运 动 控 制 器
MOTION CONTROLLER

EURA DRIVES

使用手册
EAC100系列(SP17)运动控制器

欧瑞传动
DRIVES EURA DRIVES

目 录

第一章 概述

本章简要介绍了 EAC100 SP17 系列运动控制器主要特点。主要内容为：产品特点、产品型号命名规范、产品技术参数和使用产品的注意事项等，有助于用户初步了解产品的构成和使用规范。

1.1 EAC100 SP17 系列运动控制器简述

EAC100 SP17 系列运动控制器是由欧瑞传动电气股份有限公司研发的经济型软 PLC 型运动控制器系列，支持 LD、ST、IL、FBD、SFC 和 CFC 等 6 种编程语言，符合 IEC 61131-3 标准。EAC100 系列提供了控制和可视化的一体化软件开发运行环境，并集成强大的运动控制功能。内置的高性能 EtherCAT 主站，同步时间误差<1μs。

优秀的控制性能和开放式的软件架构，使 EAC100 SP17 系列为工业自动化领域提供了一个良好的控制系统开发平台。

该系列产品具有如下特点

1. 标准化的软 PLC 开发平台

支持 LD,ST,IL,SFC,FBD 和 CFC 等 6 种编程语言，符合 IEC 61131-3 标准，支持 CODESYS

2. 丰富强大的控制功能

集成逻辑控制、运动控制和可视化功能，包含直线插补、圆弧插补、电子齿轮、电子凸轮和 CNC、机器人控制等功能模块

3. 控制与可视化的集成式开发环境

在同一平台下完成控制和可视化程序开发，可极大简化传统的“HMI+PLC”的开发模式

4. 内置高性能 EtherCAT 主站

非 DC 模式的最小通信周期 1ms；DC 模式的最小通信周期 2ms，节点间的同步时间误差<1μs

5. 开放式软硬件架构

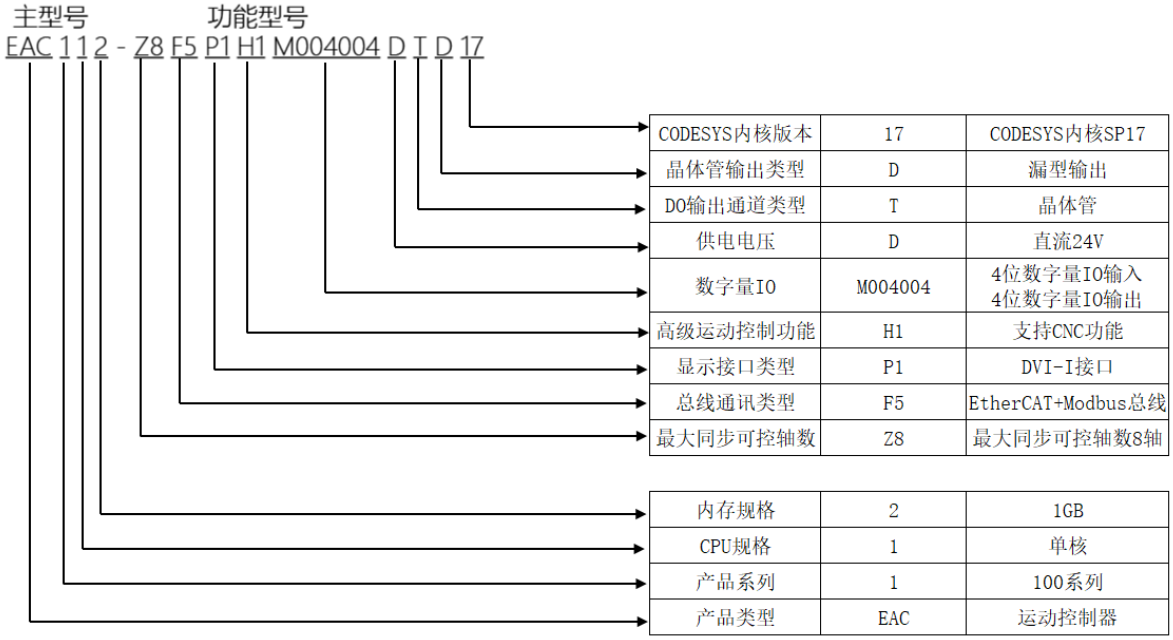
面向 OEM 用户提供计算机高级编程语言开发接口和软硬件定制服务，支持 Visual Studio 开发环境

6. 丰富的接口和协议栈

EAC100 SP17 提供 RJ45、RS232、RS485 和数字 IO 等工业通用接口，并内置 EtherCAT 主站、Modbus RTU 主站/从站，能够轻松便捷的与其他设备进行数据交互。

1.2 EAC100 SP17 系列运动控制器命名规范及订货信息

EAC100 SP17 系列运动控制器产品命名分为两部分：主型号和功能代号。其中主型号包括产品系列名称、CPU 规格和内存规格；功能代号包括最大同步控制轴数、高级软件支持等一系列功能的描述，详细可以在功能代号表中查询。标准的产品命名如下图所示。



EAC100 SP17 系列产品命名示例

主型号中各位置含义请参照下表：

功能名称	功能代码	含义
CPU 类型	1	单核 CPU
	2	双核 CPU
	4	4 核 CPU
内存规格	1	512MB
	2	1GB
	3	2GB

功能代码中各位置含义请参照下表：

功能名称	功能代码	含义
最大同步控制轴数	Z8	最大同步控制轴数为 8 轴
	Z16	最大同步控制轴数为 16 轴
	Z32	最大同步控制轴数为 32 轴
总线通讯类型	F5	EtherCAT+Modbus
	F13	EtherCAT
显示接口类型	P1	DVI-I 接口
	P3	HDMI 接口
高端运动控制功能	H1	支持 CNC 和 Robot 功能
数字量 IO	M004004	支持 4 位数字 IO 输入，4 位数字 IO 输出
	M020020	支持 20 位数字 IO 输入，20 位数字 IO 输出
	M036036	支持 36 位数字 IO 输入，36 位数字 IO 输出
	M068068	支持 68 位数字 IO 输入，68 位数字 IO 输出
供电电压	D	直流 24V
DO 输出通道类型	T	晶体管
晶体管输出类型	D	漏形输出
CODESYS 内核版本号（CODESYS Core Version）	17	CODESYS 内核版本号 SP17

现有产品的订货信息如下表所示：

订货号	规格	备注
EAC112-Z8F5P3H1M004004DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 8 轴 / EtherCAT+Modbus / HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 4 位数字 IO 输入 / 4 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	
EAC112-Z16F5P3H1M004004DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 16 轴 / EtherCAT+Modbus/	

	HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 4 位数字 IO 输入 / 4 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	
EAC112-Z32F5P3H1M004004DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 32 轴 / EtherCAT+Modbus/ HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 4 位数字 IO 输入 / 4 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	

订货号	规格	备注
EAC112-Z8F5P3H1M020020DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 8 轴 / EtherCAT+Modbus/ HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 20 位数字 IO 输入 / 20 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	
EAC112-Z16F5P3H1M020020DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 16 轴 / EtherCAT+Modbus/ HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 20 位数字 IO 输入 / 20 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	
EAC112-Z32F5P3H1M020020DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 32 轴 / EtherCAT+Modbus/ HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 20 位数字 IO 输入 / 20 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	

订货号	规格	备注
EAC112-Z8F5P3H1M036036DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 8 轴 / EtherCAT+Modbus/	

	HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 36 位数字 IO 输入 / 36 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	
EAC112-Z16F5P3H1M036036DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 16 轴 / EtherCAT+Modbus/ HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 36 位数字 IO 输入 / 36 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	
EAC112-Z32F5P3H1M036036DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 32 轴 / EtherCAT+Modbus/ HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 36 位数字 IO 输入 / 36 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	

订货号	规格	备注
EAC112-Z8F5P3H1M068068DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 8 轴 / EtherCAT+Modbus HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 68 位数字 IO 输入 / 68 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	
EAC112-Z16F5P3H1M068068DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 16 轴 / EtherCAT+Modbus/ HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 68 位数字 IO 输入 / 68 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	
EAC112-Z32F5P3H1M068068DTD17	单核 CPU / 1GB 内存/ 最大同步控制轴数 32 轴 / EtherCAT+Modbus/	

	HDMI 显示输出接口 / 支持 CNC 和 Robot 功能/ 供电电压 DC24V / 68 位数字 IO 输入 / 68 位数字 IO 输出 / IO 输出类型为晶体管漏型输出	

1.3 EAC100 SP17 系列运动控制器技术规格及工作环境参数

EAC100 SP17 系列运动控制器技术规格及工作环境参数如下表所示：

技术参数	规格
工作电源（直流）	DC24V±15%，>3.2A
工作温度	-10℃～50℃
存储温度	-25℃～70℃
相对湿度	<95%无冷凝
防护等级	IP20
工作环境	无水滴、蒸汽、腐蚀、易燃、灰尘及金属微粒的场所

1.4 安全注意事项

本节对与本系列产品相关的安全注意事项进行说明。如果不遵守这些注意事项，可能会导致死亡或重伤、并损坏本产品、相关机器及系统。因未遵守本使用说明书的内容而造成的伤害和设备损坏，本公司将不负任何责任。

1.4.1 安全信息定义

危险：如不遵守相关要求，就会造成严重的人身伤害，甚至死亡。

警告：如不遵守相关要求，可能会造成人身伤害或者设备损坏。

注意：为了确保正确的运行而采取的步骤。

培训并合格的专业人员：是指操作本设备的工作人员必须经过专业的电气培训和安全知识培训并且考试合格，已经熟悉本设备的安装，调试，投入运行以及维护保养的步骤和要求，并能避免产生各种紧急情况。

1.4.2 警告标示

警告用于对可能造成严重的人身伤亡或设备损坏的情况进行警示，给出建议以避免发生危险。本手册中使用下列警告标识：

标识	名称	说明	简写
 危险	危险	如不遵守相关要求，就会造成严重的人身伤害，甚至死亡。	

警告	警告	如不遵守相关要求，可能会造成人身伤害或者设备损坏。	
注意	注意	为了确保正确的运行而采取的步骤。	**注**

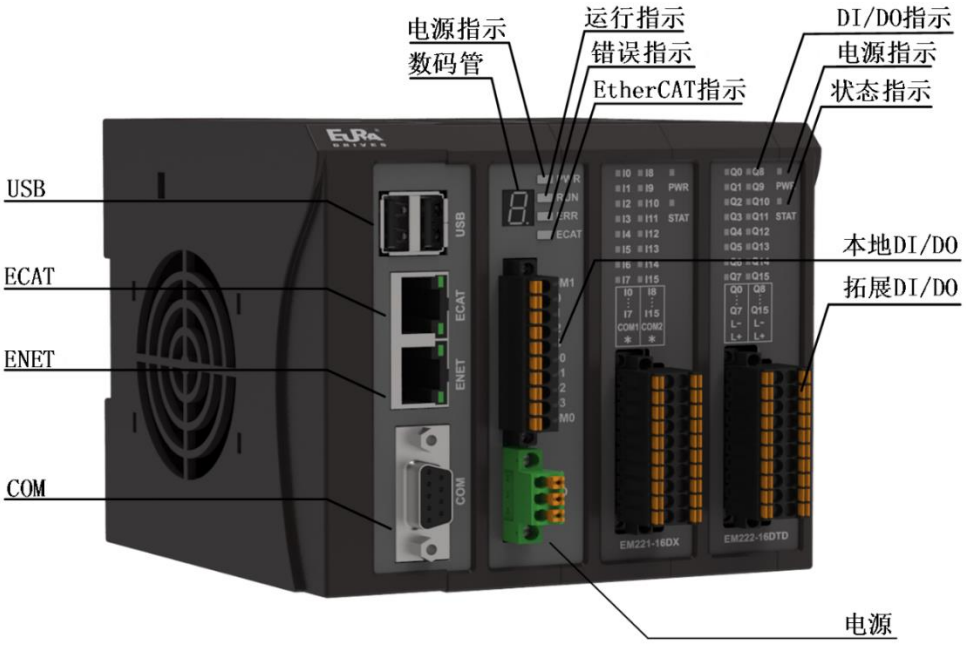
1.4.3 安全指导

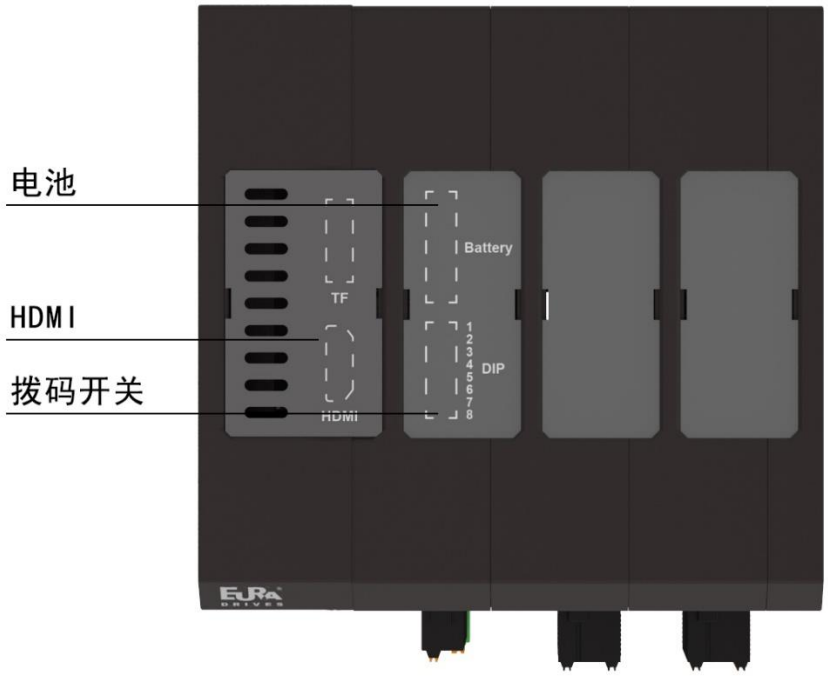
	◇只有经过培训并合格的人员才允许进行相关操作。 ◇禁止在电源接通的情况下进行接线，检查和更换器件等作业。进行接线及检查之前，必须确认所有输入电源已经断开
	◇未经授权严禁对运动控制器进行的改装和拆卸，否则可能引起火灾，触电或其他伤害。 ◇禁止将运动控制器安装在易燃物上，并避免运动控制器紧密接触或粘附易燃物。 ◇客户收到产品后，请检查控制器有无外壳损坏，包装箱内有无水渍，如有请联系当地经销商或者当地办事处。

第二章　　运动控制器硬件介绍

2.1 控制器外形结构

2.1.1 EAC112 SP17 系列

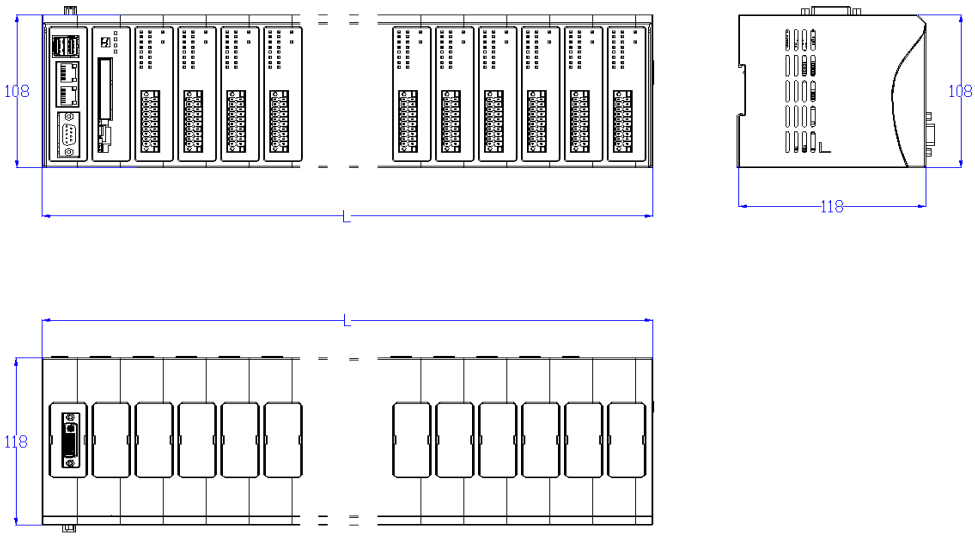




EAC112 SP17 系列运动控制器整体视图及结构名称

2.2 外形尺寸及安装方式

2.2.1 外形尺寸



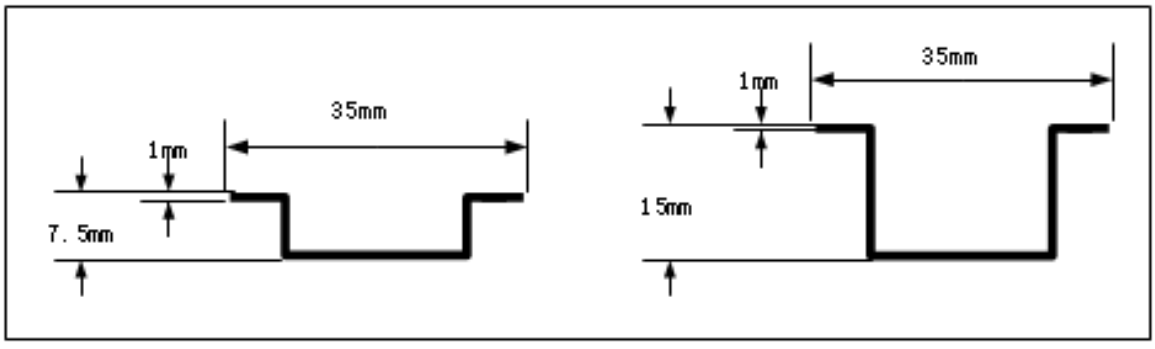
注：L=68+24*N（N=模块数）　　单位：mm

型号	L	W	H	建议安装卡扣数量
EAC112-Z8F5P3H1M004004DTD17	68	118	108	2
EAC112-Z16F5P3H1M004004DTD17				
EAC112-Z32F5P3H1M004004DTD17				
EAC112-Z8F5P3H1M020020DTD17	116	118	108	2
EAC112-Z16F5P3H1M020020DTD17				
EAC112-Z32F5P3H1M020020DTD17				
EAC112-Z8F5P3H1M036036DTD17	164	118	108	3
EAC112-Z16F5P3H1M036036DTD17				
EAC112-Z32F5P3H1M036036DTD17				
EAC112-Z8F5P3H1M068068DTD17	260	118	108	4
EAC112-Z16F5P3H1M068068DTD17				
EAC112-Z32F5P3H1M068068DTD17				

2.2.2 控制器安装方式

使用 DIN 导轨安装步骤

①准备好标准的 35mm 宽 DIN 导轨，有两种规格，如下图所示。



DIN 导轨示意图

②将导轨安装至需要的位置，将控制器卡接于导轨上。

方法：将控制器底部的 35mm 导轨卡接滑块拉下，从导轨的上部装入控制器，向前推控制器下部直到控制器紧贴导轨，然后再将卡接滑块推到原位即可。

注：为保证良好的散热和通风要求，控制器安装时四周应各留出至少 60mm 空间。

2.3 运动控制器本体部分

2.3.1 控制器本体运行状态指示及系统工作状态

控制器运行状态指示包括数码管和电源（PWR）、运行（RUN）、故障（ERR）、通讯（ECAT）四个状态指示灯，如下图所示。

设备的各个工作状态都可以通过设备上的数码管和指示灯进行显示，系统的各个工作状态及显示如下表所示。

序号	描述	数码管显示	指示灯显示				备注
			PWR	RUN	ERR	ECAT	
1	系统启动	8.	蓝色转绿色	不亮	不亮	不亮	
2	配置状态	2	绿色常亮	蓝色常亮	不亮	不亮	
3	系统程序更新中	3	绿色常亮	蓝色闪烁	不亮	不亮	
4	Runtime 启动中	4	绿色常亮	红色闪烁	不亮	不亮	
5	Runtime 启动完成，但无 PLC 程序	5	绿色常亮	红色常亮	不亮	不亮	
6	PLC 程序正常运行	6	绿色常亮	绿色常亮	不亮	N/A	ECAT 灯状态取决于程序中 EtherCAT 总线运行状态，正常运行为绿色常亮
7	PLC 程序停止	7	绿色常亮	红色常亮	不亮	N/A	ECAT 灯状态取决于程序中 EtherCAT 总线运行状态，正常运行为绿色常亮

