE

3.4.2.1.2 Instructions for the use of function block
MC_Power commands are applied to bus servo
axes and virtual axes. It's used to enable the axes.

- **Directions for use**

- Pin Description:

When Enable is True, the axis enters the enabled
state, and the PLCOpen state machine of the axis
enters the StandStill state from the Disabled state,
and the axis motion command can be executed

When Enable is False, the axis exits the enabled
state, which will interrupt the currently execut-
ing motion command, and after the axis is disen-
abled, the motion command cannot be execut-
ed, and the management command can still be
executed, and the fault reset can be performed,
and the parameters can be written and read.

When there is a error in the axis, the re-enable
MC_Power cannot control the entry of the axis to
enable, and it is necessary to call MC_Reset first
to reset the shaft fault

Valid, the signal is temporarily set to False and
then True, which means that the current output
pin status of the function block is credible;

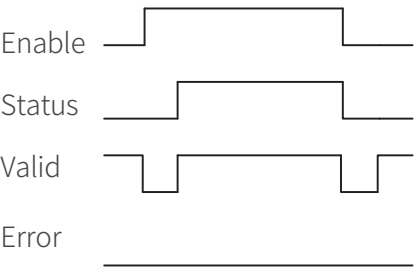
Status, If the axis is enabled successfully, set
True after Valid;

- Multiple calls

When multiple MC_Power commands control the
same axis object, the input and output of the last
MC_Power executed during the week shall prevail.

• Timing diagrams

Using a MC_Power command, the timing diagram of the axis enabled normally is as follows:



3.4.2.2 MC_Home

3.4.2.2.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_Home		<pre>fbMC_Home_1(Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Position:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(*MC_ERRORCODE*));</pre>

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the motion at rising edge
	Position	REAL	Absolute position when the reference signal is detected [u]

Pins Type	Name	Type	Description
VAR_OUTPUT	Buffer-Mode	MC_BuferMode	Defines the chronological sequence of the FB
	Done	BOOL	Reference known and set sucessfully
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis
	Command Aborted	BOOL	'Command' is aborted by switching off power (only possibility to abort)
	Error	BOOL	Signals that an error has occurred within the Function Block
	Error ID	WORD	ErrorID. See MotionControl/DataType/Numeratives/MC_ERRORCODE

3.4.2.2.2 Instructions for the use of function block

The MC_Home command is applicable to the bus servo axes and the virtual axes, and the origin return function is realized, and the rising edge of the command is valid

• Directions for use

The axis object needs to be enabled with the MC_Power command before it can be invoked.

When the bus servo axis execete this command, the user can set the Homing Method on the axis configuration interface

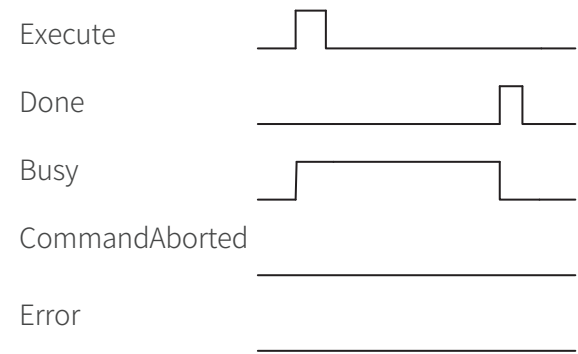
When the virtual axis invokes this command, it uses the CiA402 protocol to return to zero in mode 35

• Pin Description:

Execute is triggered, the rising edge is entered to trigger the zero return command, and the axis PLCOpen state machine enters the Homing state Zero Position, the zero offset given by the user, after the zero return is executed, the current position will be set to the value given by the user.

• Timing diagrams

MC_Home function block is triggered, the axis object executes the zeroing action, and the sequence diagram of the normal function execution is as follows:



3.4.2.3 MC_Stop

3.4.2.3.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_Stop		fbMC_Stop_1 (Axis := (* AXIS_REF *), Execute := (* BOOL *), Deceleration := (* REAL *), Jerk := (* REAL *), Done => (* BOOL *), Busy => (* BOOL *), CommandAborted=>(* BOOL *), Error => (* BOOL *), ErrorID=>(* MC_ERRORCODE *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the motion at rising edge
	Deceleration	REAL	Value of the ‘Deceleration’ [u/(s^2)]
	Jerk	REAL	Value of the ‘Jerk’ [u/(s^3)]

Pins Type	Name	Type	Description
VAR_OUTPUT	Done	BOOL	Zero velocity reached
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Command Aborted	BOOL	‘Command’ is aborted by switching off power (only possibility to abort)
	Error	BOOL	Signals that an error has occurred within the Function Block
	Error ID	WORD	ErrorID. See MotionControl/DataType/Numeratives/MC_ERRORCODE

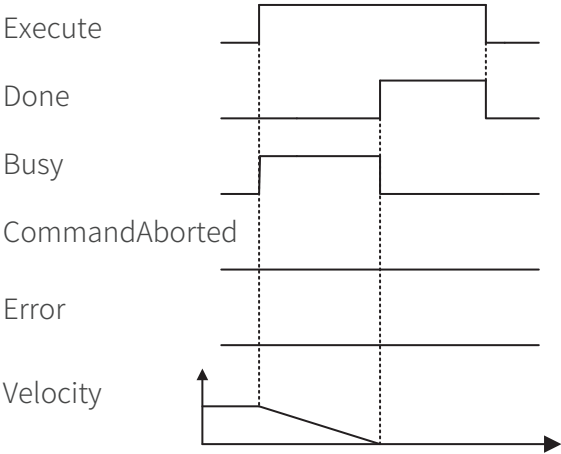
3.4.2.3.2 Instructions for the use of function block
MC_Stop command is applicable to bus servo axis and virtual axis, to achieve the stop function, and the command rising edge is valid

• Directions for use

The axis object needs to be enabled using the MC-Power instruction before it can be called.
The deceleration and acceleration given by the user will be latched at the rising edge of Execute as parameters for this stopping action.
After entering the rising edge of Execute, if Flase is not set and remains at a high level, the Done signal will remain valid after deceleration is completed, and the PLCOpen state machine of the current axis will remain in the Stopping state Afterwards, set Execute to False, and the axis state machine will switch from Stopping to Standstill state

• Timing diagrams

MC_Stop interrupt the motion sequence of MC_MoveVelocity as shown in the figure below



3.4.2.4 MC_Halt

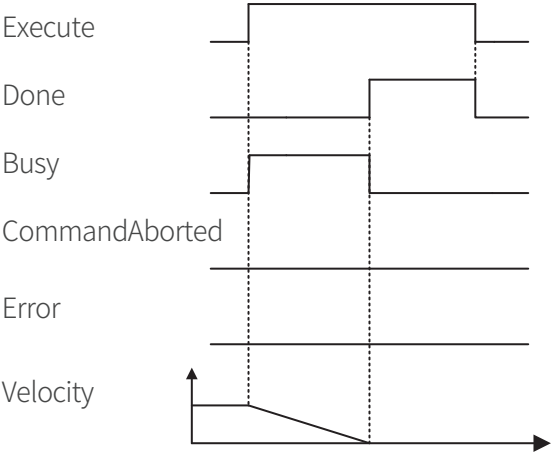
3.4.2.4.1 Description of the function block interface

Instructions	LD Performance	ST Performance	
MC_Halt		<pre>fbMC_Halt_1 (Axis := (* AXIS_REF *), Execute := (* BOOL *), Deceleration := (* REAL *), Jerk := (* REAL *), BufferMode := (* MC_BUFFER_MODE *), Done => (* BOOL *), Busy => (* BOOL *), Active => (* BOOL *), CommandAborted => (* BOOL *), Error => (* BOOL *), ErrorID => (* MC_ERRORCODE *));</pre>	
Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the action at rising edge
	Deceleration	REAL	Value of the 'Deceleration' [u/(s^2)]

Pins Type	Name	Type	Description
	Jerk	REAL	Value of the 'Jerk' [u/(s^3)]
	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB
VAR_OUTPUT	Done	BOOL	Zero velocity reached
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	Error ID	MC_ERRORCODE	ErrorID. See MotionControl/Data Type/Numeratives/MC_ERRORCODE

3.4.2.4.2 Instructions for the use of function block MC_Halt command is applicable to bus servo axis and virtual axis, to achieve the pause function, and the command rising edge is valid

- **Directions for use**
The axis object needs to be enabled using the MC-Power instruction before it can be called. Command rising edge latch, deceleration, acceleration, etc
After the function block is triggered, it takes effect and the axis state machine switches to the Discrete motion state. This state means that other instructions that comply with the PLCOpen state machine switching can interrupt this instruction.
- **Timing diagrams**
MC_Halt interrupt MC_MoveVelocity instruction sequence diagram is shown in the image



3.4.2.5 MC_MoveAbsolute

3.4.2.5.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_MoveAbsolute		<pre> fbMC_MoveAbsolute_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Position:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), Direction:= (* MC_DIRECTION *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *)); </pre>

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the motion at the rising edge
	Position	REAL	Commanded 'Position' for the motion (in technical unit [u]) (negative or positive)
	Velocity	REAL	Value of the maximum 'Velocity' (not necessarily reached) [u/s].
	Acceleration	REAL	Value of the 'Acceleration' (always positive) (increasing energy of the motor) [u/(s^2)]
	Deceleration	REAL	Value of the 'Deceleration' (always positive) (decreasing energy of the motor) [u/(s^2)]
	Jerk	REAL	Value of the 'Jerk' [u/(s^3)]. (always positive)
	Direction	MC_DIRECTION	Enum type (1-of-4 values: mcPositiveDirection, mcShortestWay, mcNegativeDirection, mcCurrentDirection)
VAR_OUTPUT	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB
	Done	BOOL	Commanded position finally reached
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block

Pins Type	Name	Type	Description
	Error ID	WORD	ErrorID. See MotionControl/DataType/Numeratives/MC_ERRORCODE

3.4.2.5.2 Instructions for the use of function block MC_MoveAbsolute command is applicable to bus servo axis and virtual axis to achieve absolute positioning function, and the command rising edge is valid

• **Directions for use**

The axis object needs to be enabled using the MC-Power instruction before it can be called.
When the rising edge of Execute is triggered, the input parameters are latched, triggering the absolute positioning function, and the axis state machine switches to Discrete motion state

• **Pin Description:**

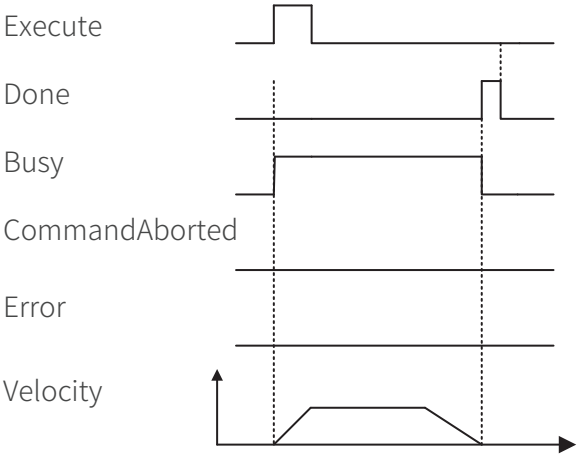
Position: The target position of absolute positioning, the way the axis reaches the target position is determined by the Direction input
Velocity: The maximum velocity during absolute positioning
Acceleration: The maximum acceleration during absolute positioning
Deceleration: The maximum deceleration during absolute positioning
Jerk: The maximum jerk acceleration during positioning
The shape of the axial velocity curve is determined by Velocity, Acceleration, Deceleration, and Jerk together

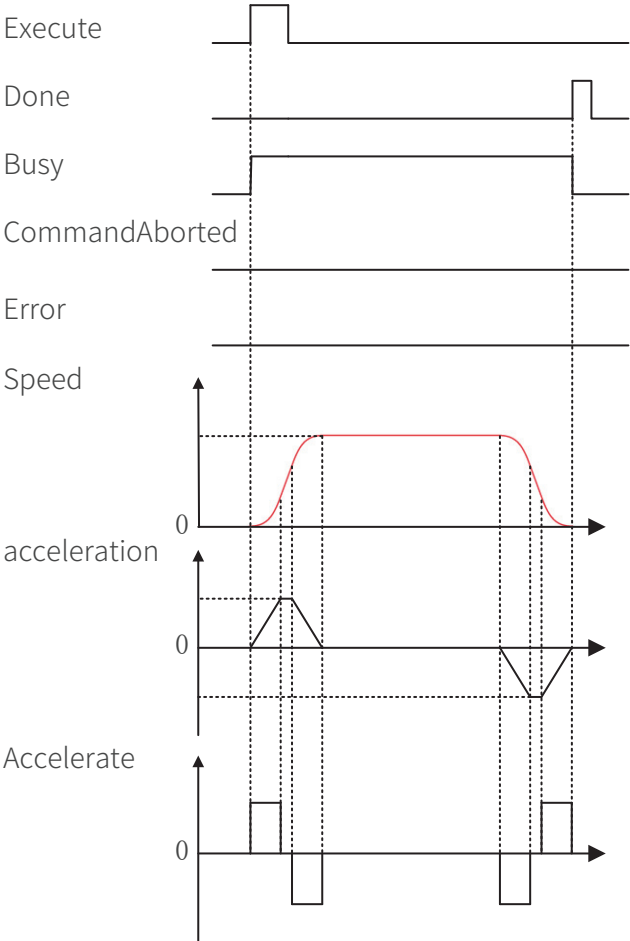
Direction direction: Contains four options that do not work when the axis is linear (in absolute mode, the direction of motion of the linear axis is determined based on the current position and the target position), while in the case of the modular axis, the same target position can be reached through different directions, divided into the following four situations

- 0-mcPositive Direction: Forward motion, reaching the target position in the positive direction along the axis
 - 1-mcShortestWay: shortest distance, the axis moves in the direction with the shortest distance to reach the target position
 - 2-mecNegativeDirection: Reverse motion, reaching the target position in the opposite direction of axial motion
 - 3-mcCurrentDirection: Reach the target position along the current direction
- If the target position of the modulus axis is greater than the modulus value, the target position is modulo the modulus value to obtain the available target position

• **Timing diagrams**

The timing diagram for calling the MC_SoveAbsolute instruction in the Standstill state of the axis can be referred to as shown in the figure





3.4.2.6 MC_MoveRelative

3.4.2.6.1 Description of the function block interface

Instruc-tions	LD Performance	ST Performance
MC_MoveRelative		<pre>fbMC_MoveRelative_1(Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:=(* MC_BUFFER_MODE *)</pre>

Instruc-tions	LD Performance	ST Performance
		<pre>BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID:=(* MC_ERRORCODE *)</pre>

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the motion at rising edge
	Distance	REAL	Relative distance for the motion (in technical unit [u])
	Velocity	REAL	Value of the maximum velocity (not necessarily reached) [u/s]
	Acceleration	REAL	Value of the acceleration (increasing energy of the motor) [u/(s^2)]
	Deceleration	REAL	Value of the deceleration (decreasing energy of the motor) [u/(s^2)]
	Jerk	REAL	Value of the Jerk [u/(s^3)]
VAR_OUTPUT	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB
	Done	BOOL	Commanded distance reached
	Busy	BOOL	The FB is not finished and new output values are to be expected

Pins Type	Name	Type	Description
	Active	BOOL	Indicates that the FB has control on the axis
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

3.4.2.6.2 Instructions for the use of function block

The MC_moveRelative instruction is suitable for bus servo axis and virtual axis, achieving relative positioning, and the rising edge of the instruction is effective

• Directions for use

The axis object needs to be enabled using the MC-Power instruction before it can be called.

When the rising edge of Execute is triggered, the input parameters are latched, triggering the absolute positioning function, and the axis state machine switches to Discrete motion state

• Pin Description:

Distance target distance: the relative movement distance of the positioning, in linear or modular mode of the axis, Distance is positive, the axis runs in the forward direction, Distance is negative, the axis runs in the reverse direction, and the running distance is | Distance | the specified distance

Velocity: The maximum speed during the positioning process

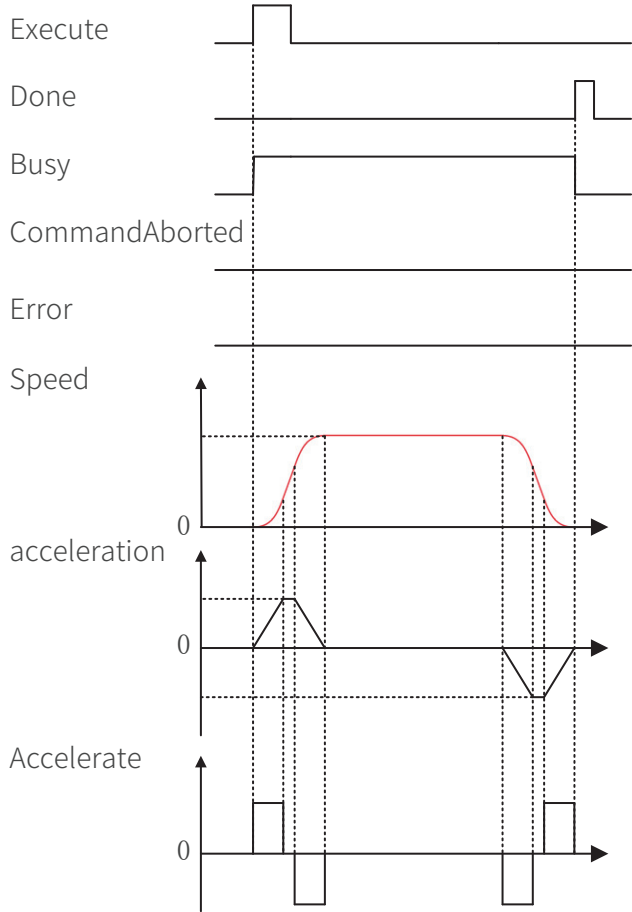
Acceleration: The maximum acceleration during the positioning process

Deceleration deceleration: the maximum deceleration during the positioning process

Jerk Acceleration: The maximum acceleration during the positioning process

The shape of the axial velocity curve is determined by Velocity, Acceleration, Deceleration, and Jerk together

• Timing diagrams



3.4.2.7 MC_MoveAdditive

3.4.2.7.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_MoveAdditive	fbMC_MoveAdditive_1 MC_MoveAdditive Axis Execute Distance Velocity Acceleration Deceleration Jerk BufferMode Axis Done Busy Active CommandAborted Error ErrorID	fbMC_MoveAdditive_1(Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *),

Instructions	LD Performance	ST Performance
		Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE*));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the motion at rising edge
	Distance	REAL	Relative distance for the motion (in technical unit [u])
	Velocity	REAL	Value of the maximum velocity (not necessarily reached) [u/s]
	Acceleration	REAL	Value of the acceleration (increasing energy of the motor) [u/(s^2)]
	Deceleration	REAL	Value of the deceleration (decreasing energy of the motor) [u/(s^2)]
	Jerk	REAL	Value of the Jerk [u/(s^3)]
	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB
VAR_OUTPUT	Done	BOOL	Commanded distance reached
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis

Pins Type	Name	Type	Description
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ERRORCODE

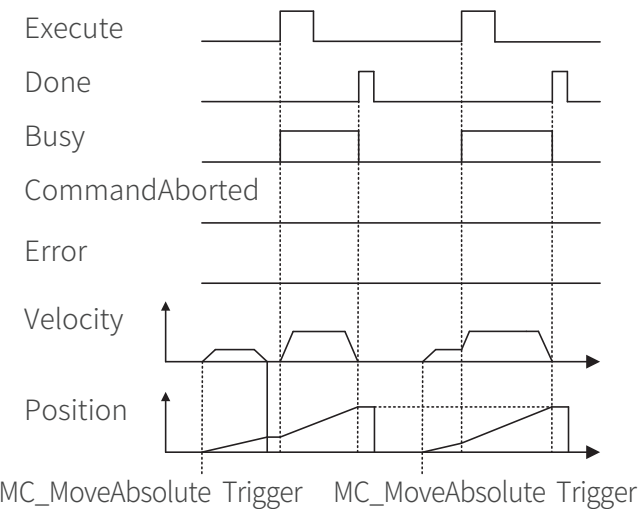
3.4.2.7.2 Instructions for the use of function block

The MC-MoveAdditive instruction is applicable to bus servo axes and virtual axes, realizing the superposition of distance on the basis of the previous discrete motion target position in the Discrete motion state machine. If the previous discrete motion is interrupted, the running result is still the superposition of the previous instruction target position and the MC-MoveAdditive target distance, and the rising edge of the instruction is valid

- **Directions for use**
The axis object needs to be enabled using the MC-Power instruction before it can be called.
- **Pin Description:**
Distance target distance: This distance will be superimposed on the previous discrete motion target position, whether it interrupts the previous motion or performs superimposed motion after the previous motion ends. When the function is executed, the axis motion distance is the superimposed value of the previous discrete motion target position and Distance.
Velocity: The maximum speed during the positioning process
Acceleration: The maximum acceleration during the positioning process
Deceleration deceleration: the maximum deceleration during the positioning process

Jerk Acceleration: The maximum acceleration during the positioning process
The shape of the axial velocity curve is determined by Velocity, Acceleration, Deceleration, and Jerk together

• Timing diagrams



3.4.2.8 MC_MoveSuperImposed
3.4.2.8.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_MoveSuperImposed		<pre>fbMC_MoveSuperimposed_1(Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), VelocityDiff:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));</pre>

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the motion at rising edge
	Distance	REAL	Additional distance that is to be superimposed (in technical unit [u])
	Velocity Diff	REAL	Value of the velocity difference of the additional motion (not necessarily reached) [u/s]
	Acceleration	REAL	Value of the acceleration (increasing energy of the motor) [u/s^2]
	Decel-eration	REAL	Value of the decelera-tion (decreasing energy of the motor) [u/s^2]
	Jerk	REAL	Value of the Jerk [u/s^3]
VAR_OUTPUT	Done	BOOL	Additional distance superimposed to the ongoing motion
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

3.4.2.8.2 Instructions for the use of function block
The MC-MoveSuperposed instruction is suitable for both bus servo axes and virtual axes, and is typically used to stack an incremental position control motion on top of a basic motion control function block (such as MC-MoveRelative). The rising

edge of the instruction is effective

• **Directions for use**

The axis object needs to be enabled using the MC-Power instruction before it can be called.

If MC_moveSuperposed is active, any command in abort mode except MC_moveSuperposed will abort two motion commands: MC_moveSuperposed and the underlying motion command. In any other mode, the underlying motion commands will not be aborted.

If MC-MoveSuperposed is active and another MC-MoveSuperposed is commanded, only the ongoing MC-MoveSuperposed command will be aborted and replaced by the new MC-MoveSuperposed, but the underlying motion command will not be aborted.

The MC_SoveSuperposed function will cause changes in the speed of the ongoing motion in all relevant states, as well as changes in the position of commands if applicable.

In the "stationary" state, the effect of FB MC-MoveSuperposed is similar to MC-MoveRelative.

The values of "acceleration", "deceleration", and "jerk" are additional values of the ongoing motion, not absolute values. In this way, regardless of whether MC-MoveSuperposed FB occurs simultaneously or not, the underlying FB always completes its work within the same time period. The behavior of outputting 'activities' is different from buffering FBs.

