BONENG



X3000系列控制器使用说明书

08/2022

目录

1 控制器技术数据及安装接线 ---------- 01
1.1 CPU模块 ---------- 01
1.2 扩展模块 ---------- 02
1.3 安装尺寸 ---------- 05
2 控制器软件安装及编程调试 ---------- 06
2.1 LogicLab软件及其插件安装 ---------- 06
2.2 新建工程 ---------- 07
2.3 编程界面介绍 ---------- 11
2.4 新建程序 ---------- 15
2.5 梯形图（LD）编辑器 ---------- 16
2.6 结构化文本（ST）编辑器 ---------- 19
2.7 背板总线配置及其使用 ---------- 20
2.8 Modbus功能使用 ---------- 25
2.9 Ethercat功能使用 ---------- 30
2.10 BonengControlLib使用 ---------- 32
2.11 BonengEmbedLogicLib使用 ---------- 36
2.12 编译 ---------- 39
2.13 启动应用程序 ---------- 39
2.14 调试 ---------- 40
3 其他说明及注意事项 ---------- 43
3.1 故障说明 ---------- 43
3.2 注意事项 ---------- 43
4 售后服务 ---------- 44

1 控制器技术数据及安装接线

1.1CPU模块

1.1.1 技术数据

功能		数据
订货号	X3010-EC-00-S	标准控制性
CPU	X3010	ARM-A8 800MHz
存储	用户程序容量	3MB
	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 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	

1.1.2 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



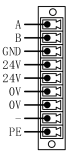
Ethernet接口

类型	带金属屏蔽的 RJ45 母座
功能	连接 EtherCAT 协议设备



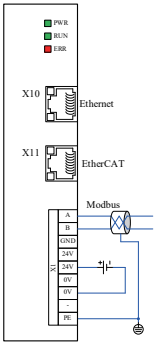
X1端子

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



1.1.3 CPU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运行正常，当RUN不在闪烁，表示CPU卡死；红色的ERR指示灯亮起时表示CPU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。



1.2扩展模块

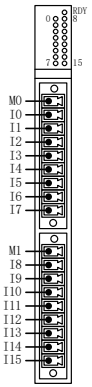
1.2.1 扩展模块通用数据

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

1.2.2 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 个共公共端，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



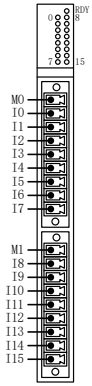
指示灯状态

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

1.2.3 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 个共公共端，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



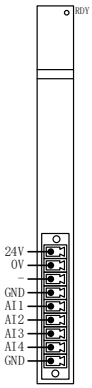
指示灯状态

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

1.2.4 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
	电子切换电压和电流模式
	电压型支持 0V…10V
	电流型支持 0mA…20mA、4mA…20mA
	电流型内阻 250Ω
	电压型：大于 50kΩ
	精度：±1%



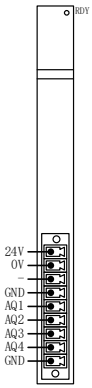
指示灯状态

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

1.2.5 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 5kΩ
	精度：±1%



指示灯状态

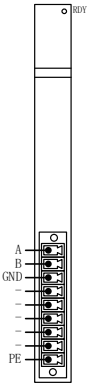
RDY	绿灯指示模块连接状态
	快闪表示等待配置中
	慢闪表示正常运行中

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2.10.1.1小节

1.2.6 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

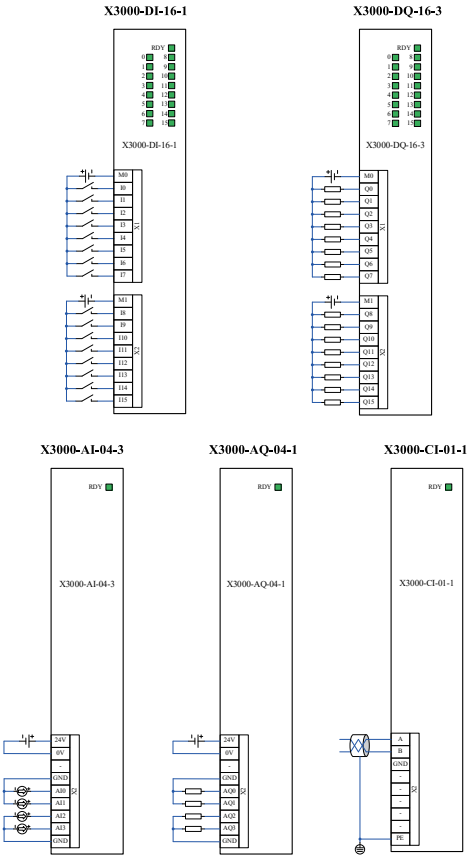


指示灯状态

RDY　绿灯指示模块连接状态

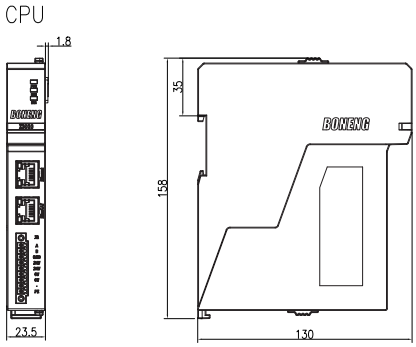
常亮表示正常运行中

1.2.7 接线示意图

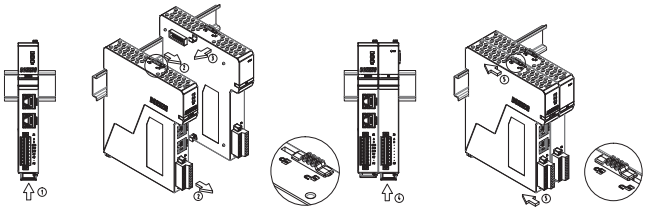


1.3 安装尺寸

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 LogicLab软件及其插件安装

安装包及插件请在博能传动（苏州）有限公司官网www.boneng.com下载最新版本，否则会导致端口IO配置异常。安装时要将插件安装在软件安装目录下，其余按照系统提示安装即可；具体安装步骤如下：

1) 运行安装包



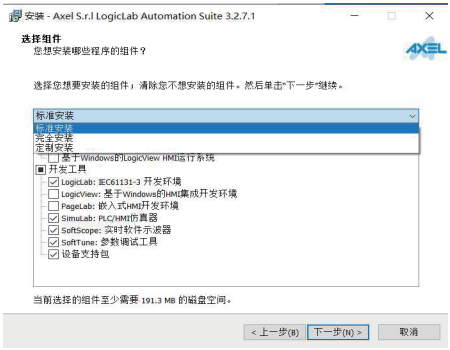
2) 选择语言：简体中文、English两种语言可选；

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；



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；



7) 打开博能X3010的插件，将其安装在LogicLab的安装目录下；



8) 根据系统提示完成插件安装；

9) 打开软件，新建工程，在选择控制器类型的选项下只有X3010表示安装完成。



2.2 新建工程

打开logiclab界面如下图所示，新建工程有两种方式。在PC与控制器硬件正确连接时，使用两种方式建立工程均可，方式一更方便；方式二可在没有控制器时进行新建工程、设置通讯参数、添加扩展模块、编辑程序等操作，在硬件连接控制器后点击连接按钮即可进行下载程序、调试等操作。