2.3.2 DIP 拨码开关

DIP 拨码开关用于在系统配置阶段对系统功能进行配置。使用时应在系统上电前将对应拨码切换到需求位置，随后正常上电即可。各拨码位定义如下表所示：

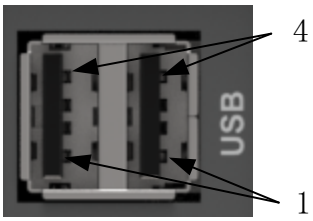
DIP 编号	拨码功能
1	ON：系统启动后自动运行 CODESYS 程序； OFF：启动后保持配置阶段
2	ON：当 DIP5 为 ON 时清除 PLC 程序
3	ON：当 DIP5 为 ON 时清除掉电保持数据
4	ON：屏蔽 RTC 时钟复位错误
5	ON：DIP2、DIP3、DIP8 功能有效
6	保留待用
7	保留待用
8	ON：复位与上位机通讯网卡的 IP 地址和 MAC 地址

2.3.3 控制器本体对外接口

2.3.3.1 USB 接口

EAC100 系列运动控制器共有 2 个 USB2.0 接口，支持标鼠标、键盘、U 盘等标准 USB 设备，并可以通过 HUB 进行扩展，USB 通讯口定义如下表所示：

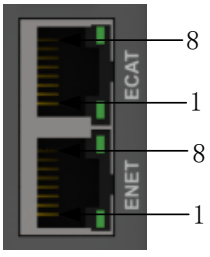
引脚号	描述	信号
1	+5V	VCC
2	Data-	D-
3	Data+	D+
4	电源地	GND



2.3.3.2 网络接口（RJ45）

EAC100 系列运动控制器带有 2 个 RJ45 接口，ECAT 接口用于连接 EtherCAT 从站设备，ENET 连接上位机，用于程序下载、在线监控、网页配置等功能，接口定义如下表所示：

引脚号	描述	信号
1	数据发送正端	TX+
2	数据发送负端	TX-
3	数据接收正端	RX+
6	数据接收负端	RX-

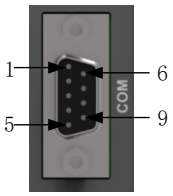


2.3.3.3 串行接口（COM）

EAC100 系列运动控制器配备有一个 RS232 和一个 RS485 串行通讯接口，两个接口共用一个 DB9 母型接口。该接口可以用作与第三方设备的通讯口。当采用屏蔽电缆时，RS232 通讯的距离建议不超过 15 米，RS485 通讯的最大距离可达 1000 米。串行接口的定义如下表
所示：

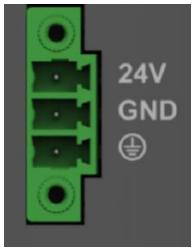
RS-232		
孔号	描述	信号
2	接收数据	RXD
3	发送数据	TXD
5	信号地	GND

RS-485		
孔号	描述	信号
7	RS485+	A+
8	RS485-	B-



2.3.3.5 电源接口

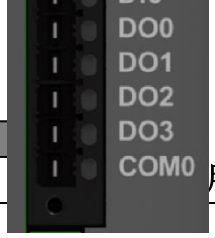
名称	描述
24V	24V 电源正
GND	24V 电源负
⏚	大地



2.3.3.6 数字 IO 接口（DI/DO）

EAC100 系列运动控制器本体配备有 4 路数字量输入和 4 路数字量输出。其接口定义 如下所示：

信号	描述
COM1	输入公共端
DI0	输入端口 0
DI1	输入端口 1
DI2	输入端口 2
DI3	输入端口 3
DO0	输出端口 0
DO1	输出端口 1
DO2	输出端口 2



DO3	输出端口 3
COM0	输出公共端

2.3.3.7 HDMI 接口

EAC112 系列运动控制器本体配备有 1 个 HDMI 输出口。其接口定义如下：

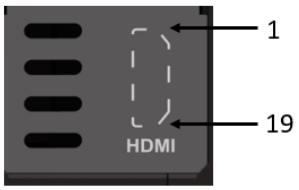
引脚号	信号	引脚号	信号
1	HOT_PLUG_DET	11	TMDS_DATA0_N
2	+5V	12	GND
3	GND	13	TMDS_DATA0_P
4	SDA	14	TMDS_DATA1_N
5	SCL	15	GND
6~7	空	16	TMDS_DATA1_P
8	TMDS_CLK_N	17	TMDS_DATA2_N
9	GND	18	GND
10	TMDS_CLK_P	19	TMDS_DATA2_P

2.3.4 控制器输入（DI）

运动控制器本体提供 4 路 DI 通道，用于普通的数字量（开关量）输入。各输入通道与内部 CPU 电路之间均有电气隔离，并有状态指示灯指示各通道的输入状态。

DI 输入通道主要特点：

- 4 路晶体管输入通道，共分成 1 组
- 各组既可接源型输入（共阴极），也可以接漏型输入（共阳极）
- 额定输入电压为 DC24V
- 现场信号与内部电路之间有电气隔离器
- 各通道有独立的状态指示灯

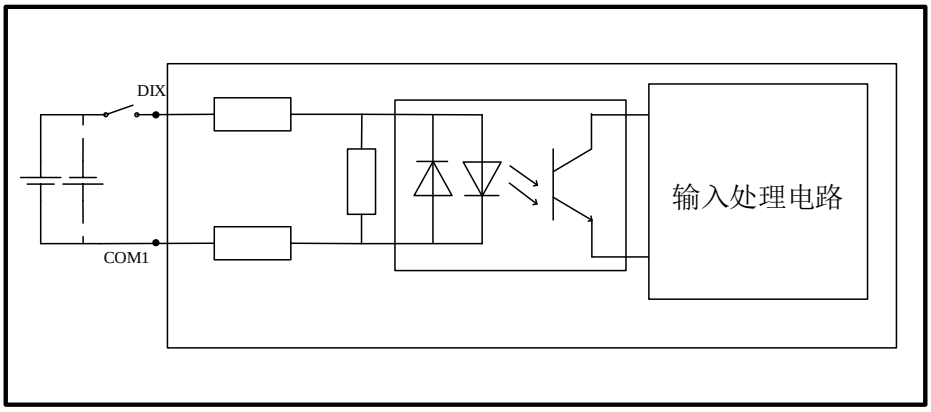


DI 通道技术参数：

项目	规格
输入类型	源型/漏型可选
额定输入电压	24VDC
额定输入电流	4.1mA@24VDC
最大输入电压	30VDC

逻辑 1 最小输入电压	15V@2.5mA
逻辑 0 最大输入电压	2.8V@0.7mA
输入与内部逻辑电路的隔离方式	光电耦合器
输入与内部逻辑电路的隔离电压	1500VAC/1 分钟
状态指示	状态指示灯亮绿色/红色

DI 输入通道电气原理图：



2.3.5 控制器输出（DO）

运动控制器本体提供 4 路 DO 通道，用于普通的数字量（开关量）输出。各输出通道与内部电路之间均有电气隔离，并有状态指示灯指示各输出通道的通断状态。

晶体管型 DO 输出通道主要特点:

- 4 路晶体管输出通道，分成 1 组
- 额定供电电压为 DC24V
- 额定输出电压为 DC24V，每通道最大输出电流为 500mA，漏型（NPN）
- 供电电源接入极性保护
- 感性负载输出保护
- 短路保护（每路输出电流大于 500mA 时）
- 同一组内通道允许并联
- 输出与内部电路之间光电隔离

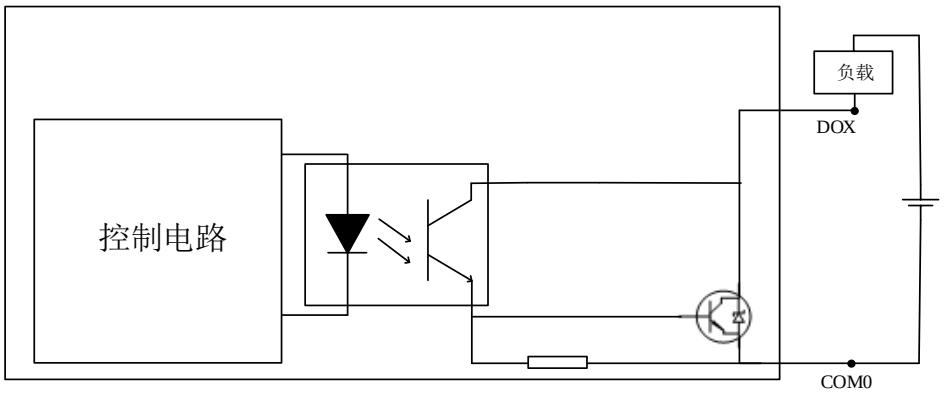
晶体管型 DO 通道技术参数

项目	规格

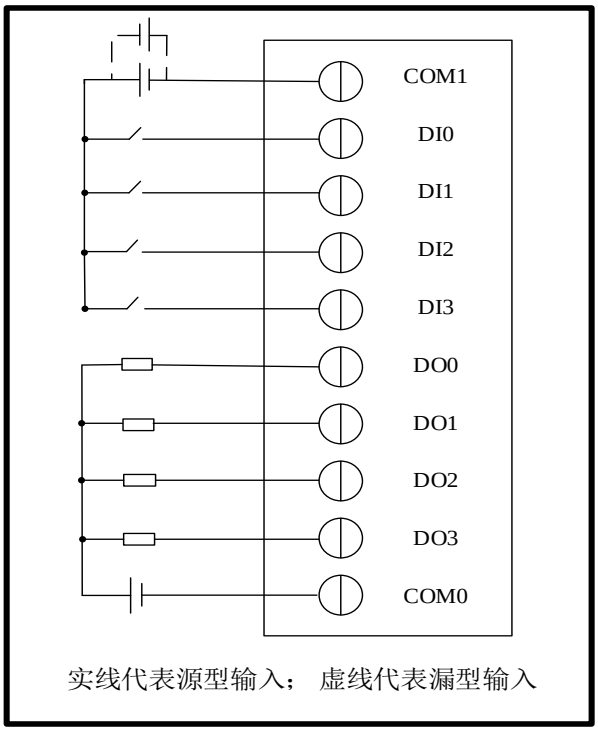
输出类型	漏型[注]
额定输出电压	24VDC
额定供电电压	24VDC
电源接入极性保护	有
每通道输出电流	最大 500mA @24VDC
输出漏电流	25μA（最大）
输出与内部逻辑电路的隔离方式	光电隔离
输出与内部逻辑电路的隔离电压	1500VAC/1 分钟
通道并联功能	有
短路保护功能	有
状态指示	状态指示灯亮绿色

注：漏型输出时无限流电阻

DO 输出通道电气原理图：



2.3.6 数字量 DI/DO 接线图



2.4 运动控制器数字量拓展 IO 部分

2.4.1 控制器数字量拓展 IO 运行状态指示及系统工作状态

控制器数字量拓展部分运行状态指示包括电源（PWR）、状态（STAT）2 个状态指示灯，如右图所示。

系统上电后各个阶段及其显示状态如下表所示：

序号		状态	描述	备注
1	PWR	绿色常亮	系统电源正常	
2		不亮	系统电源异常	
3	STAT	不亮	停止状态	
4		绿色闪烁	初始化状态，地址分配完成	
5		绿色常亮	正常运行状态	
6		红色闪烁	通信超时或地址分配错误	

2.4.2 DI 16*DC24

该模块的型号为：EM221-16XD

EM221-16XD 提供 16 路 DI 通道，用于普通的数字量（开关量）输入。各输入通道与内部 CPU 电路之间均有电气隔离，并有状态指示灯指示各通道的输入状态。

注：EM221-16XD 不可单独订货；

信号	描述	信号	描述
I0	输入端口 0	I8	输入端口 8
I1	输入端口 1	I9	输入端口 9
I2	输入端口 2	I10	输入端口 10
I3	输入端口 3	I11	输入端口 11
I4	输入端口 4	I12	输入端口 12
I5	输入端口 5	I13	输入端口 13
I6	输入端口 6	I14	输入端口 14
I7	输入端口 7	I15	输入端口 15
COM1	输入公共端 1	COM2	输入公共端 2
*	未定义	*	未定义

DI 输入通道主要特点：

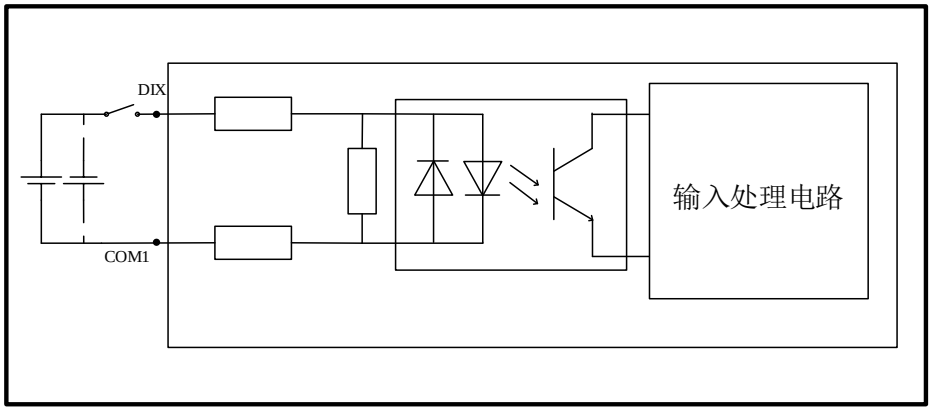
- 16 路晶体管输入通道，共分成 2 组（I0~I7 为一组，共用公共端 COM1, I8~I15 为一组，共用公共端 COM2）
- 各组既可接源型输入（共阴极），也可以接漏型输入（共阳极）
- 额定输入电压为 DC24V
- 现场信号与内部电路之间有电气隔离器
- 各通道有独立的状态指示灯

DI 通道技术参数：

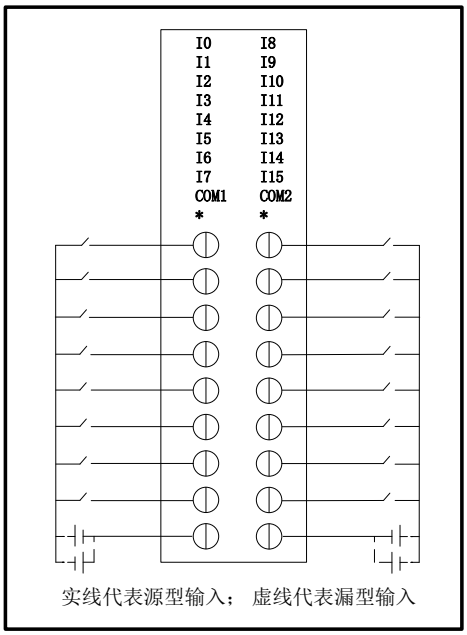
项目	规格
输入类型	源型/漏型可选
额定输入电压	24VDC
额定输入电流	4.1mA@24VDC
最大输入电压	30VDC
逻辑 1 最小输入电压	15V@2.5mA
逻辑 0 最大输入电压	2.8V@0.7mA
输入与内部逻辑电路的隔离方式	光电耦合器

输入与内部逻辑电路的隔离电压	1500VAC/1 分钟
状态指示	状态指示灯亮绿色/红色

DI 输入通道电气原理图：



接线图：



2.4.3 DO 16*DC24

该模块的型号为：EM222-16DTD

EM222-16DTD 提供 16 路 DO 通道，用于普通的数字量（开关量）输出。各输出通道与内部电路之间均有电气隔离，并有状态指示灯指示各输出通道的通断状态。

注：EM222-16DTD 不可单独订货；

信号	描述	信号	描述
Q0	输出端口 0	Q8	输入端口 8
Q1	输出端口 1	Q9	输入端口 9
Q2	输出端口 2	Q10	输入端口 10
Q3	输出端口 3	Q11	输入端口 11
Q4	输出端口 4	Q12	输入端口 12
Q5	输出端口 5	Q13	输入端口 13
Q6	输出端口 6	Q14	输入端口 14
Q7	输出端口 7	Q15	输入端口 15
L-	24V 电源负	L-	24V 电源负
L+	24V 电源正	L+	24V 电源正

注：L+、L-为外接电源，接 1 组即可；

晶体管型 DO 输出通道主要特点:

- 16 路晶体管输出通道
- 额定供电电压为 DC24V
- 额定输出电压为 DC24V，每通道最大输出电流为 500mA，漏型（NPN）
- 供电电源接入极性保护
- 感性负载输出保护
- 短路保护（每路输出电流大于 500mA 时）
- 同一组内通道允许并联
- 输出与内部电路之间光电隔离

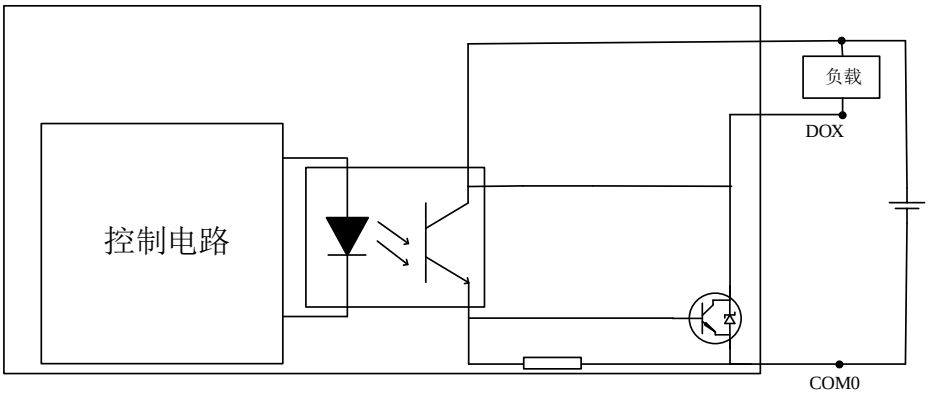
晶体管型 DO 通道技术参数

项目	规格
输出类型	漏型[注]
额定输出电压	24VDC
额定供电电压	24VDC
电源接入极性保护	有
每通道输出电流	最大 500mA @24VDC
输出漏电流	25μA（最大）

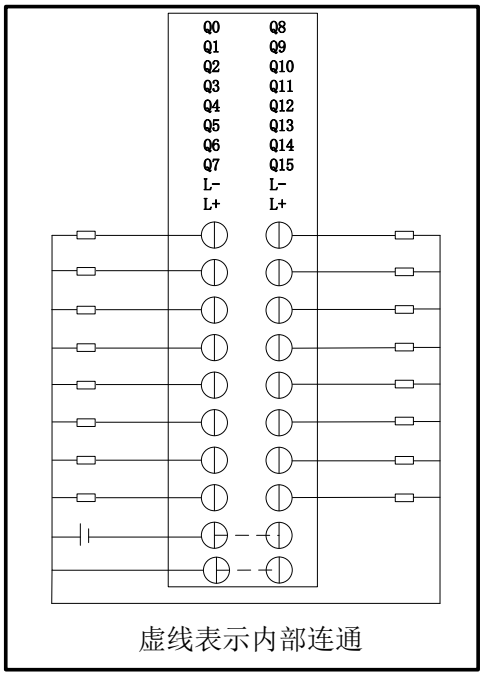
输出与内部逻辑电路的隔离方式	光电隔离
输出与内部逻辑电路的隔离电压	1500VAC/1 分钟
通道并联功能	有
短路保护功能	有
状态指示	状态指示灯亮绿色

注：漏型输出时无限流电阻

DO 输出通道电气原理图：



接线图：



2.4.4 DI 8*DC24+DO 8*DC24

该模块的型号为：EM223-16DTD

EM223-16DTD 供 8 路 DI 通道和 8 路 DO 通道，用于普通的数字量（开关量）输入、输出。各输入通道与内部 CPU 电路之间均有电气隔离，并有状态指示灯指示各通道的输入状态。各输出通道与内部电路之间均有电气隔离，并有状态指示灯指示各输出通道的通断状态。