• **Pin Description:**

Distance target distance: the additional distance obtained by superimposing motion.

VelocityDiff velocity: the maximum difference in velocity during superimposed motion

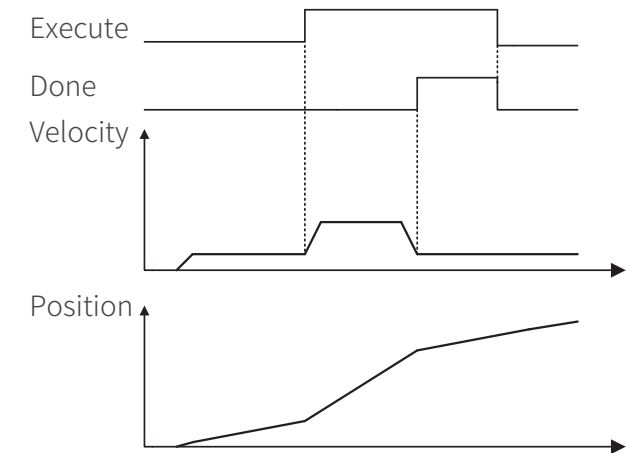
Acceleration: The maximum acceleration during the positioning process

Deceleration deceleration: the maximum deceleration during the positioning process

Jerk Acceleration: The maximum acceleration during the positioning process

The shape of the axial velocity curve is determined by Velocity, Acceleration, Deceleration, and Jerk together

• **Sequence diagram**



3.4.2.9 MC_HaltSuperImposed

3.4.2.9.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_Halt Super Imposed		fbMC_HaltSuperimposed_1(Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the action at rising edge
	Decel-eration	REAL	Value of the decelera- tion [u/s^2]
	Jerk	REAL	Value of the Jerk [u/s^3]

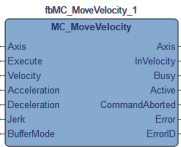
Pins Type	Name	Type	Description
VAR_OUTPUT	Done	BOOL	Superimposed motion halted
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ERRORCODE	ErrorID. See Motion-Control/DataType/Numeratives/MC_ERRORCODE

3.4.2.9.2 Instructions for the use of function block

The instruction is applicable to the bus servo axis and virtual axis, and executing MC_HaltSuperposed can just stop the motion of MC_SoveSuperposed without affecting the progress of the basic motion. This is different from executing MC_Stop, which stops all movements, including base and overlay movements, with the rising edge of the instruction being valid

3.4.2.10 MC_MoveVelocity

3.4.2.10.1 Description of the function block interface

Instruc-tions	LD Performance	ST Performance
MC_MoveVelocity		fbMC_MoveVelocity_1(Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *),

Instruc-tions	LD Performance	ST Performance
		BufferMode:= (* MC_BUFFER_MODE *), InVelocity=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *);

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the motion at the rising edge
	Velocity	REAL	Value of the maximum 'Velocity' (not necessarily reached) [u/s].
	Acceleration	REAL	Value of the 'Acceleration' (always positive) (increasing energy of the motor) [u/(s^2)] acceleration[u/s2]
	Decel-eration	REAL	Value of the 'Decelera-tion' (always positive) (decreasing energy of the motor) [u/(s^2)]
	Jerk	REAL	Value of the 'Jerk' [u/(s^3)]. (always positive)
	Buffer-Mode	MC_BUFF-ER_MODE	Defines the chronolog-ical sequence of the FB
VAR_OUTPUT	InVeloc-ity	BOOL	Commanded velocity reached
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis

Pins Type	Name	Type	Description
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ERRORCODE

3.4.2.10.2 Instructions for the use of function block

The MC_SoveVelocity instruction is applicable to both bus servo axes and virtual axes, enabling continuous motion at a user specified speed with a valid rising edge of the instruction.

• Directions for use

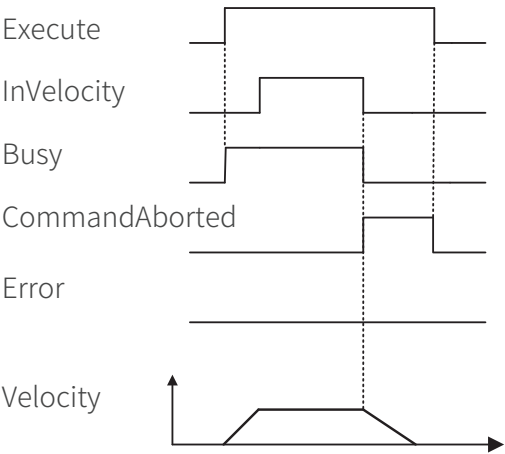
The axis object needs to be enabled using the MC-Power instruction before it can be called.

Execute input rising edge, instruction latches input parameters such as velocity and accuracy

Velocity, Acceleration, and Jerk jointly determine the shape of the curve

Switch the axis state machine to ContinuousMotion state

• Timing diagrams



3.4.2.11 MC_MoveContinuousAbsolute

3.4.2.11.1 Description of the function block interface

Instruc-tions	LD Performance	ST Performance
MC_MoveContinuousAbsolute		fbMC_MoveContinuousAbsolute_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Position:=(* REAL *), EndVelocity:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), Direction:=(* MC_DIRECTION *), BufferMode:=(* MC_BUFFER_MODE *), InEndVelocity=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the motion at the rising edge
	Position	REAL	Commanded 'Position' for the motion (in technical unit [u]) (negative or positive)
	EndVelocity	REAL	Value of the end velocity [u/s]. Signed value
	Velocity	REAL	Value of the maximum 'Velocity' (not necessarily reached) [u/s].

Pins Type	Name	Type	Description
	Acceleration	REAL	Value of the 'Acceleration' (always positive) (increasing energy of the motor) [u/(s ²)]
	Deceleration	REAL	Value of the 'Deceleration' (always positive) (decreasing energy of the motor) [u/(s ²)]
	Jerk	REAL	Value of the 'Jerk' [u/(s ³)]. (always positive)
	Direction	MC_DIRECTION	Enum type (1-of-4 values: mcPositiveDirection, mcShortestWay, mcNegativeDirection, mcCurrentDirection)
	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB
VAR_OUTPUT	InEnd Velocity	BOOL	Commanded distance reached and running at requested end velocity
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

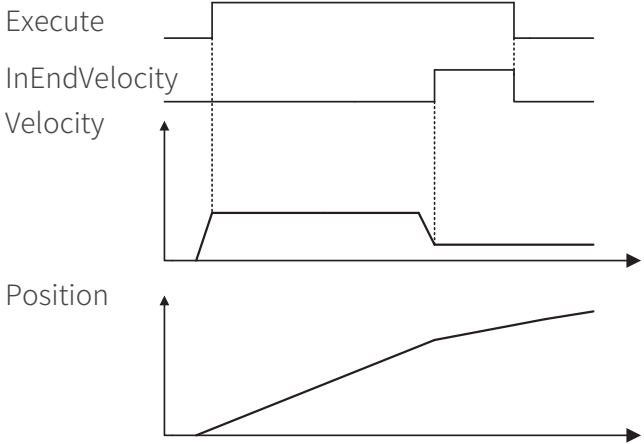
3.4.2.11.2 Instructions for the use of function block MC_MoveContinuousAbsolute command is applied to the bus servo axis and the virtual axis, to achieve a section of absolute positioning, after the completion of the set speed will continue to move, the command rising edge is valid

- **Directions for use**

The axis object needs to be enabled with the MC_Power command before it can be invoked.

The axis state machine switches from Stopping to Standstill and calls this command to enter the ContinuousMotion state

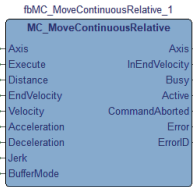
- **Timing diagrams**



3.4.2.12 MC_MoveContinuousRelative

3.4.2.12.1 Description of the function block interface

Instructions	LD Performance	ST Performance
		fbMC_MoveContinuous-Relative_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), EndVelocity:=(* REAL *), Velocity:=(* REAL *),

Instructions	LD Performance	ST Performance
MC_MoveContinuousRelative		Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:=(* MC_BUFFER_MODE *), InEndVelocity=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *);

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start the motion at the rising edge
	Distance	REAL	Relative distance for the motion (in technical unit [u])
	End Velocity	REAL	Value of the end velocity [u/s]. Signed value
	Velocity	REAL	Value of the maximum 'Velocity' (not necessarily reached) [u/s].
	Acceleration	REAL	Value of the 'Acceleration' (always positive) (increasing energy of the motor) [u/(s^2)]
	Deceleration	REAL	Value of the 'Deceleration' (always positive) (decreasing energy of the motor) [u/(s^2)]
	Jerk	REAL	Value of the 'Jerk' [u/(s^3)]. (always positive)
	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB

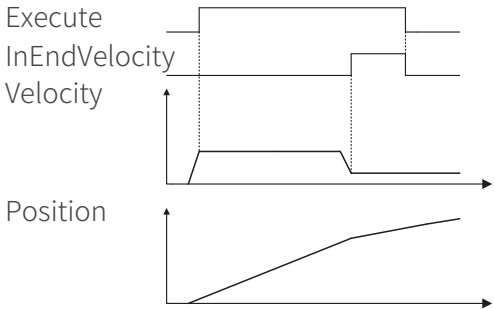
Pins Type	Name	Type	Description
VAR_OUTPUT	InEnd Velocity	BOOL	Commanded distance reached and running at requested end velocity
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See MotionControl/DataType/Numeratives/MC_ERRORCODE

3.4.2.12.2 Instructions for the use of function block MC_MoveContinuousRelative command is applied to the bus servo axis and the virtual axis, to achieve a relative positioning, after the completion of the continuous movement at a given speed, the command rising edge is valid

• **Directions for use**

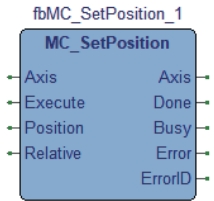
The axis object needs to be enabled with the MC_Power command before it can be invoked. The axis state machine switches from Stopping to Standstill and calls this command to enter the ContinuousMotion state

• **Timing diagrams**



3.4.2.13 MC_SetPosition

3.4.2.13.1 Description of the function block interface

Instruc-tions	LD Performance	ST Performance
MC_Set Position		fbMC_SetPosition_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Position:=(* REAL *), Relative:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Start setting position in axis
	Position	REAL	Position unit [u] (Means 'Distance' if 'Relative'= TRUE)
	Relative	BOOL	'Relative' distance if True, 'Absolute' position if False (= Default)
VAR_OUTPUT	Done	BOOL	'Position' has new value
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

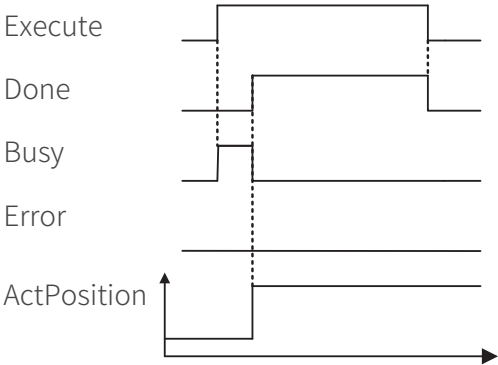
3.4.2.13.2 Instructions for the use of function block MC_SetPosition command is applied to bus servo axes and virtual axes, which are used to set the position of the motion control axes, and the command rising edge is valid

• Directions for use

After setting the position MC_SetPosition, the current position changes, but this position does not change the servo encoder count value, only as the position reference value of the current motion running, if the power is off and powered on, the previous position reference value will be lost, and the user needs to set the reference position again

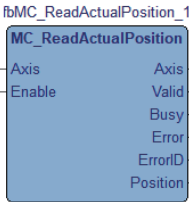
The function block does not need to be enabled by the axis to trigger the function

• Timing diagrams



3.4.2.14 MC_ReadActualPosition

3.4.2.14.1 Description of the function block interface

Instruc-tions	LD Performance	ST Performance
MC_Read Actual Position		fbMC_ReadActualPosi-tion_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *),

Instructions	LD Performance	ST Performance
		Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), Position=>(* REAL *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Enable	BOOL	Get the value of the parameter continuously while enabled
VAR_OUTPUT	Valid	BOOL	A valid output is available at the FB
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE
	Position	REAL	New absolute position (in axis' unit [u])

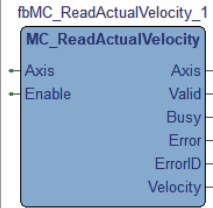
3.4.2.14.2 Instructions for the use of function block MC_ReadActualPosition commands are applied to bus servo axes and virtual axes, read the actual position, and the command is valid at a high level

• **Directions for use**

MC_ReadActualPosition function block can read the position value normally after the axis initialization is completed, and the reading value is NAN after the initialization is completed
The function block does not need to be enabled by the axis to take effect

3.4.2.15 MC_ReadActualVelocity

3.4.2.15.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_Read Actual Velocity		fbMC_ReadActualVelocity_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid:=(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), Velocity=>(* REAL *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Enable	BOOL	Get the value of the parameter continuously while enabled
VAR_OUTPUT	Valid	BOOL	A valid output is available at the FB
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE
	Velocity	REAL	The value of the actual velocity (in axis' unit [u/s])

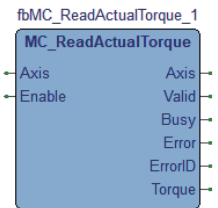
3.4.2.15.2 Instructions for the use of function block MC_ReadActualVelocity command is applied to bus servo axes and virtual axes, read the actual speed of the axis, and the command is valid at a high level

• **Directions for use**

MC_ReadActualVelocity function block can read the speed value normally only after the axis initialization is completed, and the reading value is NAN after the initialization is completed
The function block does not need to be enabled by the axis to take effect

3.4.2.16 MC_ReadActualTorque

3.4.2.16.1 Description of the function block interface

Instruc-tions	LD Performance	ST Performance
MC_Read Actual Torque		fbMC_ReadActual-Torque_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *), Torque=>(* REAL *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Enable	BOOL	Get the value of the parameter continuously while enabled
VAR_OUTPUT	Valid	BOOL	A valid output is available at the FB
	Busy	BOOL	The FB is not finished and new output values are to be expected

Pins Type	Name	Type	Description
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ER-RORCODE	ErrorID. See Motion-Control/DataType/Nu-meratives/MC_ER-RORCODE
	Torque	REAL	The value of the actual torque or force (in technical units)

3.4.2.16.2 Instructions for the use of function block MC_ReadActualTorque command is suitable for bus servo axes, which can read the actual torque in real time, and the command is valid at a high level

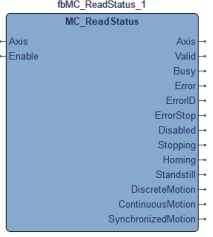
• **Directions for use**

MC_ReadActualTorque function block can read the torque normally only after the shaft initialization is completed, and the reading value is NAN after the initialization is completed
When the instruction is applied to the virtual axis, the function block does not report an error, but the output torque is always 0.
The function block does not need to be enabled by the axis to take effect

3.4.2.17 MC_ReadStatus

3.4.2.17.1 Description of the function block interface

Instruc-tions	LD Performance	ST Performance
		fbMC_ReadStatus_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *),

Instructions	LD Performance	ST Performance
MC_Read Status		ErrorID=> (* MC_ERRORCODE *), ErrorStop=>(* BOOL *), Disabled=>(* BOOL *), Stopping=>(* BOOL *), Homing=>(* BOOL *), Standstill=>(* BOOL *), DiscreteMotion=>(* BOOL *), ContinuousMotion=> (* BOOL *), SynchronizedMotion=> (* BOOL *));

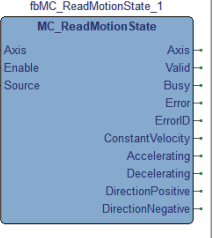
Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Enable	BOOL	Get the value of the parameter continuously while enabled
VAR_OUTPUT	Valid	BOOL	A valid set of outputs is available at the FB
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/Data Type/Numeratives/MC_ERRORCODE
	ErrorStop	BOOL	See state diagram
	Disabled	BOOL	See state diagram
	Stopping	BOOL	See state diagram
	Homing	BOOL	See state diagram
	Standstill	BOOL	See state diagram
	Discrete-Motion	BOOL	See state diagram

Pins Type	Name	Type	Description
	Continuous Motion	BOOL	See state diagram
	Synchronized-Motion	BOOL	See state diagram

3.4.2.17.2 Instructions for the use of function block MC_ReadStatus commands are available for bus servo axes and virtual axes, and the PLCOpen state machine for reading axes is valid at a high level

3.4.2.18 MC_ReadMotionState

3.4.2.18.1 Description of the function block interface

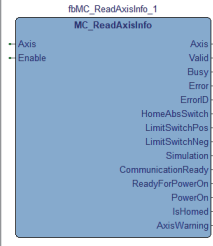
Instructions	LD Performance	ST Performance
MC_Read Motion State		fbMC_ReadMotionState_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Source:=(* MC_SOURCE *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), ConstantVelocity=> (* BOOL *), Accelerating=>(* BOOL *), Decelerating=>(* BOOL *), DirectionPositive=> (* BOOL *), DirectionNegative=> (* BOOL *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Enable	BOOL	Get the value of the parameter continuously while enabled
	Source	MC_SOURCE	Defines the source of the relevant data: mcCommandedValue, mcSetValue, mcActualValue.
VAR_OUTPUT	Valid	BOOL	True if a valid set of outputs available
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function block
	ErrorID	MC_ERRORCODE	ErrorID. See Motion-Control/DataType/Numeratives/MC_ERRORCODE
	Constant-Velocity	BOOL	Velocity is constant. Velocity may be 0. For the actual value a window is applicable (window is vendor specific)
	Accelerating	BOOL	Increasing the absolute value of the velocity
	Decelerating	BOOL	Decreasing the absolute value of the velocity
	Direction-Positive	BOOL	Signals that the position is increasing
	Direction-Negative	BOOL	Signals that the position is decreasing

3.4.2.18.2 Instructions for the use of function block MC_ReadMotionState command is applied to the bus servo axis and the virtual axis, returning the current state of the axis in motion, and the command is valid at a high level

3.4.2.19 MC_ReadAxisInfo

3.4.2.19.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_ReadAxisInfo		fbMC_ReadAxisInfo_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), HomeAbsSwitch=> (* BOOL *), LimitSwitchPos=> (* BOOL *), LimitSwitchNeg=> (* BOOL *), Simulation=>(* BOOL *), Communication Ready=> (* BOOL *), ReadyForPowerOn=> (* BOOL *), PowerOn=>(* BOOL *), IsHomed=>(* BOOL *), AxisWarning=>(* BOOL *));

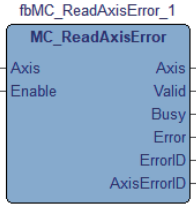
Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Enable	BOOL	Get the axis information constantly while enabled

Pins Type	Name	Type	Description
VAR_OUTPUT	Valid	BOOL	True if a valid set of outputs is available
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ER-RORCODE	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE
	Home-AbsSwitch	BOOL	Digital home switch input is active
	LimitSwitchPos	BOOL	Positive hardware end switch is active
	LimitSwitchNeg	BOOL	Negative hardware end switch is active
	Simulation	BOOL	Axis is in simulation mode (e.g. motor is simulated)
	Communication-Ready	BOOL	“Network” is initialized and ready for communication
	ReadyFor-PowerOn	BOOL	Drive is ready to be enabled (power on)
	Power On	BOOL	If TRUE shows that the power stage is switched ON
	IsHomed	BOOL	The absolute reference position is known for the axis (axis is homed)
	Axis Warning	BOOL	Warning(s) on the axis is present

3.4.2.19.2 Instructions for the use of function block MC_ReadAxisInfo commands are applied to bus servo axes and virtual axes, and read axis-related information, such as mode, enable status, etc., and the command is valid at a high level

3.4.2.20 MC_ReadAxisError

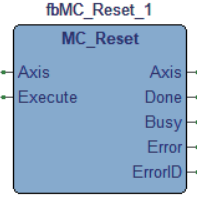
3.4.2.20.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_Read Axis Error		fbMC_ReadAxisError_1 (Axis:=(* AXIS_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *), AxisErrorID=>(* WORD *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Enable	BOOL	Get the value of the parameter continuously while enabled
VAR_OUTPUT	Valid	BOOL	True if a valid output is available at the FB
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ER-RORCODE	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE
	Axis ErrorID	WORD	The value of the axis error. These values are vendor specific

3.4.2.21 MC_Reset

3.4.2.21.1 Description of the function block interface

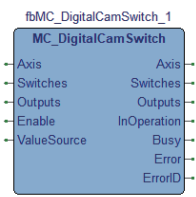
Instruc-tions	LD Performance	ST Performance
MC_Reset		fbMC_Reset_1 (Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Resets all internal axis-related errors
VAR_OUTPUT	Done	BOOL	‘Standstill’ or ‘Dis-abled’ state is reached
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

3.4.2.21.2 Instructions for the use of function block
MC_Reset command is applied to bus servo axis and virtual axis, and is used to reset motion axis faults and servo faults, and the command is valid on the rising edge

3.4.2.22 MC_DigitalCamSwitch

3.4.2.22.1 Description of the function block interface

Instruc-tions	LD Performance	ST Performance
MC_DigitalCamSwitch		fbMC_DigitalCamSwitch_1 (Axis:=(* AXIS_REF *), Switches:= (* MC_CAMSWITCH_REF *), Outputs:=(* BOOL *), Enable:=(* BOOL *), ValueSource:= (* MC_SOURCE *), InOperation=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Switch-es	MC_CAMS-WITCH_REF	Reference to the switching actions.
	Outputs	BOOL	Reference to the signal outputs, directly related to the referenced tracks. (max. 32 per function block) (First output = first TrackNumber)
VAR_OUTPUT	Enable	BOOL	Enables the ‘Switch-es’ outputs
	Value-Source	MC_SOURCE	Defines the source for axis values (e.g. positions): mcSetValue - Synchronization on set value mcActualValue - Synchronization on actual value
	InOper-ation	BOOL	The commanded tracks are enabled
	Busy	BOOL	The FB is not finished and new output values are to be expected

Pins Type	Name	Type	Description
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ER-RORCODE	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

3.4.2.22.2 Instructions for the use of function block MC_DigitalCamSwitch command is applicable to bus servo axes and virtual axes, and functions like a position switch on the axis, outputting a given signal at a given position and direction, and the command is valid at a high level

• **Directions for use**

The user can set the input Switches according to the actual needs, so as to complete the setting of the position of the switch point and the running direction

The output of the Outputs can be used to obtain the output switching signal

The structure of the input type MC_CAMSWITCH_REF switches is as follows

Name	Type	Description
NoOfSwitches	BYTE	The number of switching signal groups
CamSwitchPtr	@MC_CAMSWITCH_TR	Switch signal pointer

MC_CAMSWITCH_TR the structure is as follows

Name	Type	Description
FirstOnPosition	REAL	Lower boundary where the switch is ON
LastOnPosition	REAL	Upper boundary where the switch is ON
AxisDirection	INT	Both (=0; Default); Positive (1); Negative (2)

Name	Type	Description
CamSwitch-Mode	INT	Position based (=0; Default); Time based (=1)
Duration	TIME	Coupled to time based CamSwitchMode
bOn	BOOL	internal; used by MC_DigitalCamSwitch
CounterOff	UDINT	internal; used by MC_DigitalCamSwitch

Users can set a variable of the MC_CMSWITCH-TR structure array type, and then point the CamSwitchPtr pointer of MC_CMSWITCHREF to this array. The number of switch points can be filled in NoOfSwitches to complete the switch point setting.