2.2.1 方式一

点击扫描网络，弹出以下界面；IP地址初始为：192.168.0.123；



点击新建工程按钮，弹出新建工程窗口；



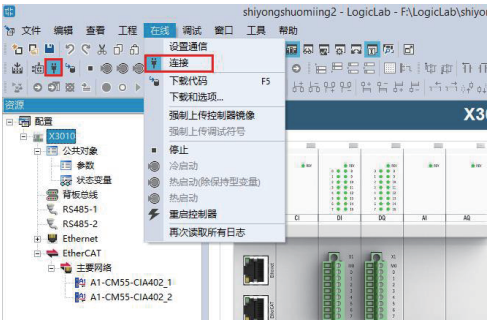
1) 名称：此工程名称，工程名称要为英文，中文会导致程序编译不通过；

2) 目录：此工程保存位置；

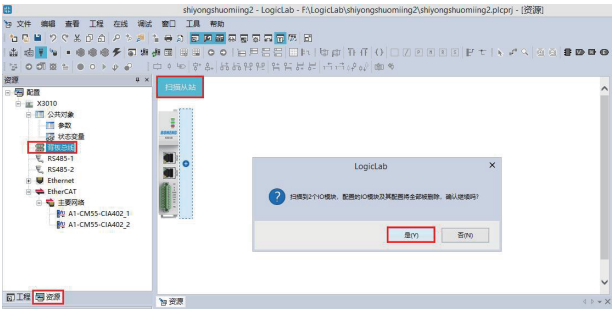
3) 控制器选择：选择使用控制器型号 为X3010 1.0；

选项：区分大小写：需要注意，默认情况下在IEC 61131-3标准中，这个选项是不被激活的，也就是说当您选择创建一个区分大小写的项目，它将是不符合标准的。

点击确定后，点击连接按钮“ ”或“在线--连接”与控制器进行通讯连接，如下图所示：



组态扩展模块：依次点击“资源--背板总线--扫描从站”；当扩展模块中有X3000-CI-01-1模块时，将会弹出一个扫描到RS485的窗口，但不会组态到背板总线中；当扩展模块中没有X3000-CI-01-1模块时，会弹出未扫描到RS485窗口。



扫描完成，点击“是”，工程建立完成，返回“工程”编写程序。

2.2.2 方式二

点击新建工程；



1) 名称：此工程名称，工程名称要为英文，中文会导致程序编译不通过；

2) 目录：此工程保存位置；

3) 控制器选择：选择使用控制器型号；

选项：区分大小写：需要注意，默认情况下在IEC 61131-3标准中，这个选项是不被激活的，也就是说当您选择创建一个区分大小写的项目，它将是不符合标准的。

此时Logiclab建立的工程未与控制器通信连接，需要先对通信进行设置。

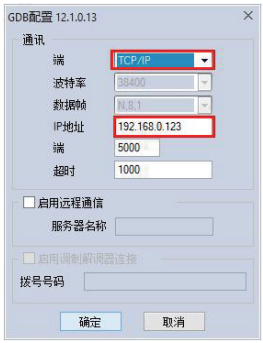
设置通信的步骤：依次点击导航栏“在线--设置通信”



依次点击“GDB--属性”；



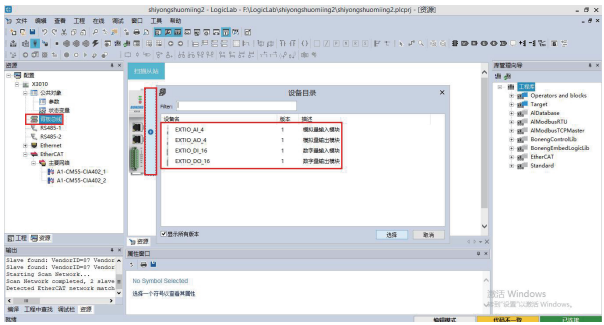
端选择“TCP/IP”，输入IP地址，和端口号，点击确定；



此时界面会返回“设备链接管理器配置，点击“激活”，“GDB”右侧激活栏显示已激活，然后点击“确定”；



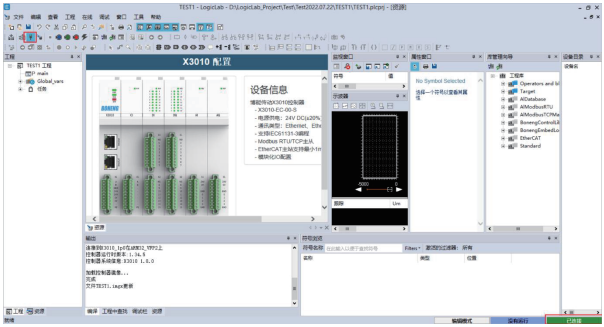
使用方式二建立工程，需要手动添加组态扩展模块；
依次点击“资源--背板总线—添加拓展模块”，选择需要添加的模块；



添加完成后即可返回“工程”编写程序。

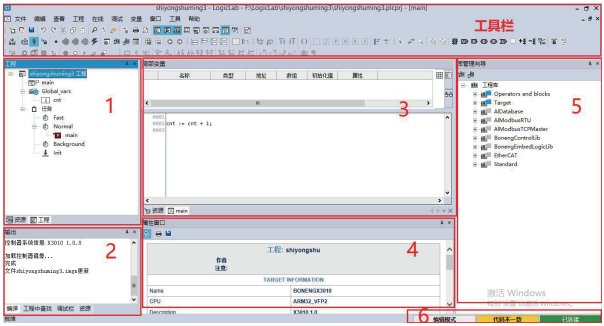
如果控制器与PC硬件连接好，点击连接按钮“”或“在线--连接”已经可以与控制器进行通信连接。

连接成功后现象



2.3编程界面介绍

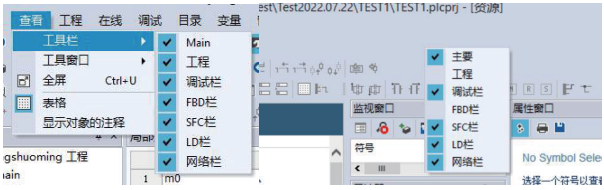
下图为LogicLab工作区视图，包含常用的几个组件。



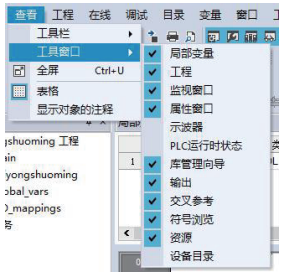
1.工作区窗口 2.输出窗口 3.代码编辑区 4.属性窗口 5.库管理窗口 6.状态栏

2.3.1 工具栏和工具窗口

Logiclab的“工具栏”调用可通过导航栏，点击“查看”，然后点击“工具栏”对要使用的进行勾选，或者直接在“工具栏”位置点击鼠标右键，也可进行勾选。



Logiclab的“工具窗口”可以在导航栏点击“查看”点击工具窗口”调取相应的工具窗口。

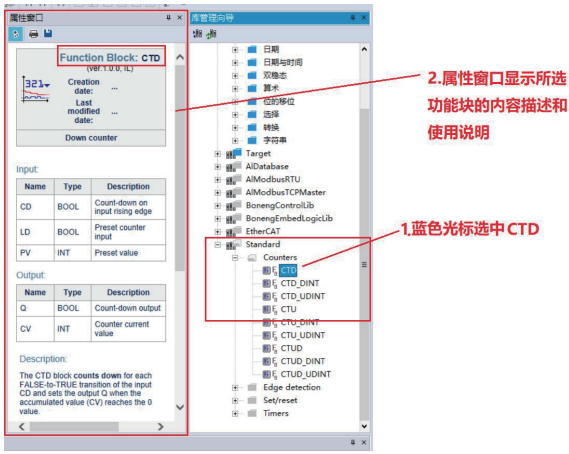


2.3.2 库管理窗口

此面板列出了基本的语言元素，IEC 61131-3标准定义的运算符和函数。PLC指令在“库管理向导窗口”，这些指令按照一定的逻辑被分组到不同的文件夹中，以方便使用和查找。



点击一个指令在“属性窗口”可看到其详细信息；

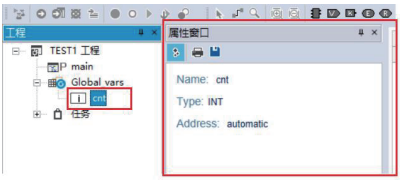


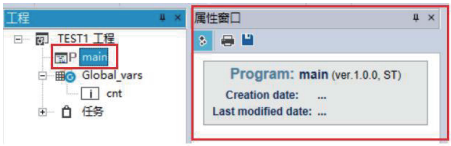
添加/删除库：点击“库管理向导”窗口内“ ”，点击“增加/删除”选中库文件即可添加/删除库。