注：EM223-16DTD 不可单独订货；

信号	描述	信号	描述
I0	输入端口 0	Q0	输出端口 0
I1	输入端口 1	Q1	输出端口 1
I2	输入端口 2	Q2	输出端口 2
I3	输入端口 3	Q3	输出端口 3
I4	输入端口 4	Q4	输出端口 4
I5	输入端口 5	Q5	输出端口 5
I6	输入端口 6	Q6	输出端口 6
I7	输入端口 7	Q7	输出端口 7
COM1	输入公共端 1	L-	24V 电源负
*	未定义	L+	24V 电源正

注：L+、L-为外接电源

DI 输入通道主要特点：

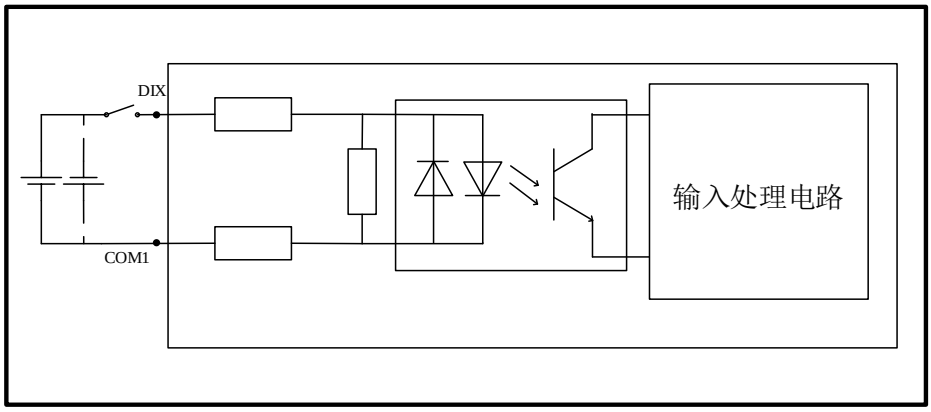
- 8 路晶体管输入通道，共分成 1 组，共用 COM1 端口
- 各组既可接源型输入（共阴极），也可以接漏型输入（共阳极）
- 额定输入电压为 DC24V
- 现场信号与内部电路之间有电气隔离器
- 各通道有独立的状态指示灯

DI 通道技术参数：

项目	规格
输入类型	源型/漏型可选
额定输入电压	24VDC
额定输入电流	4.1mA@24VDC
最大输入电压	30VDC

逻辑 1 最小输入电压	15V@2.5mA
逻辑 0 最大输入电压	2.8V@0.7mA
输入与内部逻辑电路的隔离方式	光电耦合器
输入与内部逻辑电路的隔离电压	1500VAC/1 分钟
状态指示	状态指示灯亮绿色/红色

DI 输入通道电气原理图：



晶体管型 DO 输出通道主要特点:

- 8 路晶体管输出通道，分成 1 组
- 额定供电电压为 DC24V
- 额定输出电压为 DC24V，每通道最大输出电流为 500mA，漏型（NPN）
- 供电电源接入极性保护
- 感性负载输出保护
- 短路保护（每路输出电流大于 500mA 时）
- 同一组内通道允许并联
- 输出与内部电路之间光电隔离

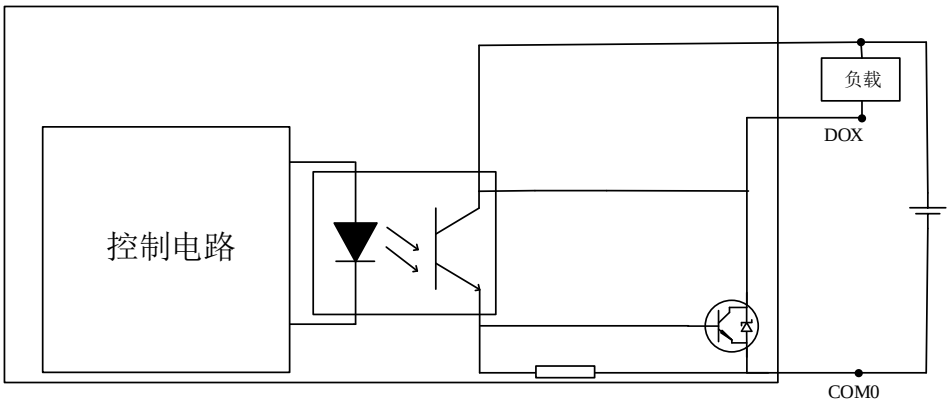
晶体管型 DO 通道技术参数

项目	规格
输出类型	漏型[注]
额定输出电压	24VDC
额定供电电压	24VDC

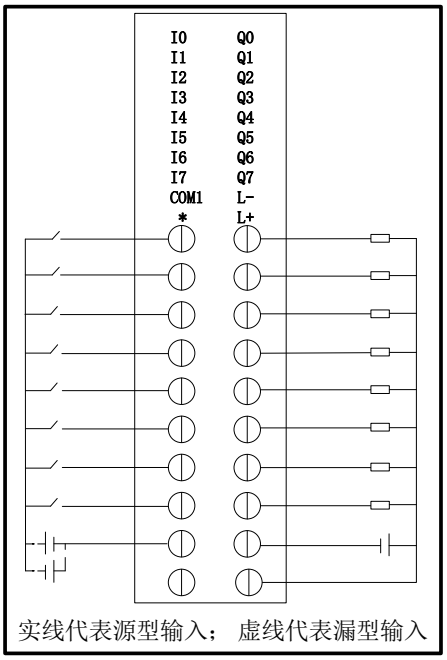
电源接入极性保护	有
每通道输出电流	最大 500mA @24VDC
输出漏电流	25μA（最大）
输出与内部逻辑电路的隔离方式	光电隔离
输出与内部逻辑电路的隔离电压	1500VAC/1 分钟
通道并联功能	有
短路保护功能	有
状态指示	状态指示灯亮绿色

注：漏型输出时无限流电阻

DO 输出通道电气原理图：



接线图



第三章　　软件功能介绍

本章主要介绍 EAC100 系列运动控制器的软件功能，包括可编程功能、Web 管理和 U 盘更新功能。

3.1 可编程功能

EAC100 以 CoDeSys 作为可编程环境，支持 6 种编程语言，涵盖梯形图、结构化文本等。此外，涵盖丰富的软件功能块，包括 EtherCAT、运动控制以及人机交互等。

3.2 Web 管理功能

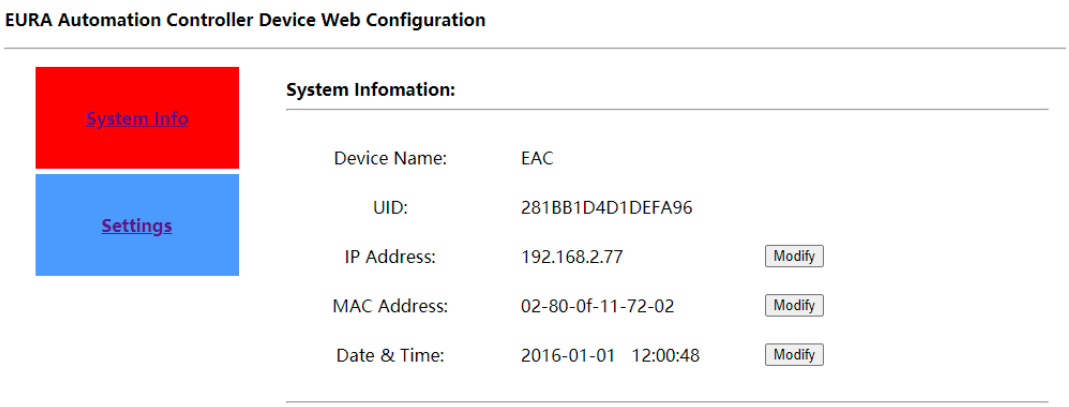
用户可以通过 PC 连接 EAC 设备，并通过浏览器对设备的系统时间、IP 地址等参数进行查看和修改。

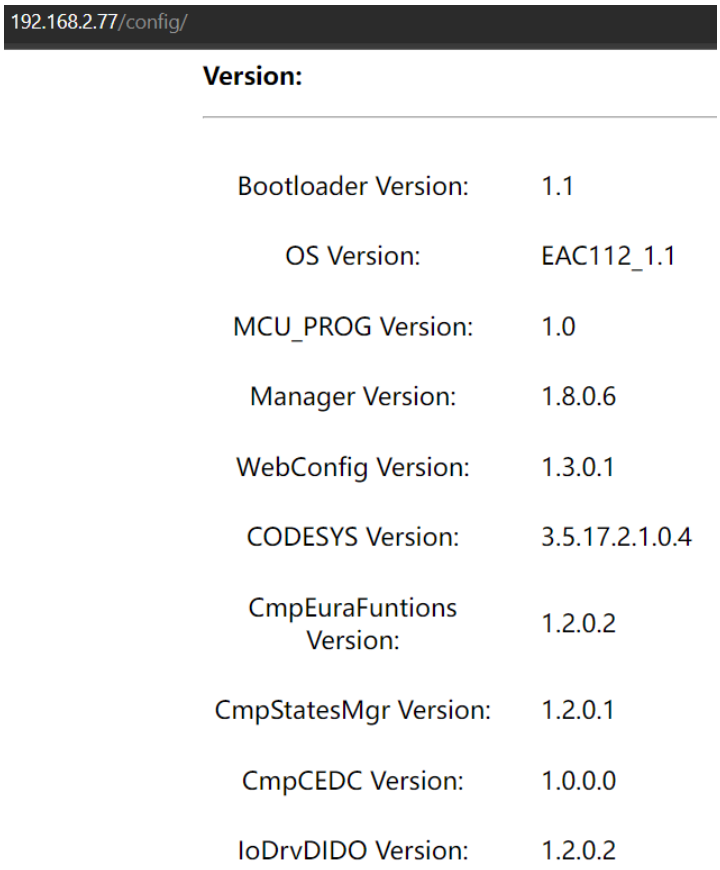
3.2.1 连接并进入设置

1. 使用网线连接 PC 机网口和设备的 ENET 网口；
2. 请确认 PC 机网口的 IP 地址与设备的 IP 地址在同一网段上，如不在同一网段请修改 PC 机 IP 地址。

注：设备默认 IP 地址为 192.168.2.77。如果无法确定设备当前 IP 地址，可通过将拨码开关的 DIP8 位置于 ON 状态来使设备的 IP 地址重置为默认 IP 地址。

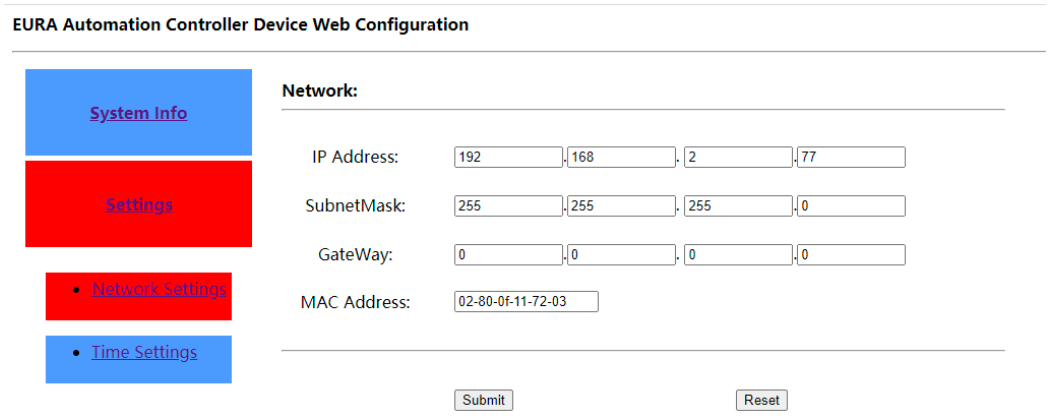
3. 打开浏览器，在浏览器地址栏中输入“设备 IP 地址/config”(默认为 192.168.2.77/config)，点击回车键进入网页设置，
4. 网页设置主界面如下图所示，主界面上包含控制器系统信息和控制器固件、软件的版本号，便于用户了解控制器固件、软件当前的升级情况。





3.2.2 修改 IP 地址/MAC 地址

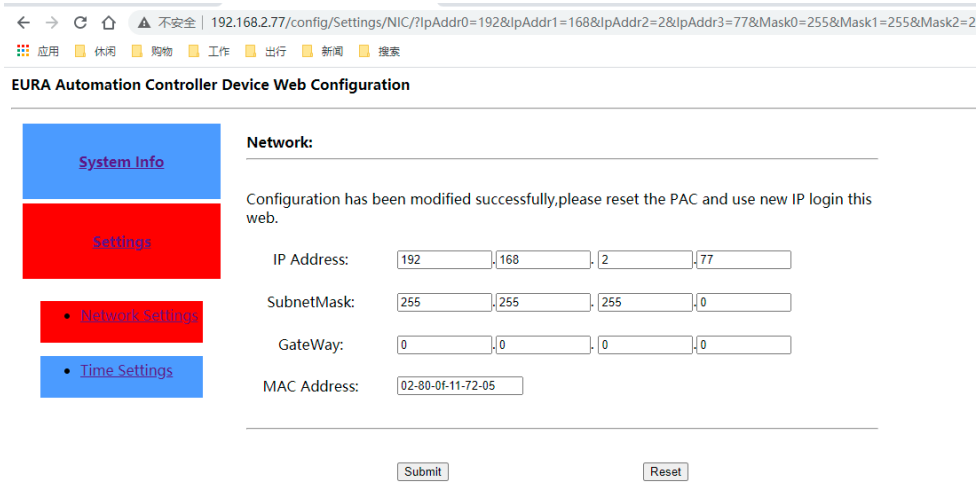
点击左侧的 Settings 按钮或者点击主页 IP 地址或 MAC 地址右侧的 Modify 按钮，可以进入网络设置页面，如下图所示。



修改完成后点击 Submit 按钮，成功修改后系统会提示重启设备使新地址设置生效，如下图所示。

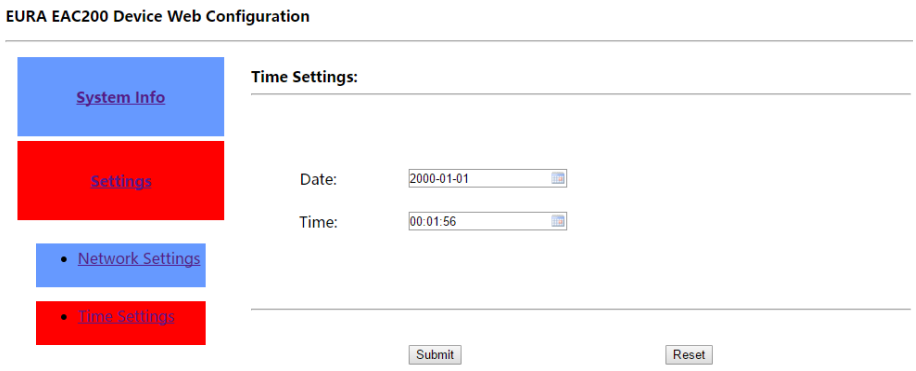
注 1：重启前请检查 IP 地址复位拨码（DIP8）是否置于 OFF 状态，否则新设置的 IP 地址会被还原；

注 2：重启后请使用新设置的 IP 登陆网页设置界面，参见 3.2.1.1 节。

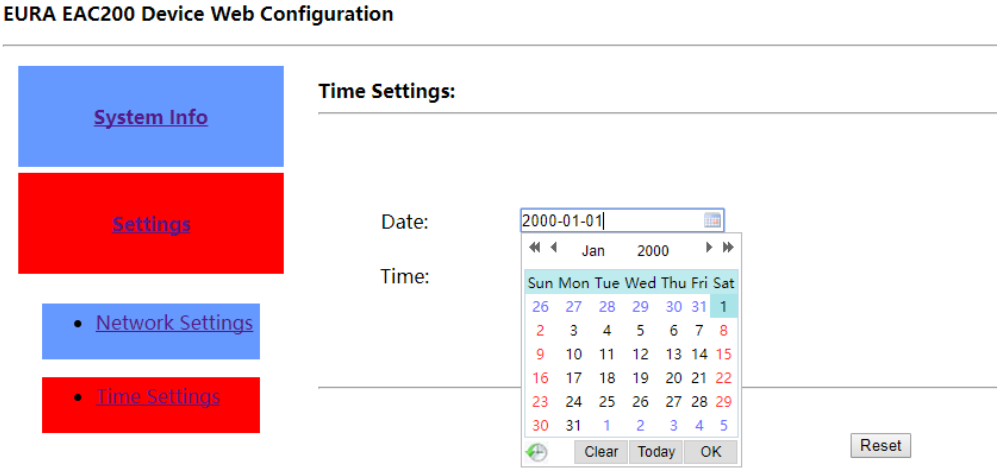


3.2.3 修改系统时间

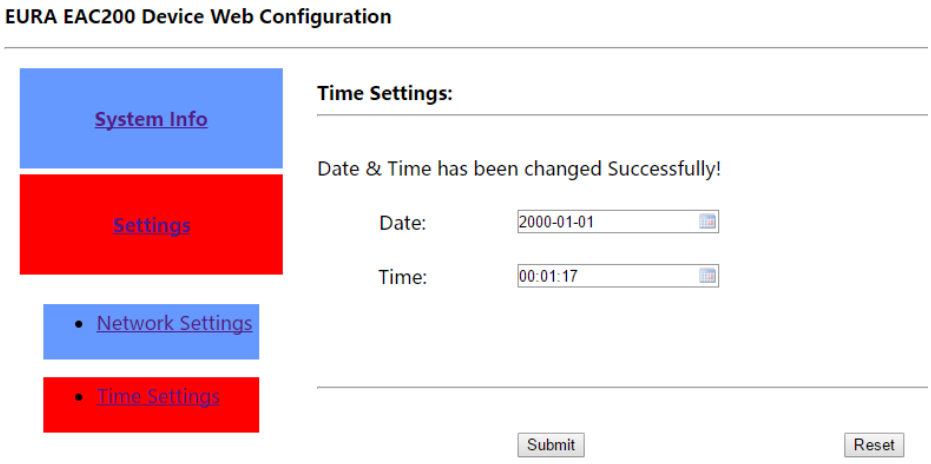
网页设置提供对系统时间的修改，点击时间右侧的 Modify 按钮或者点击左侧 Settings 再点击子菜单中的 Time Settings 即可进入时间设置。



点击日期或者时间文本框，会弹出日期/时间选择对话框，在对话框中选择需要的日期或者时间。



完成日期/时间设置后点击 Submit 提交，提交成功出现对应文字提示。



3.2 U 盘更新功能

U 盘更新功能用于对 EAC 设备中的软件进行更新或者重置。

使用时将需要更新的文件或文件夹放入 U 盘根目录，在设备断电状态下将 U 盘插入设备 USB 接口，上电后设备自动运行更新程序，数码管由原状态转为显示“3”。视更新文件大小及数目影响，更新时间为 1-5 秒，更新完成后重新读取 DIP 拨码开关状态并执行相应操作。

注 1：更新前请仔细检查 U 盘根目录下是否只包含需要更新的文件或文件夹，防止将设备中其他软件还原为旧版本。

U 盘更新文件如下表所示：

序号	程序或文件夹名称	说明
1	DeviceUpdate.exe	U 盘更新程序
2	Update	包含有更新文件和更新配置文件的文件夹

第四章　　软件编程向导

本章将介绍软件编程环境的建立以及功能应用。

4.1 软件编程环境建立

软件编程环境的建立步骤，包括如下 2 部分：

1. CoDeSys 编程软件安装。
2. 工程软件包导入。

4.1.1 CoDeSys 编程软件安装

CoDeSys 是德国 CoDeSys 公司推出的一款基于 IEC-61131 标准的工业自动化开发编程系统，利用 CoDeSys 软件可以快速实现对 EAC 设备的编程、程序下载、程序在线监控等功能。

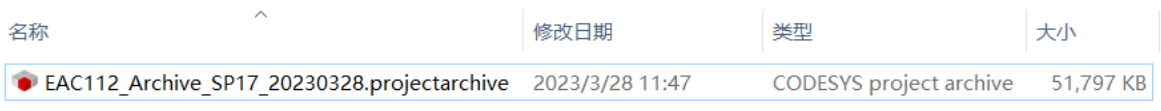
CoDeSys 的安装程序可以在 CoDeSys 官网进行下载或从本公司索取，推荐使用版本为 **V3.5 SP17**。