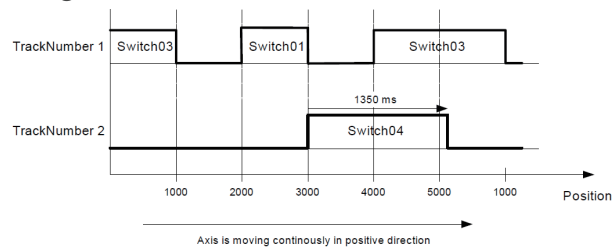
Enable to achieve switch output in a given direction and position

• **Timing diagrams**

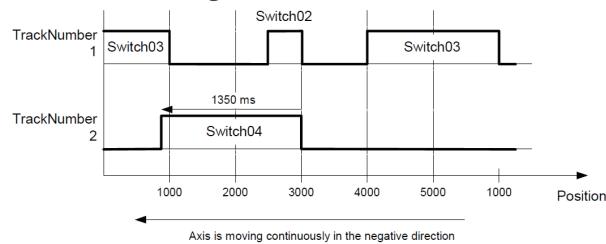
Take the data in a table as an example

Parameter	Type	Switch 01	Switch 02	Switch 03	Switch 04	...	Switch N
Track-Number	INTEGER	1	1	1	2		
FirstOn-Position [u]	REAL	2000	2500	4000	3000		
LastOn-Position [u]	REAL	3000	3000	1000	--		
AxisDirection	INTEGER	1=Pos	2=Neg	0=Both	0=Both		
CamSwitchMode	INTEGER	0=Position	0=Position	0=Position	1=Time		
Duration	TIME	--	--	--	1350		

If it runs in the positive direction, you can refer to the figure below



If it runs in the negative direction



In actual operation, due to the different shaft speeds, there may be different degrees of error in the switching signal

3.4.2.23 MC_TouchProbe

3.4.2.23.1 Description of the function block interface

Instruc-tions	LD Performance	ST Performance
MC_Tou- chProbe		<pre>fbMC_TouchProbe_1 (Axis:=(* AXIS_REF *), TriggerInput:= (* MC_TRIGGER_REF *), Execute:=(* BOOL *), WindowOnly:=(* BOOL *), FirstPosition:=(* REAL *), LastPosition:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), RecordedPosition=> (* REAL *));</pre>

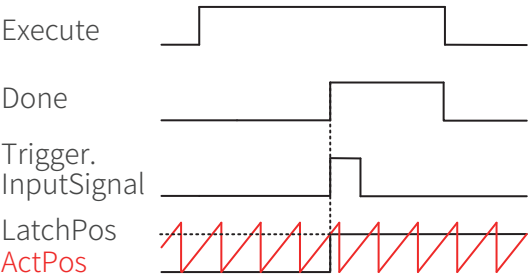
Pins Type	Name	Type	Description
VAR_IN_OUT	Accxis	AXIS_REF	Reference to the axis
	Trigger-Input	MC_TRIGGER_REF	Reference to the trigger signal source. Trigger input may be
VAR_INPUT	Execute	BOOL	Starts touch probe recording at rising edge
	Window Only	BOOL	If SET, only use the window (defined hereunder) to accept trigger events
	FirstPo-sition	REAL	Start position from where (positive direc-tion) trigger events are accepted (in technical units [u]). Value included in window.
	LastPo-sition	REAL	Stop position of the window (in technical units [u]). Value included in window.
VAR_OUTPUT	Done	BOOL	Trigger event recorded
	Busy	BOOL	The FB is not finished and new output values are to be expected
	C o m -mand Aborted	BOOL	'Command' is aborted by another command (MC_AbortTrigger)
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Nu-meratives/MC_ER-RORCODE
	Recorded Position	REAL	Position where trigger event occurred (in technical units [u])

3.4.2.23.2 Instructions for the use of function block MC_TouchProbe commands are available for bus servo axes and virtual axes, and are used to record the axis position of the triggering event, and the command is valid on the rising edge

• Directions for use

When the function block is triggered, the position of the function block latch is cleared, and when the rising edge of the input Trigger.InputSignal is detected, the position of the axis is recorded, and the function block is executed
The input pin WindowOnly enables the window limit of the trigger event, which is set by the input pins FirstPosition and LastPosition

• Timing diagrams



3.4.2.24 MC_AbortTrigger

3.4.2.24.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_AbortTrigger		<pre>fbMC_AbortTrigger_1 (Axis:=(* AXIS_REF *), TriggerInput:= (* MC_TRIGGER_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
	Trigger Input	MC_TRIGGER_REF	Reference to the trigger signal source. 'TriggerInput' may be specified by the AXIS_REF.
VAR_INPUT	Execute	BOOL	Aborts trigger event at rising edge
VAR_OUTPUT	Done	BOOL	Trigger functionality aborted
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ERRORCODE	ErrorID. See Motion-Control/DataType/Numeratives/MC_ERRORCODE

3.4.3 Cam and Gear

3.4.3.1 MC_CamTableSelect

3.4.3.1.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_CamTableSelect		<pre>fbMC_CamTableSelect_1 (Master:=(* AXIS_REF *), Slave:=(* AXIS_REF *), CamTable:= (* MC_CAM_REF *), Execute:=(* BOOL *), Periodic:=(* BOOL *), MasterAbsolute:= (* BOOL *), SlaveAbsolute:=(* BOOL *),</pre>

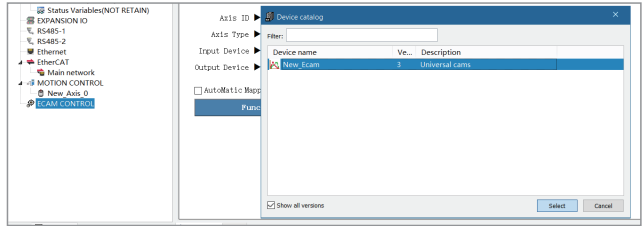
Instructions	LD Performance	ST Performance
		Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), CamTableID=> (* MC_CAM_ID *);

Pins Type	Name	Type	Description
VAR_IN_OUT	Master	AXIS_REF	Reference to the master axis
	Slave	AXIS_REF	Reference to the slave axis
	Cam Table	MC_CAM_REF	Reference to CAM description
VAR_INPUT	Execute	BOOL	Selection at rising edge
	Periodic	BOOL	1 = periodic, 0 = non periodic (single-shot)
	Master Absolute	BOOL	1 = absolute 0 = relative coordinates
	Slave Absolute	BOOL	1 = absolute 0 = relative coordinates
	Done	BOOL	Pre-selection done
VAR_OUTPUT	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/Data Type/Numeratives/MC_ERRORCODE
	Cam Table ID	MC_CAM_ID	Identifier of CAM Table to be used in the MC_CamIn FB

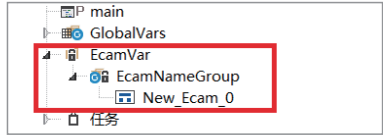
3.4.3.1.2 Instructions for the use of function block MC_CamTableSelect command converts the CamTable generated by the cam table interface into a CamTableID that can be used by MC_CamIn, and the command is triggered by the rising edge.

• Directions for use

The CamTable pin of the function block enters the CamTable variable, and the Cam Table can be created from the interface



After creation, a cam table variable will be generated in the project bar that can be used by the user



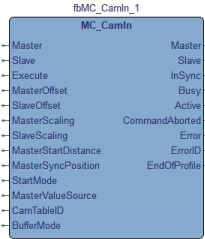
Relative absolute position mode selection

Spindle Absolute MasterAbsolute: When this option is True, the target position is obtained from the actual position of the axis reference spindle corresponding to the spindle position on the cam table, otherwise the target position is obtained by referring to the relative position of the spindle

SlaveAbsolute from Axis: If this option is True and the corresponding MC_CamIn Startmode is selected for mcAbsolute, the target position of the slave axis is the same as the corresponding position of the cam table, otherwise it is the relative value of the corresponding position of the cam table

3.4.3.2 MC_CamIn

3.4.3.2.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_CamIn		<pre>fbMC_CamIn_1 (Master:=(* AXIS_REF *), Slave:=(* AXIS_REF *), Execute:=(* BOOL *), MasterOffset:=(* REAL *), SlaveOffset:=(* REAL *), MasterScaling:=(* REAL *), SlaveScaling:=(* REAL *), MasterStartDistance:= (* REAL *), MasterSyncPosition:= (* REAL *), StartMode:= (* MC_START_MODE *), MasterValueSource:= (* MC_SOURCE *), CamTableID:= (* MC_CAM_ID *), BufferMode:= (* MC_BUFFER_MODE *), InSync=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), EndOfProfile=>(* BOOL *));</pre>

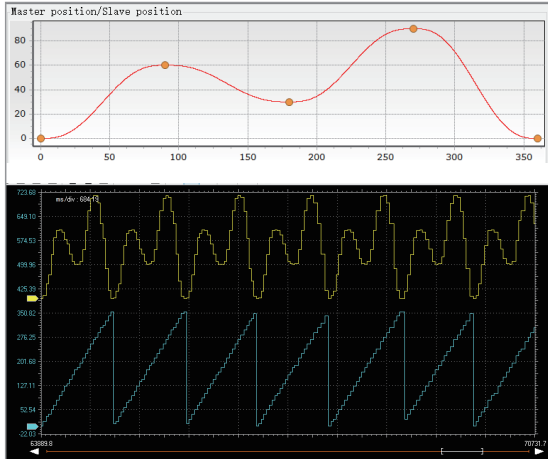
Pins Type	Name	Type	Description
VAR_IN_OUT	Master	AXIS_REF	Reference to the master axis
	Slave	AXIS_REF	Reference to the slave axis
VAR_INPUT	Execute	BOOL	Start at rising edge
	Master-Offset	LREAL	Offset of the master shaft to cam.
	Slave-Offset	LREAL	Offset of slave table.
	Master-Scaling	LREAL	Factor for the master profile (default = 1.0). From the slave point of view the master overall profile is multiplied by this factor
	SlaveScaling	LREAL	Factor for the slave profile (default = 1.0). The overall slave profile is multiplied by this factor.
	Master-Start-Distance	LREAL	The master distance for the slave to start to synchronize to the master.
	Master-SyncPosition	LREAL	The position of the master in the CAM profile where the slave is in-sync with the master.
	Start-Mode	MC_START_MODE	Start mode: mcAbsolute, mcRelative, or mcRampln
	Master-Value-Source	MC_SOURCE	Defines the source for synchronization: mcSetValue - Synchronization on master set value mcActualValue - Synchronization on master actual value
	Cam-TableID	MC_CAM_ID	Identifier of CAM Table to be used, linked to output of MC_CamTableSelect
	Buffer-Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB.

Pins Type	Name	Type	Description
VAR_OUTPUT	InSync	BOOL	Is TRUE if the set value = the commanded value.
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ERRORCODE
	EndOfProfile	BOOL	Pulsed output signaling the cyclic end of the CAM Profile It is displayed every time the end of the cam profile is reached.

3.4.3.2.2 Instructions for the use of function block MC_CamIn command to engage the cam to synchronize the shaft into the cam,

- **Directions for use**
- Pin Description:
MC_CamIn function block execution does not require the spindle state
When executing a MC_CamIn function block, if the master-slave position does not correspond to the user-set offset, the deviation will be automatically handled during the transition phase
- Relative vs. Absolute
Please refer to the instructions in the MC_Cam-TableSelect

- **Scale & Offset:**
Spindle scale and offset: At this time, the spindle position is multiplied by the spindle scale, and the spindle offset is added, and the position corresponding to the slave shaft can be obtained according to the cam table
Scale and offset from the axis: After the position of the slave shaft is preliminarily obtained according to the cam table, the position value is processed, scaled and offset, and the output slave axis position is obtained
- MasterValueSource: Source of spindle position
- **Timing diagrams**



3.4.3.3 MC_CamOut

3.4.3.3.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_CamOut	fbMC_CamOut_1 MC_CamOut Slave Execute Slave Done Busy Error ErrorID	fbMC_CamOut_1 (Slave:=(* AXIS_REF *), Execute:=(* BOOL *), Done:=(* BOOL *), Busy:=(* BOOL *), Error:=(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Slave	AXIS_REF	Reference to the slave axis
VAR_INPUT	Execute	BOOL	Start to disengage the slave from the master
VAR_OUTPUT	Done	BOOL	Disengaging completed
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ERRORCODE

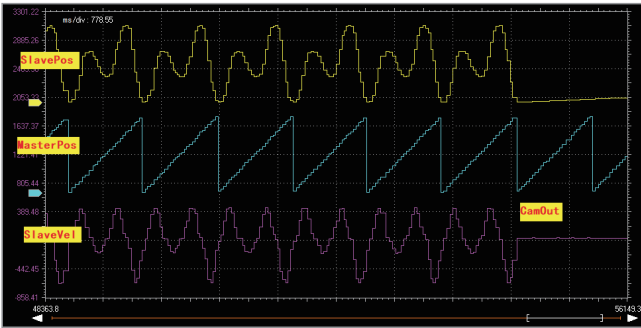
3.4.3.3.2 Instructions for the use of function block

- **Directions for use**

MC_CamOut command is applied to bus servo axes and virtual axes, which are used to interrupt the cam synchronization state of the slave axis, the Execute rising edge triggers, the slave axis will continue to run at the speed at the time of triggering, and the axis state machine jumps from synchronous motion to continuous motion.

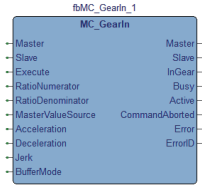
Note: After the MC_CamOut is triggered, although the slave axis is still moving at a certain speed, in fact, the axis movement will not continue to be controlled after the function of the function MC_CamOut function is executed.

- **Timing diagrams**



3.4.3.4 MC_GearIn

3.4.3.4.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_GearIn		fbMC_GearIn_1 (Master:=(* AXIS_REF *), Slave:=(* AXIS_REF *), Execute:=(* BOOL *), RatioNumerator:=(* DINT *), RatioDenominator:= (* UDINT *), MasterValueSource:= (* MC_SOURCE *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), InGear=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Master	AXIS_REF	Reference to the master axis
	Slave	AXIS_REF	Reference to the slave axis
VAR_INPUT	Execute	BOOL	Start the gearing process at the rising edge
	Ratio Numerator	LREAL	Gear ratio Numerator
	Ratio Denominator	LREAL	Gear ratio Denominator
	MasterValueSource	MC_SOURCE	Defines the source for synchronization: mcSetValue - Synchronization on master set value mcActualValue - Synchronization on master actual value
	Acceleration	LREAL	Acceleration for gearing in
	Deceleration	LREAL	Deceleration for gearing in
	Jerk	LREAL	Jerk of Gearing
VAR_OUTPUT	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB
	InGear	BOOL	Is TRUE if the set value = the commanded value.
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/Data Type/Numeratives/MC_ERRORCODE

3.4.3.4.2 Instructions for the use of function block MC_GearIn directives apply to bus servo axes and virtual axes

• Directions for use

After the operation starts, the acceleration and deceleration of the slave shaft is performed with the speed of the spindle multiplied by the gear ratio as the target speed.

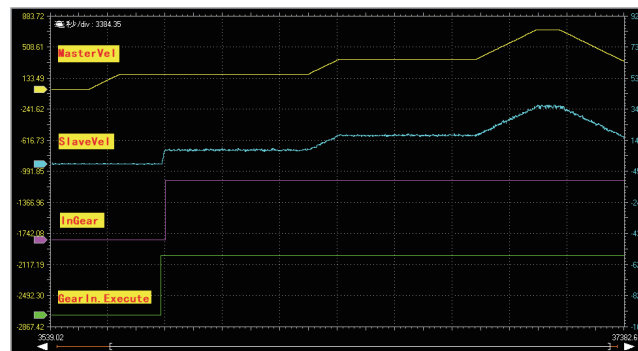
The stage before the speed of the slave shaft and the speed of the main shaft meet the gear ratio is the transition stage, and the speed of the master and slave shaft enters the synchronization stage after the gear ratio is met

The gear ratio is positive and moves in the same direction as the main shaft from the shaft, and the gear ratio is negative and moves in the opposite direction from the shaft to the main shaft

Both the block input acceleration and deceleration and the jerk and acceleration are specified as the maximum acceleration and deceleration before entering the synchronization, that is, the acceleration and deceleration in the transition phase

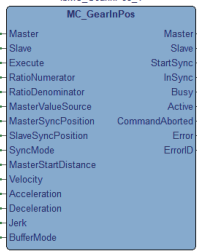
When entering synchronization with the movement of the spindle from the axis, the acceleration and deceleration should not be restricted

• Timing diagrams



3.4.3.5 MC_GearInPos

3.4.3.5.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_GearInPos		<pre>fbMC_GearInPos_1 (Master:=(* AXIS_REF *), Slave:=(* AXIS_REF *), Execute:=(* BOOL *), RatioNumerator:=(* DINT *), RatioDenominator:= (* UDINT *), MasterValueSource:= (* MC_SOURCE *), MasterSyncPosition:= (* REAL *), SlaveSyncPosition:= (* REAL *), SyncMode:= (* MC_SYNC_MODE *), MasterStartDistance:= (* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), StartSync=>(* BOOL *), InSync=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error => (* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

Pins Type	Name	Type	Description
VAR_IN_OUT	Master	AXIS_REF	Reference to the master axis
	Slave	AXIS_REF	Reference to the slave axis
VAR_INPUT	Execute	BOOL	Start the gearing process at the rising edge
	RatioNumerator	LREAL	Gear ratio Numerator
	RatioDenominator	LREAL	Gear ratio Denominator
	Master-Value-Source	MC_SOURCE	Defines the source for synchronization: mcSetValue - Synchronization on master set value mcActualValue - Synchronization on master actual value
	Master-SyncPosition	LREAL	The position of the master in the CAM profile where the slave is in-sync with the master.
	SlaveSyncPosition	LREAL	Slave Position at which the axes are running in sync
	Sync-Mode	MC_SYNC_MODE	Defines the way to synchronize
	Master-StartDistance	LREAL	Master Distance for gear in procedure
	Velocity	LREAL	Maximum Velocity during the time difference 'StartSync' and 'InSync'
	Acceleration	LREAL	Maximum Acceleration during the time difference 'Start-Sync' and 'InSync'

Pins Type	Name	Type	Description
	Deceleration	LREAL	Maximum Deceleration during the time difference 'Start-Sync' and 'InSync'
	Jerk	LREAL	Maximum Jerk during the time difference 'Start-Sync' and 'InSync'
	Buffer-Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB.
VAR_OUTPUT	Start-Sync	BOOL	Commanded gearing starts
	InGear	BOOL	Is TRUE if the set value = the commanded value
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Active	BOOL	Indicates that the FB has control on the axis
	Command Aborted	BOOL	'Command' is aborted by another command
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ERRORCODE

3.4.3.5.2 Instructions for the use of function block MC_GearInPos directives apply to bus servo axes and virtual axes

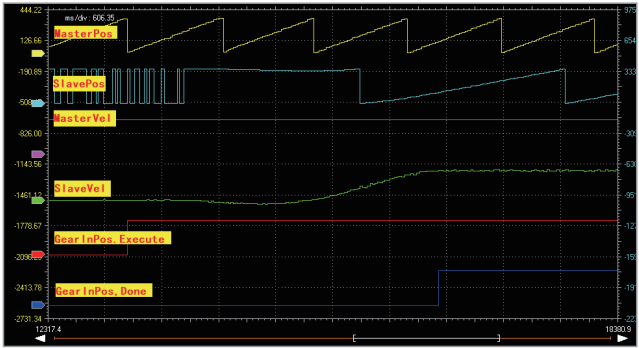
• **Directions for use**

Similar to MC_GearIn, the goal of MC_GearInPos

is to control the electronic gear coupling from the shaft to the spindle, operating at a given gear ratio

In contrast to MC_GearIn, the MC_GearInPos specifies the position of the spindle entering the synchronization, and the synchronization distance from the shaft position and the spindle starts to complete the action of cutting into the electronic gear

• **Timing diagrams**



3.4.3.6 MC_GearOut

3.4.3.6.1 Description of the function block interface

Instructions	LD Performance	ST Performance
MC_GearOut		fbMC_GearOut_1 (Slave:=(* AXIS_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

Pins Type	Name	Type	Description
VAR_IN_OUT	Slave	AXIS_REF	Reference to the slave axis
VAR_INPUT	Execute	BOOL	Start disengaging process at the rising edge

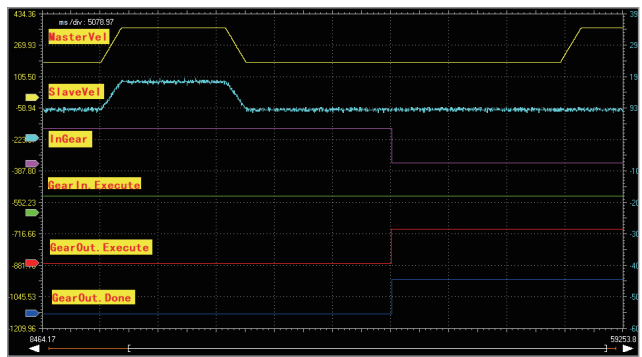
Pins Type	Name	Type	Description
VAR_OUTPUT	Done	BOOL	Disengaging completed
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	WORD	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

3.4.3.6.2 Instructions for the use of function block MC_GearOut directives apply to both bus servo axes and virtual axes

• Directions for use

The MC_GearOut command is used to separate the master-slave shafts that are coupled to each other in an electronic gear manner, and when the slave shaft is separated from the spindle, it will continue to operate at the speed at the time of the MC_GearOut command, and the MC_GearOut function block will no longer control the action of the shaft during subsequent continuous operation

• Timing diagrams



3.4.4 Encoder

The encoder Axis can be used as an axis object using the MC basic function block, and can only be used as the main axis of the Cam and Gear function block, or the dedicated ENC function block can be used to set and read the position

3.4.4.1 ENC_Counter

ENC_Counter function block is used to read the encoder shaft position, speed and limit information, and the direction of position change can be set, and the high level is active

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Enable	BOOL	Get the axis counter value constantly while enabled
	Direction	BOOL	If the value is true, the count value increases otherwise, the count value decreases
VAR_OUTPUT	Valid	BOOL	True if a valid set of outputs is available
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Position	REAL	Current counter value
	Velocity	REAL	Current Velocity
	Positive Limit	BOOL	Linear mode reaches positive limit value
	Negative Limit	BOOL	Linear mode reaches negative limit value
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ER-RORCODE	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

3.4.4.2 ENC_Preset

ENC_Preset function block is used for setting the initial value preset value of encoder shaft, after the rising edge is triggered, this position value is set to enter the initial value preset value of encoder shaft, at this moment, this value still does not take effect, and the user needs to call ENC_Reset function block before the current position can be reset to the preset initial position value.