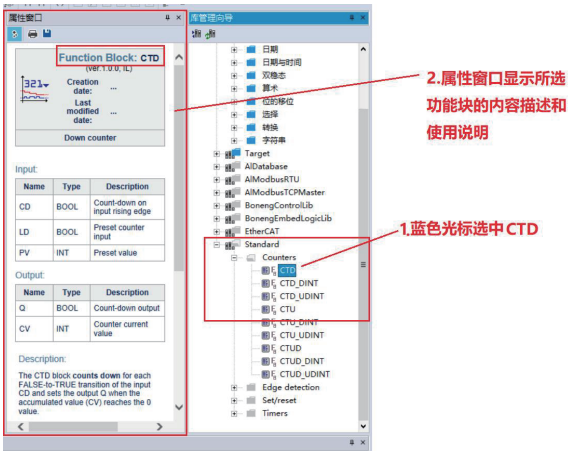


2.3.3 属性窗口

此窗口能够查看这里应该是所选内容（包括工程栏，局部、全局变量，函数，功能块，库管理指令等）的详细信息，右击指令点击对象属性即可。点击程序中的变量在“属性窗口”也可看到起详细信息。







2.3.4 输出窗口

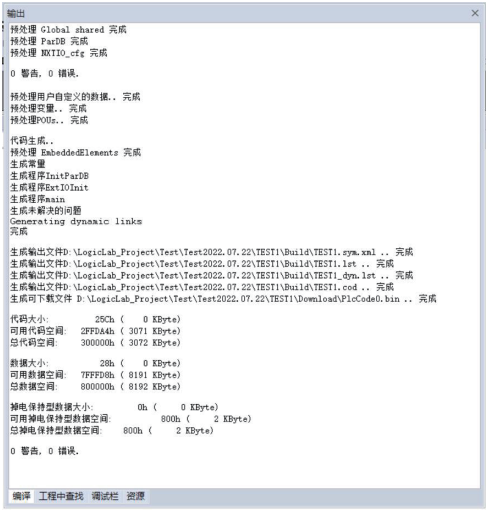
输出窗口是输出控制器信息的窗口，该窗口有四个选项：编译，工程中查找，调试栏，资源。

编译：可以看到编译的信息。编译时详细信息在输出窗口显示，双击在输出栏中对应的错误行，即可定位到具体的编译错误位置上，编译有警告不影响程序的下载；

工程中查找：显示项目中查找活动的结果；

调试栏：面板显示高级调试活动（例如：断点调试）的信息。根据所使用目标设备上的接口，LogicLab可以通过这个输出窗口上打印每一个PLC运行时错误（例如：除零错误），由此定位错误发生的确切位置；

资源：正在连接的目标设备的相关消息。



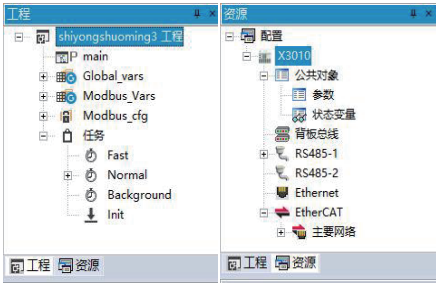
2.3.5 状态栏

状态栏分为三栏，从左到右显示控制器所处的模式应用程序的状态，然后是控制器内代码和代码编辑区代码的匹配状态，最右侧是通信状态。



2.3.6 工作区窗口

工作区窗口包含两个不同的面板，如下图所示；



工程

工程面板包含工程中的所有元素，以树形组织的形式表现，可将自定义文件夹添加到项目树中，以使用户操作。

任务列表它表示可用的任务和相关分配的程序。

资源

资源面板的内容取决于目标设备如何与logiclab进行连接，如连接HMI、Modbus-TCP/RTU设备、Etherecat设备。

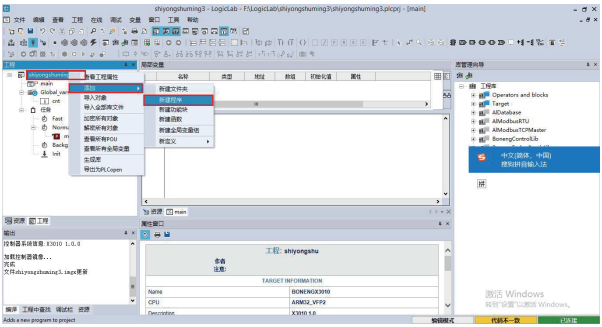
2.4新建程序

LogicLab包含五种源代码编辑器，它支持IEC 61131-3标准的五种编程语言：指令表（IL），结构化文本（ST），梯形图（LD），功能块图（FBD），和顺序功能图（SFC）。LogicLab具有网格状编辑器，支持用户自定义变量。

可以在工程下按照需求建立文件夹，管理建立的程序、功能块、函数等。



如下图所示添加所需程序类型：





新建程序流程：选择程序使用的语言类型、输入程序名称（英文），最后按照需求将程序分配给对应任务。

删除程序：右击工程项目树中程序，点击删除程序即可。

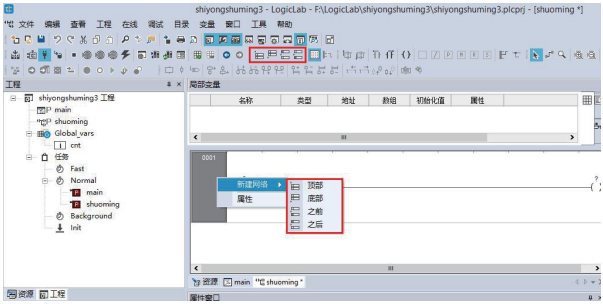
2.5梯形图（LD）编辑器

2.5.1 添加/删除网格

每个网络的上限和下限由两条直线确定，每个网络的左侧由包含网络编号的灰度凸按钮，灰色按钮上有网格编号。



添加新的空白网络请右击网络左侧灰色按钮点击新建网络选择所需新建的位置即可，或者点击工具栏中的相应按钮，如下图所示；



删除网络：点击所要删除网络左侧灰色按钮，按键盘“Delete”。

2.5.2 新建变量

新建变量有两种方式：

方式一：在网络中对未命名的触点、线圈、引脚进行命名并编辑其范围、地址、属性、类型、初始值、类型、描述，以触点为例如下图所示：

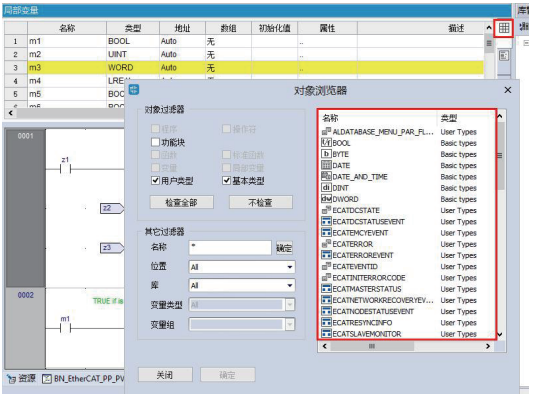


点击确定，弹出以下窗口：

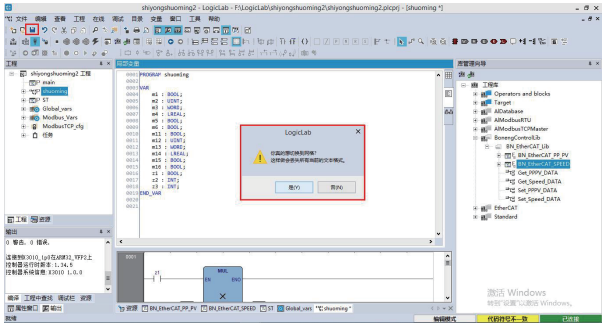


确定变量名称，选取变量类型，变量的地址和属性一般默认即可，变量的范围选取：Local为局部变量，只能在此程序中调用；Global为全局变量可以在项目中的任何模块中调用。建立的变量按照全局变量和局部变量将出现在对应的变量表中。

方式二：直接在局部变量表和全局变量表中右击插入变量，在变量表中对变量的各参数进行编辑，然后点击变量名，将建立的变量拖到需要的位置即可。在拖动变量时鼠标变为“⊞”才能将其放在程序。