提示：同一台计算机可并存多个不同版本的 CoDeSys，因此无需卸载已安装的 CoDeSys 软件。

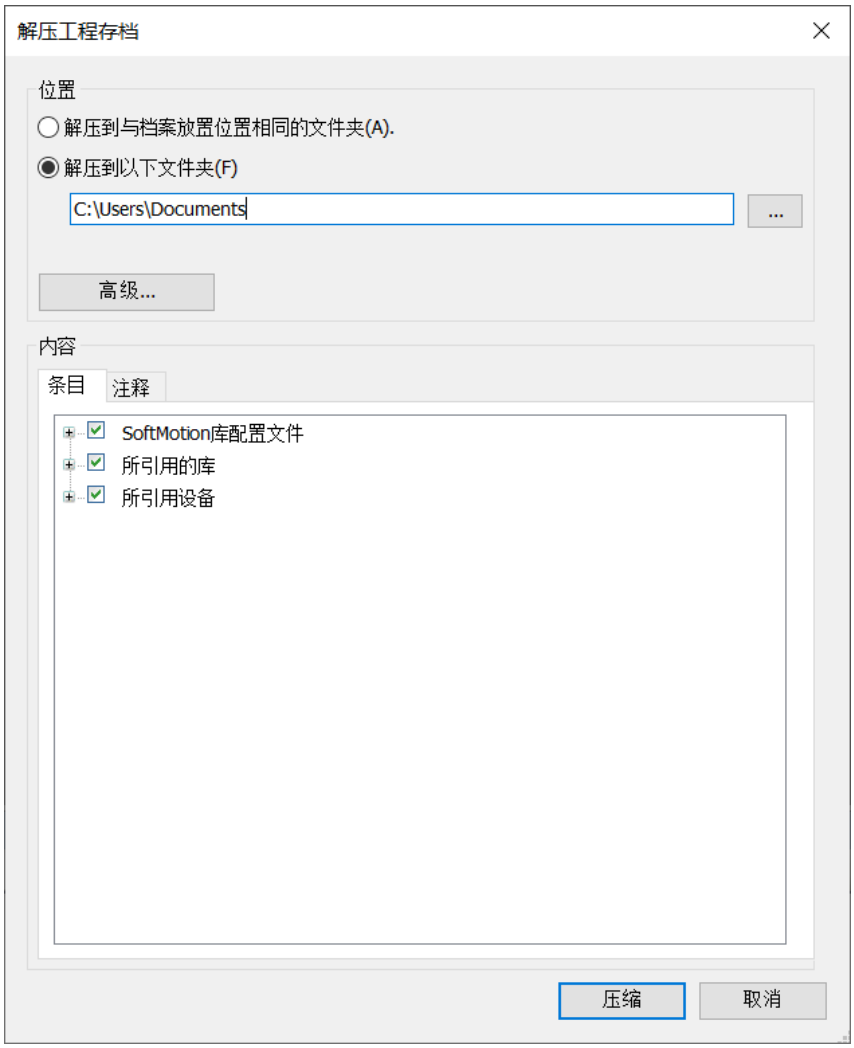
注意： 推荐使用 Windows 10 以上的版本安装 CODESYS v3.5 SP17。

4.1.2 工程软件包导入

工程软件包可以在官网进行下载，里面包含 EAC100 设备相关的设备描述文件和库文件，如下图所示。



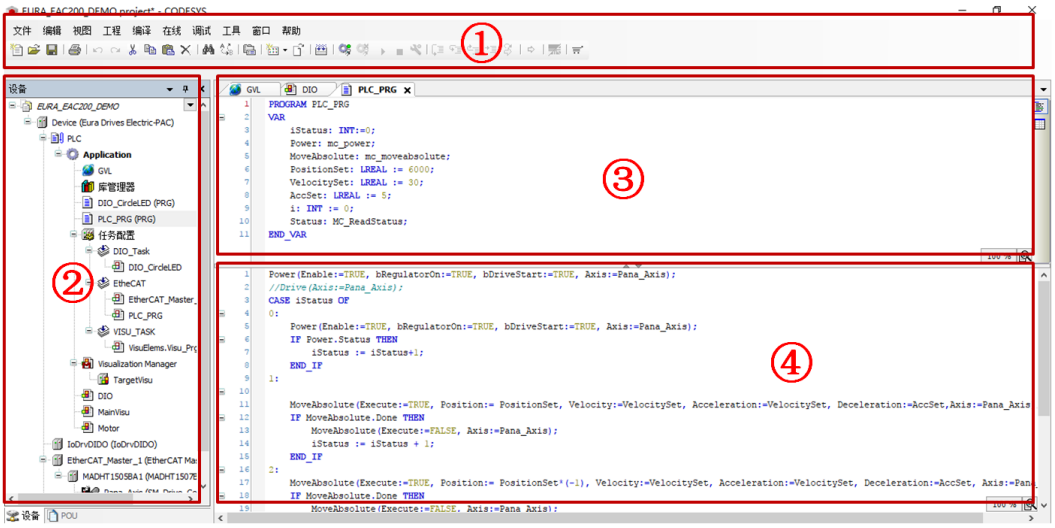
双击该文件之后，系统会调用 CoDeSys 打开该文件，并对内部的设备描述文件和库文件进行安装，如下图所示。



4.6 软件编程功能应用

本节将对 EAC112 的各个软件功能做入门介绍，具体的使用方法可以参考附录以及官网的例程。

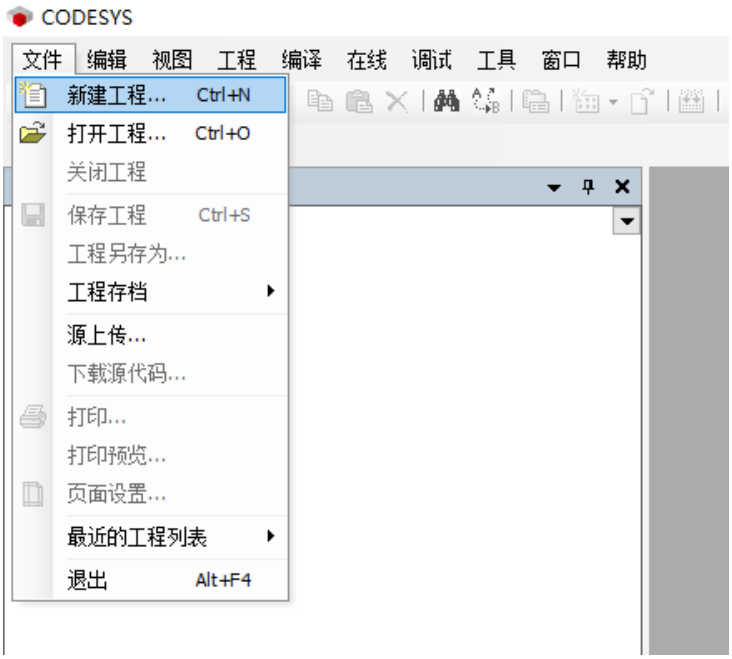
使用 CODESYS 软件打开工程之后，基本界面如下图所示，各部分功能如下表所述。



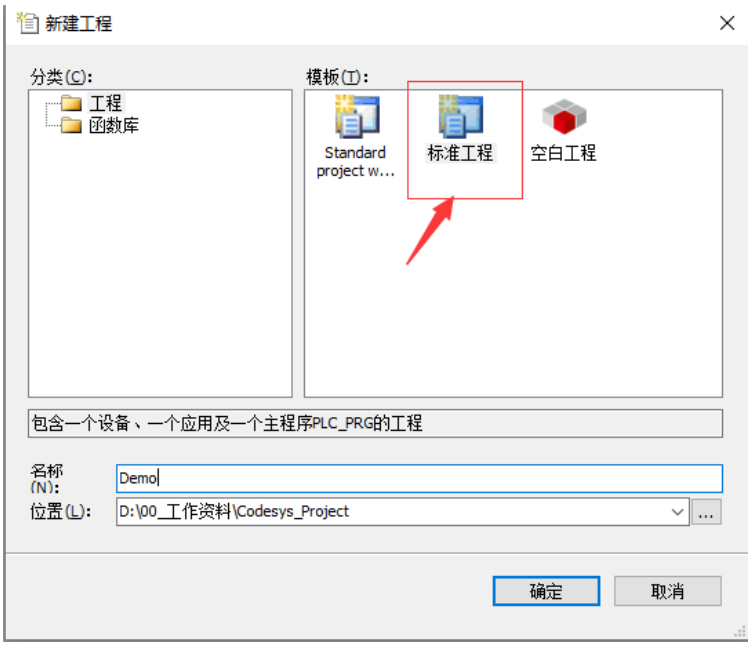
区域标号	说明
1	菜单栏和工具栏区，集成常用命令及操作
2	设备树区，显示整个工程的设备结构和程序组织结构
3	变量定义区，显示适用于该段程序中的变量定义
4	编程区，显示程序实现

3.3.7.1 工程创建、编译与下载调试

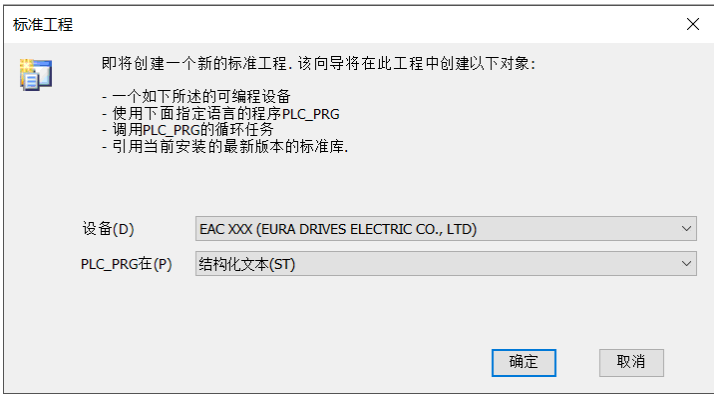
启动 CODESYS，在菜单栏中选择“文件”——“新建工程”。



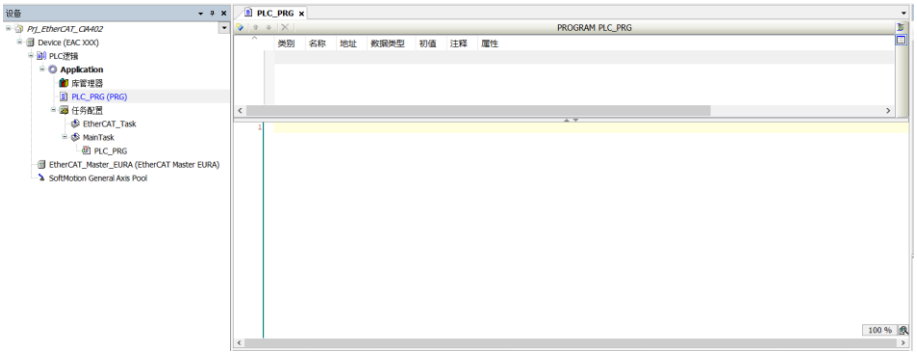
在弹出的对话框中选择“标准工程”，输入工程名称和存储路径，点击确定。



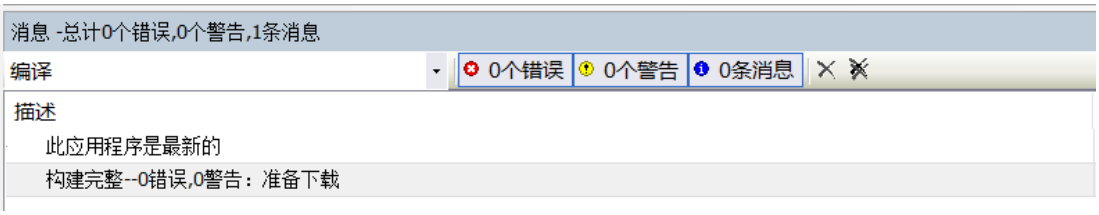
在弹出的对话框中设备一栏选择“EAC XXX(EURA DRIVES ELECTRIC CO.,LTD)”，主程序编程语言 PLC_PRG 选项选择“结构化文本（ST）”，点击确定按钮。



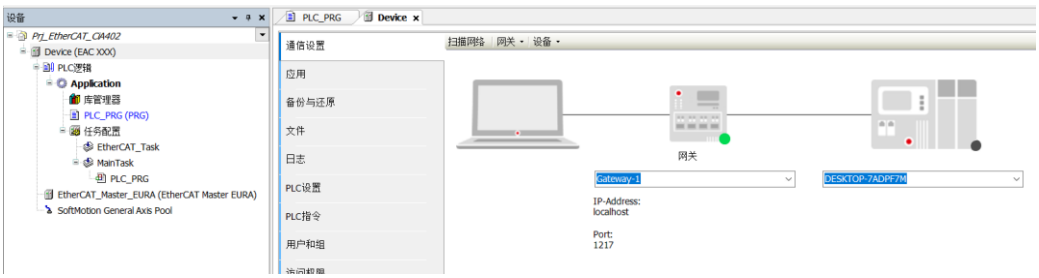
创建成功的工程，如下图所示：



对代码进行编译，编译成功，无错误和警告，如下图所示：

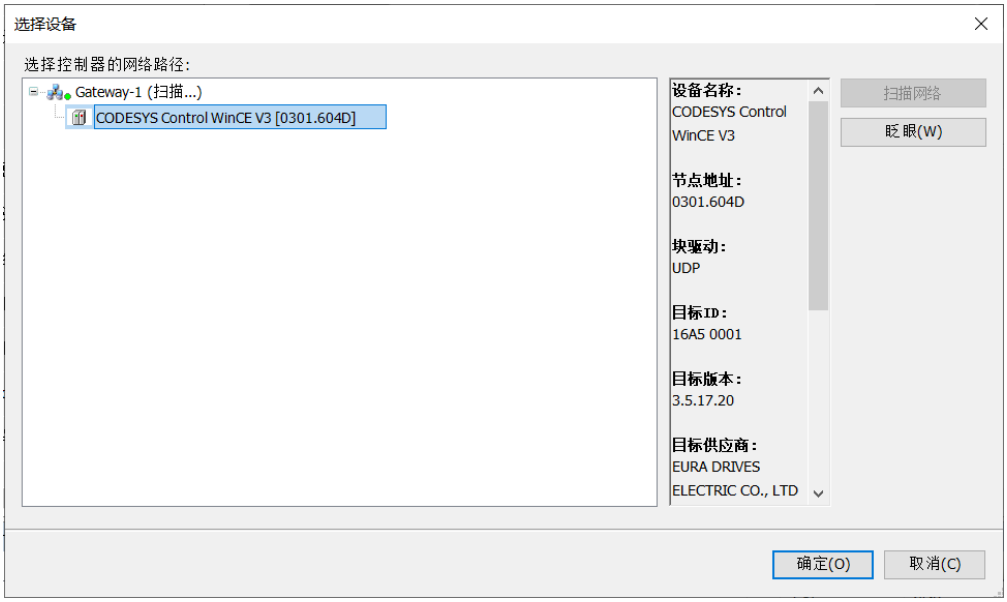


点击 Device(EAC XXX)，在通信设置选项卡中，点击扫描网络，如下图所示：



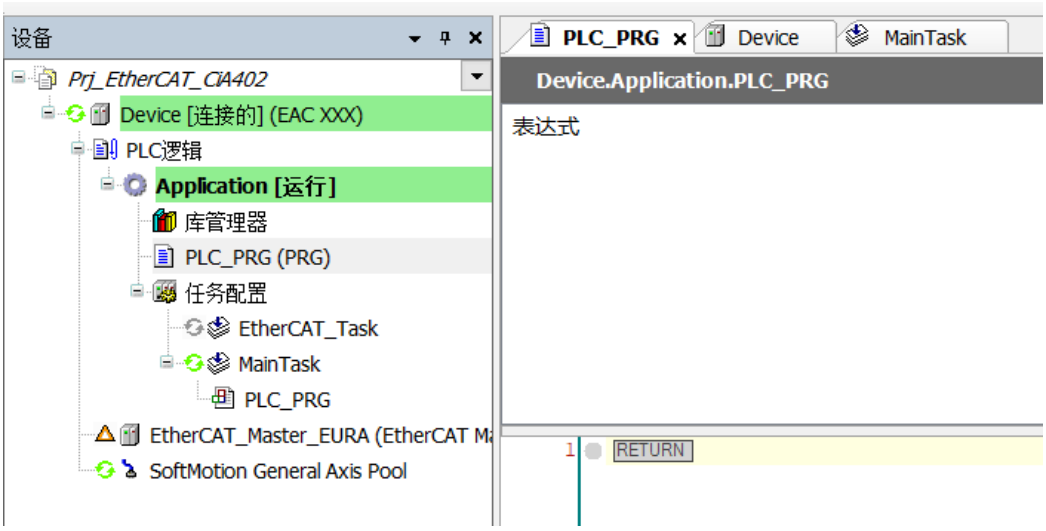
在弹出的扫描网络对话中，找到已发现的匹配设备，选择设备并点击确定。

注意：当前以太网的 IP 地址与目标设备的 IP 地址位于同一网段内。



在选定设备之后，点击登录按钮，即可登录到目标设备并自动下载当前工程的代码。

下载完成之后，点击运行，则当前工程中的代码即在目标设备中运行，如下图所示：

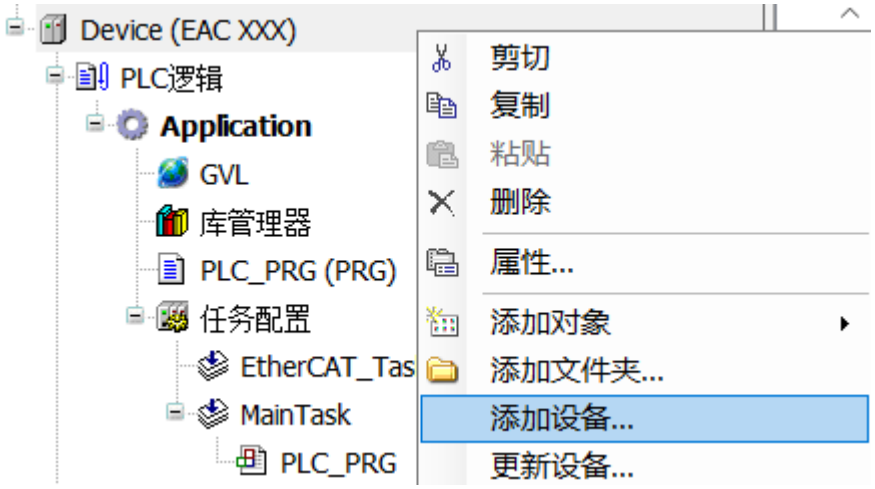


3.3.7.2 拓展数字量 I/O（DIO）

不同型号的控制器带有不同数量的拓展数字量 I/O。在编程时，根据实际使用的控制器型号，向工程中添加匹配的 I/O 模块，然后建立变量映射以供软件编程使用。

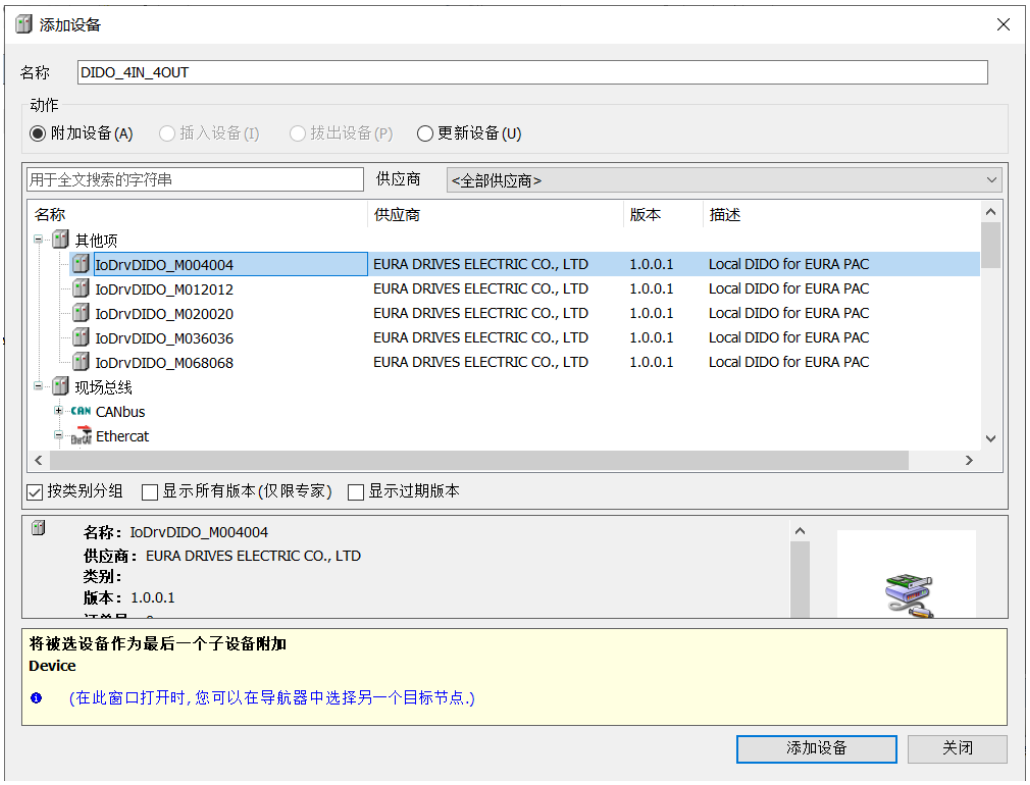
添加设备：

右击设备树“Device”节点，在右键菜单中选择“添加设备”，在弹出的对话框中的供应商选项中选择“EURA DRIVES ELECTRIC CO., LTD”后，根据实际应用的扩展模块选择相应的设备，点击添加设备，如下图所示。