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis
VAR_INPUT	Execute	BOOL	Set value to the preset value
	Position	REAL	Value of preset position
VAR_OUTPUT	Done	BOOL	Set position to the preset value
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ER-RORCODE	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

3.4.4.3 ENC_Reset

ENC_Reset function block is used to reset the encoder shaft position, setting the current position to the preset position value, note that this function block is triggered by the rising edge and needs to be triggered when the enable pin of the ENC_Counter function block is low, that is, the position of the encoder shaft cannot be read when the encoder shaft position is reset.

Pins Type	Name	Type	Description
VAR_IN_OUT	Axis	AXIS_REF	Reference to the axis

Pins Type	Name	Type	Description
VAR_INPUT	Execute	BOOL	Resets counter value with preset value
VAR_OUTPUT	Done	BOOL	Set the count value to the preset value
	Busy	BOOL	The FB is not finished and new output values are to be expected
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ER-RORCODE	ErrorID. See Motion-Control/DataType/Numeratives/MC_ER-RORCODE

3.4.5 Axes Group Control

3.4.5.1 Axes Group instructions

The Motion Control library includes BoNeng's independently developed axis group MC function blocks, which comply with PLCOpen standards.

3.4.5.2 Instruction List

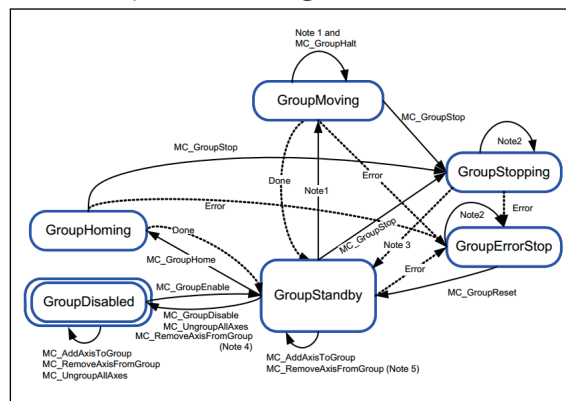
The basic commands for axis group motion control include the following:

Category	Function	
Admin-istrative	MC_AddAxisToGroup	add one axis to a group
	MC_GroupReadActualPosition	Get the position in the corresponding coordinate system.
	MC_GroupReadActualVelocity	Get the velocity in the corresponding coordinate system.
	MC_GroupSetPosition	Set current position of axis group
	MC_GroupReset	Reset Axis Group
	MC_ReadCoordinateTransform	Read the coordinate transformation between MCS and PCS

Category	Function	
	MC_ReadKinTransform	Read the kinematic transformations between ACS and MCS activities
	MC_GroupReadConfiguration	Read Axis Group Configuration
	MC_GroupReadError	Read axis group error code
	MC_GroupReadStatus	Read Axis Group Status
	MC_SetCoordinateTransform	Set the coordinate transformation between MCS and PCS
	MC_SetkinTransform	Set the kinematic transformation between ACS and MCS
	MC_UngroupAllAxes	Remove all axes from the axis group
	MC_RemoveAxisFromGroup	Remove the corresponding shaft from the shaft group
	MC_GroupContinue	The axle group continues to operate along its original trajectory.
Motion	MC_GroupDisable	Disable axes group
	MC_GroupEnable	Enable axes group
	MC_GroupHalt	Halt axis group movement
	MC_GroupStop	Stop axis group movement
	MC_GroupHome	The axis group returns to the origin
	MC_Groupinterrupt	Interrupt the original movement and perform a new movement
	MC_MoveCircularAbsolute	The axis group performs arc movement in absolute mode

Category	Function
	MC_MoveCircularRelative The axis group performs arc movement in relative mode
	MC_MoveDirectAbsolute The axis group performs point-to-point movement in absolute mode
	MC_MoveDirectRelative The axis group performs point-to-point movement in relative mode
	MC_MoveLinearAbsolute The axis group performs linear motion in absolute mode
	MC_MoveLinearRelative The axis group performs linear motion in relative mode

3.4.5.3 PLCopen State Diagram



The state machine is described in detail as follows:

name	description
GroupDisabled	the initial state at power up where a group can be created
GroupErrorStop	an error occurred in group
GroupStandby	group is enabled and no function block has control on one of the axes in the group

name	description
GroupHoming	A group going home
GroupStopping	A group stopping
GroupMoving	A group moving

Note 1: Applicable for all non-administrative (moving) function blocks.

Note 2: In the states GroupErrorStop or GroupStopping, all Function Blocks can be called, although they will not be executed, except MC_GroupReset for GroupErrorStop and any occurring Error – they will generate the transition to GroupStandby or GroupErrorStop respectively

Note 3: MC_GroupStop.DONE AND NOT MC_GroupStop.EXECUTE

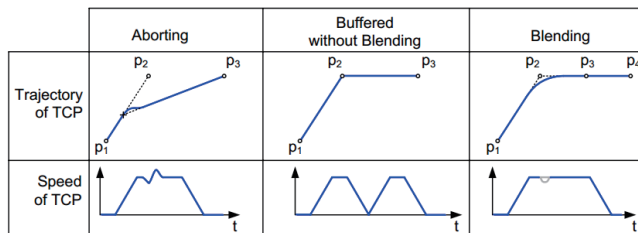
Note 4: Transition is applicable if last axis is removed from the group

Note 5: Transition is applicable while group is not empty.

Note 6: MC_GroupDisable and MC_UngroupAllAxes can be issued in all states and will change the state to GroupDisabled.

3.4.5.4 BufferMode And TransitionModes Explanation

There is an over-mode between the two functional blocks of the axis groups, which are influenced by the BufferMode and TransitionModes interfaces, respectively. The main effects are as follows:



BufferMode Interface Description:

Some motion function blocks have pins with BufferMode, which means buffer mode, specifying the connection mode between two motion function blocks.

Name	Value	Description
Aborting	0	Start FB immediately (default mode)
Buffered	1	Start FB after current motion has finished
BlendingLow	2	The velocity is blended with the lowest velocity of both FBs
BlendingPrevious	3	The velocity is blended with the velocity of the first FB
Blending-Next	4	The velocity is blended with velocity of the second FB
Blending-High	5	The velocity is blended with highest velocity of both FBs

TransitionModes Interface Description:

According to the transition mode, different transition parameters can be given, which characterize the contour curve.

Name	Value	Description
TMNone	0	Insert no transition curve (default mode)
TMStartVelocity	1	Transition with given start velocity
TMConstantVelocity	2	Transition with given constant velocity
TMCornerDistance	3	Transition with given corner distance
TMCornerDistance	4	Transition with given maximum corner deviation

The available conversion modes under the above interface are as follows:

Buffer Mode Transition Mode	Aborting	Buffered	Blending Low	Blending Previous	Blending Next	Blending High
TMNone	A	A	N	N	N	N
TMMax Velocity	D	D	D	D	D	D
TMDefined Velocity	A	N	A	A	A	A
TMCorner Distance	N	N	A	A	A	A
TMMaxCorner Deviation	N	N	A	A	A	A

A: Available
N: Not possible
D: BlendingMode is dispensable

3.4.5.5 CoordSystem Interface Description

This is the coordinate system interface, indicating that the input parameters are relative to this coordinate system.

ACS: Axis related
MCS: Machine related
PCS: Product or Workpiece related

3.4.5.6 CircMode And PathChoice Interface Description

This interface is provided for arc planning. It is mainly used to define AuxPoint and EndPoint. Among them, the three points consisting of the current point at the end of the axis group, AuxPoint, and EndPoint cannot be collinear, as collinearity will trigger an error. CircMode provides the following parameters:

BORDER: three-point arc mode, where AuxPoint is a point that the arc passes through; the arc constructed from three points will go through this point, and PathChoice is ignored.

CENTER: center arc mode, where AuxPoint is the center of the circle and EndPoint is the terminal point. PathChoice determines whether the arc is clockwise or counterclockwise.

SEMI: semi-circle mode, where AuxPoint determines the plane in which the circle lies. The arc does not pass through this point nor is it the center; the center is the midpoint between the start point and EndPoint. PathChoice determines whether the arc is clockwise or counterclockwise.

FULL: full circle mode, where AuxPoint determines the center and EndPoint determines the plane of the circle. PathChoice determines whether the arc is clockwise or counterclockwise.

3.4.5.7 Fault Code Description

Users can find the description of fault codes through the enumeration types in the data types of the MotionControl library, through the properties window, or by double-clicking to open and view them.

</

can also refer to the table below

Error name	Value	Description
PERMISSION_DENIED	5000	Permission denied
GROUP NOT EXIST	5001	Axis group does not exist
GROUP AXIS ARRAY FULL	5002	Axis group space is full
AXIS ALREADY IN GROUP	5003	Axis already exists in the group
AXIS NOT IN GROUP	5004	Axis group does not contain this axis
GROUP QUEUE FULL	5005	Axis group queue is full
GROUP DISABLED	5006	Axis group not enabled
GROUP_AXIS_DISABLED	5007	Axis member of the group is not enabled

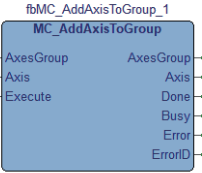
Error name	Value	Description
GROUP_AXIS_ERR	5008	Axis member of the group has an error
GROUP_AXIS_0_ERR	5009	Axis 0 member of the group has an error
GROUP_AXIS_1_ERR	5010	Axis 1 member of the group has an error
GROUP_AXIS_2_ERR	5011	Axis 2 member of the group has an error
GROUP_AXIS_3_ERR	5012	Axis 3 member of the group has an error
GROUP_SYNC_AXIS_0_ERR	5013	Synchronous axis 0 member of the group has an error
GROUP_SYNC_AXIS_1_ERR	5014	Synchronous axis 1 member of the group has an error
GROUP_SYNC_AXIS_2_ERR	5015	Synchronous axis 2 member of the group has an error
GROUP_AXIS_0_HOME_FAILED	5016	Axis 0 member of the group failed to home
GROUP_AXIS_1_HOME_FAILED	5017	Axis 1 member of the group failed to home
GROUP_AXIS_2_HOME_FAILED	5018	Axis 2 member of the group failed to home
GROUP_AXIS_3_HOME_FAILED	5019	Axis 3 member of the group failed to home
GROUP_SYNC_AXIS_0_HOME_FAILED	5020	Synchronous axis 0 member of the group failed to home
GROUP_SYNC_AXIS_1_HOME_FAILED	5021	Synchronous axis 1 member of the group failed to home
GROUP_SYNC_AXIS_2_HOME_FAILED	5022	Synchronous axis 2 member of the group failed to home
GROUP_AXIS_MODULO_NOT_ALLOWED	5023	Axis member of the group is not allowed to be a modulo axis

Error name	Value	Description
GROUP_FSM_DISABLED	5100	PLCOpen state machine: DISABLED
GROUP_FSM_STANDBY	5101	PLCOpen state machine: STANDBY
GROUP_FSM_HOMING	5102	PLCOpen state machine: HOMING
GROUP_FSM_MOVING	5103	PLCOpen state machine: MOVING
GROUP_FSM_STOPPING	5104	PLCOpen state machine: STOPPING
GROUP_FSM_ERRORSTOP	5105	PLCOpen state machine: ERRORSTOP
KINEMATIC_DEFINED	5140	Kinematic transformation already defined
KINEMATIC_NOT_ALLOWED	5141	Kinematic mechanism type not allowed
KIN_TRANSFORM_NOT_CONSTANT	5142	Kinematic transformation is not unique
GROUP_AIMPOUT_LIMIT	5150	Target position exceeds limit
GROUP_KINPARAM_ERROR	5151	Kinematic parameter limits are unreasonable
GROUP_TRANMODEERROR	5152	Transition mode error
GROUP_CIRCLEPARAERROR	5153	Circle mode parameter error
GROUP_PLANFAIL	5154	S-curve planning failed
GROUP_PARAMERROR	5155	Parameter error
GROUP_ACC_DEC_ILLEGAL	5156	Acceleration/Deceleration is illegal
GROUP_JERK_ILLEGAL	5157	Jerk is illegal
GROUP_AIMEQUAL	5158	Target positions coincide
GROUP_BLENDING_ERROR	5159	Trajectory blending error
GROUP_MOTION_LIMIT_AVOUT	5160	Joint angular velocity and angular acceleration exceed limits during motion
GROUP_MOTION_LIMIT_POSOUT	5161	Joint position exceeds limit during motion
GROUP_PLAN_ERROR	5162	Axis group planning error

Error name	Value	Description
GROUP_INTERPOLATE_ERROR	5180	Axis group interpolator error
GROUP_SYNC_POSLAGOVERLIMIT	5181	Synchronous axis following error exceeds limit
GROUP_UNAVALID_COMBINE_PARAMETER	5182	Unsupported parameter combination
GROUP_INTERRUPT_NODE_NOT_EXIST	5183	Axis group interrupt motion does not exist

3.4.5.8 Administrative Command

3.4.5.8.1 MC_AddAxisToGroup add one axis to a group

Instruc-tions	LD Performance	ST Performance
MC_AddAxisToGroup		fbMC_AddAxisToGroup_1(AxesGroup:= (* AXES_GROUP_REF *), Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

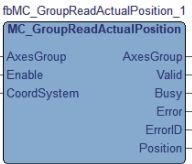
Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GRP- OUP_REF	Reference to a group of axes
	Axis	AXIS_REF	Reference to a axis
VAR_INPUT	Execute	BOOL	Start the grouping process at the rising edge
VAR_OUTPUT	Done	BOOL	AXES_GROUP_REF is valid and the axis added
	Busy	BOOL	The FB is not finished
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ER- RORCODE	Error code

FB Explanation

This Function Block adds one axis to a group in a structure AxesGroup. This is an administrative FB, since no movement is generated. The command cannot be buffered.

3.4.5.8.2 MC_GroupReadActualPosition

Get the position in the corresponding coordinate system.

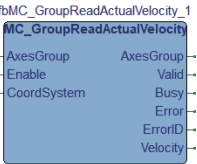
Instruc-tions	LD Performance	ST Performance
MC_GroupReadActualPosition		fbMC_GroupReadActualPosition_1(AxesGroup:= (* AXES_GROUP_REF *), Enable:=(* BOOL *), CoordSystem:= (* MC_COORD_SYSTEM *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *), Position=>(* REAL *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GRP- OUP_REF	Reference to a group of axes
VAR_INPUT	Enable	BOOL	Get the actual position in the selected coordinate system of the group
	Coord System	MC_COORD- SYSTEM	Reference to the coordinate system (ACS, MCS, PCS)
VAR_OUTPUT	Valid	BOOL	True if valid outputs are available
	Busy	BOOL	The FB is not finished
	Error	BOOL	Signals that an error has occurred within the Function Block

Pin	Name	Type	Description
	ErrorID	MC_ER-RORCODE	Error code
	Position	REAL	Current position of the group

FB Explanation
This Function Block returns the actual position in the selected coordinate system of an axes group. This is an administrative FB, since no movement is generated.

3.4.5.8.3 MC_GroupReadActualVelocity
Get the velocity in the corresponding coordinate system.

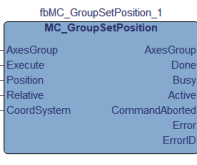
Instruc-tions	LD Performance	ST Performance
MC_GroupReadActualPosition		fbMC_GroupReadActualVelocity_1(AxesGroup:= (* AXES_GROUP_REF *), Enable:=(* BOOL *), CoordSystem:= (* MC_COORD_SYSTEM *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ER-RORCODE *), Velocity=>(* REAL *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GRP-ROUP_REF	Reference to a group of axes
VAR_INPUT	Enable	BOOL	Get the actual velocity in the selected coordinate system of the group

Pin	Name	Type	Description
	Coord System	MC_COORD_SYSTEM	Reference to the coordinate system (ACS, MCS, PCS)
VAR_OUTPUT	Valid	BOOL	True if valid outputs are available
	Busy	BOOL	The FB is not finished
	Error	BOOL	Signals that an error has occurred within the Function Block
	ErrorID	MC_ER-RORCODE	Error code
	Velocity	REAL	Current velocity of the group

FB Explanation
This Function Block returns the actual velocity in the selected coordinate system of an axes group. This is an administrative FB, since no movement is generated.

3.4.5.8.4 MC_GroupSetPosition

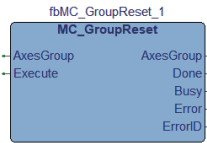
Instruc-tions	LD Performance	ST Performance
MC_GroupSetPosition		fbMC_GroupSetPosition_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Position:=(* REAL *), Relative:=(* BOOL *), CoordSystem:= (* MC_COORD_SYSTEM *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
	Position	REAL	Array of coordinates, incl. positions and orientations (Means 'Distance' if Mode = RELATIVE)
	Relative	BOOL	Mode of position inputs - RELATIVE = True, ABSOLUTE = False (Default)
	Coord System	MC_COORD_SYSTEM	Reference to the coordinate system used: ACS, MCS, or PCS
VAR_OUTPUT	Done	BOOL	Commanded end positions reached for all axes
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This function block sets the Position of all axes in a group without moving the axes. The new coordinates are described in an array. With the coordinate system input the according coordinate system is selected. It can be seen as a way of referencing or a transformation. MC_GroupSetPosition shifts the position of the addressed coordinate system and affects the higher level coordinate systems (so if ACS is selected, MCS and PCS are affected).

3.4.5.8.5 MC_GroupReset

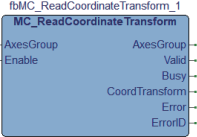
Instructions	LD Performance	ST Performance
MC_GroupReset		<pre>fbMC_GroupReset_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
VAR_OUTPUT	Done	BOOL	Reset for axes group and all axes in this group done.
	Busy	BOOL	The FB is not finished
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This function block makes the transition from the state GroupErrorStop to GroupStandby by resetting all internal group-related errors – it does not affect the output of the FB instances. This function block also resets all axes in this group like MC_Reset.

3.4.5.8.6 MC_ReadCoordinateTransform

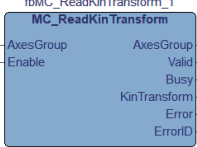
Instructions	LD Performance	ST Performance
MC_GroupReset		fbMC_ReadCoordinateTransform_1(AxesGroup:= (* AXES_GROUP_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), CoordTransform=> (* MC_COORD_REF *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Enable	BOOL	Get the actual coordinate transformation reference of the axes
VAR_OUTPUT	Valid	BOOL	True if valid outputs are available
	Busy	BOOL	The FB is not finished
	Coord Transform	MC_COORD_REF	Reference to a Coordinate Transformation. Vendor specific datatype.
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This Function Block reads the coordinate transformation that is active between the MCS and PCS.

3.4.5.8.7 MC_ReadKinTransform

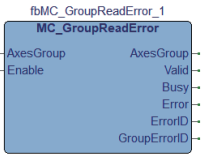
Instructions	LD Performance	ST Performance
MC_GroupReset		fbMC_ReadKinTransform_1(AxesGroup:= (* AXES_GROUP_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), KinTransform=> (* MC_KIN_REF *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Enable	BOOL	Get the actual kinematic transformation reference of the axes group continuously while enabled
VAR_OUTPUT	Valid	BOOL	True if valid outputs are available
	Busy	BOOL	The FB is not finished
	Coord Transform	MC_COORD_REF	Reference to a Kinematic Model. Vendor specific datatype.
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This Function Block reads the kinematic transformation that is active between the ACS and MCS.

3.4.5.8.8 MC_GroupReadError

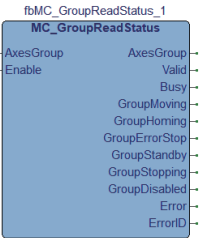
Instruc-tions	LD Performance	ST Performance
MC_GroupReadError		fbMC_GroupReadError_1(AxesGroup:=(* AXES_GROUP_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *), GroupErrorID=>(* WORD *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Enable	BOOL	Get the actual kinematic transformation reference of the axes group continuously while enabled
VAR_OUTPUT	Valid	BOOL	True if valid outputs are available
	Busy	BOOL	The FB is not finished
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification
	Group ErrorID	WORD	The value of the axes group error.

FB Explanation

This Function Block describes general axes group errors not relating to the Function Blocks. This is an administrative FB, since no movement is generated.

3.4.5.8.9 MC_GroupReadStatus

Instruc-tions	LD Performance	ST Performance
MC_GroupReadStatus		fbMC_GroupReadStatus_1(AxesGroup:=(* AXES_GROUP_REF *), Enable:=(* BOOL *), Valid=>(* BOOL *), Busy=>(* BOOL *), GroupMoving=>(* BOOL *), GroupHoming=>(* BOOL *), GroupErrorStop=>(* BOOL *), GroupStandby=>(* BOOL *), GroupStopping=>(* BOOL *), GroupDisabled=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

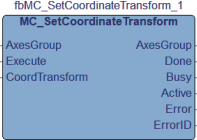
Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Enable	BOOL	Get the status of the axes group continuously while enabled
VAR_OUTPUT	Valid	BOOL	True if valid outputs are available
	Busy	BOOL	The FB is not finished
	Group Moving	BOOL	See group state diagram
	Group Homing	BOOL	See group state diagram
	Group ErrorStop	BOOL	See group state diagram
	Group Standby	BOOL	See group state diagram
	Group Stopping	BOOL	See group state diagram
	Group Disabled	BOOL	See group state diagram

Pin	Name	Type	Description
	Group-Disabled	BOOL	See group state diagram
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This Function Block returns the status of an axes group according to the active Group-FB. This is an administrative FB, since no movement is generated.

3.4.5.8.10 MC_SetCoordinateTransform

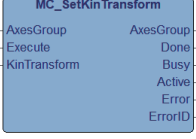
Instructions	LD Performance	ST Performance
MC_SetCoordinateTransform		fbMC_SetCoordinateTransform_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), CoordTransform:= (* MC_COORD_REF *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Sets the coordinate transformation on rising edge
	CoordTransform	MC_COORD_REF	Reference to a Coordinate Transformation.

FB Explanation

This Function Block sets a coordinate transformation between the MCS and PCS.

3.4.5.8.11 MC_SetkinTransform

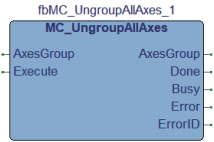
Instructions	LD Performance	ST Performance
MC_SetkinTransform		fbMC_SetKinTransform_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), KinTransform:= (* MC_KIN_REF *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Sets the kinematic model transformation on rising edge
	KinTransform	MC_KIN_REF	Reference to a Kinematic Model. Vendor specific datatype.
VAR_OUTPUT	Done	BOOL	Transformation is set successfully
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB is processed
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This Function Block sets a kinematic transformation between the ACS and MCS based on the pre-defined kinematic model.