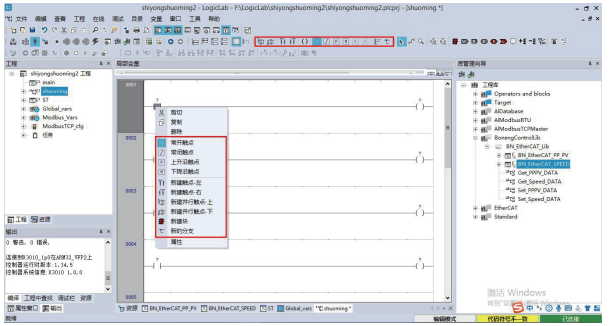
当建立的变量重复性较高时变量表可以转换为ST语言式进行复制、粘贴，在退出前需要点击保存按钮，保存后点击是，否则会丢失更改的数据，如下图所示：



2.5.3 程序编辑

1) 插入触点/线圈

在程序编辑过程右击触点/线圈或点击工具栏按钮可以改变触点线圈/类型和添加触点；



2) 添加功能块

在右侧工程库中将需要的功能块拖至网络中，需要注意在拖动功能块时鼠标变为“⊞”才能将其放在程序。

2.5.4 分配标签

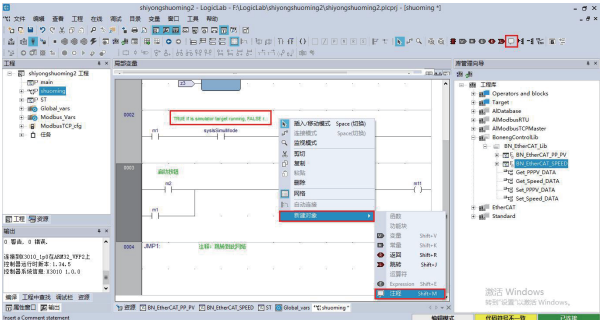
您可以通过跳转语句来改变网络的顺序执行，该语句将程序的网络控制权转移到网络的标号。要将标签分配给网络，请双击左侧带有网络编号的的灰色按钮，会弹出一个对话框出现，输入要与所选网络关联的标签。



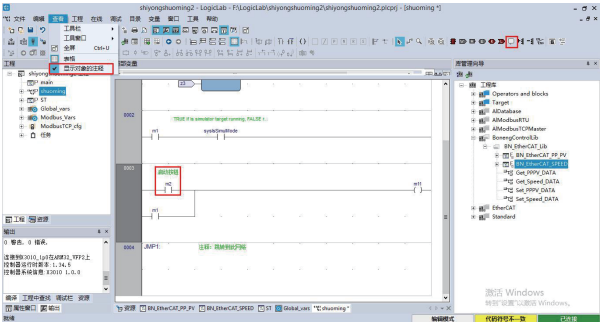
2.5.5 添加注释

注释分为两种：网络注释、对象注释；

网络注释：右击网络点击新建对象选择注释即可添加注释，或点击工具栏中对应按钮，如下图所示：



对象注释：若要使用对象注释需要通过“查看”勾选“显示对象的注释”，在点击工具栏中注释按钮即可输入注释。



2.6结构化文本（ST）编辑器

2.6.1 编辑代码

在ST代码中添加到您现有PLC对象的引用，则有两个选择：

1)直接输入PLC对象的名称；

2)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。

2.6.2 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。

2.6.3 书签

您可以设置书签，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。当你不再需要，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

2.7背板总线配置及其使用

创建一个新的工程后，在资源列表里面可以看到背板总线：如下图所示：



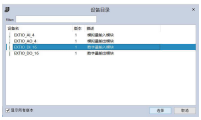
双击打开背板总线，此时可以看到如下操作页面：



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 

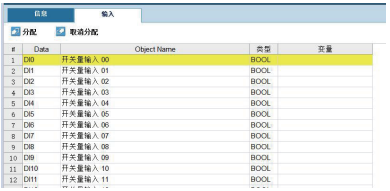
。即可将背板总线上配置，与实际安装的扩展槽号对应。

2.7.1 DI扩展的使用

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



#	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，我们设置DIO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_O_mappings，并在其栏目下有刚才设置的SB1变量。



点击I_O_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，)

	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。

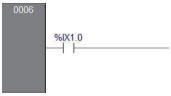
1） 直接访问如所示，只需要在梯形的触点中直接更该名称为地址即可，选择一个触点更改名称为%IX1.0，点击确定。



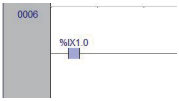
点击确定后会弹出以下对话框，在直接访问时，直接×掉即可。(点击确会报错，这是在建立变量时使用的对话框)



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。



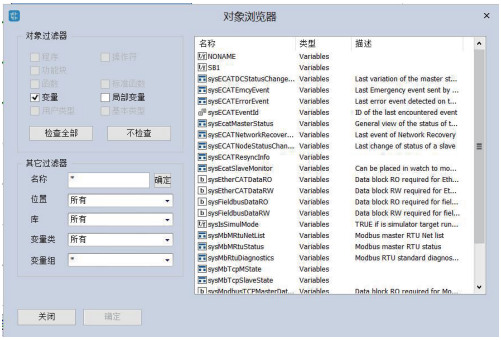
当打开，开关时该触点会导通，如下图所示：



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“…”

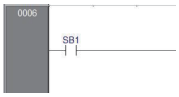


点击后会出现以下对话框。(对象浏览器)



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，即可出出现下图。

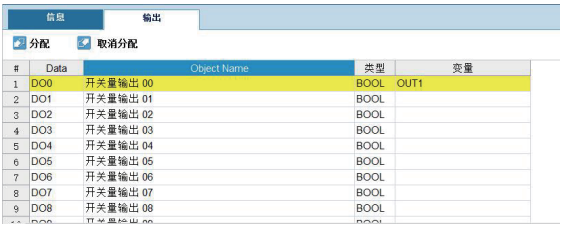


重新编译，下载，调试。



2.7.2DQ 扩展的使用

DQ模块的使用，类似DI，在背板总线中，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。



#	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_O_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	%QX2.0	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

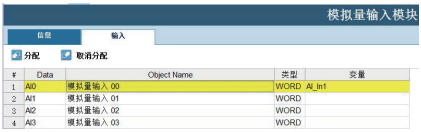


2) 间接调用如下图所示，其调用方法参见DI的调用方法



2.7.3 AI扩展的使用

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，



#	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_O_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	%QX2.0	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。

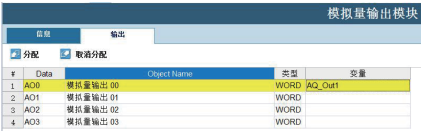
电压和电流模式的切换，使用博能的库函数实现，具体实现方法看2.11.1.1。若不设置是电压检测还是电流检测，则会默认电压检测。

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；

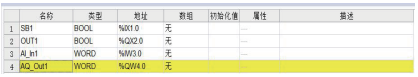
2.7.4 AQ扩展的使用

AQ扩展模块的使用类似AI扩展，创建变量，AQ_Out1，点击保存



#	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_O_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	%IX1.0	无			
2	OUT1	BOOL	%QX2.0	无			
3	AI_In1	WORD	%IW3.0	无			
4	AQ_Out1	WORD	%QW4.0	无			

将变量AQ_Out1的值方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。

2.8 Modbus功能使用

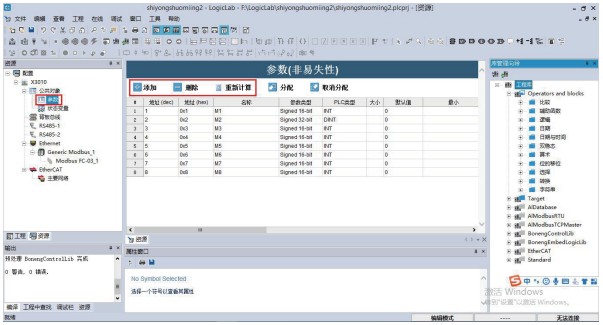
创建PLC工程以后，可以将RS485以及EtherNet配置为Modbus接口，进行Modbus通信。