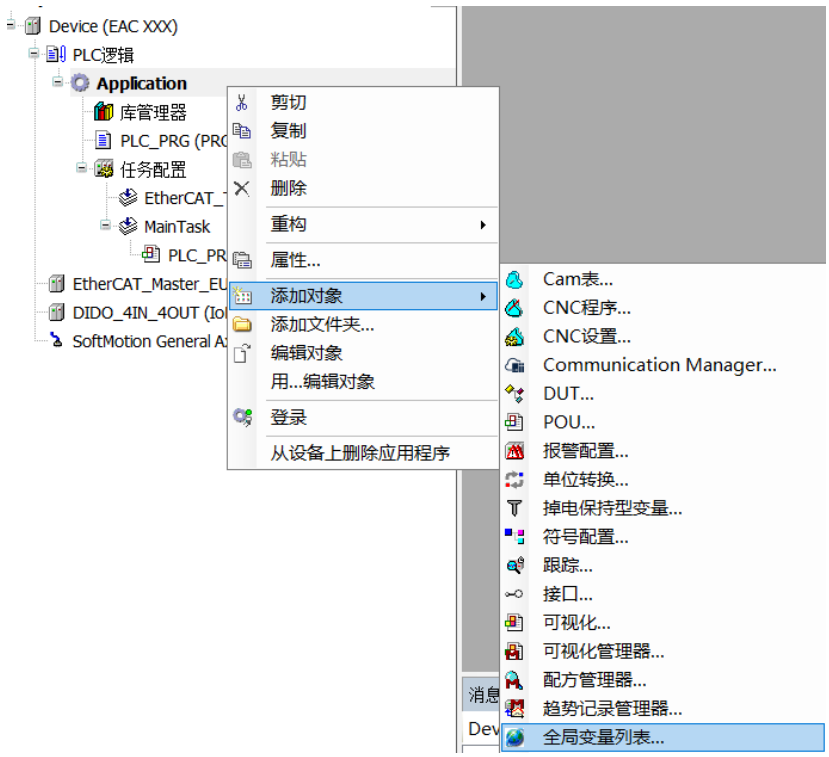
选择设备：

从弹出的对话框中，选择对应的 I/O 设备，这里以具有 4 个 DI 和 4 个 DO 的控制器为例，选择 IoDrvDIDO_M004004，点击确定以将该设备添加到工程中。

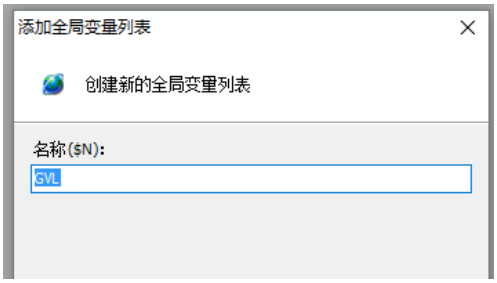


添加全局变量：

创建全局变量用于映射到新添加的设备（DIDO_4IN_4OUT）中，通过在右侧设备树选项卡中的“Application”节点右击，在弹出的菜单中选择“全局变量列表”选项。

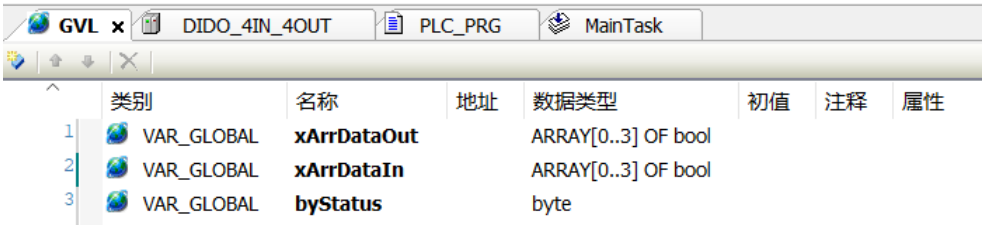


填写全局变量列表名称，点击打开。



编辑全局变量：

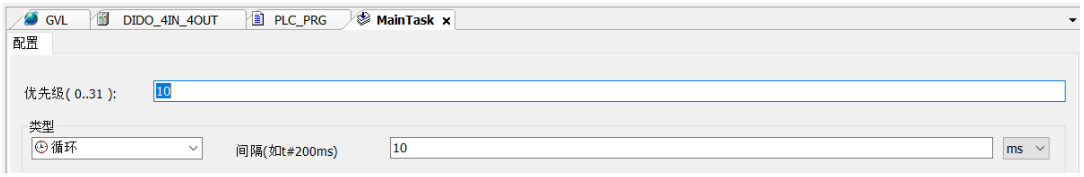
在右侧的设备树中找到建立的全局变量列表，双击打开编辑器，编辑变量区代码，如下图所示。



创建和编辑设备关联任务：

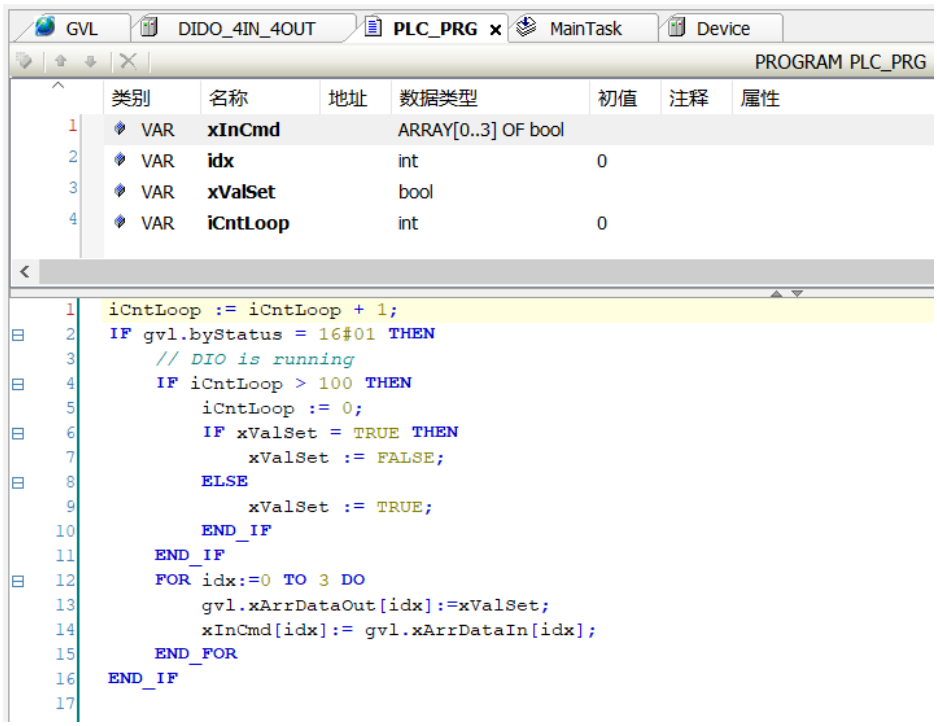
每种设备都需要关联一个指定的任务，用于明确访问的频率和时间等，可以通过创建一个任务或者修改一个任务属性，以使其符合该设备对任务的需求。

这里通过修改工程中自带的 MainTask 为例，修改其任务类型、优先级和周期（间隔），按照下图所示设置参数：



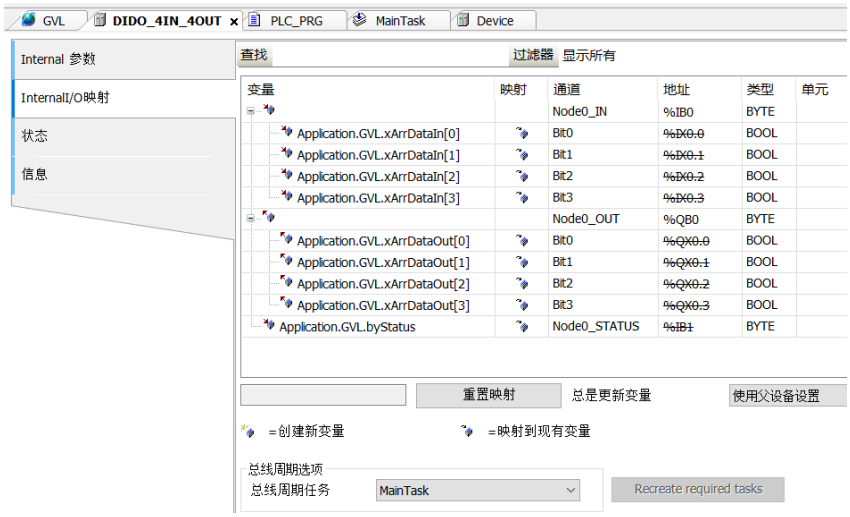
创建和编辑任务内的程序：

任务中的程序内容定义了对设备的操作逻辑，这里通过修改 MainTask 内的 PLC_PRG，实现对功能中能。如下图所示，该代码在检测到 DIO 进入正常启动状态之后，实现闪烁控制，周期为 1s，同时可以采集到 DI 信号的变化。



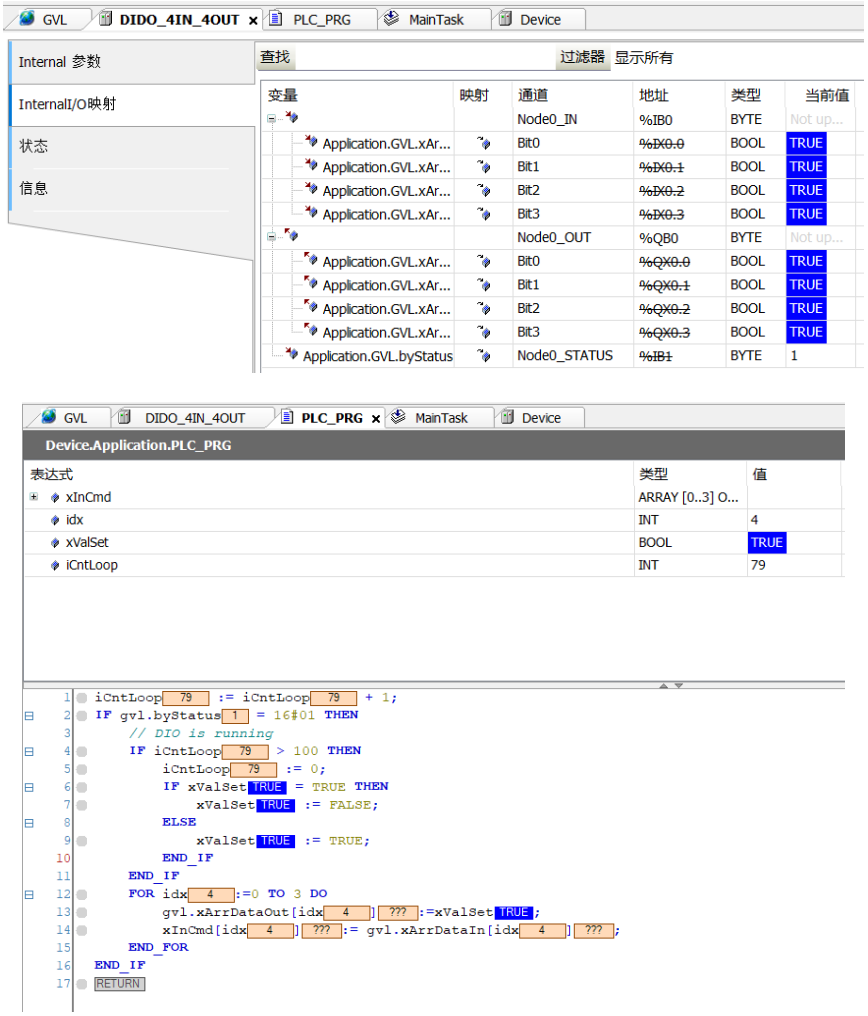
设备的变量与任务映射：

打开新添加的设备（DIDO_4IN_4OUT），添加变量映射和任务关联，如下图所示。



编译、下载和调试：

对工程进行编译，在编译成功之后，登录到设备将工程代码下载到目标设备中，然后使程序运行以在线调试，如下图所示。



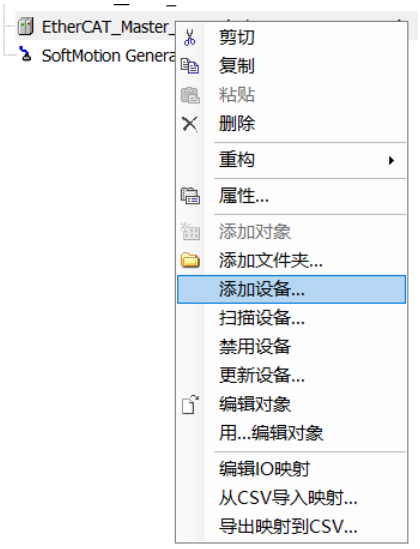
3.3.7.3 EtherCAT

本节将介绍如何创建 EtherCAT 主站，并通过 CiA402 协议控制 EtherCAT 型伺服驱动器运行。

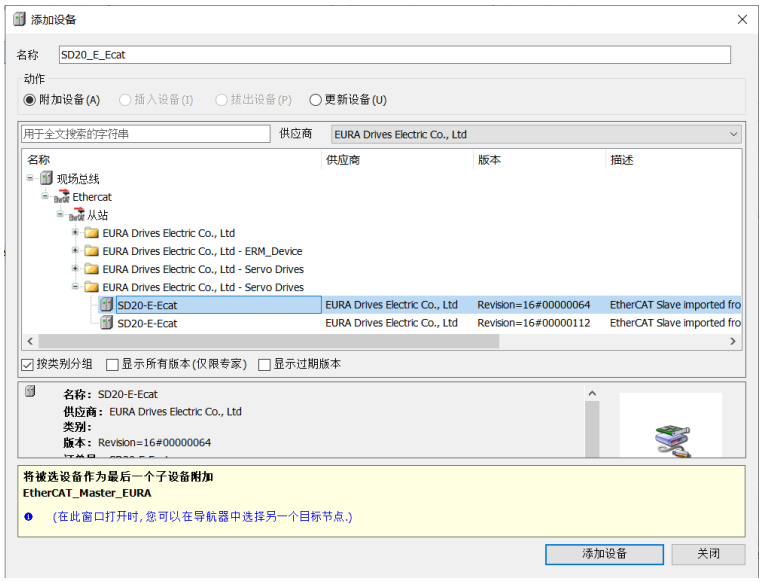
在创建工程时，已经自动添加 EtherCAT 主站设备，如下图所示。

添加 EtherCAT 从站：

这里以 SD20-E 型 EtherCAT 交流伺服驱动器为例，右击 EtherCAT_Master_EURA，点击添加设备，如下图所示



从弹出的对话框中，选择版本匹配的 SD20-E 设备，如下图所示。

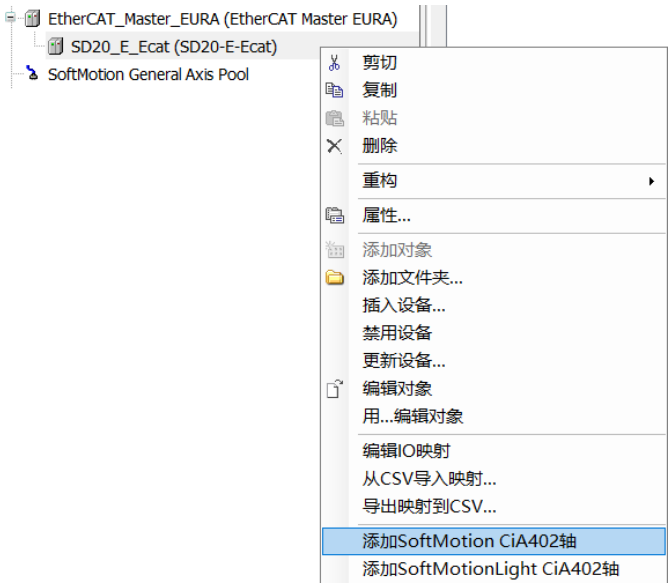


修改 EtherCAT 从站配置：

如下图所示，开启从站的分布式时钟（DC-Synchron）。

添加 CiA 402 轴：

向新添加的 EtherCAT 从站设备（SD20_E_Ecat）中添加 CiA 402 轴，用于在 CoDeSys Softmotion 中使用，如下图所示。



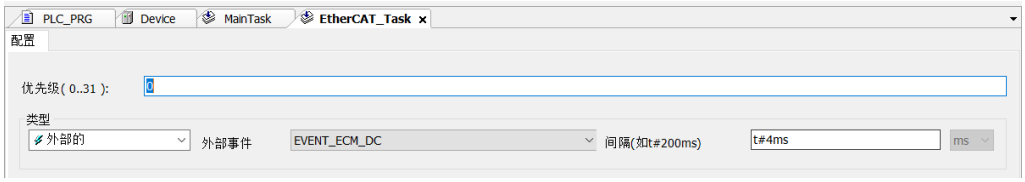
配置 CiA 402 轴：

修改新添加的 CiA 402 轴的名称为 AxisCtrl 以便于编程，同时修改缩放比例，这里使用的比例为 1:1:1，如下图所示。

创建与配置 EtherCAT 任务：

在工程创建时，已经创建一个 EtherCAT 任务，根据实际需求修改该任务的属性。如下图所示，任务类型为外部的（精密定时器任务），外部事件为 EVENT_ECM_DC，周期为 4ms，优先级为 0（最高优先级）。

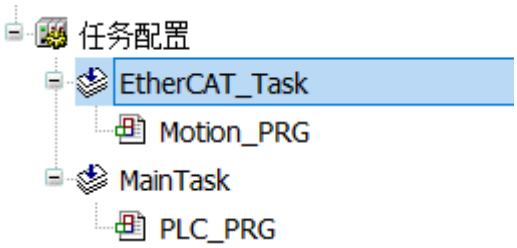
注意：任务周期应与 EtherCAT 主站配置的周期值一致



创建与编写运动控制代码：

向工程中添加运动控制程序，通过右击添加 POU 的方式，添加一个名为 Motion_PRG 的程序，如下图所示。

将 Motion_PRG 添加到 EtherCAT_Task 中，如下图所示：



Motion_PRG 中的代码如下图所示，使用 MC_JOG 功能块控制 CiA402 轴执行正反转运行。

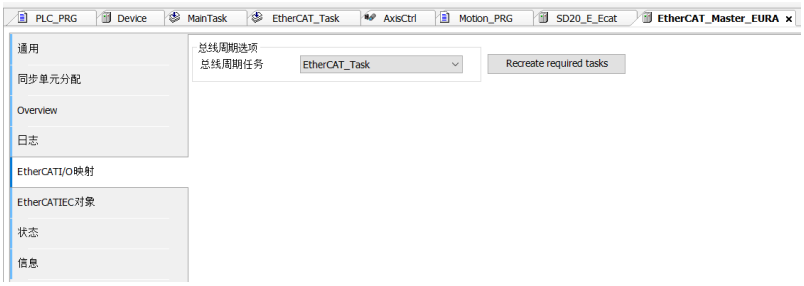
PLC_PRG | AxisCtrl | Motion_PRG | EtherCAT_Master_EURA

```
VAR
    fbPower:mc_power;
    fbJog:mc_jog;
    xEnPwr:BOOL:=TRUE;
    xStPwr:BOOL:=FALSE;
    xFwCmd:BOOL:=FALSE;      // 正转
    xBwCmd:BOOL:=FALSE;      // 反转
    lfVel:LREAL:=1000.0;
    lfAcc:LREAL:=1000.0;
END_VAR
```

```
fbPower(
    Axis:= AxisCtrl,
    Enable:= xEnPwr,
    bRegulatorOn:= xEnPwr,
    bDriveStart:= xEnPwr,
    Status=> xStPwr,
    bRegulatorRealState=> ,
    bDriveStartRealState=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
IF xStPwr THEN
    xFwCmd := gvl.xFwIn;
    xBwCmd := gvl.xBwIn;
ELSE
    xFwCmd := FALSE;
    xBwCmd := FALSE;
END_IF
fbJog(
    Axis:= AxisCtrl,
    JogForward:= xFwCmd,
    JogBackward:= xBwCmd,
    Velocity:= lfVel,
    Acceleration:= lfAcc,
    Jerk:= ,
    Deceleration:= lfAcc,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorId=> );
gvl.lfPosRead := AxisCtrl.fActPosition;
```