3.4.5.8.12 MC_UngroupAllAxes

Instructions	LD Performance	ST Performance
MC_SetKinTrans- form		fbMC_UngroupAllAxes_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GRP- OUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the process at the rising edge
VAR_OUTPUT	Done	BOOL	All axes are removed
	Busy	BOOL	The FB is not finished
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ER- RORCODE	Error identification

FB Explanation

This Function Block removes all axes from the group AxesGroup. This is an administrative FB, since no movement is generated. The command cannot be buffered. After finalization the state is changed to GroupDisabled

3.4.5.8.13 MC_RemoveAxisFromGroup

Instruc- tions	LD Performance	ST Performance
MC_Re- moveAx- isFrom Group		fbMC_RemoveAxis- FromGroup_1(AxesGroup:= (* AXES_GROUP_REF *), Axis:=(* AXIS_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

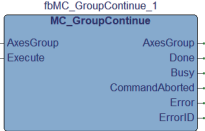
Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GRP- OUP_REF	Reference to a group of axes
	Axis	AXIS_REF	Reference to a axis
VAR_INPUT	Execute	BOOL	Start the axis removal process at the rising edge
VAR_OUTPUT	Done	BOOL	AXES_GROUP_REF is valid and the axis removed
	Busy	BOOL	The FB is not finished
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ER- RORCODE	Error code

FB Explanation

This Function Block removes one axis from the group AxesGroup. This is an administrative FB, since no movement is generated. The command cannot be buffered. If there is no axis left in the group, the state changes to GroupDisabled.

3.4.5.9 Motion Commad

3.4.5.9.1 MC_GroupContinue

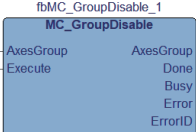
Instruc-tions	LD Performance	ST Performance
MC_GroupContinue		fbMC_GroupContinue_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GR- OUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
VAR_OUTPUT	Done	BOOL	Control transferred back to original FB
	Busy	BOOL	The FB is not finished
	C o m - mand Aborted	BOOL	Command is aborted by another com- mand
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	M C _ E R - R O R C O D E	Error identification

FB Explanation

This function block transfers the program back to the situation at issuing MC_GroupInterrupt. It uses internally the data set as stored at issuing MC_GroupInterrupt, and at the end (output DONE set) transfer the control on the group back to the original FB doing the movement on the axes group, meaning also that at the originally interrupted FB the output Busy is still high and Active is set again.

3.4.5.9.2 MC_GroupDisable

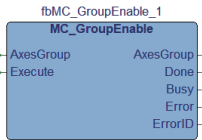
Instruc-tions	LD Performance	ST Performance
MC_GroupDisable		fbMC_GroupDisable_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GR- OUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
VAR_OUTPUT	Done	BOOL	AxesGroup in state GroupDisabled
	Busy	BOOL	The FB is not finished
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	M C _ E R - R O R C O D E	Error identification

FB Explanation

This Function Block changes the state for a group to GroupDisabled, although it is an administrative FB, since no movement is generated. If the axes are not standing still while issuing this command, it is up to the application to take the necessary precautions.

3.4.5.9.3 MC_GroupEnable

Instruc-tions	LD Performance	ST Performance
MC_GroupDis-able		fbMC_GroupEnable_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Done=>(* BOOL *), Busy=>(* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

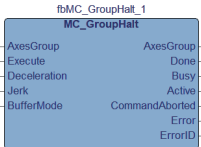
Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GRP- OUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
VAR_OUTPUT	Done	BOOL	AxesGroup in state GroupStandby
	Busy	BOOL	The FB is not finished
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ER- RORCODE	Error identification

FB Explanation

This Function Block changes the state for a group from GroupDisabled to GroupStandby. This is an administrative FB, since no movement is generated.

3.4.5.9.4 MC_GroupHalt

Instruc-tions	LD Performance	ST Performance
		fbMC_GroupHalt_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *),

Instruc-tions	LD Performance	ST Performance
MC_GroupHalt		Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

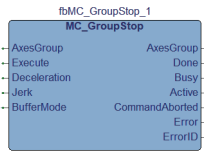
Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GRP- OUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
	Deceleration	REAL	Value of the deceleration
	Jerk	REAL	Value of the jerk
	Buffer Mode	MC_BUFF- ER_MODE	Refer to Buffer Modes
VAR_OUTPUT	Done	BOOL	Zero velocity reached
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group
	C o m - mand - Aborted	BOOL	Command is aborted by another com- mand
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ER- RORCODE	Error identification

FB Explanation

This function block commands a controlled motion stop. It aborts any ongoing function block execution. AxesGroup is moved to the state “GroupMoving”, until the velocity is zero. With the DONE output set, the state is transferred to GroupStandby.

- MC_GroupHalt is used to stop the axes group under normal operation conditions. In non-buffered mode: during deceleration of the axes group it is possible to set another motion command, which will abort the MC_GroupHalt and will be executed immediately.
- If this command is active the next command can be issued. E.g. a driverless vehicle detects an obstacle and needs to stop. MC_GroupHalt is issued. Before the standstill is reached the obstacle is removed and the motion can be continued by setting another motion command, so the vehicle does not stop.
- The relevant axes stay on the same path which would have been executed without having issued MC_GroupHalt.

3.4.5.9.5 MC_GroupStop

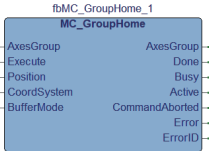
Instruc-tions	LD Performance	ST Performance
MC_GroupHalt		<pre>fbMC_GroupStop_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
	Deceleration	REAL	Value of the deceleration
	Jerk	REAL	Value of the jerk
	Buffer Mode	MC_BUFFER_MODE	Refer to Buffer Modes
VAR_OUTPUT	Done	BOOL	Stop for all axes done.
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This Function Block commands a controlled motion stop and transfers the axes group to the state “GroupStopping” . It aborts any ongoing Function Block execution. While the axes group is in state GroupStopping, no other FB can perform any motion on the same axes group. After the axes group has reached velocity zero, the Done output is set to TRUE immediately. The axes group remains in the state “GroupStopping” as long as Execute is still TRUE or velocity zero is not yet reached. As soon as “Done” is SET and “Execute” is FALSE the axes group goes to state “GroupStandBy” . The command can only be aborted by MC_GroupDisable.

3.4.5.9.6 MC_GroupHome

Instruc-tions	LD Performance	ST Performance
MC_GroupHome		fbMC_GroupHome_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Position:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

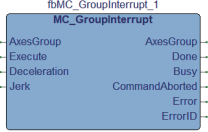
Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GRP- OUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the homing sequence at the rising edge
	Position	REAL	Array of coordinates incl. Positions and orientations
	Coord System	MC_COORD- D_SYSTEM	Reference to the applicable coordinate system: ACS, MCS, PCS
	Buffer Mode	MC_BUFF- ER_MODE	Refer to Buffer Modes
VAR_OUTPUT	Done	BOOL	Homing sequence ended successfully
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group

Pin	Name	Type	Description
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ER- RORCODE	Error identification

FB Explanation

This Function Block commands the AxesGroup to perform the «search home» sequence. The details of this sequence are manufacturer dependent and can be set by the axis parameters. The “Position” input is used to set the absolute position when reference signal is detected. This Function Bock completes at “GroupStandby” .

3.4.5.9.7 MC_GroupInterrupt

Instruc-tions	LD Performance	ST Performance
MC_GroupInterrupt		fbMC_GroupInterrupt_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
	Deceleration	REAL	Value of the deceleration
	Jerk	REAL	Value of the jerk
VAR_OUTPUT	Done	BOOL	Zero velocity reached
	Busy	BOOL	Indicates that the FB has control on the axes group
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This function block interrupts the on-going motion and stops the group from moving, however does not abort the interrupted motion (meaning that at the interrupted FB the output CommandAborted will not be Set, Busy is still high and Active is reset). It stores all relevant track or path information internally at the moment it becomes active.

The AxesGroup stays in the original state even if the velocity zero is reached and the DONE output set.

This FB is coupled to MC_GroupContinue. Issuing MC_GroupContinue transfers the program back to the situation at issuing MC_GroupInterrupt.

Further motion commands may be accepted by the group.

3.4.5.9.8 MC_MoveCircularAbsolute

Instructions	LD Performance	ST Performance
MC_MoveCircularAbsolute fbMC_MoveCircularAbsolute_1 MC_MoveCircularAbsolute AxesGroup Execute CircMode AuxPoint EndPoint PathChoice Velocity Acceleration Deceleration Jerk CoordSystem BufferMode TransitionMode TransitionParameter AxesGroup Done Busy Active CommandAborted Error ErrorID		AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), CircMode:= (* MC_CIRC_MODE *), AuxPoint:=(* REAL *), EndPoint:=(* REAL *), PathChoice:= (* MC_CIRC_PATHCHOICE *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), TransitionMode:= (* MC_TRANSITION_MODE*), TransitionParameter:= (* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
	Execute	BOOL	Start the action at rising edge

Pin	Name	Type	Description
VAR_INPUT	C i r c - Mode	MC_CIRC_MODE	Specifies the meaning of the input signals 'AuxPoint' and 'CircDirection'
	AuxPoint	REAL	Array [1..N] of absolute positions for each dimension in the coordinate system specified by the input signal
	EndPoint	REAL	Array [1..N] of absolute positions for each dimension in the coordinate system specified by the input signal
	Path Choice	MC_CIRC_PATHCHOICE	Choice of path: CLOCKWISE or COUNTERCLOCKWISE
	Velocity	REAL	Maximum Velocity for the path for the coordinate system in which the path is defined. Always positive. Not necessarily reached
	Acceleration	REAL	Maximum acceleration.
	Deceleration	REAL	Maximum deceleration
	Jerk	REAL	Maximum jerk
	Coord System	MC_COORD_SYSTEM	Reference to the applicable coordinate system:ACS, MCS, PCS
	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB relative to the previous block.
	TransitionMode	MC_TRANSITION_MODE	Overview of Transition Mode

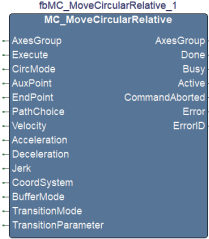
Pin	Name	Type	Description
	Transition Parameter	REAL	Overview of Transition Mode
VAR_OUTPUT	Done	BOOL	Commanded end positions reached for all axes
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This function block commands an interpolated circular movement on an axes group from the actual position of the TCP. The end point as well as the auxiliary point (meaning depending on applied mode, see below) are defined absolutely in the specified coordinate system.

3.4.5.9.9 MC_MoveCircularRelative

Instructions	LD Performance	ST Performance
		fbMC_MoveCircularRelative_1(AxesGroup:=(*AXES_GROUP_REF*), Execute:=(*BOOL*), CircMode:=(*MC_CIRC_MODE*), AuxPoint:=(*REAL*), EndPoint:=(*REAL*), PathChoice:=(*MC_CIRC_PATHCHOICE*),

Instructions	LD Performance	ST Performance
MC_MoveCircularRelative		Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), CoordSystem:=(* MC_COORD_SYSTEM *), BufferMode:=(* MC_BUFFER_MODE *), TransitionMode:=(* MC_TRANSITION_MODE *), TransitionParameter:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
	Execute	BOOL	Start the action at rising edge
	CircMode	MC_CIRC_MODE	Specifies the meaning of the input signals 'AuxPoint' and 'CircDirection'
	AuxPoint	REAL	Array [1..N] of absolute positions for each dimension in the coordinate system specified by the input signal

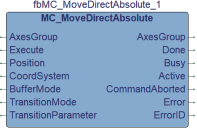
Pin	Name	Type	Description
VAR_INPUT	EndPoint	REAL	Array [1..N] of absolute positions for each dimension in the coordinate system specified by the input signal
	Path Choice	MC_CIRC_PATH_CHOICE	Choice of path: CLOCKWISE or COUNTERCLOCKWISE
	Velocity	REAL	Maximum Velocity for the path for the coordinate system in which the path is defined. Always positive. Not necessarily reached
	Acceleration	REAL	Maximum acceleration.
	Deceleration	REAL	Maximum deceleration
	Jerk	REAL	Maximum jerk
	Coord System	MC_COORD_SYSTEM	Reference to the applicable coordinate system:ACS, MCS, PCS
	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB relative to the previous block.
	TransitionMode	MC_TRANSITION_MODE	Overview of Transition Mode
	TransitionParameter	REAL	Overview of Transition Mode
VAR_OUTPUT	Done	BOOL	Commanded end positions reached for all axes
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group

Pin	Name	Type	Description
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This function block commands an interpolated circular movement on an axes group from the actual position of the TCP. The end point as well as the auxiliary point (meaning depending on applied mode, see below) are defined in the specified coordinate system relatively to the starting point

3.4.5.9.10 MC_MoveDirectAbsolute

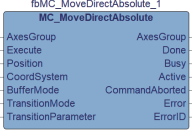
Instructions	LD Performance	ST Performance
MC_MoveDirectAbsolute		<pre>fbMC_MoveDirectAbsolute_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Position:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), TransitionMode:= (* MC_TRANSITION_MODE *), TransitionParameter:= (* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));</pre>

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
	Position	REAL	Array [1..N] of end position for each dimension in the specified coordinate system.
	Coord System	MC_COORD_SYSTEM	Reference to the applicable coordinate system: ACS, MCS, PCS
	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB relative to the previous block.
	TransitionMode	MC_TRANSITION_MODE	Overview of Transition Mode
	TransitionParameter	REAL	Overview of Transition Mode
VAR_OUTPUT	Done	BOOL	Commanded end positions reached for all axes
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This function block commands a movement of an axes group to the specified absolute position in the specified coordinate system without taking care of how (on which path) the target position is reached.

3.4.5.9.11 MC_MoveDirectRelative

Instruc-tions	LD Performance	ST Performance
MC_Move Direct Relative		fbMC_MoveDirectRelative_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), TransitionMode:= (* MC_TRANSITION_MODE *), TransitionParameter:= (* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=> (* BOOL *), Error=>(* BOOL *), ErrorID=> (* MC_ERRORCODE *));

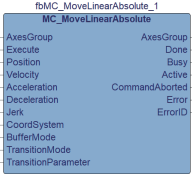
Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GRP- OUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
	Distance	REAL	Array [1..N] of end position for each dimension in the specified coordinate system.
	Coord System	MC_COORD_SYS- TEM	Reference to the applicable coordinate system: ACS, MCS, PCS

Pin	Name	Type	Description
VAR_OUTPUT	Buffer Mode	MC_BUFFER- ER_MODE	Defines the chronological sequence of the FB relative to the previous block.
	TransitionMode	MC_TRANSITION- MODE	Overview of Transition Mode
	Transition Parameter	REAL	Overview of Transition Mode
	Done	BOOL	Commanded end positions reached for all axes
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ER- RORCODE	Error identification

FB Explanation
This function block commands a movement of an axes group to a relative position without taking care of how (on which path) the target position is reached. Start position is the actual position of the TCP.

3.4.5.9.12 MC_MoveLinearAbsolute

Instruc-tions	LD Performance	ST Performance
		fbMC_MoveLinearAbsolute_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Position:=(* REAL *),

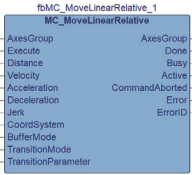
Instructions	LD Performance	ST Performance
MC_MoveLinearAbsolute		Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), CoordSystem:=(* MC_COORD_SYSTEM *), BufferMode:=(* MC_BUFFER_MODE *), TransitionMode:=(* MC_TRANSITION_MODE *), TransitionParameter:=(* REAL *), Done=>(* BOOL *), Busy=>(* BOOL *), Active=>(* BOOL *), CommandAborted=>(* BOOL *), Error=>(* BOOL *), ErrorID=>(* MC_ERRORCODE *));

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
VAR_INPUT	Execute	BOOL	Start the action at rising edge
	Position	REAL	absolute end positions for each dimension in the specified coordinate system.
	Velocity	REAL	Maximum Velocity [u/s] for the path for the coordinate system in which the path is defined.
	Acceleration	REAL	Maximum acceleration. Always positive.

Pin	Name	Type	Description
	Deceleration	REAL	Maximum deceleration. Always positive.
	Jerk	REAL	Maximum jerk. Always positive.
	Coord System	MC_COORD_SYSTEM	Reference to the applicable coordinate system: ACS, MCS, PCS
	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB relative to the previous block.
	TransitionMode	MC_TRANSITION_MODE	Overview of Transition Mode
	TransitionParameter	REAL	Overview of Transition Mode
VAR_OUTPUT	Done	BOOL	Commanded end positions reached for all axes
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation
This function block commands an interpolated linear movement on an axes group from the actual position of the TCP to an absolute position in the specified coordinate system.

3.4.5.9.13 MC_MoveLinearRelative

Instruc-tions	LD Performance	ST Performance
MC_MoveLinearRelative		<pre>fbMC_MoveLinearRelative_1(AxesGroup:= (* AXES_GROUP_REF *), Execute:=(* BOOL *), Distance:=(* REAL *), Velocity:=(* REAL *), Acceleration:=(* REAL *), Deceleration:=(* REAL *), Jerk:=(* REAL *), CoordSystem:= (* MC_COORD_SYSTEM *), BufferMode:= (* MC_BUFFER_MODE *), TransitionMode:= (* MC_TRANSITION_MODE *), TransitionParameter := (* REAL *), Done => (* BOOL *), Busy => (* BOOL *), Active => (* BOOL *), CommandAborted => (* BOOL *), Error => (* BOOL *), ErrorID => (* MC_ERRORCODE *));</pre>

Pin	Name	Type	Description
VAR_IN_OUT	Axes Group	AXES_GROUP_REF	Reference to a group of axes
	Execute	BOOL	Start the action at rising edge
	Distance	REAL	relative distances for each dimension in the specified coordinate system.

Pin	Name	Type	Description
VAR_INPUT	Velocity	REAL	Maximum Velocity [u/s] for the path for the coordinate system in which the path is defined.
	Acceleration	REAL	Maximum acceleration. Always positive.
	Deceleration	REAL	Maximum deceleration. Always positive.
	Jerk	REAL	Maximum jerk. Always positive.
	Coord System	MC_COORD_SYSTEM	Reference to the applicable coordinate system: ACS, MCS, PCS
	Buffer Mode	MC_BUFFER_MODE	Defines the chronological sequence of the FB relative to the previous block.
	Transition Mode	MC_TRANSITION_MODE	Overview of Transition Mode
VAR_OUTPUT	Transition Parameter	REAL	Overview of Transition Mode
	Done	BOOL	Commanded end positions reached for all axes
	Busy	BOOL	The FB is not finished
	Active	BOOL	Indicates that the FB has control on the axes group
	Command Aborted	BOOL	Command is aborted by another command
	Error	BOOL	Signals that an error has occurred within the function block
	ErrorID	MC_ERRORCODE	Error identification

FB Explanation

This function block commands an interpolated linear movement on an axes group from the actual position of the TCP to a relative position in the specified coordinate system.

MC_GroupContinue
MC_GroupDisable
MC_GroupEnable
MC_GroupHalt
MC_GroupStop
MC_GroupHome
MC_GroupInterrupt
MC_MoveCircularAbsolute
MC_MoveCircularRelative
MC_MoveDirectAbsolute
MC_MoveDirectRelative
MC_MoveLinearAbsolute
MC_MoveLinearRelative

3.5 EtherCAT

3.5.1 Definitions

This definition folder contains the definitions used by the EtherCAT library, including enums and structures.

3.5.1.1 Enumerations

ECATDCSTATE		
Name	Type	Description
ECAT_DC-STATE_IDLE	0	The master clock is completely independent from the slaves
ECAT_DC-STATE_SYN-CHRONIZING	1	The master is making preliminary measurements to synchronize with the slaves
ECAT_DC-STATE_WAITING_START_TIME	2	After the master has synced with the slaves, you have to wait for the START_TIME, which is the first time the slaves implement I/O

ECATDCSTATE		
Name	Type	Description
ECAT_DC-STATE_COM-PENSATING	3	If the master clock is synchronous with that of the slaves and after START_TIME has passed, the management goes into this state, which means that the Master is constantly tracking the clocks of the net
ECAT_STATE_NOT_SYN-CHRONIZED	4	If for any reason the master fails to follow the network clock, then management goes into this state and all the slaves involved go to the SAFE_OP state

ECATERORR		
Name	Type	Description
ECAT_ERROR_KE00000_ERROR_OK	0	No errors
ECAT_ERROR_KE10000_FRAME_CORRUPTED	10000	Corrupted frame
ECAT_ERROR_KE10001_MASTER_NETWORK_CHECK	10001	Master found network check error
ECAT_ERROR_KE10002_FRAME_CORRUPTED_WCK	10002	Corrupted frame, error in Wkc
ECAT_ERROR_KE10003_REGISTER_ACCESS	10003	Register access failed
ECAT_ERROR_KE10004_SLAVE_NOT_FOUND	10004	Slave not found for variable
ECAT_ERROR_KE10005_CONFIG_FILE_NOT_FOUND	10005	Configuration file not found
ECAT_ERROR_KE10006_CONFIG_FILE_PARSE	10006	Configuration file parsing error
ECAT_ERROR_KE10007_SDO_RESPONSE	10007	Failed to get SDO response
ECAT_ERROR_KE10008_INTERNAL	10008	Internal error

ECATERORR		
Name	Type	Description
ECAT_ERROR_KE10009_SDO_INTERNAL	10009	SDO internal error
ECAT_ERROR_KE10010_SDO_WRITE	10010	SDO Write error
ECAT_ERROR_KE10011_SDO_WRITE	10011	SDO Write error
ECAT_ERROR_KE10012_SDO_RESPONSE	10012	Failed to get SDO response
ECAT_ERROR_KE10013_PDO_RX_MAPPING	10013	RX PDO mapping error
ECAT_ERROR_KE10014_PDO_TX_MAPPING	10014	TX PDO mapping error
ECAT_ERROR_KE10015_PDO_RX_MAPPING_CNT	10015	RX PDO mapping error setting mapped object count
ECAT_ERROR_KE10016_PDO_TX_MAPPING_CNT	10016	TX PDO mapping error setting mapped object count
ECAT_ERROR_KE10017_PDO_ASSIGN_SMANAG	10017	Failed to assign Syn- cManager to PDO
ECAT_ERROR_KE10018_PDO_CNT_SMANAG	10018	Failed to set Sync- Manager mapped PDO count
ECAT_ERROR_KE10019_SDO_WRITE	10019	SDO Write error
ECAT_ERROR_KE10020_SLAVE_STATE_CHANGE	10020	Slave failed to change state
ECAT_ERROR_KE10021_EOE_VIRTUAL_MAC	10021	EOE: VirtualMAC not assigned
ECAT_ERROR_KE10022_EOE_INITCMD	10022	EOE: InitCmd failed
ECAT_ERROR_KE10023_EOE_IP_MAC	10023	EOE: error setting IP and MAC
ECAT_ERROR_KE10024_EOE_PARAMS	10024	EOE: error setting params

ECATERORR		
Name	Type	Description
ECAT_ERROR_KE10025_EOE_MISS_PREVF	10025	EOE: previous frag- ment missing
ECAT_ERROR_KE10026_EOE_FRAGMENT_LEN	10026	EOE: fragment exceeds maximum length
ECAT_ERROR_KE10027_EOE_UNKNOWN	10027	EOE: unknown frame
ECAT_ERROR_KE10028_EOE_CHANGE_STATE	10028	EOE: failed to change state
ECAT_ERROR_KE10029_FOE_TIMEOUT	10029	FOE: timeout
ECAT_ERROR_KE10030_FOE_TIMEOUT	10030	FOE: timeout
ECAT_ERROR_KE10031_FOE_SEQUENCE	10031	FOE: sequence error
ECAT_ERROR_KE10032_FOE_GENERAL	10032	FOE: general error
ECAT_ERROR_KE10033_FOE_OPCODE	10033	FOE: invalid opCode
ECAT_ERROR_KE10034_FOE_TIMEOUT	10034	FOE: timeout
ECAT_ERROR_KE10035_INTERNAL	10035	Internal error
ECAT_ERROR_KE10036_SSC	10036	SSC write error
ECAT_ERROR_KE10037_MAILBOX_REPLY	10037	Mailbox error reply during polling
ECAT_ERROR_KE10038_MAILBOX_UNKNOWN	10038	Mailbox unknown protocol
ECAT_ERROR_KE10039_MAC	10039	Error retrieving net- work mac address
ECAT_ERROR_KE10040_JOIN_MULTICAST	10040	Error joining multi- cast address
ECAT_ERROR_KE10041_NET_IFACE	10041	Error opening net- work interface
ECAT_ERROR_KE10042_NET_GLB_STAT	10042	Global network status not found in any cyclic frame