2.8.1Modbus-TCP

2.8.1.1 Modbus TCP从站

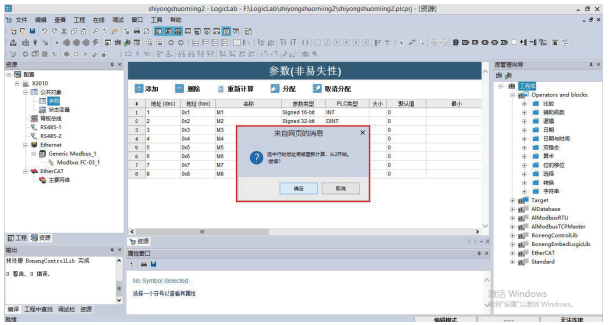
点击Ethernet，可以看到Modbus TCP slave是默认启动的，设备地址是255；

与TCP主站通讯参数建立：点击“参数-添加”，添加通讯参数，输入变量名称，选择参数类型、PLC类型等。



“地址（dec）”栏中为Modbus TCP通讯地址；

需要注意当参数中参数类型有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。



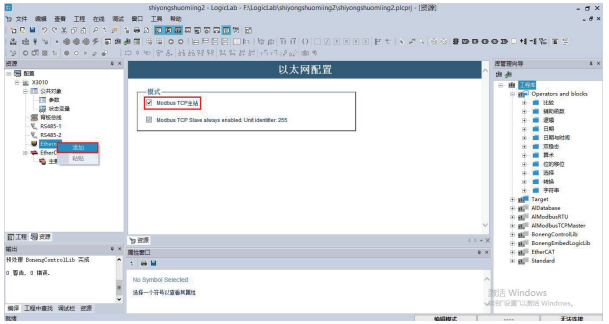
计算后结果如图：



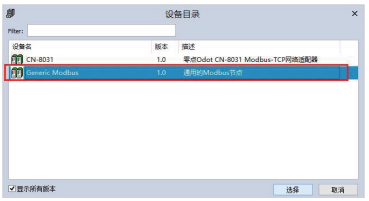
#	地址 (dec)	地址 (hex)	名称	参数类型	PLC类型	大小	默认值	最小
1	1	0x1	M1	Signed 16-bit	INT		0	
2	2	0x2	M2	Signed 16-bit	INT		0	
3	3	0x3	M3	Signed 16-bit	INT		0	
4	4	0x4	M4	Signed 32-bit	DINT		0	
5	6	0x6	M5	Signed 16-bit	INT		0	
6	7	0x7	M6	Signed 16-bit	INT		0	
7	8	0x8	M7	Signed 16-bit	INT		0	
8	9	0x9	M8	Signed 32-bit	DINT		0	
9	11	0xB	M9	Signed 16-bit	INT		0	
10	12	0xC	M10	Signed 16-bit	INT		0	

2.8.1.2 Modbus TCP主站

勾选Modbus-TCP主站，可以启动Modbus-TCP master；选中Ethernet点击右键，选择添加，这样可以添加modbus从站；选择通用类型Modbus从站节点，这样就完成了从站节点的添加。

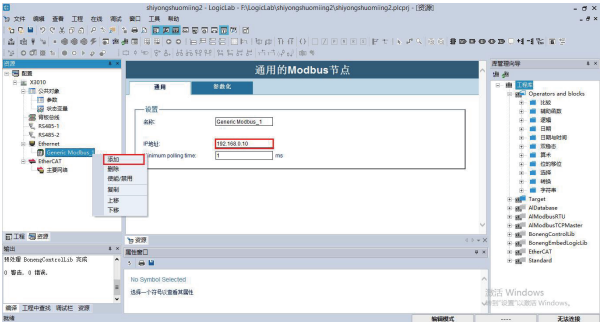


使能modbus-TCP主站



选择modbus TCP从站类型

添加modbus从站节点以后，在IP地址内添加从站的IP地址，默认从站地址是1，然后再添加该从站读写的数据；点击添加的从站节点，右键添加从站的数据类型。

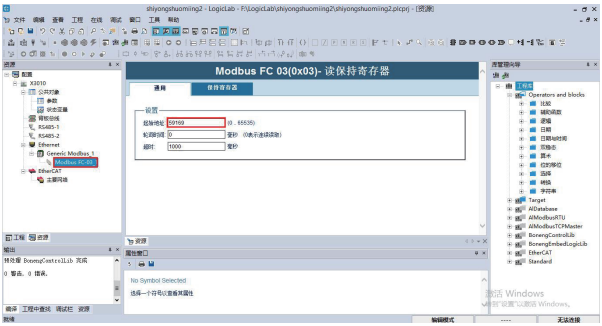


配置从站信息



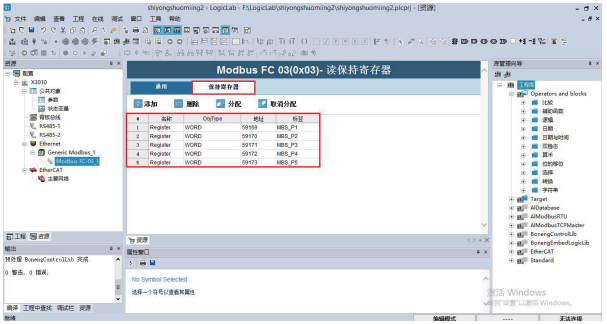
Modbus TCP从站数据类型

添加从站数据类型后，设置数据的起始地址；



配置参数的起始地址

选择保持寄存器窗口，点击添加来不同地址的参数以及参数名称；



配置读写参数

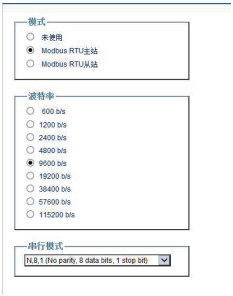
上述操作完成以后，编译并下载PLC程序，设备就可以和modbus TCP从站进行通信。

2.8.2 Modbus-RTU

在设备资源列表中，RS485-1表示控制器自带的接口，RS485-2表示扩展板的RS485接口。如下图所示：



选中任意RS485接口以后，可以看到模式选择的接口，可以配置接口为modbus RTU主站或者从站。如下图所示：(这里需要配置波特率，串行模式)



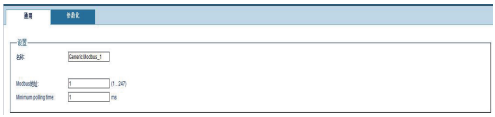
右击RS485-1，添加



这里选择通用的Modbus节点，



选择添加后，在资源中，可以看到，GenericModbus-1,点击后会出现以下页面



这里可以配置从机的地址，从1开始。
在资源中，右击GenericModbus-1添加



点击添加后如下图所示



在这里以选择读保持寄存器为例，读取从站的信息。点击选择。在资源中GenericModbus-1,目录下会出现以下页面



点击Modbus FC-03_1,这里可以配置起始地址，也就是驱动器功能码的地址，功能码的地址是16进制的数，需要将16进制的功能码转化为10进制的数后加1填写到起始地址中。在这里以读取开机显示选择2（0x202A）的信息为例



选择保持寄存器窗口，点击添加来不同地址的参数以及参数名称；



Modbus-RTU写入参数功能，与读参数功能类似。这里就不在一一赘述。

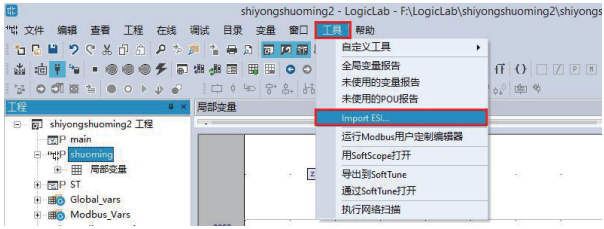
2.9 Ethercat功能使用

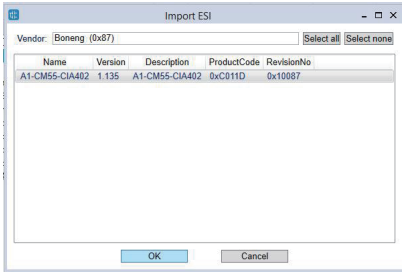
Ethercat通信需要首先添加从站设备，然后编译并下载PLC程序，就可以进行EtherCAT通信。