至此，工程编译完毕，在成功编译之后下载到目标设备上调试运行。

注意：务必保持 EtherCAT 主站关联到正确的任务，如下图所示，关联到本工程中的 EtherCAT_Task 任务中。



3.3.7.4 Modbus RTU/TCP

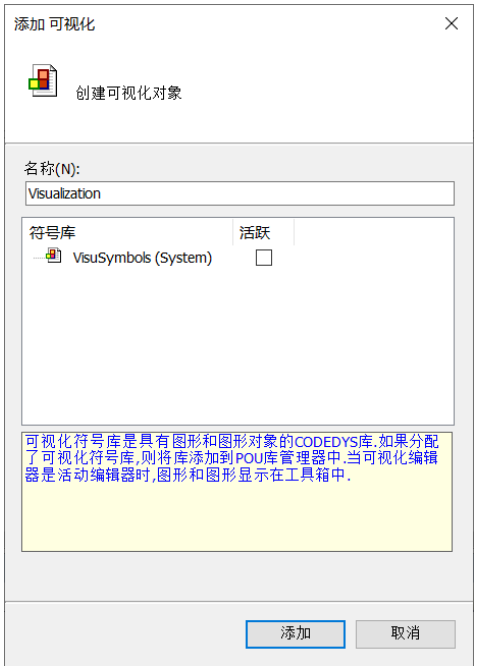
EAC112 支持 Modbus RTU/TCP 主站（客户端）和从站（服务端），具体的使用方法，请参照附录 2。

3.3.7.5 可视化功能

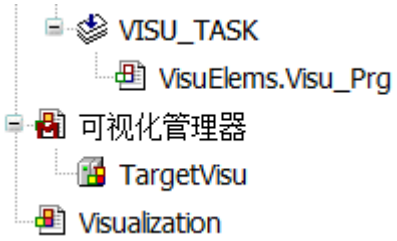
可视化功能，用于做人机交互，通过 EAC112 的 HDMI 接口连接显示器，将显示内容呈现给用户，同时可以连接触摸板、鼠标和键盘等用户输入设备。

向工程中添加视图：

在右侧设备树选项卡 Application 节点右击，选择“添加对象”—“视图”，如下图所示。



添加视图之后，同时会添加一个可视化管理器和 VISU_TASK 的任务。可视化管理器，用于配置视图的显示样式，而 VISU_TASK 定义了视图的更新周期。

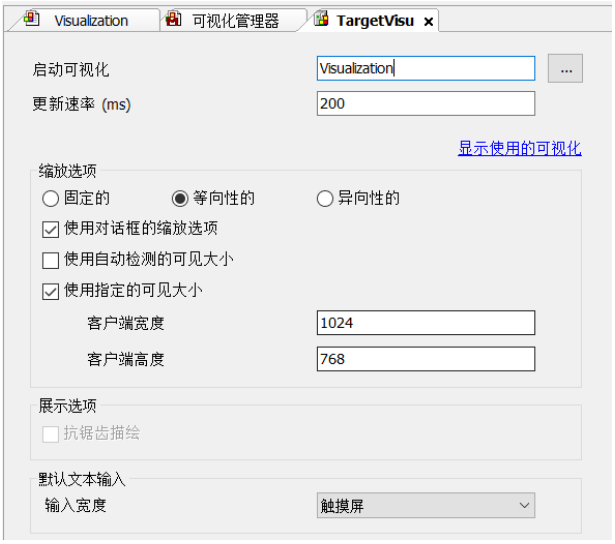


中文显示支持：

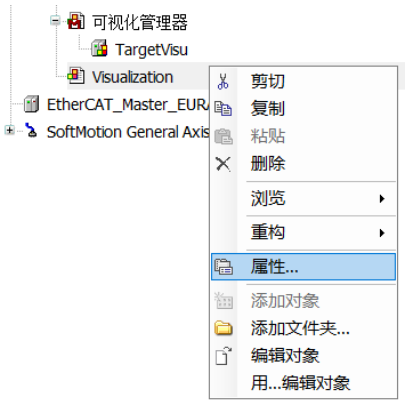
双击可视化管理器，开启“使用 Unicode”字符串以支持中文显示。如下图所示，使用 Basic Style 作为显示样式。

修改显示尺寸：

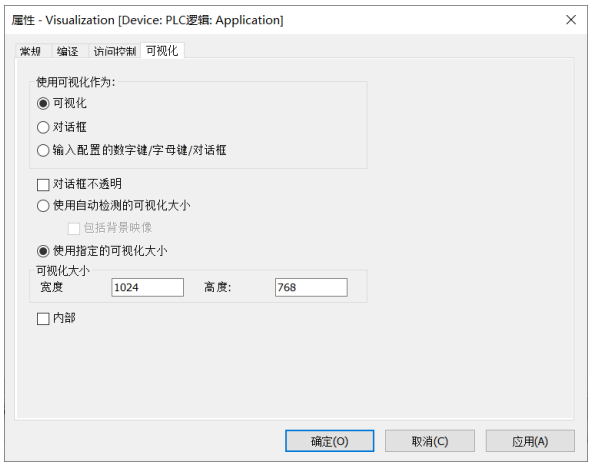
打开 TargetVisu，按照下图所示修改显示尺寸，分辨率为 1024×768。



右击 Visualization，点击属性，如下图所示。

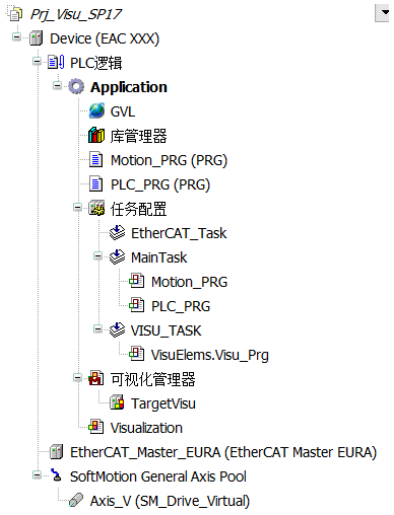


在打开的对话框中，切换到可视化选项卡，按照下图配置可视化尺寸 1024×768。

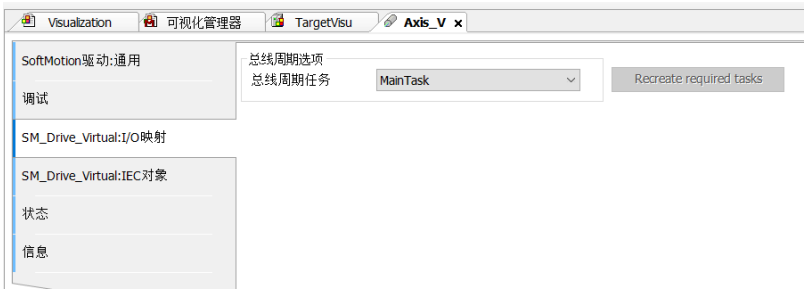


编辑程序代码：

本部分将创建一个虚拟的运动控制程序和 I/O 控制程序，工程组态如下入所示：

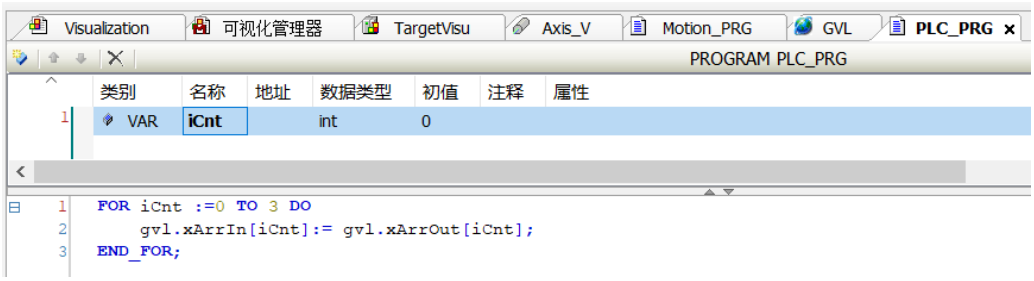


修改 Axis_V 设备关联的任务为 MainTask



GVL 的数据区，如下图所示：

PLC_PRG 的代码，如下图所示，模拟将输出信号接入到输入信号。



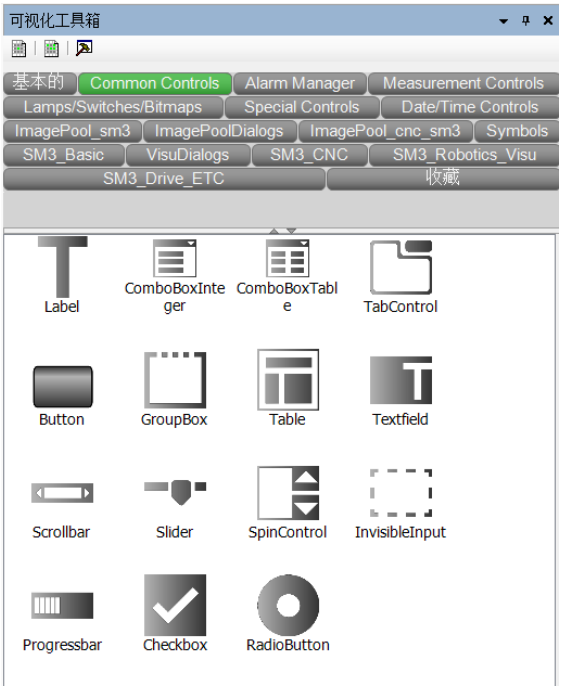
Motion_PRG 的代码，如下图所示，用于控制虚拟轴（Axis_V）正反转。

```
PROGRAM Motion_PRG
VAR
    fbPower:mc_power;
    fbJog:mc_jog;
    xEnPwr:BOOL:=TRUE;
    xStPwr:BOOL:=FALSE;
    xFwCmd:BOOL:=FALSE;      // 正转
    xBwCmd:BOOL:=FALSE;      // 反转
    lfVel:LREAL:=1000.0;
    lfAcc:LREAL:=1000.0;
END_VAR
```

```
fbPower(
    Axis:= Axis_V,
    Enable:= xEnPwr,
    bRegulatorOn:= xEnPwr,
    bDriveStart:= xEnPwr,
    Status=> xStPwr,
    bRegulatorRealState=> ,
    bDriveStartRealState=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
IF xStPwr THEN
    xFwCmd := gvl.xFwIn;
    xBwCmd := gvl.xBwIn;
ELSE
    xFwCmd := FALSE;
    xBwCmd := FALSE;
END_IF
fbJog(
    Axis:= Axis_V,
    JogForward:= xFwCmd,
    JogBackward:= xBwCmd,
    Velocity:= lfVel,
    Acceleration:= lfAcc,
    Deceleration:= lfAcc,
    Jerk:= ,
    Busy=> ,
    CommandAborted=> ,
    Error=> ,
    ErrorId=> );
gvl.lfPosRead := axis_v.fActPosition;
```

编辑可视化界面：

通过可视化工具箱，向 Visualization 页面文件中添加图形元素，如下图所示。

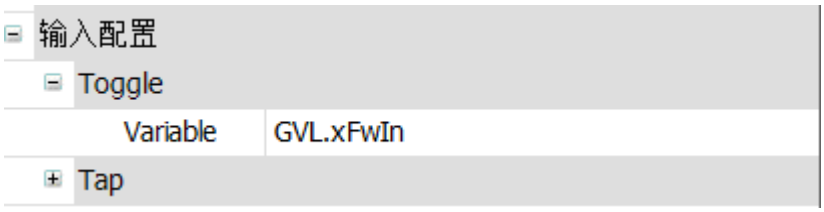


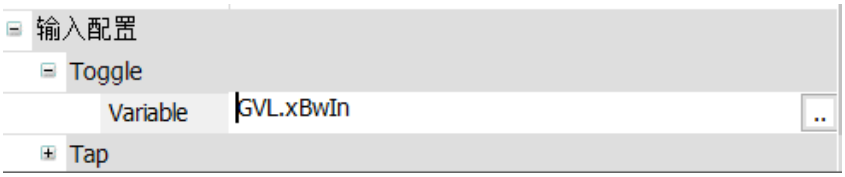
图形元素列表和位置，如下图所示。

类型	X	Y	宽度	高度	ID	名称	访...	选项卡命令
#0 Button	52	77	150	30	0	Gen...		0
#1 Button	53	143	150	30	1	Gen...		1
#2 Label	270	69	172	49	2	Gen...		2
#3 Textfield	280	143	150	30	3	Gen...		3
#4 Lamp1	55	208	70	70	4	Gen...		4
#5 Lamp1	194	208	70	70	5	Gen...		5
#6 Lamp1	324	208	70	70	6	Gen...		6
#7 Lamp1	453	208	70	70	7	Gen...		7
#8 DipSwitch	55	305	70	70	8	Gen...		8
#9 DipSwitch	194	305	70	70	9	Gen...		9
#10 DipSwitch	324	305	70	70	10	Gen...		10
#11 DipSwitch	453	305	70	70	11	Gen...		11

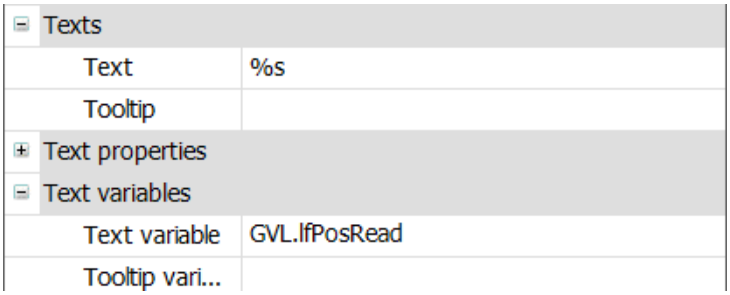
图形元素映射变量：

Button 图形元素，用于控制电机正反转，映射变量见下图。





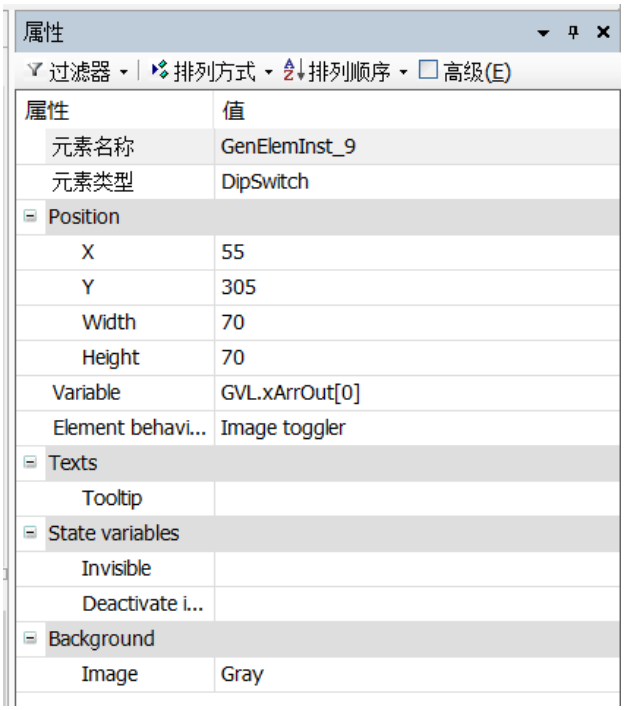
Textfield 图形元素，用于显示实际的电机位置，变量映射见下图。



Lamp 图形元素，用于指示接收的输入信号，变量映射见下图。

属性	值
元素名称	GenElemInst_5
元素类型	Lamp1
Position	
X	55
Y	208
Width	70
Height	70
Variable	GVL.xArrIn[0]
Texts	
Tooltip	
State variables	
Invisible	
Background	
Image	Yellow

DipSwitch 图形元素，用于模拟输出信号，变量映射见下图。



属性	值
元素名称	GenElemInst_9
元素类型	DipSwitch
Position	
X	55
Y	305
Width	70
Height	70
Variable	GVL.xArrOut[0]
Element behavi...	Image toggler
Texts	
Tooltip	
State variables	
Invisible	
Deactivate i...	
Background	
Image	Gray

编译和调试：

完成以上编辑操作，在成功编译程序之后，下载到目标设备中运行调试。

第五章 故障分析与解决

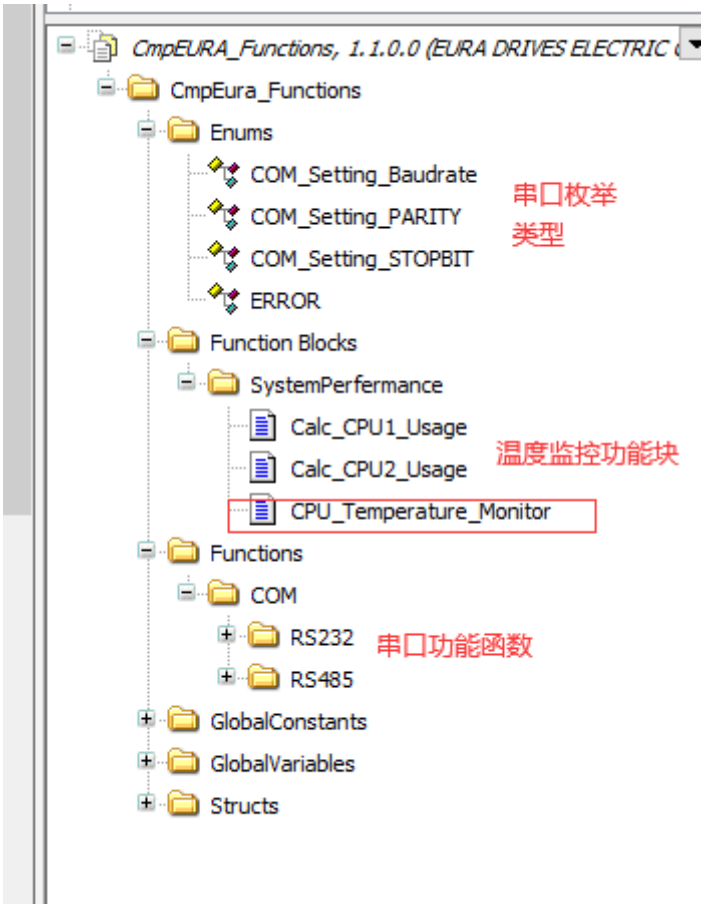
序号	现象	数码管显示	指示灯	原因	解决方案
1	系统上电后不启动	无显示	无显示	供电电源无电压	断电检查外部电源
2		无显示	PWR 红灯常亮	电源接反	检查电源极性
3	系统启动后不进入 CODESYS	2	PWR 绿灯常亮 ERR 灯不亮	DIP 拨码设置为不自动启动 CODESYS	根据需求设置拨码后重启
4		d	ERR 红灯常亮	因 RTC 电池掉电或其他原因导致系统时钟复位（系统年时间<2016）	可通过网页配置重新设置时间，重启后仍提示该错误请更换电池，可以通过拨码设置屏蔽该错误
5		E	ERR 红灯常亮	系统欠压	断电检查外部电源，修复后上电
6	设备程序下载后不运行或运行过程突然停止	A	ERR 红灯常亮	程序中存在死循环等导致系统卡死的代码	断电重启，并使用拨码开关清除设备内现有程序，重新下载修改后的程序
7		b	ERR 红灯常亮	EtherCAT 总线通讯中断或者出错	检查网线是否存在松动，重新连接后重启
8		C	ERR 红灯常亮	程序或通讯中出现异常导致 CODESYS 退出	断电重启，并使用拨码开关清除设备内现有程序，重新下载修改后的程序
9		E	ERR 红灯常亮	系统欠压	断电检查外部电源，修复后重新上电
10		F	ERR 红灯常亮	系统掉电	检查外部电源，修复后重新上电
11	系统正常运行时	h 闪烁	ERR 红灯闪烁	高温预警	
12		H	ERR 红灯常亮	运动控制器芯片温度过高	
13	上电后扩展 IO 模块无效	N/A	扩展 IO 模块 STAT 灯红色闪烁	本体与扩展 IO 通讯异常	重新上电