ECATERERROR		
Name	Type	Description
ECAT_ERROR_KE10043_MASTER_INITCMD	10043	Master initCmd validation failed
ECAT_ERROR_KE10044_MASTER_ACYCLIC	10044	Master received a wrong acyclic frame
ECAT_ERROR_KE10045_REGISTER_ACCESS	10045	Register access failed
ECAT_ERROR_KE10046_MASTER_WKC	10046	Master detected a Wkc mismatch for the SyncUnit
ECAT_ERROR_KE10047_MASTER_PROP_DELAY	10047	Master couldn't calculate propagation delays properly
ECAT_ERROR_KE10048_REGISTER_ACCESS	10048	Register access failed
ECAT_ERROR_KE10049_MASTER_DC_DELAY	10049	Master delay calculation for Distributed Clock failed: wrong net configuration
ECAT_ERROR_KE10050_MASTER_PROP_DELAYS	10050	Master detected a propagation delay calculation error for slave
ECAT_ERROR_KE10051_MASTER_PROP_DELAYS	10051	Master detected a propagation delay calculation error for slave
ECAT_ERROR_KE10052_MASTER_PROP_DELAYS	10052	Master detected a propagation delay calculation error for slave
ECAT_ERROR_KE10053_REGISTER_ACCESS	10053	Register access failed
ECAT_ERROR_KE10054_REGISTER_ACCESS	10054	Register access failed
ECAT_ERROR_KE10055_REGISTER_ACCESS	10055	Register access failed
ECAT_ERROR_KE10056_REGISTER_ACCESS	10056	Register access failed
ECAT_ERROR_KE10057_REGISTER_ACCESS	10057	Register access failed

ECATERERROR		
Name	Type	Description
ECAT_ERROR_KE10058_NET_SYNC_ACYCLIC	10058	The network has NOT been synchronized after # acyclic frames
ECAT_ERROR_KE10059_REGISTER_ACCESS	10059	Register access failed
ECAT_ERROR_KE10060_MASTER_CLOCK_64	10060	Master detected an error while reading 64bit clock acyclically
ECAT_ERROR_KE10061_RESET_COUNT	10061	Reset procedure failed
ECAT_ERROR_KE10062_MASTER_INITCMD	10062	Master initCmd: failed to change state
ECAT_ERROR_KE10063_MASTER_SLAVES_COUNT	10063	Master detected wrong number of slaves
ECAT_ERROR_KE10064_SLAVE_PARENT	10064	Cannot find slave parent
ECAT_ERROR_KE10065_SLAVE_PORT	10065	Slave port not opened but required as previous port from slave
ECAT_ERROR_KE10067_MASTER_INITCMD	10067	Master initCmd validation failed
ECAT_ERROR_KE10068_MASTER_INITCMD	10068	Master initCmd validation failed
ECAT_ERROR_KE10069_SLAVE_MISSING	10069	Slave missing
ECAT_ERROR_KE10070_FILE_TRANSF_ACCESS	10070	File transfer access error
ECAT_ERROR_KE10071_INTERNAL	10071	Internal error
ECAT_ERROR_KE10072_SLAVE_MISSING_ERROR	10072	Slave comes back from missing with an error
ECAT_ERROR_KE10073_SLAVE_LEN	10073	Slave requires a wrong length read command for the mailbox
ECAT_ERROR_KE10074_SLAVE_RESET	10074	Slave reset failed

ECATERERROR		
Name	Type	Description
ECAT_ERROR_KE10075_SLAVE_DC	10075	Slave has been configured to use DC but doesn't support DC
ECAT_ERROR_KE10077_INTERNAL	10077	Internal error
ECAT_ERROR_KE10078_INTERNAL	10078	Internal error
ECAT_ERROR_KE10079_INTERNAL	10079	Internal error
ECAT_ERROR_KE10080_INTERNAL	10080	Internal error
ECAT_ERROR_KE10081_ACYCLIC	10081	Acyclic frame has not been received yet
ECAT_ERROR_KE10082_SLAVE_COUNT	10082	Slave count mismatch
ECAT_ERROR_KE10083_EOE_OVER_EOE	10083	Error sending EoE packet through EoE service for slave
ECAT_ERROR_KE10084_MASTER_DC_SYNCM	10084	Master uses DC but SyncMaster not found
ECAT_ERROR_KE10085_INTERNAL	10085	Internal error
ECAT_ERROR_KE19999_INTERNAL	19999	Internal error

ECATEVENTID		
Name	Type	Description
ECAT_EVENT_NONE	0	NO Event
ECAT_EVENT_ERROR	1	Error from a slave or from the master
ECAT_EVENT_EMERGENCY	2	Emergency event from a CoE slave
ECAT_EVENT_NODE_STATUSCHANGED	3	Signaled when a slave changes its state
ECAT_EVENT_DC_STATUSCHANGED	4	Signaled when the master changes its state while tracking the Distributed Clock

ECATEVENTID		
Name	Type	Description
ECAT_EVENT_NETWORKRECOVERY	5	Network restarted, to recover from a very critical error
ECAT_EVENT_NETWORKRESYNC	6	Signaled when all the slaves are ready to enter in operational state
ECAT_EVENT_SLAVEVERESYNC	7	Signaled when a slave is ready to enter in operational state

ECATINITERRORCODE		
Name	Type	Description
ECAT_INITERR_OK	0	No error
ECAT_INITERR_DISABLED	500	EtherCAT is disabled
ECAT_INITERR_SYS	501	Internal system error
ECAT_INITERR_XML	502	Error in the XML parsing phase
ECAT_INITERR_EMPTYCFG	503	Empty configuration file
ECAT_INITERR_OUTOFMEMORY	504	Memory exhausted
ECAT_INITERR_FILE	505	IO error in reading
ECAT_INITERR_NO_CONFIG	506	Config node not present
ECAT_INITERR_NO_EN-NAME	507	No ENI name
ECAT_INITERR_NO_DBTYPE	508	No datablock type
ECAT_INITERR_NO_DBNUM	509	No datablock index
ECAT_INITERR_NO_DBOFF	510	No datablock offset
ECAT_INITERR_NO_BITLEN	511	Missing length in bits
ECAT_INITERR_NO_DB-BYTOFF	512	Missing byte offset
ECAT_INITERR_DB-MAPDBTYPEINV	513	Invalid I/O mapping
ECAT_INITERR_ENIVAR-INV	514	Configured variable doesn't match with ENI fileMax number of variables exceededDistributed clock is not supportedError in array length

ECATINITERERRORCODE		
Name	Type	Description
ECAT_INITERR_TOOMANYVARIABLES	515	Configuration file corrupted
ECAT_INITERR_DC_UN-SUPPORTED	516	Configuration check for ethernet interface failed
ECAT_INITERR_WRONGGARRAYLEN	517	No error
ECAT_INITERR_CONFIG_CORRUPTED	518	EtherCAT is disabled
ECAT_INITERR_WRONG_INTF_CHECK	519	Internal system error

ECATSTATE		
Name	Type	Description
ECAT_STATE_UNKNOWN	0	Unknown State
ECAT_STATE_INIT	1	Init State
ECAT_STATE_PREOP	2	PreOperational State-Boot State
ECAT_STATE_BOOT	3	SafeOperational State
ECAT_STATE_SAFEOP	4	Operational State
ECAT_STATE_OP	8	Stopped State
ECAT_STATE_STOPPED	254	Fatal State
ECAT_STATE_FATAL	255	Unknown State

3.5.1.2 Structures

ECATDCSTATUSEVENT		
Name	Type	Description
event_id	SINT	Its value is fixed: 4 corresponding to DCSTATUSCHANGED
ecat_chn	USINT	Master channel, for multi-master devices
from	USINT	Previous State
to	USINT	Actual State

ECATEMCEVENT		
Name	Type	Description
event_id	SINT	This value is fixed: 2, corresponding to EMERGENCY
ecat_chn	USINT	Master channel, for multi-master devices
node	UINT	Physical address of the slave that signaled the event
emcyCode	UINT	CoE Emergency Code
emcyReg	USINT	CoE Emergency Register
emcyData	ARRAY[0..4] OF USINT	Emergency Data

ECATERROREVENT		
Name	Type	Description
event_id	SINT	Its value is fixed: 1, corresponding to ERROR
ecat_chn	USINT	Master channel, for multi-master devices The slave's physical address that signaled the event. If the error is an error from the master, this field assumes the value 0
node	UINT	Error Code
errorCode	ECATERROR	Error specific code parameters
iP1	DINT	Error specific code parameters
iP2	DINT	Error specific code parameters
iP3	DINT	Its value is fixed: 1, corresponding to ERROR

ECATMASTERSTATUS		
Name	Type	Description
error_code	INT	Last error code in the initialization phase of the master stack (see ECATINITERERRORCODE enum)
ecat_enable	BOOL	EtherCAT stack enabled

ECATMASTERSTATUS		
Name	Type	Description
data_exch_enable	BOOL	Data exchange enabled: when a valid mapping has been made between the exchanged objects and the associated PLC vars. This is optional, the network can even go into operational mode without configuring the objects
chn	USINT	Channel number, for systems with more than one EtherCAT network
eni_ok	BOOL	ENI read and parsed successfully
xml_ok	BOOL	XML with additional configuration read and parsed successfully
init_ok	BOOL	Init of the stack completed successfully
num_slaves	UINT	Number of defined slaves
dc_sync_master	UINT	Distributed Clock Sync Master's Physical Address
net_stat	ECAT-STATE	Current network state
target_net_stat	ECAT-STATE	State that the master set for all the slaves, regardless of they reached it or not
dc_stat	ECAT-DC-STATE	Master sync state with sync master (only if DC is used). Master must be in 3-COMPENSATING before slaves can reach SAFEOP/OP states
use_dc	BOOL	DC is in use and set for at least one slave
drivesClock	BOOL	EtherCAT stack gives the clock to whole system including PLC. Important to detect failure in the first slave, because in that case the EtherCAT stack may lose packets and synchronization with the network
num_slaves_cfg	UINT	Number of slaves correctly configured by init commands
num_pdo_tx	UDINT	Number of transmit PDOs
num_pdo_rx	UDINT	Number of receive PDOs
network_ok	BOOL	EtherCAT Network is running with no errors: all the slaves reach the OP state. In case of any error will switch to false

ECATMASTERSTATUS		
Name	Type	Description
network_starting	BOOL	EtherCAT Network is still starting
num_frame_tx	ULINT	Total frames sent
num_frame_tx_cyclic	ULINT	Periodic frames sent
num_frame_tx_acyclic	ULINT	Non-periodic frames sent
num_frame_rx	ULINT	Total frames received
num_frame_rx_cyclic	ULINT	Periodic frames received
num_frame_rx_acyclic	ULINT	Non-periodic frames received
num_frame_lost	ULINT	Total frames lost
num_frame_lost_cyclic	ULINT	Periodic frames lost
num_frame_lost_acyclic	ULINT	Non-periodic frames lost

ECATNETWORKRECOVERYEVENT		
Name	Type	Description
event_id	SINT	Its value is fixed: 8 which corresponds to NETWORKRECOVERY
ecat_chn	USINT	Master channel, for multi-master devices

ECATNODESTATUSUSEVENT		
Name	Type	Description
event_id	SINT	Its value is fixed: 3, corresponding to NODESTATUSCHANGED
ecat_chn	USINT	Master channel, for multi-master devices
node	UINT	The slave's physical address that signaled the event
from	UINT	Previous State
to	UINT	Actual State

ECATRESYNCINFO		
Name	Type	Description
event_id	SINT	Its value can be 6 or 7: depending respectively of which is ready to enter in operational state: network/master or a slave
ecat_chn	USINT	Master channel, for multi-master devices
node	UINT	The slave's physical address that signaled the event
is_master	BOOL	TRUE if it is a NETWORKRESYNC event, FALSE if it is a SLAVERESYNC event
ready_to_start	BOOL	FALSE if there is an error inhibiting the slave to goes in operational state

ECATSLAVEMONITOR		
Name	Type	Description
idx_requested	UINT	Writable field to select the slave to monitor
idx_actual	UINT	If the value in idx_requested is valid, this field will assume the same value, and the relative status structure is monitored; otherwise if idx_requested is not valid, will assume the value 65535
status	ECAT-SLAVE-STATUS	Actual state of slave of idx_actual

ECATSLAVESTATUS		
Name	Type	Description
cfg	BOOL	Configured: slave has received all the init commands
connected	BOOL	Present and physically connected
address	UINT	Physical address, assigned by default from master or configured by user

ECATSLAVESTATUS		
Name	Type	Description
state	ECAT-STATE	Current state of the slave
error_cnt	UDINT	Number of errors detected
vendor_id	UDINT	Vendor ID; valid only if Check VendorID or Always read identification information is enabled
product_code	UDINT	Product code; valid only if Check ProductCode or Always read identification information is enabled
revision_number	UDINT	Revision Number; valid only if Check RevisionNo or Always read identification information is enabled
serial_number	UDINT	Serial Number; valid only if Check SerialNo or Always read identification information is enabled
lastEmergencyCode	UINT	Last Emergency Code
lastEmergencyReg	USINT	Last Emergency Register
lastEmergencyData	ARRAY [0..4] OF USINT	Emergency Data
index	UINT	Position in network
name	STRING [32]	Name
lastErrorCode	UINT	Last Error Code
lastErrorReg	USINT	Last Error Register
lastErrorData	ARRAY [0..4] OF USINT	Error specific Data
AIStatuscode	UINT	AIStatusCode register

3.5.2 Master control functions

This folder contains the control functions for the main station, including stopping, starting, and resetting the main station.

3.5.2.1 sysECAT_Recovery

Recover EtherCAT Network.This is a non blocking function which could be called in any task.. But this function cannot be called recursively, otherwise abnormal behavior may occur.

Pin Type	Name	Type	Description
VAR_INPUT	dummy	UINT	Dummy

Result Type	Description
BOOL	TRUE:Succeed FALSE:Failed

3.5.2.2 sysECAT_StartMaster

Starts a stopped EtherCAT master.
This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	dummy	UINT	Dummy

Result Type	Description
BOOL	TRUE:Succeed FALSE:Failed

3.5.2.3 sysECAT_StopMaster

Stops a running EtherCAT master.
This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	dummy	UINT	Dummy

Result Type	Description
BOOL	TRUE:Succeed FALSE:Failed

3.5.3 Master diagnostic functions

3.5.3.1 sysECAT_GetSlaveCfgByAddress

This function returns the field 'cfg' of the ECAT-SLAVESTATUS structure for the slave selected by its physical address.

This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	address	UINT	Slave Physical Address

Result Type	Description
BOOL	TRUE:Succeed FALSE:Failed

3.5.3.2 sysECAT_GetSlaveCfgByIndex

This function returns the field 'cfg' of the ECAT-SLAVESTATUS structure for the slave selected by positional index in the network.

This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	index	UINT	Slave Index

Result Type	Description
BOOL	TRUE:Succeed FALSE:Failed

3.5.3.3 sysECAT_GetSlaveDiagByAddress

This function returns a pointer to the ECATSLAVESTATUS structure for the slave selected by its physical address. It returns a value of type @ECATSLAVESTATUS, or NULL in case of error.

This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	address	UINT	Slave Index

Result Type	Description
@ECATSLAVESTATUS	It returns a value of type @ECAT-SLAVESTATUS, or NULL in case of error.

3.5.3.4 sysECAT_GetSlaveDiagByIndex

This function returns a pointer to the ECATSLAVES-TATUS structure for the slave selected by positional index in the network. This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	index	UINT	Slave Index

Result Type	Description
@ECATSLAVES-TATUS	It returns a value of type @ECAT-SLAVESTATUS, or NULL in case of error.

3.5.3.5 sysECAT_GetSlavePdoVarByAddress

This function returns PDO mapped variable address by slave address

Pin Type	Name	Type	Description
VAR_INPUT	slave_address	UINT	Slave Physical Address
VAR_INPUT	index	UINT	Object Index
VAR_INPUT	sindex	USINT	Object Subindex

Result Type	Description
PVOID	Returns a pointer of type void, which points to the address of a variable mapped from the slave device, and returns NULL in case of an error.

3.5.3.6 sysECAT_GetSlavePdoVarByIndex

This function returns PDO mapped variable address by slave index

Pin Type	Name	Type	Description
VAR_INPUT	slave_index	UINT	Slave Index
VAR_INPUT	index	UINT	Object Index
VAR_INPUT	sindex	USINT	Object Subindex

Result Type	Description
PVOID	Returns a pointer of type void, which points to the address of a variable mapped from the slave device, and returns NULL in case of an error.

3.5.3.7 sysECAT_GetSlavePresByAddress

This function returns the field 'connected' of the ECATSLAVESTATUS structure for the slave selected by its physical address.
This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	address	UINT	Slave Index

Result Type	Description
BOOL	TRUE:Succeed FALSE:Failed

3.5.3.8 sysECAT_GetSlavePresByIndex

This function returns the field 'connected' of the ECATSLAVESTATUS structure for the slave selected by positional index in the network.
This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	index	UINT	Slave Index

Result Type	Description
BOOL	TRUE:Succeed FALSE:Failed

3.5.3.9 sysECAT_GetSlaveStateByAddress

This function returns the field 'state' of the ECAT-SLAVESTATUS structure for the slave selected by its physical address.
This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	address	UINT	Slave Index

Result Type	Description
ECATSTATE	status of the slave

3.5.3.10 sysECAT_GetSlaveStateByIndex

This function returns the field 'state' of the ECAT-SLAVESTATUS structure for the slave selected by positional index in the network.

This is a non blocking function which could be called in any task.

Pin Type	Name	Type	Description
VAR_INPUT	index	UINT	Slave Index

Result Type	Description
ECATSTATE	status of the slave

3.5.4 SDO function blocks

3.5.4.1 sysECAT_Sdo_Read

This function block can be used to read object values from a slave device that supports the CoE protocol, and the device uses SDO commands. It cannot be called from a timed task, but it can be called from a background task or an event callback task. Only object lengths less than or equal to 4 bytes are supported. This is a blocking function block, and it returns the result after a single execution.

Pin Type	Name	Type	Description
VAR_INPUT	slave_ad- dress	UINT	Slave Physical Ad- dress
VAR_INPUT	index	UINT	Object Index
VAR_INPUT	sindex	USINT	Object Subindex
VAR_INPUT	lenght	UDINT	Object Length
VAR_OUTPUT	value	DINT	Value read
VAR_OUTPUT	errorCode	DINT	Error code - see function block de- scription for details
VAR_OUTPUT	result	BOOL	TRUE if read success

3.5.4.2 sysECAT_Sdo_Read2

This function block can be used to read object values in a slave supporting CoE protocol with SDO commands.

It can be called by any task, because it is not blocking and allows underlying parallel execution.

Only object lengths <= 4 bytes are supported.

Multiple instances are allowed.

The typical use is to instance many function blocks and call them in every iteration of a specified task.

The rising edge of the enable variable determines the start of processing.

The enable variable must be kept active at each subsequent invocation until processing is terminated.

Changing any value of the function block parameters (also the enable) during its execution has no effect.

Processing is finished when:

- 1)The done or error variables are set true by the underlying engine
- 2)The caller invokes the function with the enable variable false

Pin Type	Name	Type	Description
VAR_INPUT	slave_ address	UINT	Slave Physical Ad- dress
VAR_INPUT	index	UINT	Object Index
VAR_INPUT	sindex	USINT	Object Subindex
VAR_INPUT	length	USINT	Object Length (1 - 4)
VAR_INPUT	enable	BOOL	Enable on rising edge
VAR_OUTPUT	active	BOOL	TRUE when FB is executing and pro- cessing the SDO
VAR_OUTPUT	done	BOOL	Function block pro- cessed
VAR_OUTPUT	error	BOOL	If done, error = TRUE if errors occured
VAR_OUTPUT	value	DINT	Value read
VAR_OUTPUT	error Code	DINT	Error code (valid if error = TRUE) - see function block de- scription for details

3.5.4.3 sysECAT_Sdo_Write

This function block can be used to write object values in a slave supporting CoE protocol with SDO commands.It can not be called from a timed task, it can be called from the background task or from an event callback task.Only object lengths <= 4 bytes are supported.

It is a blocking function block which returns results after a single execution.

Pin Type	Name	Type	Description
VAR_INPUT	slave_address	UINT	Slave Physical Address
VAR_INPUT	index	UINT	Object Index
VAR_INPUT	sindex	USINT	Object Subindex
VAR_INPUT	length	UDINT	Object Length
VAR_INPUT	value	DINT	Value to be written
VAR_OUTPUT	error Code	DINT	Error code - see description for details
VAR_OUTPUT	result	BOOL	TRUE if write success

3.5.4.4 sysECAT_Sdo_Write2

This function block can be used to write object values in a slave supporting CoE protocol with SDO commands. It can be called by any task, because it is not blocking and allows underlying parallel execution. Only object lengths <= 4 bytes are supported. Multiple instances are allowed.

The typical use is to instance many function blocks and call them in every iteration of a specified task. The rising edge of the enable variable determines the start of processing.

The enable variable must be kept active at each subsequent invocation until processing is terminated. Changing any value of the function block parameters (also the enable) during its execution has no effect. Processing is finished when:

The done or error variables are set true by the underlying engine

The caller invokes the function with the enable variable false

Pin Type	Name	Type	Description
VAR_INPUT	slave_address	UINT	Slave Physical Address
VAR_INPUT	index	UINT	Object Index
VAR_INPUT	sindex	USINT	Object Subindex
VAR_INPUT	value	DINT	Value to be written
VAR_INPUT	length	USINT	Object Length (1 - 4)
VAR_INPUT	enable	BOOL	Enable on rising edge
VAR_OUTPUT	active	BOOL	TRUE when FB is executing and processing the SDO
VAR_OUTPUT	done	BOOL	Function block processed
VAR_OUTPUT	error	BOOL	If done, error = TRUE if errors occurred
VAR_OUTPUT	error Code	DINT	Error code (valid if error = TRUE) - see function block description for details

errorCode	
Value	Description
0	in case of success
<0 internal errors:	
-11	Out of memory
-520	Master disabled or stopped
-521	Slave not present or not configured
-522	Internal timeout
-523	Errors in internal queue
-524	Parameter not valid
>0 slave errors (vendor specific codes are allowed):	
0x05030000	Toggle bit not changed
0x05040000	SDO timeout
0x05040001	Command specifier unknown
0x05040005	Out of memory
0x06010000	Unsupported Access
0x06010001	Write only entry
0x06010002	Read only entry
0x06010003	Entry can not be written because Subindex0 is not 0

errorCode	
Value	Description
0x06010004	The object can not be accessed via complete access
0x06020000	Object not existing
0x06040041	Object can not be mapped to PDO
0x06040042	Mapped Object exceeds PDO
0x06040043	Parameter is incompatible
0x06040047	Device incompatibility
0x06060000	Hardware error
0x06070010	Parameter length error
0x06070012	Parameter is too long
0x06070013	Parameter is too short
0x06090011	Subindex (Entry) not exists
0x06090030	Value exceeds
0x06090031	Value is too great
0x06090032	Value is too small
0x06090033	Detected Module Ident List (0xF030) and Configured Module Ident list (0xF050) does not match
0x06090036	Value is less than minimum value
0x08000000	General error
0x08000020	Data can not be read or written
0x08000021	Data can not be accessed because of local control
0x08000022	Data can not be read or written in the current state
0x08000023	Object is not in the object dictionary

3.5.5 Vars(EtherCAT)

The following variables are global variables in the EtherCAT library, indicating that these variables can be directly called in any task to check the status of the master or the slave.