2.9.1 导入EtherCAT从站ESI文件

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--import ESI，然后选择对应的XML文件，选择文件中保存的设备即可。



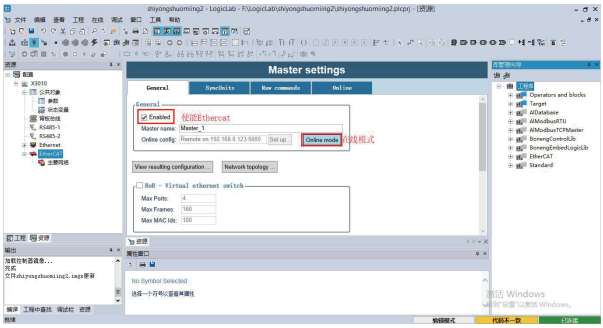




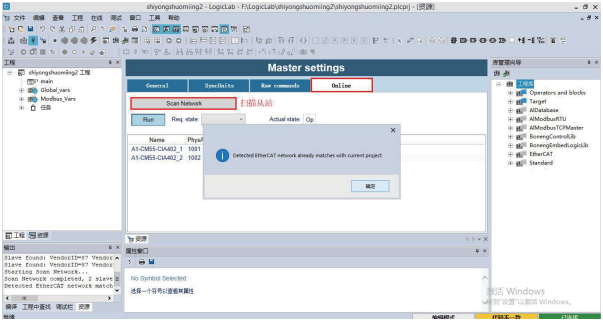
2.9.2 添加设备

2.9.2.1 自动扫描EtherCAT从站

点击Online mode，可以看到新增Online栏，然后点击Online栏，可以看到从站扫描按钮；

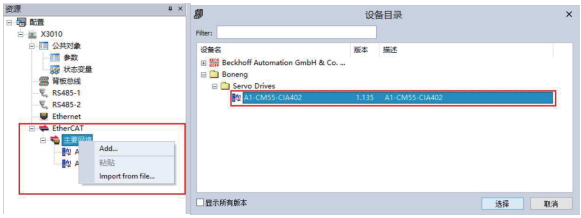


点击Scan Network按钮扫描从站，弹出窗口点击确定，就可以看到网络上连接的从站设备信息，如下图所示：



2.9.2.2 手动添加EtherCAT从站

在EtherCAT主站，右键选择add，然后在从站设备列表中选择相应的从站进行添加即可。





EtherCAT从站配置好以后，编译并下载PLC程序，就可以看到主从站正常进入OP模式，这时就可以在线实时查看通信数据值。

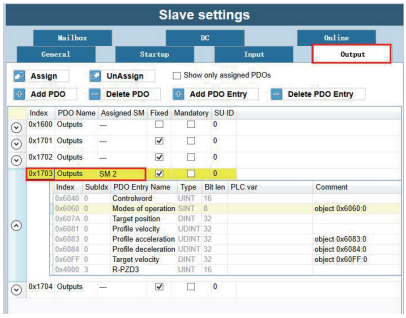
2.10 BonengControlLib使用

当前库包含BN_EtherCAT_PP_PV、BN_EtherCAT_SPEED功能块；BN_EtherCAT_PP_PV功能块可以驱动PP、PV模式，BN_EtherCAT_SPEED功能块驱动变频模式。库添加方式参考2.3.2库管理方式。

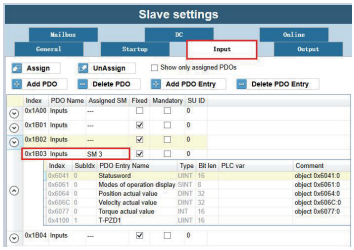
2.10.1 BN EtherCat_PP_PV模块使用

2.10.1.1 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：



2.10.1.2 引脚说明

BN EtherCat_PP_PV功能块引脚说明如下表所示：

Name	Range	DataType	Units	Explain
AxisIndex	EtherCat 轴 ID	UINT		0--15
OperationMode	1：相对位置	INT	\	1：6060 = 1 ； 6040 |= 0x40
	2：绝对位置			2：6060 = 1 ； 6040 &= -0x40
	3：速度模式			3：6060 = 3
				4：6060 = 4
	6：回零模式			6：6060 = 6
bAxisEnable	1：使能	BOOL	\	1：6040 |= 0x0F；0:6040 &= 0x7
	0：未使能			
bAxisHalt	1：暂停	BOOL	\	1：6040 |= 0x100
	0：正常			0：6040 &= -0x100
bAxisReset	1：故障复位	BOOL	上升沿触发	1：6040 |= 0x80；
	0：无故障			0：6040 &= -0x80
bAxisExecute	1：触发指令	BOOL	上升沿触发	当 OperationMode = 1、2、6 时生效
	0：正常			1：6040 |= 0x10
				0：6040 &= ~0x10
TargetPosition	-2147483647 ——2147483647	DINT	p	当 OperationMode = 1、2 时作为目标位置
TargetVel	-2147483647 ——2147483647	DITN	p/s	当 OperationMode = 1、2 、3 时速度指令
ProfileAcc	0——4294967295	UDINT	p/s^2	当 OperationMode = 1、2 、3 时加速度指令
ProfileDec	0——4294967295	UDINT	p/s^2	当 OperationMode = 1、2 、3 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			
bAxisError	1：轴有错误	BOOL	\	\
	0：轴无错误			
AxisError	轴错误代码	INT	\	\
bFault	1：功能块有错误	BOOL	\	\
	0：功能块无错误			
FBError	功能块错误代码	UINT	\	\
bTargetReach	1：目标到达	BOOL	\	\
	0：未到达			
ActMode	当前模式	INT	\	\
ActPosition	实际位置	DINT	\	\
ActVelocity	实际速度	DINT	\	\
PZD1	用户自定义 PZD1	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，则加加速度10000指令脉冲/s2 = 1r/s2=60rpm/s2,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；

3)减减速度，同上；

4)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；

2.10.1.3 驱动器参数设置

根据模块的输入接口给相应变量赋值，尤其注意需要设置伺服驱动器电子齿轮，否则伺服可能运行不起来，或者运行不正常。EtherCAT的电子齿轮请按分子分母的方式设置，目前只支持这一种方式。电子齿轮可以通过SDO设置，也可以直接在驱动器中设置，如果在驱动器中设置，则需要断电上电，否则不生效。

在开始运行前检查驱动器状态：

1)驱动器进入OP指示灯LNK1灯常亮；

2)驱动器参数“C06.13EC通讯状态”为“8：OP”；

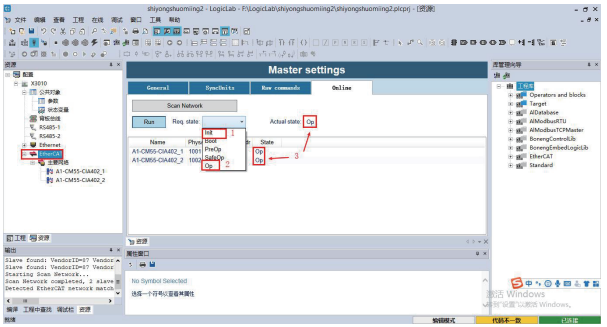
3)C06.21当前操作模式、C06.22指定操作模式保持一致且不能为“0：NA”；

驱动器参数不准确需要重新进入OP，操作步骤如下图所示：

1)点击init；

2)OP；

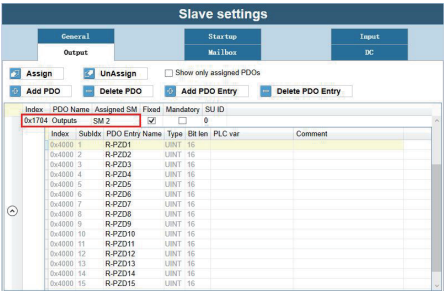
检查标记3都变OP。



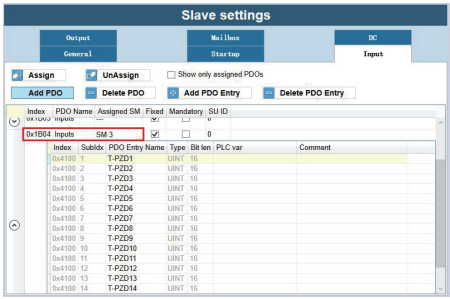
2.10.2 BN EtherCat_SPEED模块使用

2.10.2.1 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：



2.10.2.2 引脚说明

BN EtherCat_SPEED功能块引脚说明如下表所示：

Name	DataType	Explain	Other
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	\
bMotorRotation	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
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	\
DriveErrorCode	WORD	驱动器故障代码	\
ActSpeed	INT	反馈速度：基准值为 RefSpead	\
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。