附录 1：CmpEuraFunctions 组件库说明：

CmpEuraFunctions 组件库用于提供欧瑞自己的函数和功能块实现，目前实现的功能有以下两种：

1.串口自由通讯相关枚举及功能函数。

2.系统温度监控功能块。



附录 1.1 串口自由通讯相关枚举及功能函数

枚举数据定义：

COM_Setting_Baudrate

说明：该枚举类型用于提供串口波特率值的选择

枚举值：

名称	值
CBR_4800	4800
CBR_9600	9600
CBR_19200	19200
CBR_38400	38400
CBR_57600	57600
CBR_115200	115200

COM_Setting_PARITY

说明：该枚举类型用于提供串口校验类型的选择

枚举值：

名称	值	说明
NOPARITY	0	无校验
ODDPARITY	1	奇校验
EVENPARITY	2	偶校验

COM_Setting_STOPBIT

说明：该枚举类型用于提供串口停止位类型的选择

枚举值：

名称	值	说明
ONESTOPBIT	0	1 位停止位
TWOSTOPBITS	2	2 位停止位

结构体数据定义：

COM_Setting

说明：该结构体整合串口设置需要的波特率、数据位、停止位和校验类型，用于提供给功能块进行串口配置。

结构成员：

名称	类型	说明
Baudrate	COM_Setting_Baudrate	用于给定串口的波特率
Parity	COM_Setting_PARITY	用于给定串口的校验类型
Databits	DINT	用于给定串口的数据位
StopBit	COM_Setting_STOPBIT	用于给定串口的停止位

函数定义：

OpenCom_RS232

说明：该函数用于打开 RS232 串口。

返回值：TRUE 表示串口打开成功，FALSE 表示打开失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
Result	输出	UDINT	返回串口失败代码，0 表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

CloseCom_RS232

说明：该函数用于关闭 RS232 串口。

返回值：TRUE 表示串口关闭成功，FALSE 表示关闭失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
Result	输出	UDINT	返回串口失败代码，0 表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

SetCom_RS232

说明：该函数用于对 RS232 串口参数进行设置。

返回值：TRUE 表示串口设置成功，FALSE 表示关闭失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
ComSetting	输入	COM_Setting/	用于串口通讯参数的给定
Result	输出	UDINT	返回串口失败代码，0 表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

SendCom_RS232

说明：该函数用于 RS232 串口向外发送数据。

返回值：TRUE 表示数据发送成功，FALSE 表示发送失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
pbyWriteBuffer	输入	POINTER TO BYTE	数据指针，指向存有需要发送数据的字节型数组。
ulNumOfBytesToSend	输入	UDINT	需要发送的字节数
ulActualBytesSend	输出	UDINT	实际发送的字节数
Result	输出	UDINT	返回串口失败代码，0 表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

注：当串口连接的设备对于接受到数据有返回时，发送函数 SendCom_RS232 和接收函数 ReceiveCom_RS232 应成对使用，或者延时足够长时间后才能再次调用发送函数，否则发送数据和和设备返回数据形成干扰，无法正常通信。

RecevieCom_RS232

说明：该函数用于接收 RS232 串口获得的数据。

返回值：TRUE 表示数据接收成功，FALSE 表示接收失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
pbyReadBuffer	输入	POINTER TO BYTE	数据指针，指向需要存放接收数据的字节型数组。
ulNumOfBytesToRecv	输入	UDINT	需要接收的字节数
ulActualBytesRecv	输出	UDINT	实际接收的字节数
Result	输出	UDINT	返回串口失败代码，0 表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

OpenCom_RS485

说明：该函数用于打开 RS485 串口。

返回值：TRUE 表示串口打开成功，FALSE 表示打开失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
Result	输出	UDINT	返回串口失败代码，0 表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

CloseCom_RS485

说明：该函数用于关闭 RS485 串口。

返回值：TRUE 表示串口关闭成功，FALSE 表示关闭失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
Result	输出	UDINT	返回串口失败代码，0 表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

SetCom_RS485

说明：该函数用于对 RS485 串口参数进行设置。

返回值：TRUE 表示串口设置成功，FALSE 表示关闭失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
ComSetting	输入	COM_Setting	用于串口通讯参数的给定
Result	输出	UDINT	返回串口失败代码，0 表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

SendCom_RS485

说明：该函数用于 RS485 串口向外发送数据。

返回值：TRUE 表示数据发送成功，FALSE 表示发送失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
pbyWriteBuffer	输入	POINTER TO BYTE	数据指针，指向存有需要发送数据的字节型数组。
ulNumOfBytesToSend	输入	UDINT	需要发送的字节数
ulActualBytesSend	输出	UDINT	实际发送的字节数
Result	输出	UDINT	返回串口失败代码，0 表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

注：当串口连接的设备对于接受到数据有返回时，发送函数 SendCom_RS485 和接收函数 ReceiveCom_RS485 应成对使用，或者延时足够长时间后才能再次调用发送函数，否则发送数据和和设备返回数据形成干扰，无法正常通信。

RecevieCom_RS485

说明：该函数用于接收 RS485 串口获得的数据。

返回值：TRUE 表示数据接收成功，FALSE 表示接收失败，失败代码通过参数 Result 给出。

输入输出参数：

名称	输入输出类型	类型	说明
pbyReadBuffer	输入	POINTER TO BYTE	数据指针，指向需要存放接收数据的字节型数组。
ulNumOfBytesToRecv	输入	UDINT	需要接收的字节数
ulActualBytesRecv	输出	UDINT	实际接收的字节数
Result	输出	UDINT	返回串口失败代码，0表示无错误，其余值含义可以查询微软标准的系统错误代码（System Error Codes）

附录 1.2 系统温度监控功能块

CPU_Temperature_Monitor (FB)

说明：该功能块用于监控系统温度，并进行高温报警和预警输出提示。

输入输出参数：

名称	输入输出类型	类型	说明
bStart	输入	BOOL	为 True 时启动功能块功能
bError	输出	BOOL	错误标志位
uiErrorCode	输出	UINT	错误代码： 0：无错误 1：传感器打开失败， 2 传感器读取温度失败
uiCPUTemperature	输出	UINT	CPU 当前温度
bHighTemperature_Waring	输出	BOOL	CPU 高温警告 t>105 ℃
bHighTemperature_PreWaring	输出	BOOL	CPU 高温预报警 t>90℃

附录 2：Modbus RTU/TCP 使用说明：

Modbus 是一种工业标准通讯协议，广泛用于 PLC 、控制器、触摸屏等设备间的数据交互。Modbus RTU 以 RS232/RS485 作为通信接口，Modbus TCP 使用以太网作为通信接口。

EAC112 运动控制器具有 1 个 RS232 接口、1 个 RS485 接口和 1 个以太网（ENET）接口，均可以独立的作为 Modbus 主站/从站使用。

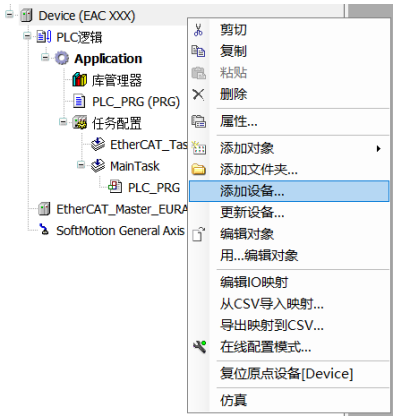
Modbus RTU 主站的使用方法：

本节将以 RS485 接口为例，详细说明如何在 CODESYS 编程软件中开发 Modbus RTU 主站功能，操作步骤如下：

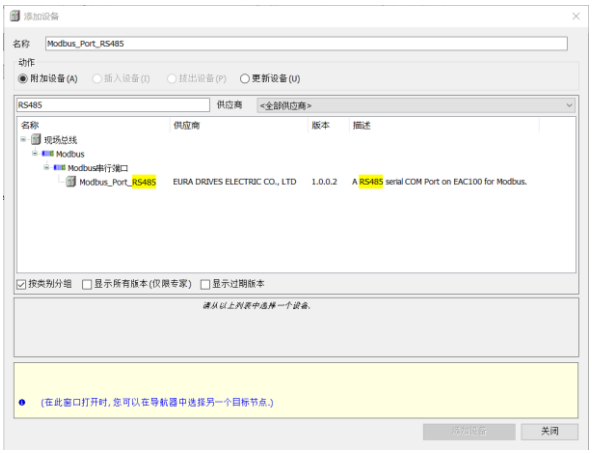
设备组态：

向工程中添加串口设备：

右击 Device(EAC XXX)点击“添加设备”，如图所示。

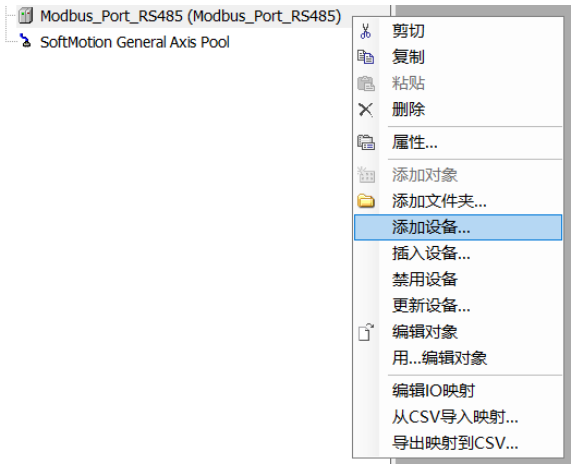


在弹出的对话框中，选择 Modbus_PORT_RS485，供应商为 EURA DRIVES ELECTRIC CO.,LTD

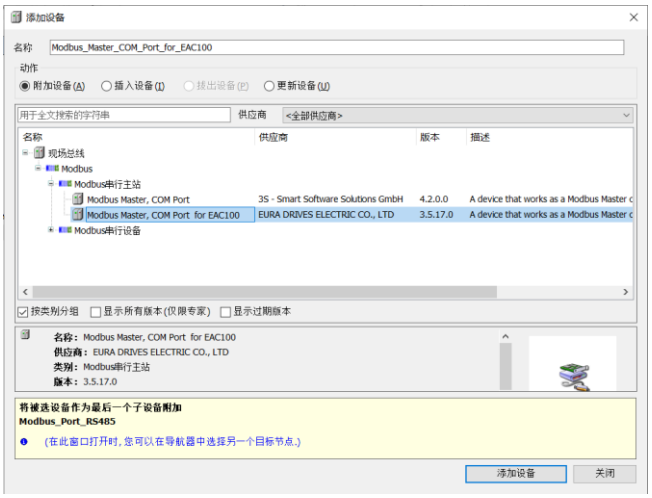


添加 Modbus RTU 主站：

向新添加的 RS485 的串口设备，添加 Modbus RTU 主站，如图所示。

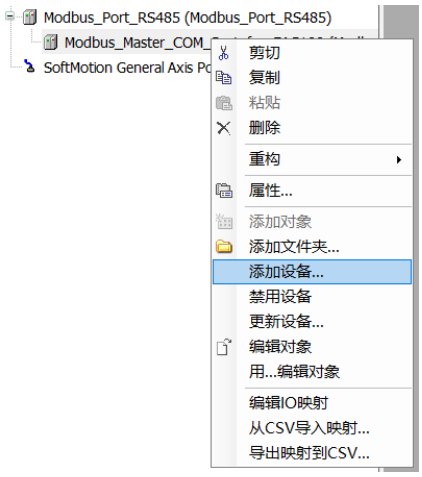


在弹出的对话框中，选择位于 Modbus 串行主站中的 Modbus Master,COM Port for EAC100，供应商为 EURA DRIVES ELECTRIC CO.,LTD，如图所示。

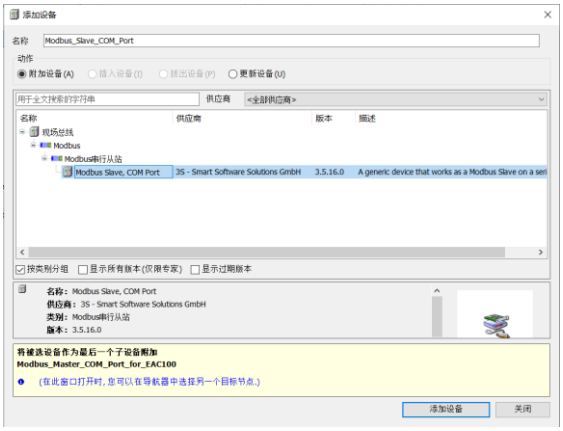


添加待访问的 Modbus 从站：

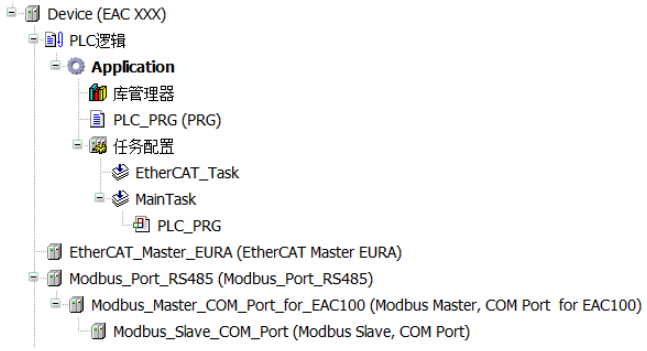
右击 Modbus_Master_COM_Port_for_EAC100，点击“添加设备”，如图所示。



在弹出的对话框中，选择 Modbus Slave,COM Port。

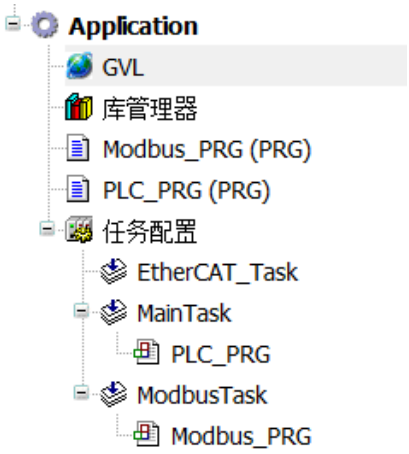


至此，设备组态完成，工程树结构如图所示。



软件开发

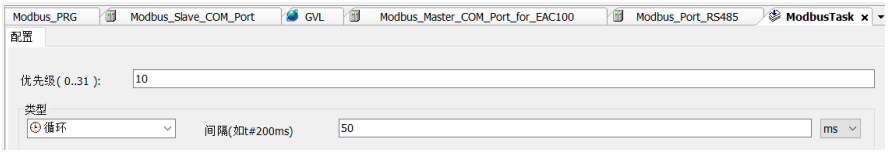
工程的任务和程序部分，实现如图所示的配置。



配置任务和程序：

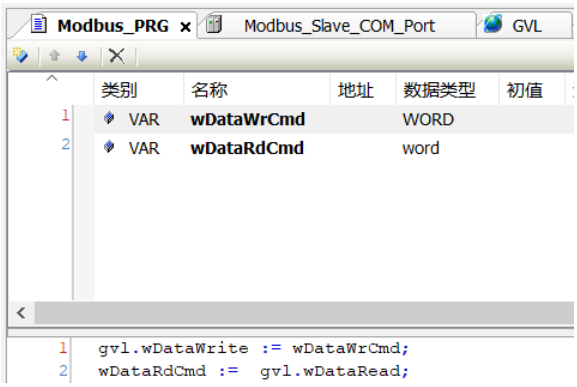
GVL 数据区间的配置，用于映射到 Modbus 通信设备中，如图所示。

新建的 Modbus 任务配置，如图所示。



编写程序代码：

Modbus_PRG 中的程序代码，如图所示。

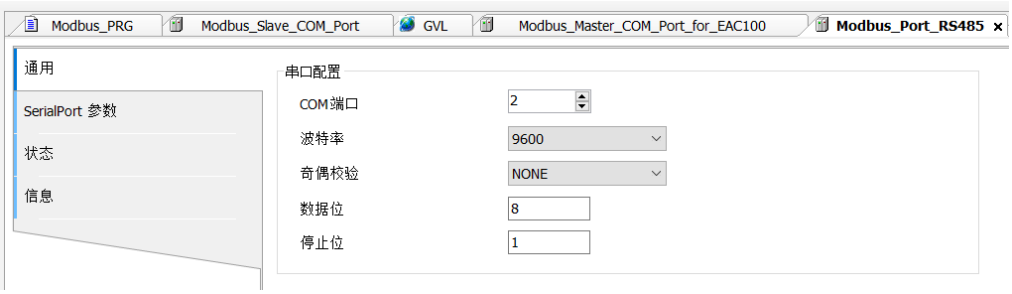


设备配置与数据映射：

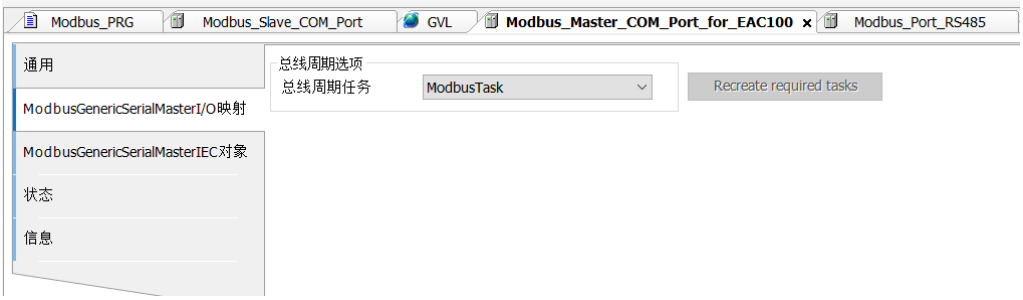
调整设备通信参数，使其符合实际网络通信需求，并将程序数据映射设备中。

配置通信参数：

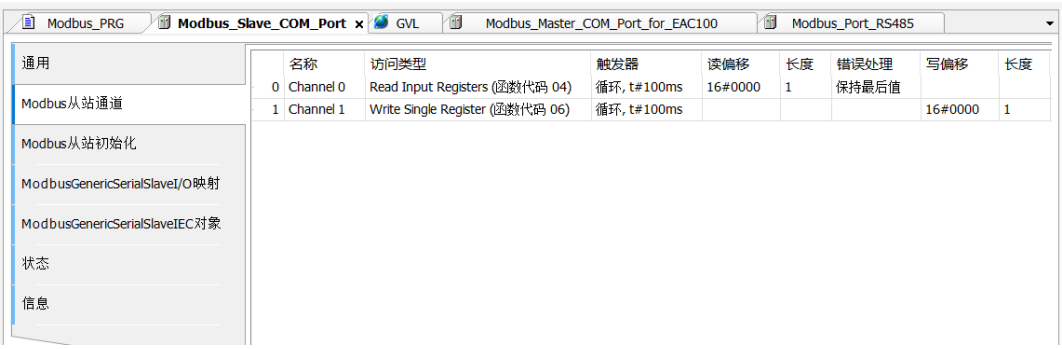
修改串口通信参数，以满足实际通信需求，如图所示。



调整设备的关联任务，使设备的总线周期任务配置为 ModbusTask。



配置待访问的 Modbus 从站的地址，如图所示，配置为 1 个 16 位数据的输入寄存器和 1 个 16 位数据的保持寄存器区。



数据映射：

将 GVL 数据区间的数据与 Modbus 从站设备中的数据建立映射，如图所示。

至此，工程开发完毕，可以从本公司官网获得工程案例，编译成功之后下载到目标设备上运行调试。

Modbus RTU 从站的使用方法：

本节将详细说明在上位机软件 CODESYS 中如何配置 Modbus 从站。在 CODESYS 中 Modbus RTU 从站使用带有掉电保持的全局数组和功能块进行实现。