Name	Type	Description
sysECATDCStatusChangedEvent	ECATDCSTATUSEVENT	Last variation of the master state, related to its synchronization with the Distributed Clock
sysECATEmergencyEvent	ECATEMERGENCYEVENT	Last Emergency event sent by a CoE slave
sysECATErrorEvent	ECATERROREVENT	Last error event detected on the EtherCAT
sysECATEventId	ECATEVENTID	ID of the last encountered event
sysEcatMasterStatus	ECATMASTERSTATUS	General view of the status of the EtherCAT master and network
sysECATNetworkRecoveryEvent	ECATNETWORKRECOVERYEVENT	Last event of Network Recovery
sysECATNodeStatusChangedEvent	ECATNODESTATUSEVENT	Last change of status of a slave
sysECATResyncInfo	ECATRESYNCFINFORMATION	Resynchronize information
sysEcatSlaveMonitor	ECATSLAVEMONITOR	Can be placed in watch to monitor the particular status of any slave. You can select the slave to monitor by writing its index (in the network, starting from 0) in the field idx_requested

3.6 ModbusRTU

3.6.1 Definitions

3.6.1.1 Structures

MbInpOutCmdStruct		
Name	Type	Description
StartupCmd	MbInpOutCmdMode	Startup command
CurrentCmd	MbInpOutCmdMode	Current command
FnzRead	INT	Function used to read
FnzWrite	INT	Function used to write
Flags	DWORD	Internal flags
PValue	DWORD	Pointer to current value
Reserved	DWORD	Reserved for further usage
ImageRead	LWORD	Image read
ImageWrite	LWORD	Image write

MbMRtuNetList		
Name	Type	Description
chn	USINT	Modbus master RTU channel, returned from sysMbMRtu_OpenChn
adr	USINT	Modbus slave address
cfg	BOOL	Modbus slave is added to netlist
pres	BOOL	Modbus slave is present on network
active	BOOL	Modbus slave is concurrent in communication
swapWords-Mode	USINT	Swap Mode field
numParamMsg	UINT	Total number of Parametrization Slave Message
numQueuedMsg	UINT	Total number of Configured Slave Message
minPolling-Time	UINT	Minimum polling time (ms)
status	M b - Status	Modbus slave state machine, referred to enumerative MbStatus

MbMRtuNetList		
Name	Type	Description
error	MbError	Modbus slave last error, referred to enumerative MbError
exception	MbException	Modbus slave last exception, referred to enumerative MbException
slaveExceptionCount	UINT	Slave Exception Error Count
slaveMessageCount	UINT	Slave Message Count
slaveNoResponseCount	UINT	Slave No Response Count
slaveNoAckCount	UINT	Slave NAK Count
slaveBusyCount	UINT	Slave Busy Count
slaveChrOvrCount	UINT	Slave Character Overrun Count
maxRetries	UINT	Maximum number of retries
disableWrite	BOOL	TRUE if write commands are disabled
reserved	USINT	reserved

MbMRtuStatus		
Name	Type	Description
cfg	BOOL	Modbus Master configured
chn	USINT	Modbus master RTU channel, returned from sysMbMRtu_OpenChn
numSlaves	USINT	Total number of Configured Slave in Net list
reserved	USINT	Reserved field
status	M b - Status	Modbus master state machine, referred to enumerative MbStatus
error	MbError	Modbus master last error, referred to enumerative MbError
numMsg	UINT	Total number of Configured Broadcast and Single Slave Message

MbMRtuStatus		
Name	Type	Description
numBroadcastMsg	UINT	Total number of Configured Broadcast Message
numBusMsgCount	UINT	Bus Message Count
numBusCommErrCount	UINT	Bus Communication Error Count

MbRtuDiagnostics		
Name	Type	Description
busMsgCount	UINT	Generic counter for messages successfully sent/received on the bus
busCommErrCount	UINT	Generic counter for failed messages sent/received on the bus
exceptionCount	UINT	Generic exception error counter
messageCount	UINT	If master: counter of messages sent. If slave: counter of messages received
noResponseCount	UINT	Counter of messages without response
slaveRspNoAckCount	UINT	(Only if master) Counter of slave NAK response messages
slaveRspBusyCount	UINT	(Only if master) Counter of slave BUSY response messages
chrOvrCount	UINT	Counter of character overrun

3.6.1.2 Enumerations

MB_RESULT_VALUE		
Name	Value	Description
MB_E_OK	0	Successful command
MB_E_WRITE_DISABLED	-349	Write commands are disabled
MB_E_POLLINGTIME	-348	Polling time not respected
MB_E_REQUESTMISMATCH	-347	Request mismatch
MB_E_NOCONNSYSTEMBUSY	-346	System busy
MB_E_NOCONNSLOTAVAILABLE	-345	Connection error

MB_RESULT_VALUE		
Name	Value	Description
MB_E_CMDPARAMERROR	-344	Invalid parameter specified
MB_E_CMDSAMEBUFFER	-343	Same buffer specified
MB_E_CMDBUFFERTOOSMALL	-342	Buffer too small
MB_E_CMDBUFFERERROR	-341	Invalid command buffer error
MB_E_CMDBUFFERSIZEERROR	-340	Invalid command buffer size error
MB_E_NETWORK	-339	Invalid network id specified
MB_E_CHECKSUM	-338	Checksum
MB_E_NEGATIVEACKNOWLEDGE	-337	Negative acknowledge
MB_E_NOTALLOWEDINTIMEDTASK	-336	Function cannot be called in a timed task
MB_E_SWITCHEDOFF	-335	Node is switched off
MB_E_ILLEGALDATAPROTOCOL	-332	Illegal data value in protocol
MB_E_TIMEOUT	-331	Time out request
MB_E_GATEWAYDEVICEFAILED	-330	Gateway target device failed to respond
MB_E_GATEWAYPATHUNAVAIL	-329	Gateway path unavailable
MB_E_MEMORYPARITYERROR	-328	Memory parity error
MB_E_SLAVEDEVICEBUSY	-327	Slave device busy
MB_E_ACKNOWLEDGE	-326	Acknowledge
MB_E_SLAVEDEVICEFAILURE	-325	Slave device failure
MB_E_ILLEGALDATAVALUE	-324	Illegal data value
MB_E_ILLEGALDATAADDR	-323	Illegal data address
MB_E_ILLEGALFUNCTION	-322	Illegal function
MB_E_IDE	-321	Ide doesn't match
MB_E_PROTOCOL	-320	Protocol error
MB_E_STOPPED	-317	Master not running
MB_E_DATATYPE	-311	Data type parameter not valid
MB_E_NODENOTEXIST	-303	Not assigned node
MB_E_NODE	-301	Not valid node

MB_RTU_NETWORK_ID		
Name	Value	Description
MB_RTU_NETWORK_0	0	Net 0
MB_RTU_NETWORK_1	1	Net 1
MB_RTU_NETWORK_2	2	Net 2
MB_RTU_NETWORK_3	3	Net 3
MB_RTU_NETWORKS	4	Maximum number of networks supported

MbBaudrate		
Name	Value	Description
MbBaud_600	600	600 bit/s
MbBaud_1200	1200	1200 bit/s
MbBaud_2400	2400	2400 bit/s
MbBaud_4800	4800	4800 bit/s
MbBaud_9600	9600	9600 bit/s
MbBaud_19200	19200	19200 bit/s
MbBaud_38400	38400	38400 bit/s
MbBaud_57600	57600	57600 bit/s
MbBaud_76800	76800	76800 bit/s
MbBaud_115200	115200	115200 bit/s
MbBaud_Auto	0	Selected by low level driver

MbChn		
Name	Value	Description
MbChn_0	0	Handler for use serial 1
MbChn_1	1	Handler for use serial 2

MbError		
Name	Value	Description
MbError_Ok	0	No error
MbError_System	1	System, internal error
MbError_CommChn	8	Error on communication channel
MbError_CommHdl	11	Invalid communication handle
MbError_CommTx	12	Impossible to send request
MbError_CommOpen	19	Cannot open communication port
MbError_SlaveMaxNum	20	Maximum number of slave

MbError		
Name	Value	Description
MbError_SlaveAddressInvalid	21	Invalid slave address
MbError_SlaveNotCfg	22	Slave not exist in netlist
MbError_SlaveInUse	23	Slave address already in use
MbError_SlaveLoss	24	Slave loss
MbError_MsgMaxNum	30	Maximum number of modbus message
MbError_MsgFncInvalid	31	Invalid modbus function
MbError_MsgFncSize	32	Invalid modbus function dimension
MbError_MsgDataBlockInvalid	33	Invalid datablock
MbError_ParamInvalid	40	System, internal error
MbError_ParamException	41	Exception occurred
MbError_ParamTimeout	42	Timeout

MbException		
Name	Value	Description
MbExc_Ok	0	No Error
MbExc_IllegalFunction	1	Illegal function
MbExc_IllegalDataAddress	2	Illegal data address
MbExc_IllegalDataValue	3	Illegal data value
MbExc_SlaveDeviceFailure	4	Slave device failure
MbExc_Acknowledge	5	Acknowledge
MbExc_SlaveDeviceBusy	6	Slave device busy
MbExc_MemoryParityError	7	Memory parity error
MbExc_GatewayPathUnavailable	10	Gateway path unavailable
MbExc_GatewayTargetDevice	11	Gateway target device failed to respond

MbFunction		
Name	Value	Description
MbFunc_ReadCoils	1	0x01 Read Coils
MbFunc_ReadDiscretInputs	2	0x02 Read Discrete Inputs
MbFunc_ReadHoldingRegs	3	0x03 Read Holding Registers
MbFunc_ReadInputRegs	4	0x04 Read Input Registers
MbFunc_WriteSingleCoil	5	0x05 Write Single Coil
MbFunc_WriteSingleReg	6	0x06 Write Single Register
MbFunc_WriteMultipleCoils	15	0x0F Write Multiple Coils
MbFunc_WriteMultipleRegs	16	0x10 Write Multiple Registers

MbInpOutCmdMode		
Name	Value	Description
MbInpOutCmdMode_Read	0	Read mode
MbInpOutCmdMode_Write	1	Write mode

MbMode		
Name	Value	Description
MbMode_8N1	0	8 bit, no parity, 1 stop bit
MbMode_8E1	1	8 bit, even parity, 1 stop bit
MbMode_8O1	2	8 bit, odd parity, 1 stop bit
MbMode_8N2	3	8 bit, no parity, 2 stop bit
MbMode_8E2	4	8 bit, even parity, 2 stop bit
MbMode_8O2	5	8 bit, odd parity, 2 stop bit
MbMode_8S1	6	8 bit, space parity, 1 stop bit
MbMode_8M1	7	8 bit, mark parity, 1 stop bit
MbMode_8S2	8	8 bit, space parity, 2 stop bit
MbMode_8M2	9	8 bit, mark parity, 2 stop bit
MbMode_Auto	255	Selected by low level driver

MbParam		
Name	Value	Description
MbParam_Init	0	Initialization
MbParam_Busy	1	Busy
MbParam_SendRequest	2	Send request
MbParam_WaitResponse	3	Wait response
MbParam_Ended	4	Terminated

MbRqErr		
Name	Value	Description
MbRqErr_Ok	0	No error occurred
MbRqErr_IllegalFunction	1	Invalid request message
MbRqErr_IllegalDataAddress	2	Illegal data address
MbRqErr_IllegalDataValue	3	Illegal data value
MbRqErr_SlaveDeviceFailure	4	Slave device failure
MbRqErr_Acknowledge	5	Acknowledge error
MbRqErr_SlaveDeviceBusy	6	Slave device busy
MbRqErr_ParityError	7	Memory parity error
MbRqErr_CommChn	8	Communication channel configuration error
MbRqErr_InvalidSlaveAddress	9	Invalid slave address (address range 0 - 247)
MbRqErr_GatewayPathUnavailable	10	Gateway path unavailable
MbRqErr_GatewayDeviceFailed	11	Gateway target device failed to respond
MbRqErr_MsgMaxElements	12	Invalid number of element for this function
MbRqErr_MsgRequestInvalid	13	Cannot correctly build the request message
MbRqErr_MsgResponseInvalid	14	Invalid response message
MbRqErr_MsgSize	15	Response size error
MbRqErr_Timeout	16	Timeout reached
MbRqErr_TaskInvalid	17	Function invoked by an invalid task
MbRqErr_Sys	18	Generic modbus error

MbRqErr		
Name	Value	Description
MbRqErr_CommOpen	19	Communication port not yet opened
MbRqErr_CommTx	20	Failed to send message
MbRqErr_SlaveMax	21	Maximum number of slave has been reached
MbRqErr_SlaveNotExists	22	Could not get requested slave handle
MbRqErr_SlaveExists	23	Slave already existing in the same network
MbRqErr_SlaveLoss	24	Slave loss
MbRqErr_MSGDB	33	Invalid datablock physical address
MbRqErr_PrmInvalid	40	Invalid parametrization
MbRqErr_PrmException	41	Invalid parametrization
MbRqErr_PrmTmo	42	Parametrization timeout
MbRqErr_MemoryCfg	43	Not enough memory available
MbRqErr_PollingTime	44	Time since last request is not respecting the minimum polling time
MbRqErr_Stopped	45	Request made before modbus is fully running
MbRqErr_WriteDisabled	46	Write commands are disabled for the slave

MbRSMMode		
Name	Value	Description
RSMMode_Auto	0	Automatic configuration
RSMMode_485	1	RS485 mode
RSMMode_422	2	RS422 mode
RSMMode_232	3	RS232 mode

MbSlaveAddressConfigMode		
Name	Value	Description
MbSlaveAddressConfig-Mode_Static	0	Address specified by static address
MbSlaveAddressConfig-Mode_ByVariableAddress	1	Address specified by variable address

MbSlaveAddressConfigMode		
Name	Value	Description
MbSlaveAddressConfig-Mode_By16BitKeyValue	2	Address specified by 16 bit key value
MbSlaveAddressConfig-Mode_By32BitKeyValue	4	Address specified by 32 bit key value

MbStatus		
Name	Value	Description
MbStatus_NotConfigured	0	Not configured
MbStatus_Configured	1	Configured, waiting for go running
MbStatus_Configuring	2	Configuration in progress
MbStatus_Stopped	3	Stopped
MbStatus_Starting	4	Start, waiting for running
MbStatus_Running	5	Running

3.6.2 ModbusRTU_ReadCoils2

Read coils from a Modbus slave using FC 01 Modbus function.Coil values will be written to the provided pDataOut buffer.The slave does not need to configured in Modbus master network.This function block can be used in Background task only.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NET_WORK_ID	Modbus serial port number
VAR_INPUT	modbusAddr	UINT	Modbus node address
VAR_INPUT	coilAddrStart	UINT	Coil start address
VAR_INPUT	coilsNum	UINT	Number of coils to read
VAR_INPUT	pollingTime	UINT	Minimum polling time in ms. 0=read continuously; if>0, command will be executed only if this minimum time has passed after the previous call

Pin Type	Name	Type	Description
VAR_INPUT	timeoutMs	UINT	Timeout in ms
VAR_INPUT	waitBeforeSendMs	UINT	Wait before send in ms
VAR_INPUT	pDataOut	PVOID	Address of the output buffer (@BOOL or @WORD, see packedBits)
VAR_INPUT	sizeIn	UINT	Size in bytes of the output buffer pDataOut
VAR_INPUT	packedBits	BOOL	If FALSE, pDataOut should be a @BOOL and will receive each splitted bit; if TRUE, pDataOut should be a @WORD and will receive raw unsplitted values (NOT IMPLEMENTED YET)
VAR_INPUT	_reserved	ARRAY[0..2] OF BYTE	reserved
VAR_OUTPUT	sizeOut	UINT	Size in bytes of the data written to the pDataOut buffer
VAR_OUTPUT	result	MB_RESULT_VALUE	Result of the command

3.6.3 ModbusRTU_ReadDeviceIdent

Read device identification of a Modbus slave using FC 43/14 Modbus function. Only individual object access is implemented. The slave does not need to be configured in Modbus master network. This function block can be used in Background task only.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus serial port number
VAR_INPUT	modbusAddr	UINT	Modbus node address
VAR_INPUT	readDeviceIDCode	USINT	Read access type (NOT IMPLEMENTED, only 4=individual access)

Pin Type	Name	Type	Description
VAR_INPUT	objectId	USINT	Object to read
VAR_INPUT	pDataOut	PVOID	Address of the output buffer
VAR_INPUT	sizeIn	UINT	Size in bytes of the output buffer pDataOut
VAR_INPUT	timeoutMs	UINT	Timeout in ms
VAR_INPUT	waitBeforeSendMs	UINT	Wait before send in ms
VAR_OUTPUT	sizeOut	UINT	Size in bytes of the data written to the pDataOut buffer
VAR_OUTPUT	result	MB_RESULT_VALUE	Result of the command

3.6.4 ModbusRTU_ReadDiscreteInputs2

Read discrete inputs from a Modbus slave using FC 02 Modbus function. Input values will be written to the provided pDataOut buffer. The slave does not need to be configured in Modbus master network. This function block can be used in Background task only.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus serial port number
VAR_INPUT	modbusAddr	UINT	Modbus node address
VAR_INPUT	discreteInputStart	UINT	Discrete input start address
VAR_INPUT	discreteInputsNum	UINT	Number of discrete inputs to read
VAR_INPUT	pollingTime	UINT	Minimum polling time in ms. 0=read continuously; if >0, command will be executed only if this minimum time has passed after the previous call
VAR_INPUT	timeoutMs	UINT	Timeout in ms

Pin Type	Name	Type	Description
VAR_INPUT	waitBeforeSendMs	UINT	Wait before send in ms
VAR_INPUT	pDataOut	PVOID	Address of the output buffer (@BOOL or @WORD, see packed-Bits)
VAR_INPUT	sizeIn	UINT	Size in bytes of the output buffer pDataOut
VAR_INPUT	packed-Bits	BOOL	If FALSE, pDataOut should be a @BOOL and will receive each splitted bit; if TRUE, pDataOut should be a @WORD and will receive raw unsplitted values (NOT IMPLEMENTED YET)
VAR_INPUT	_reserved	ARRAY[0..2] OF BYTE	reserved
VAR_OUTPUT	sizeOut	UINT	Size in bytes of the data written to the pDataOut buffer
VAR_OUTPUT	result	MB_RESULT_VALUE	Result of the command

3.6.5 ModbusRTU_ReadHoldingRegs2

Read holding registers from a Modbus slave using FC 03 Modbus function. Register values will be written to the provided pDataOut buffer. The slave does not need to be configured in Modbus master network. This function block can be used in Background task only.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus serial port number
VAR_INPUT	modbusAddr	UINT	Modbus node address
VAR_INPUT	regAddrStart	UINT	Register start address

Pin Type	Name	Type	Description
VAR_INPUT	regNum	UINT	Number of registers
VAR_INPUT	polling-Time	UINT	Minimum polling time in ms. 0=read continuously; if >0, command will be executed only if this minimum time has passed after the previous call
VAR_INPUT	timeoutMs	UINT	Timeout in ms
VAR_INPUT	waitBeforeSendMs	UINT	Wait before send in ms
VAR_INPUT	pDataOut	PVOID	Address of the output buffer
VAR_INPUT	sizeIn	UINT	Size in bytes of the output buffer pDataOut
VAR_OUTPUT	sizeOut	UINT	Size in bytes of the data written to the pDataOut buffer
VAR_OUTPUT	result	MB_RESULT_VALUE	Result of the command

3.6.6 ModbusRTU_ReadInputRegs2

Read input registers from a Modbus slave using FC 04 Modbus function. Register values will be written to the provided pDataOut buffer. The slave does not need to be configured in Modbus master network. This function block can be used in Background task only.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus serial port number
VAR_INPUT	modbusAddr	UINT	Modbus node address
VAR_INPUT	regAddrStart	UINT	Register start address
VAR_INPUT	regNum	UINT	Number of registers

Pin Type	Name	Type	Description
VAR_INPUT	polling-Time	UINT	Minimum polling time in ms. 0=read continuously; if >0, command will be executed only if this minimum time has passed after the previous call
VAR_INPUT	timeoutMs	UINT	Timeout in ms
VAR_INPUT	waitBeforeSendMs	UINT	Wait before send in ms
VAR_INPUT	pDataOut	PVOID	Address of the output buffer
VAR_INPUT	sizeIn	UINT	Size in bytes of the output buffer pDataOut
VAR_OUTPUT	sizeOut	UINT	Size in bytes of the data written to the pDataOut buffer
VAR_OUTPUT	result	MB_RESULT_VALUE	Result of the command

3.6.7 ModbusRTU_WriteMultCoils2

Write multiple coils to a Modbus slave using FC 15 Modbus function. Values will be taken from the provided pDataIn buffer. The slave does not need to be configured in Modbus master network. This function block can be used in Background task only.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus serial port number
VAR_INPUT	modbusAddr	UINT	Modbus node address
VAR_INPUT	coilStart	UINT	First coil address
VAR_INPUT	coilsNum	UINT	Number of coils
VAR_INPUT	sizeIn	UINT	Size in bytes of the output buffer pDataOut
VAR_INPUT	pDataIn	PVOID	Address of the input buffer

Pin Type	Name	Type	Description
VAR_INPUT	polling-Time	UINT	Minimum polling time in ms. 0=write continuously; if >0, command will be executed only if this minimum time has passed after the previous call
VAR_INPUT	timeoutMs	UINT	Timeout in ms
VAR_INPUT	waitBeforeSendMs	UINT	Wait before send in ms
VAR_INPUT	packed-Bits	BOOL	If FALSE, pDataIn should be a @BOOL and contains splitted bits to write; if TRUE, pDataIn should be a @WORD and contains unsplit raw values to write (NOT IMPLEMENTED YET)
VAR_INPUT	_reserved	BYTE	reserved
VAR_OUTPUT	result	MB_RESULT_VALUE	Result of the command

3.6.8 ModbusRTU_WriteMultRegs2

Write multiple registers to a Modbus slave using FC 16 Modbus function.

Values will be taken from the provided pDataIn buffer.

The slave does not need to be configured in Modbus master network.