2.10.2.3 驱动器参数设置

在开始运行前检查驱动器状态：

1)驱动器进入OP指示灯LNK1灯常亮；

2)A04.14驱动器驱动模式选择设为“0：变频模式”；

3)C06.04博能控制器功能块设为“1”；

4)驱动器参数“C06.13EC通讯状态”为“8：OP”；

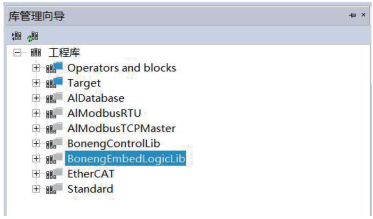
5)C06.21当前操作模式、C06.22指定操作模式保持一致且不能为“0：NA”；驱动器参数不准确需要重新进入OP；点击init、OP。

2.11 BonengEmbedLogicLib使用

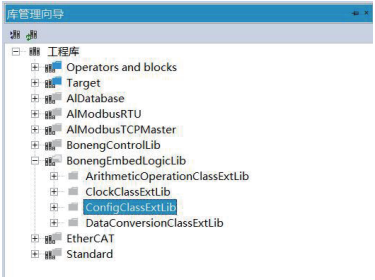
2.11.1 模拟量设置

2.11.1.1 模拟量输入电压电流的切换

首先在库管理向导中找到博能自定义库,找到BonengEmbedLogicLib 如下图所示:



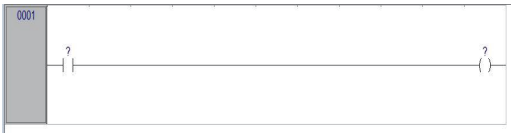
点击加号展开文件夹找到ConfigClassExtLib点击加号继续展开文件夹



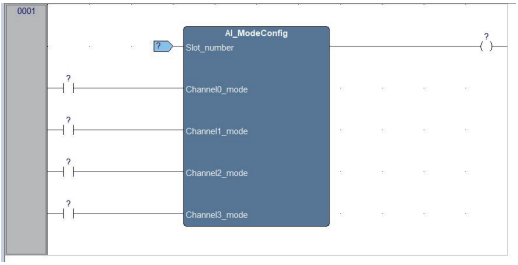
找到如右图所示函数 AI_ModeConfig 

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;

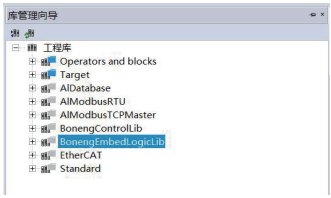


该函数共有5个输入参数,需要设置对应的槽号,和通道的模式即可切换电压,电流之间的输入检测转化,如果不调用该函数,默认电压模式输入。

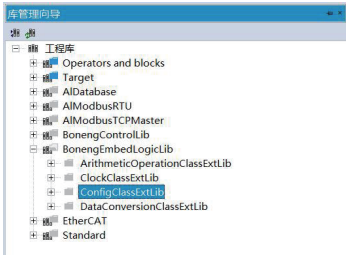
Slot_number:UINT; (*AI04扩展模块的槽号,在背板组态图中查询,范围1-15*)
Channel0_mode : BOOL; (* AI0模式选择,0:电压模式,1:电流模式 *)
Channel1_mode : BOOL; (* AI1模式选择,0:电压模式,1:电流模式 *)
Channel2_mode : BOOL; (* AI2模式选择,0:电压模式,1:电流模式 *)
Channel3_mode : BOOL; (* AI3模式选择,0:电压模式,1:电流模式 *)

2.11.1.2 用户自定义模拟量输入曲线

首先在库管理向导中找到博能自定义库,找到BonengEmbedLogicLib 如下图所示:



点击加号展开文件夹找到ConfigClassExtLib点击加号继续展开文件夹



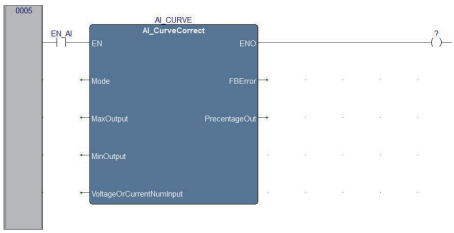
找到如右图所示功能块 

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_CurveCorrect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。

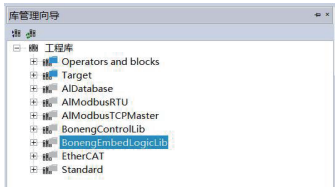
该功能块有两个输出的参数:

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

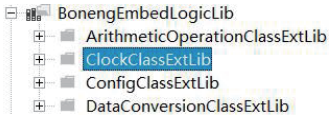
PrecentageOut : DINT;(* 特性曲线数值输出 *)

2.11.2 设置实时时间与获取实时时间

首先在库管理向导中找到博能自定义库,找到BonengEmbedLogicLib如下图所示:



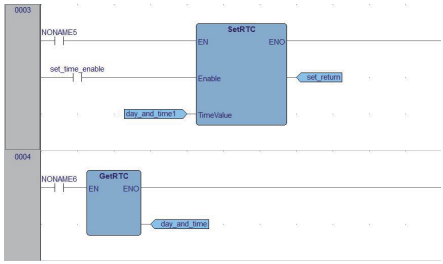
点击加号展开BonengEmbedLogicLib能看到ClockClassExtLib库



继续点击加号展开ClockClassExtLib可看到SetRTC和GetRTC



将这两个函数都添加到程序中,具体添加函数的方法见2.10.1.1节,添加完成后如下图所示:



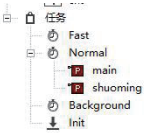
在SetRTC中EN引脚是该函数块的使能,Enable是该函数的写入数据的触发引脚,强调:(该引脚需要上升沿触发),TimeValue是该函数的时间设置引脚,该引脚的变量类型是TIME_AND_DAY,该引脚可设置如→1920-01-01-00:00:00格式的时间,该时间的设置一般来说是当前的北京时间,该函数的返回值不用关注,

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。

2.12编译

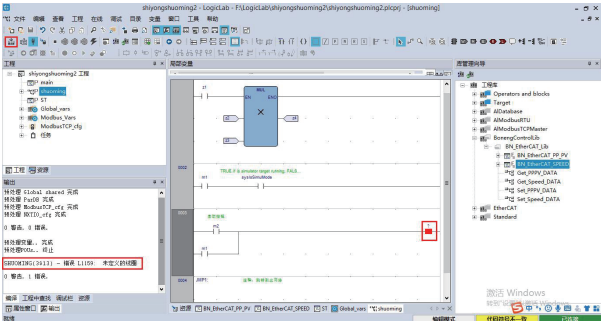
编译过程包括获取PLC源代码并将其自动转换为二进制代码，代码将在设备上的处理器执行。

在开始实际编译之前，确保至少已将程序分配给一个任务。需要注意LogicLab开始编译前会自动保存项目的所有变更。



2.12.1 编译项目

点击工具栏中编译按钮，编译的详细信息将在输出窗口显示，双击在输出栏中对应的错误行，即可定位到具体的编译错误位置上，编译有警告不影响程序的下载。



在工程中有不需要编译的程序时，工程项目树右击该程序，点击禁止编译。

2.13启动应用程序

为了下载和调试应用程序，必须与目标设备建立连接

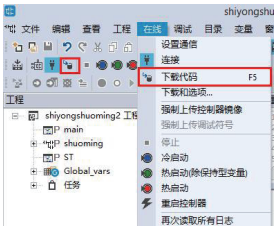
2.13.1 在线状态

在程序下载前状态栏必须是“已连接”如下图所示状态，如果显示未连接请查看“1.2 方式二”



2.13.2 下载程序

程序编译无错误（警告不影响程序下载）、在线状态正确可以下载程序，下载程序可以在工具栏点击下载按钮或者点击“在线”“下载代码”，如下图所示：

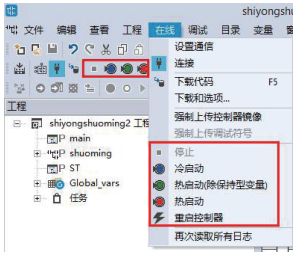


代码下载成功后状态栏显示状态应如下图所示：



2.13.3 控制PLC执行

可以使用工具栏中的相关功能或通过在线菜单中显示的命令来控制PLC应用程序的执行。



停止：单击停止PLC执行；

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；

热启动（除保持性变量）：PLC应用程序将被重新启动，并且非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；

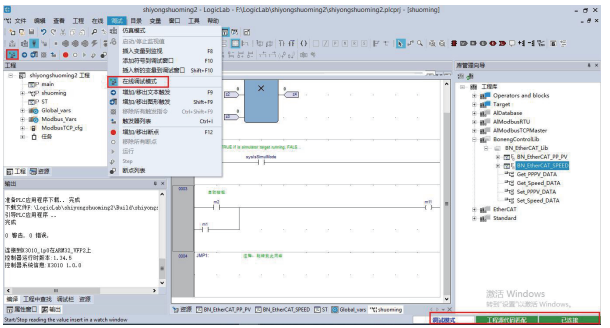
重启目标设备：以重新启动目标设备。

2.14 调试

LogicLab提供了调试工具，可帮助开发人员检查应用程序的行为是否符合预期。调试工具都允许开发人员在PLC应用程序运行时观察所选变量的值。

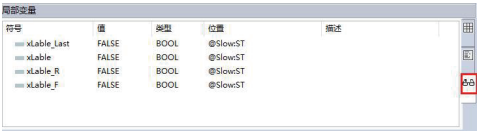
2.14.1 在线调试模式

进入在线调试模式后将无法更改程序，只能更改程序中变量的数值、状态。点击在线调试按或调试栏在线调试模式进入此模式，进入此模式状态栏显示调试模式，如下图所示；



进入在线调试模式双击程序中变量，输入要改为的状态、数值，点击置位即可改变变量的状态或数值。

在局部变量表和全局变量表中有监视按钮，可以监视变量表中的变量的值和在PLC程序中的位置，如下图所示：



2.14.2 监视窗口

监视窗口可以监视所选变量的当前值，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。监视窗口中显示的信息包括变量的名称、值、类型及其在PLC应用程序中的位置。

2.14.2.1 开启和关闭监视窗口

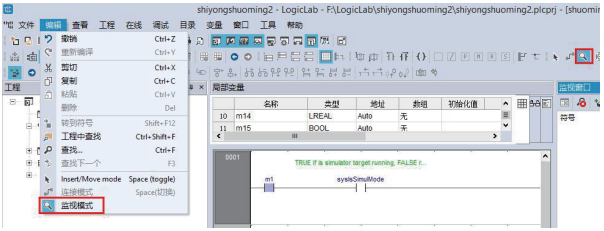
如果要打开或者关闭监视窗口中，单击导航栏“查看”“工具窗口”开启和关闭监视窗口。关闭监视窗口意味着仅隐藏它，而不是对其进行重置，关闭监视窗口，然后再次打开，仍然包含添加到其中的所有变量。

2.14.2.2 编辑器源代码添加监视变量

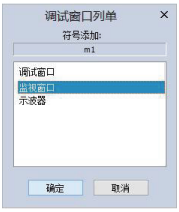
梯形图（LD）编辑器和结构化文本（ST）编辑器都可以在变量表中将变量拖至监视窗口进行监视。

从梯形图（LD）的源代码编辑器向监视窗口中添加变量：

点击工具栏监视按钮或导航栏监视模式，再点击需要监视的变量或块。



弹出一个对话框，列出所有当前存在的调试窗口实例。选择监视窗口，点击确定，变量的名称、值以及位置将会显示在监视窗口的新行中。



从文本（ST）的源代码编辑器向监视窗口中添加变量：需要先关闭在线调试模式，选择一个变量，通过双击，然后将其拖动到监视窗口。

2.14.2.3 删除监视变量

删除单个变量：单击其名称以选择变量，然后按键盘上的Delete键；

删除全部变量：点击监视窗口移除从监视列表中的所有对象如下图所示：



2.14.3 示波器

示波器允许绘制一组变量值的变化。示波器是一种异步工具，不能保证变量值的实时同步。

2.14.3.1 开启和关闭示波器

打开或关闭示波器，点击导航栏“查看--工具窗口”勾选或取消示波器即可；

关闭示波器只不过是将其隐藏而不是复位它；在关闭示波器后再次打开它，将重新看到添加所有变量的曲线图。

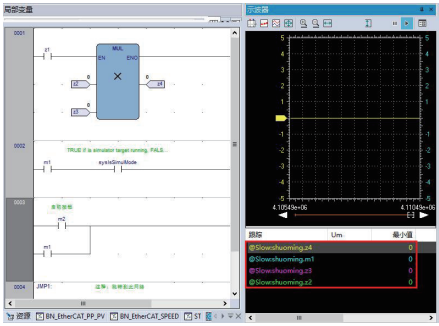
2.14.3.2 添加/删除对象到示波器

为了绘制一个变量值的变化，将其添加到示波器中。无论变量对象在何处，都可以将其添加到示波器中。梯形图（LD）编辑器和结构化文本（ST）编辑器都可以在变量表中将变量拖至示波器窗口查看波形。

从梯形图（LD）的源代码编辑器向示波器窗口中添加变量：

点击工具栏监视按钮或导航栏监视模式，再点击需要监视的变量，弹出一个对话框，列出所有当前存在的调试窗口实例。选择示波器，点击确定，变量将会显示在示波器窗口；





从文本（ST）的源代码编辑器向监视窗口中添加变量：需要先关闭在线调试模式窗口，选择一个变量，通过双击，然后将其拖动到监视；

删除对象：单击其名称以选择它，然后按键盘上的Delete键。

3.其他说明及注意事项

3.1故障说明

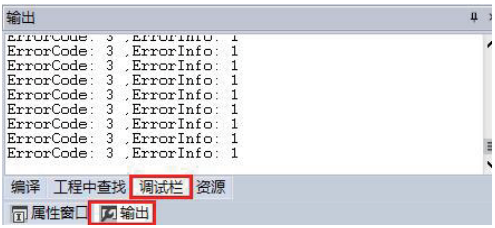
当CPU ERR指示灯红灯亮起时表示出现错误，引起错误的原因有两种：扩展模块组态错误（多组态模块、模块顺序错误）或者扩展模块本身错误。

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。

3.2注意事项

1)工程名和新建程序名称必须为英文，中文会编译不通过；

2)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；

3)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；

4)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。

5)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；

6)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；

7)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；

8)开始实际编译之前，确保至少已将程序分配给一个任务；

9)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。

10)在变量声明处（ST语言/变量表）给变量赋初始值时，需要将设备冷启动，才能将变量的初始值对应改变。

11) LogicLab中组态的EtherCAT从站类型和数量应该与实际物理连接的类型和数量保持一致，否则主站无法进入OP。

4 售后服务

各种传动设备，客户发现有质量问题时，不要先拆卸零件，应说明以下情况然后与本公司售后服务部联系，说明现象后确认问题所在，再采用较理想的方法处理。

型号规格: _____

出厂日期: _____

编　号: _____

已使用时间: _____

使用场合或主机名称: _____

主机生产单位: _____

质量问题描述: _____

_____

_____

_____

_____

用户单位: _____

地　址: _____

电　话: _____　　传 真: _____

邮　编: _____　　联系人: _____

博能售后服务电话: _____

传 真: _____

NOTE: _____

_____

_____

_____

_____

X3000系列控制器使用说明书

BONENG

博能传动(苏州)有限公司
苏州市相城区如元路100号
www.boneng.com

BONENG