This function block can be used in Background task only.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus serial port number
VAR_INPUT	modbusAddr	UINT	Modbus node address

Pin Type	Name	Type	Description
VAR_INPUT	regAddrStart	UINT	First register address
VAR_INPUT	regNum	UINT	Number of registers
VAR_INPUT	sizeIn	UINT	Size in bytes of the output buffer pDataOut
VAR_INPUT	pDataIn	PVOID	Address of the input buffer
VAR_INPUT	polling-Time	UINT	Minimum polling time in ms. 0=write continuously; if >0, command will be executed only if this minimum time has passed after the previous call
VAR_INPUT	timeoutMs	UINT	Timeout in ms
VAR_INPUT	waitBeforeSendMs	UINT	Wait before send in ms
VAR_INPUT	_reserved	WORD	Reserved field
VAR_OUTPUT	result	MB_RESULT_VALUE	Result of the command

3.6.9 ModbusRTU_WriteSingleCoil2

Write single coil to a Modbus slave using FC 05 Modbus function.

The slave does not need to be configured in Modbus master network.

This function block can be used in Background task only.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus serial port number
VAR_INPUT	modbusAddr	UINT	Modbus node address
VAR_INPUT	coilAddr	UINT	Coil address
VAR_INPUT	coilValue	BOOL	Coil value
VAR_INPUT	_reserved	BYTE	

Pin Type	Name	Type	Description
VAR_INPUT	polling-Time	UINT	Minimum polling time in ms. 0=write continuously; if >0, command will be executed only if this minimum time has passed after the previous call
VAR_INPUT	timeoutMs	UINT	Timeout in ms
VAR_INPUT	waitBeforeSendMs	UINT	Wait before send in ms
VAR_OUTPUT	result	MB_RESULT_VALUE	Result of the command

3.6.10 ModbusRTU_WriteSingleReg2

Write single register to a Modbus slave using FC 06 Modbus function.

The slave does not need to be configured in Modbus master network.

This function block can be used in Background task only.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus serial port number
VAR_INPUT	modbusAddr	UINT	Modbus node address
VAR_INPUT	regAddr	UINT	Register address
VAR_INPUT	regValue	UINT	Register value
VAR_INPUT	polling-Time	UINT	Minimum polling time in ms. 0=write continuously; if >0, command will be executed only if this minimum time has passed after the previous call
VAR_INPUT	timeoutMs	UINT	Timeout in ms
VAR_INPUT	waitBeforeSendMs	UINT	Wait before send in ms
VAR_OUTPUT	result	MB_RESULT_VALUE	Result of the command

3.6.11 sysMbMRtu_ChangePortSettings

Change serial configuration at runtime (close and reopen channel with new settings)
Returns TRUE if change happens, FALSE else.

Pin Type	Name	Type	Description
VAR_INPUT	netId	MB_RTU_NETWORK_ID	Modbus serial port number
VAR_INPUT	new_baudrate	MbBaudrate	New value for baudrate
VAR_INPUT	new_frameMode	MbMode	New value for frame mode
VAR_INPUT	new_RS-Mode	M b R S - Mode	New value for serial mode

3.6.12 sysMbMRtu_DisableWriteForSlave

Enable or disable write commands for a slave.
Current value can be found in the corresponding sysMbMRtuNetList[] entry.

Pin Type	Name	Type	Description
VAR_INPUT	netId	USINT	Modbus serial port number, referred to enumerative MB_RTU_NETWORK_ID
VAR_INPUT	handle	USINT	Modbus slave handle, returned from sysMbMRtu_GetSlaveHandle
VAR_INPUT	disable-Write	BOOL	TRUE if write commands must be disabled

3.6.13 sysMbMRtu_GetMasterStatus

Modbus master RTU get status of master

Pin Type	Name	Type	Description
VAR_INPUT	netId	USINT	Modbus serial port number, referred to enumerative MB_RTU_NETWORK_ID

Pin Type	Name	Type	Description
VAR_OUTPUT	cfgOk	BOOL	Modbus Master configured
VAR_OUTPUT	status	MbStatus	Modbus Master state machine, referred to enumerative MbStatus
VAR_OUTPUT	lastError	MbError	Modbus Master last error, referred to enumerative MbError
VAR_OUTPUT	nSlaves	USINT	Total number of Configured Slave in Net list
VAR_OUTPUT	nQueuedMsg	UINT	Total number of Configured Broadcast and Single Slave Message
VAR_OUTPUT	nQueuedBroadcastMsg	UINT	Total number of Configured Broadcast Message
VAR_OUTPUT	nBusMsgCount	UINT	Bus Message Count
VAR_OUTPUT	nBus-CommErrCount	UINT	Bus Communication Error Count

3.6.14 sysMbMRtu_GetSlaveHandle

Modbus master RTU get offset in Netlist of corresponding slave

Pin Type	Name	Type	Description
VAR_INPUT	netId	USINT	Modbus serial port number, referred to enumerative MB_RTU_NETWORK_ID
VAR_INPUT	adr	USINT	Modbus slave address

3.6.15 sysMbMRtu_GetSlaveStatus

Modbus master RTU get status of corresponding slave

Pin Type	Name	Type	Description
VAR_INPUT	netId	USINT	Modbus serial port number, referred to enumerative MB_RTU_NETWORK_ID
VAR_INPUT	hdl	USINT	Modbus slave handle, returned from sysMbMRtu_GetSlaveHandle
VAR_OUTPUT	adr	USINT	Modbus slave address
VAR_OUTPUT	cfg	BOOL	Modbus slave is added to netlist
VAR_OUTPUT	pres	BOOL	Modbus slave is present on network
VAR_OUTPUT	active	BOOL	Modbus slave is concurrent in communication
VAR_OUTPUT	n_prm	UINT	Total number of Parametrization Slave Message
VAR_OUTPUT	n_qmsg	UINT	Total number of Configured Slave Message
VAR_OUTPUT	min_poll_time	UINT	Minimum polling time (ms)
VAR_OUTPUT	status	Mb Status	Modbus slave state machine, referred to enumerative MbStatus
VAR_OUTPUT	error	MbError	Modbus slave last error, referred to enumerative MbError
VAR_OUTPUT	exception	MbException	Modbus slave last exception, referred to enumerative MbException
VAR_OUTPUT	n_SlaveExceptionCount	UINT	Slave Exception Error Count
VAR_OUTPUT	n_SlaveMessageCount	UINT	Slave Message Count
VAR_OUTPUT	n_SlaveNoResponseCount	UINT	Slave No Response Count

Pin Type	Name	Type	Description
VAR_OUTPUT	n_SlaveNAKCount	UINT	Slave NAK Count
VAR_OUTPUT	n_SlaveBusyCount	UINT	Slave Busy Count
VAR_OUTPUT	n_SlaveChrOverCount	UINT	Slave Character Overrun Count

3.6.16 sysMbMRtu_SetMaxRetriesForSlave

Set maximum number of retries for a slave: after this value a slave will declared as not active and lost. Current value can be found in the corresponding sysMbMRtuNetList[] entry.

Pin Type	Name	Type	Description
VAR_INPUT	netId	USINT	Modbus serial port number, referred to enumerative MB_RTU_NETWORK_ID
VAR_INPUT	handle	USINT	Modbus slave handle, returned from sysMbMRtu_GetSlaveHandle
VAR_INPUT	maxRetries	UINT	Max number of retries; use 16#FFFF for infinite retries

3.6.17 sysMbMRtu_SlaveActivation

Modbus master RTU Activate/Deactivate Slave communication

Pin Type	Name	Type	Description
VAR_INPUT	netId	USINT	Modbus serial port number, referred to enumerative MB_RTU_NETWORK_ID
VAR_INPUT	handle	USINT	Modbus slave handle, returned from sysMbMRtu_GetSlaveHandle
VAR_INPUT	activate	BOOL	If TRUE, Slave is concurrent in communication

3.6.18 sysMbMRtu_SlaveChangeAddress

Modbus master RTU change slave address. Address can be changed only in init task

Pin Type	Name	Type	Description
VAR_INPUT	netId	USINT	Modbus serial port number, referred to enumerative MB_RTU_NETWORK_ID
VAR_INPUT	handle	USINT	Modbus slave handle, returned from sysMbMRtu_GetSlaveHandle
VAR_INPUT	new_address	USINT	New slave address

3.6.19 ModbusVars

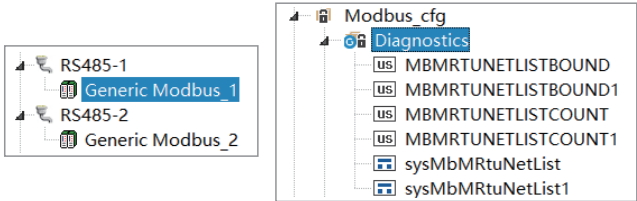
Variable Description:

This variable is a global variable in the ModbusRTU library. Its function is to simply view the master station status and perform basic diagnostics on the master station status. For more master control and status queries, it is necessary to find them in the sysMbMRtuNetList[] element.

Name	Type	Description
sysMbMRtuStatus	MbMRtuStatus	Modbus master RTU status (network 0)
sysMbRtuDiagnostics	MbRtuDiagnostics	Modbus RTU standard diagnostics

3.6.20 sysMbMRtuNetList

Description: This variable is a system global variable, automatically generated by the configuration page in the resource bar. The path is 'Project Bar Window -> Modbus_cfg -> Diagnostics'. When slave nodes are added in the resource bar and saved and compiled, this variable will appear in the project section. Since the X3000 series supports dual masters, if the RS485-2 module is also configured as a master in the resource bar and slave nodes are added, an additional global variable named sysMbMRtuNetList1 will appear in the project section. As shown in the figure below:



Name	Type	Description
sysMbMRtuNetList	MbMRtuNetList	Modbus master RTU status list, corresponding to RS485-1 in the resource view. This variable can be accessed directly in the program.
sysMbMRtuNetList1	MbMRtuNetList	Modbus master RTU status list, corresponding to RS485-2 in the resource view.
M B M R - TUNETLIST-BOUND	USINT	This variable is a constant, configured in the resource view and generated after compilation. It indicates the accessible index range (upper bound) of the sysMbMRtuNetList array. Corresponds to RS485-1.
M B M R - TUNETLIST-COUNT	USINT	This variable is a constant, indicating the number of slave stations under the RS485-1 master network node.
M B M R - TUNETLIST-BOUND1	USINT	This variable is a constant, configured in the resource view and generated after compilation. It indicates the accessible index range (upper bound) of the sysMbMRtuNetList1 array. Corresponds to RS485-2.
M B M R - TUNETLIST-COUNT1	USINT	This variable is a constant, indicating the number of slave stations under the RS485-2 master network node.

3.7 ModbusTCP

3.7.1 Enumerations

sysMbMTcpEsi		
Name	Value	Description
MB_TCP_E_OK	0	Invalid network id specified
MB_TCP_E_NETWORK	-339	Checksum
MB_TCP_E_CHECKSUM	-338	Negative acknowledge
MB_TCP_E_NEGATIVEACKNOWLEDGE	-337	Function cannot be called in a timed task
MB_TCP_E_NOTALLOWEDINTIMEDTASK	-336	Node is switched off
MB_TCP_E_SWITCHEDOFF	-335	Illegal data value in protocol
MB_TCP_E_ILLEGALDATAAPROTOCOL	-332	Time out richiesta
MB_TCP_E_TIMEOUT	-331	Gateway target device failed to respond
MB_TCP_E_GATEWAYDEVICEFAILED	-330	Gateway path unavailable
MB_TCP_E_GATEWAYPATHUNAV	-329	Memory parity error
MB_TCP_E_MEMORYPARITYERROR	-328	Slave device busy
MB_TCP_E_SLAVEDEVICEBUSY	-327	Acknowledge
MB_TCP_E_ACKNOWLEDGE	-326	Slave device failure
MB_TCP_E_SLAVEDEVICEFAILURE	-325	Illegal data value
MB_TCP_E_ILLEGALDATAVALUE	-324	Illegal data address
MB_TCP_E_ILLEGALDATAADDR	-323	Illegal function
MB_TCP_E_ILLEGALFUNCTION	-322	Ide doesn't match
MB_TCP_E_IDE	-321	Protocol error
MB_TCP_E_PROTOCOL	-320	Data type parameter not valid
MB_TCP_E_DATATYPE	-311	Not assigned node
MB_TCP_E_NODENOTEXIST	-303	Not valid node
MB_TCP_E_NODE	-301	Successful command

sysMbMTcpSlaveAddressConfigMode		
Name	Value	Description
MB_TCP_CONFIG_MODE_STATIC	0	Address specified by static address
MB_TCP_CONFIG_MODE_BY_VAR_ADDRESS	1	Address specified by variable address
MB_TCP_CONFIG_MODE_BY_16BIT_KEY	2	Address specified by 16 bit key value
MB_TCP_CONFIG_MODE_BY_32BIT_KEY	4	Address specified by 32 bit key value

sysMbMTcpTypeD		
Name	Value	Description
MB_TCP_DATA_T_SINT	0	SINT
MB_TCP_DATA_T_USINT	1	USINT
MB_TCP_DATA_T_BYTE	2	BYTE
MB_TCP_DATA_T_INT	3	INT
MB_TCP_DATA_T_UINT	4	UINT
MB_TCP_DATA_T_WORD	5	WORD
MB_TCP_DATA_T_DINT	6	DINT
MB_TCP_DATA_T_UDINT	7	UDINT
MB_TCP_DATA_T_DWORD	8	DWORD
MB_TCP_DATA_T_REAL	9	REAL
MB_TCP_DATA_T_BOOL	10	BOOL
MB_TCP_DATA_T_NULL	255	NULL

3.7.2 Structures

MbTcpSlaveState		
Name	Value	Description
cfg	BOOL	configured
pres	BOOL	present
active	BOOL	active node
miss	BOOL	delay cycle
IP	STRING[19]	IP address
address	UINT	Subnet modbus address
missCnt	UINT	delay cycles counter
state	INT	last modbus error on slave
netStat	UINT	slave status

MbTcpState		
Name	Value	Description
netConfigOk	BOOL	Net configuration OK
clientOK	BOOL	driver client initialized
clientRun	BOOL	driver client is running
taskRun	BOOL	communicaton task is running
sysMemoryError	BOOL	memory configuration error. this field is used for debug

3.7.3 sysMbMTcp_ReadReg_RO

Read RO modbus register using ReadInputRegister modbus command

Pin Type	Name	Type	Description
VAR_INPUT	node	UINT	The index of the node in the network
VAR_INPUT	reg_addr	UINT	The address of the register to read
VAR_INPUT	data_type	sysMbMTcp- TypeD	The type of the data to read
VAR_INPUT	tmo	UDINT	Timeout in ms
VAR_OUTPUT	esi	sysMbMTcpEsi	Operation result
VAR_OUTPUT	res_dword	DWORD	Result is always put in a dword

3.7.4 sysMbMTcp_ReadReg_RW

Read RW modbus register using ReadHoldingRegister modbus command

Pin Type	Name	Type	Description
VAR_INPUT	node	UINT	The index of the node in the network
VAR_INPUT	reg_addr	UINT	The address of the register to read
VAR_INPUT	data_type	sysMbMTcp- TypeD	The type of the data to read
VAR_INPUT	tmo	UDINT	Timeout in ms
VAR_OUTPUT	esi	sysMbMTcpEsi	Operation result
VAR_OUTPUT	res_dword	DWORD	Result is always put in a dword

3.7.5 sysMbMTcp_WriteReg_M

Write modbus register using WriteMultipleRegisters modbus command

Pin Type	Name	Type	Description
VAR_INPUT	node	UINT	The index of the node in the network
VAR_INPUT	reg_addr	UINT	The address of the register to read
VAR_INPUT	data_type	sysMbMTcp- TypeD	The type of the data to read
VAR_INPUT	data_raw	DWORD	Data to be written
VAR_INPUT	tmo	UDINT	Timeout in ms
VAR_OUTPUT	esi	sysMbMTcp- pEsi	Operation result

3.7.6 sysMbMTcp_WriteReg_S

Write modbus register using WriteSingleRegister modbus command

Pin Type	Name	Type	Description
VAR_INPUT	node	UINT	The index of the node in the network
VAR_INPUT	reg_addr	UINT	The address of the register to read
VAR_INPUT	data_type	sysMbMTcp- TypeD	The type of the data to read
VAR_INPUT	data_raw	DWORD	Data to be written
VAR_INPUT	tmo	UDINT	Timeout in ms
VAR_OUTPUT	esi	sysMbMTcp- pEsi	Operation result

3.7.7 Diagnostics

Explanation: This variable is a global variable in the ModbusTCP library, and the variables in the following table can be used directly in the program to obtain the status of the master and slave stations.

Name	Type	Description
sysMbMTcpMState	MbTcp- State	System variables, can directly query the relevant status of the ModbusTCP master station

Name	Type	Description
sysMb - TcpSlaveState	Mb - TcpSlaveState	System variables, can directly query the relevant status of the ModbusTCP slave station

4 Other Notes and Precautions

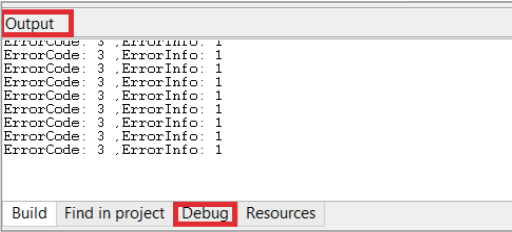
4.1 Fault description

When the ERR indicator of the CPU module turns red, an error occurs due to two reasons: the extension module configuration error (multi-configuration module, module sequence error) or the extension module itself error.

The debug bar of the output window allows you to see the type of error and the slave number:

- 1) Error Code: 3 indicates an extension configuration error,
- 2) Error Code: 1 indicates an error in the extension module itself;
- 3) ErrorInfo indicates the configuration error of the first extension module.

The following diagram illustrates the configuration error of the first extension module:



Note: The X3000-CI-01-1 expansion module does not participate in the configuration, and the wrong extension module number does not contain this module, that is, when this module is installed in the first position on the right side of the CPU module, the ErrorInfo:1 actual configuration error module is the first module on the right side of X3000-CI-01-1.

After the error is eliminated, re-download the program and power it back on, and the controller resets incorrectly.

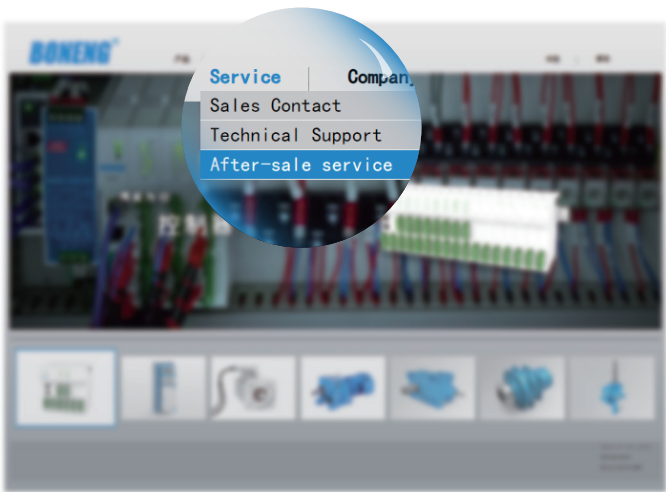
4.2 Precautions

- 1) When editing a program in the Ladder (LD) editor, when dragging a variable or function block into the program, you need to pay attention to the mouse change to "+" in order to place it in the program;
- 2) If you create a new variable in ST language, you need to click the save button when exiting, otherwise the changed data will be lost;
- 3) If the controller uses a Modbus TCP slave to create a 32-bit parameter in the parameters, direct compilation will cause an address conflict, and the 32-bit parameter and the following parameters need to be recalculated. Press the keyboard shift or Ctrl to select the parameter to be recalculated, and click Recalculate.
- 4) When using the function block in the Boneng-ControlLib library, some pins must have initial values, otherwise the function block or driver will report a fault; The parameters in the drive also need to be set, and the relevant parameters need to be correct to ensure normal operation;
- 5) If a variable has been called in the program, deleting the variable in the variable table will not be deleted in the program, resulting in a compilation error.
- 6) When deleting objects in the program, you need to pay attention to whether there are variables in the variable table that are selected (the yellow part in the variable table) will be deleted, otherwise it will be deleted by mistake;
- 7) Before you start the actual compilation, make sure that you have assigned the program to at least one task;
- 8) The watch window can monitor the current value of the added variable, or you can change the value of the variable to check that the application is behaving as expected.
- 9) If the initial value of a variable needs to be changed when using the variable table, the device needs to be cold-started after the program is compiled to change the initial value of the variable.

After-sale service

After-sales Service Please follow the steps below to submit the after-sales process.

- Login “www.boneng.com”
- Click “Service” and “After-sale Service”



Login system

[Email sign](#)

[New user registration](#)

Please enter email address

Please enter your password

Please enter the verification code



Sign in

Other District

Controller/Drive : 0512-66182005
Motor/Gear motor/Gearbox : 0512-66189918

BONENG TRANSMISSION(SHEN YANG)CO.,LTD

Controller/Drive : 024-31271571
Motor/Gear motor/Gearbox : 024-31292571

BONENG TRANSMISSION(TIAN JIN)CO.,LTD

Controller/Drive : 022-86928559
Motor/Gear motor/Gearbox : 022-26929558

BONENG TRANSMISSION(KAIFENG)CO.,LTD

Controller/Drive : 0371-23335230
Motor/Gear motor/Gearbox : 0371-23277771

BONENG TRANSMISSION(WEIFANG)CO.,LTD

Controller/Drive : 0536-4699687
Motor/Gear motor/Gearbox : 0536-4699667

BONENG TRANSMISSION(CHANGSHA)CO.,LTD

Controller/Drive : 0731-88386958
Motor/Gear motor/Gearbox : 0731-88380725

BONENG TRANSMISSION EQUIPMENT (CHENGDU)CO.,LTD

Controller/Drive : 028-87740066
Motor/Gear motor/Gearbox : 028-87740066

BONENG TRANSMISSION(ZHAOQING)CO.,LTD

Controller/Drive : 0757-86719757
Motor/Gear motor/Gearbox : 0758-2699830

BONENG TRANSMISSION(SUZHOU)CO.,LTD

Controller/Drive Southern Jiangsu : 0512-66182005
Motor/Gear motor/Gearbox Southern Jiangsu : 0512-66189918
Controller/Drive Zhejiang-Shanghai : 0512-66182005
Motor/Gear motor/Gearbox Zhejiang-Shanghai : 0512-66189918
Controller/Drive Jiangsu-Anhui District : 0512-66182005
Motor/Gear motor/Gearbox Jiangsu-Anhui : 025-52171612

BONENG TRANSMISSION(USA/Canada)CO.,LTD

Technical Support/Debugging/After-Sales Service :
1250 E 222nd Euclid, OH 44117, United States
Email: America@boneng.com
Office Tel: 1-216-618-3099 / 1-216-618-0138

BONENG TRANSMISSION(India)CO.,LTD

Technical Support/Debugging/After-Sales Service :
Plot No. E-10/3, MIDC sinner (Malegaon) Industrial Area,
Nashik, 422123, Maharashtra, India.
Email: india@boneng.com
Tel: +91-22-2781 3385 / +91-11- 4507 6293

EN

EN