本节将以 RS485 接口为例，详细说明如何在 CODESYS 编程软件中开发 Modbus RTU 从站功能，操作步骤如下：

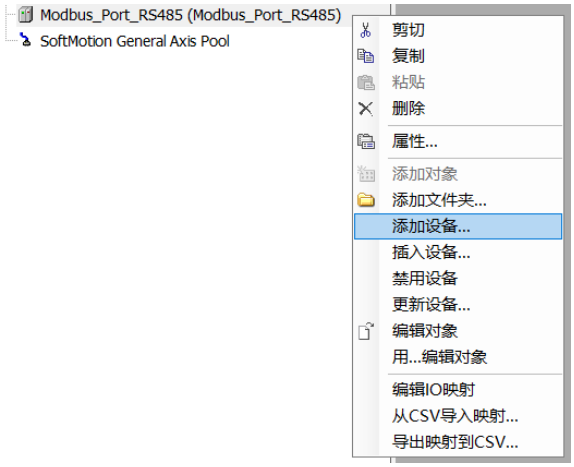
设备组态：

向工程中添加串口设备：

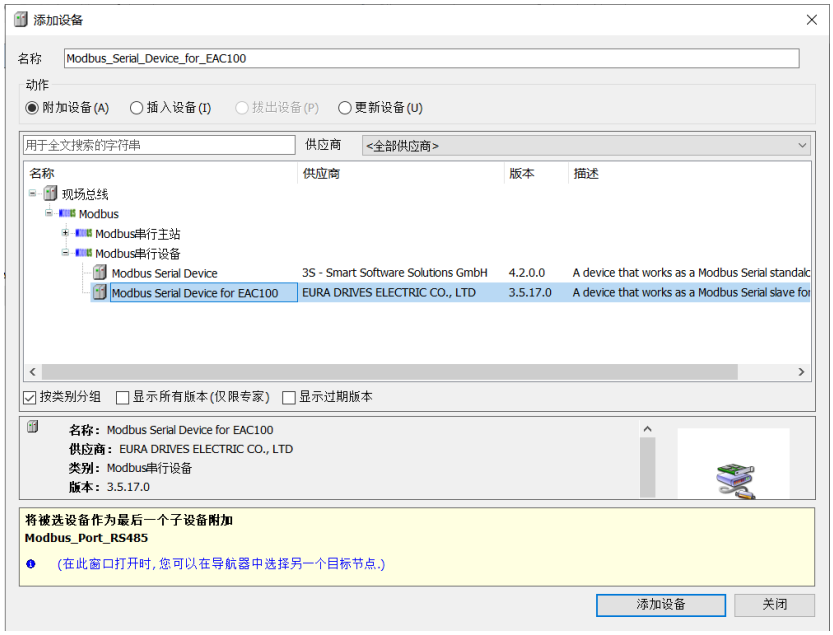
参照 Modbus RTU 主站的使用方法相关章节。

添加 Modbus RTU 从站：

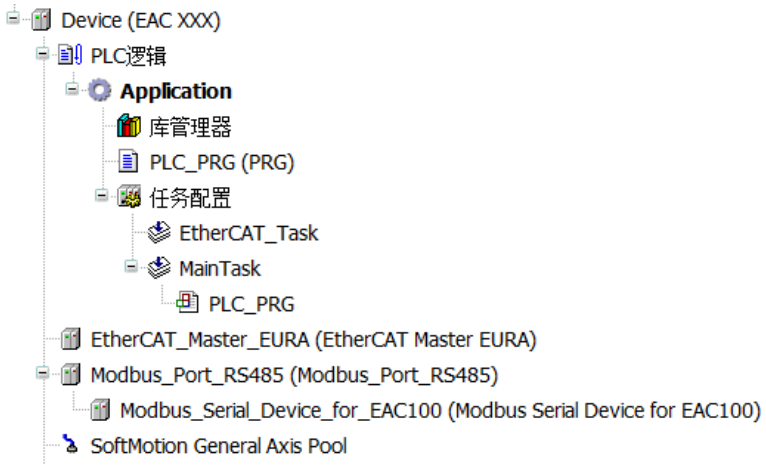
向新添加的 RS485 的串口设备，添加 Modbus RTU 从站，如图所示。



在弹出的对话框中，选择位于 Modbus 串行主站中的 Modbus_Serial_Device_for_EAC100，供应商为 EURA DRIVES ELECTRIC CO.,LTD，如图所示。

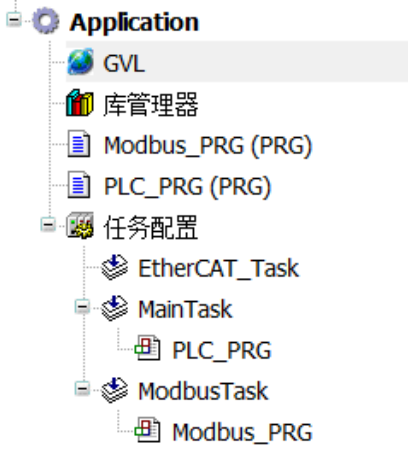


完成设备组态之后的工程树结构，如图所示。



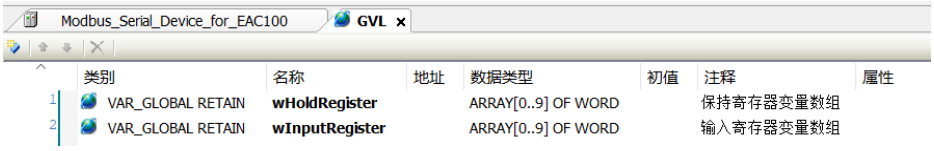
软件开发

工程的任务和程序部分，实现如图所示的配置。

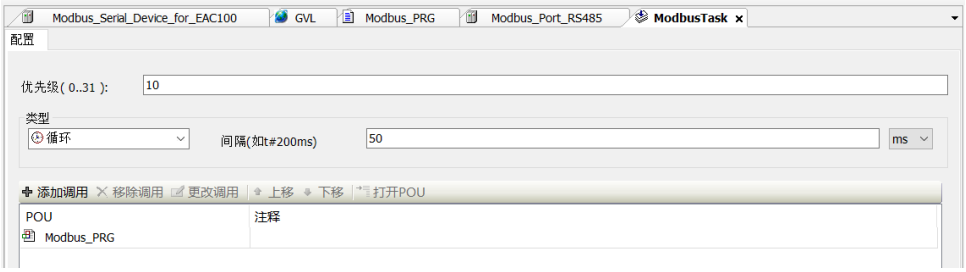


配置任务和程序：

GVL 数据区间的配置，用于映射到 Modbus 通信设备中，如图所示。

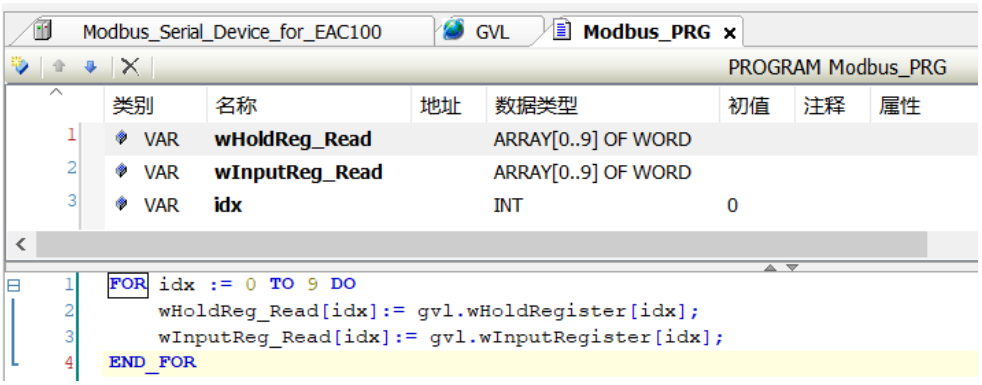


新建的 Modbus 任务配置，如图所示。



编写程序代码：

Modbus_PRG 中的程序代码，如图所示。

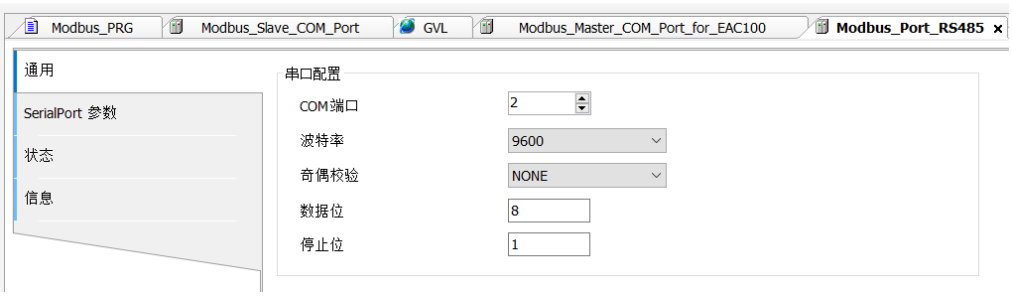


设备配置与数据映射：

调整设备通信参数，使其符合实际网络通信需求，并将程序数据映射设备中。

配置通信参数：

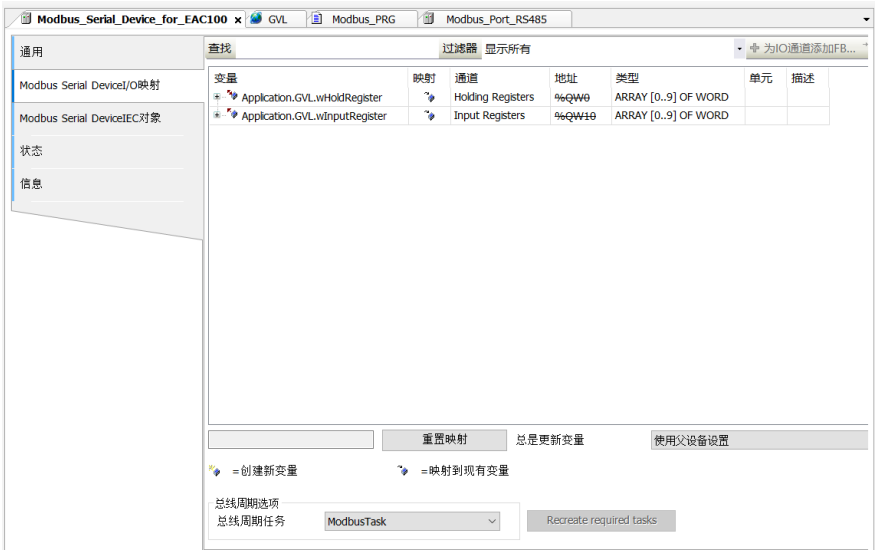
修改串口通信参数，以满足实际通信需求，如图所示。



调整设备的 ID、保持寄存器和输入寄存器的数量等，如图所示。

调整设备的关联任务，使设备的总线周期任务配置为 ModbusTask，并将 Modbus 通信数据映

射到 GVL 数据区。



至此，工程开发完毕，可以从本公司官网获得工程案例，编译成功之后下载到目标设备上运行调试

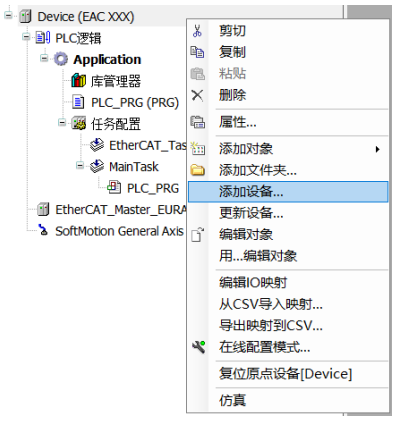
Modus TCP 主站的使用方法

本节将详细说明如何在 CODESYS 编程软件中开发 Modbus TCP 主站功能，操作步骤如下：

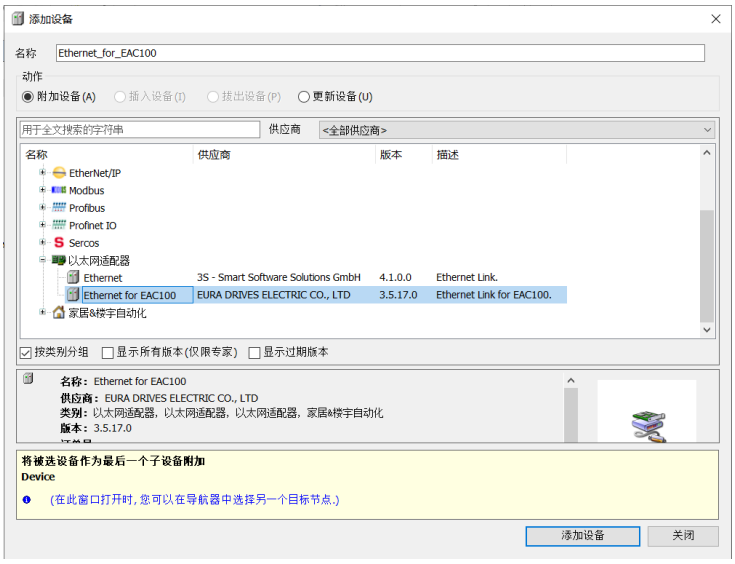
设备组态：

向工程中添加以太网设备：

右击 Device(EAC XXX)点击“添加设备”，如图所示。

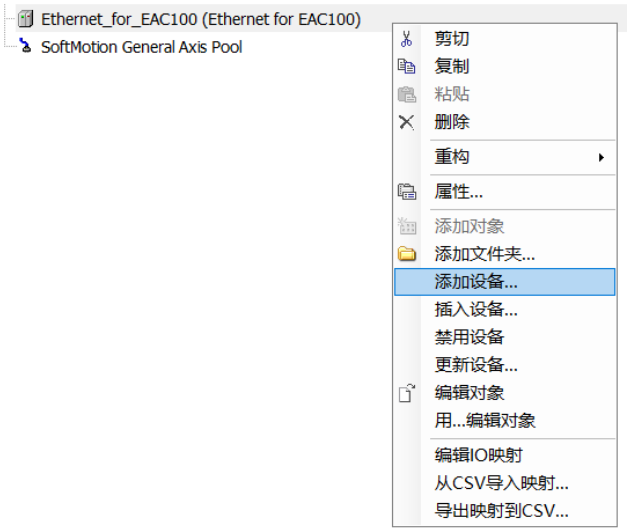


在弹出的对话框中，选择 Ethernet_for_EAC100，供应商为 EURA DRIVES ELECTRIC CO.,LTD

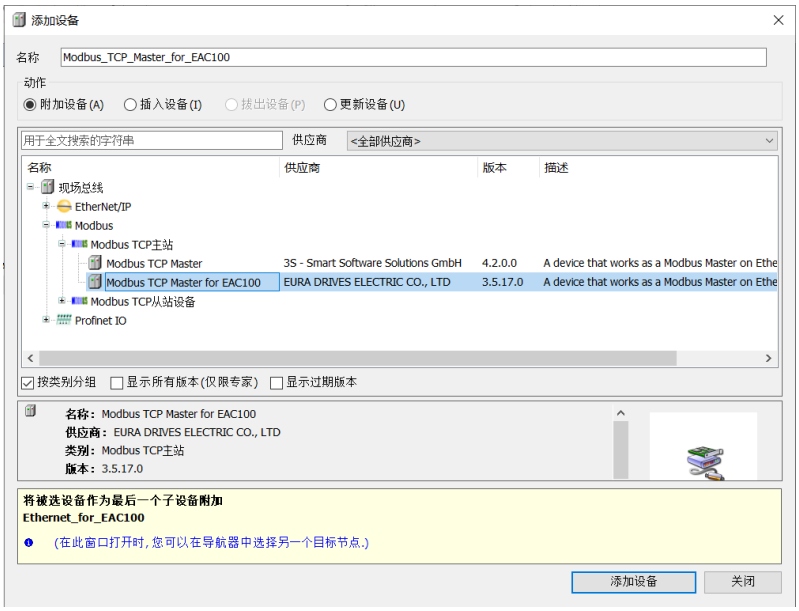


添加 Modbus TCP 主站：

向新添加的 Ethernet_for_EAC100 设备，添加 Modbus TCP 主站，如图所示。

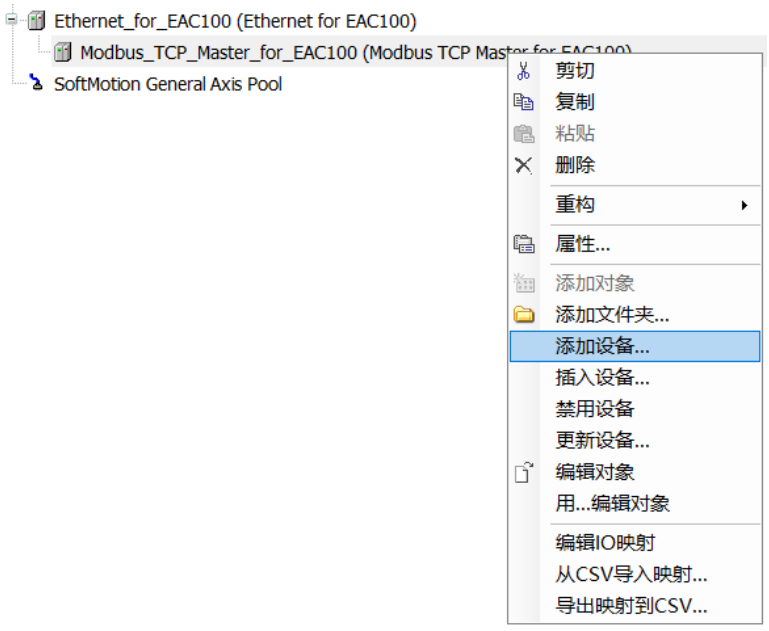


在弹出的对话框中，选择位于 Modbus 串行主站中的 Modbus_TCP_Master_for_EAC100，供应商为 EURA DRIVES ELECTRIC CO.,LTD，如图所示。

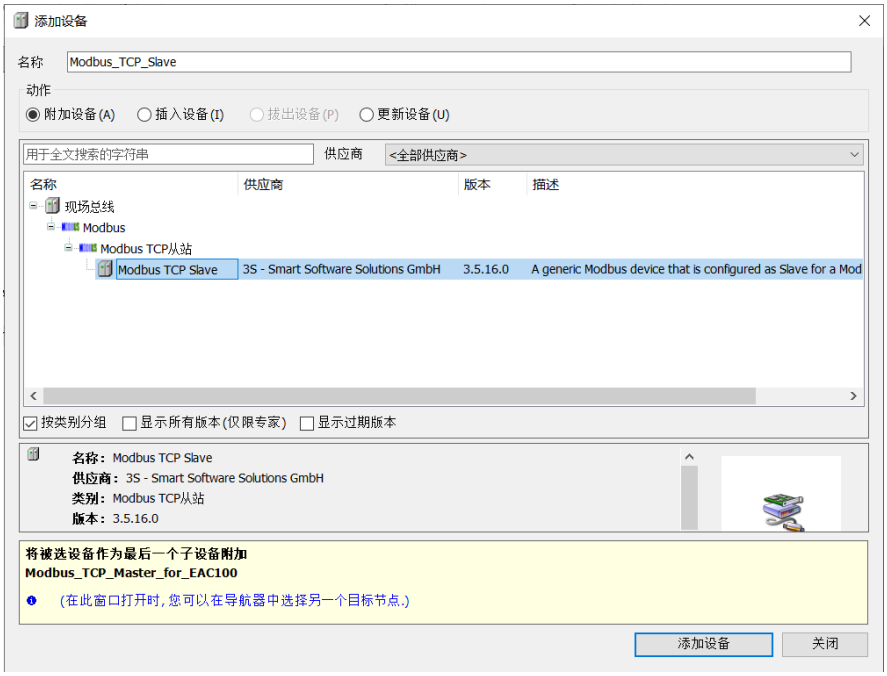


添加待访问的 Modbus 从站：

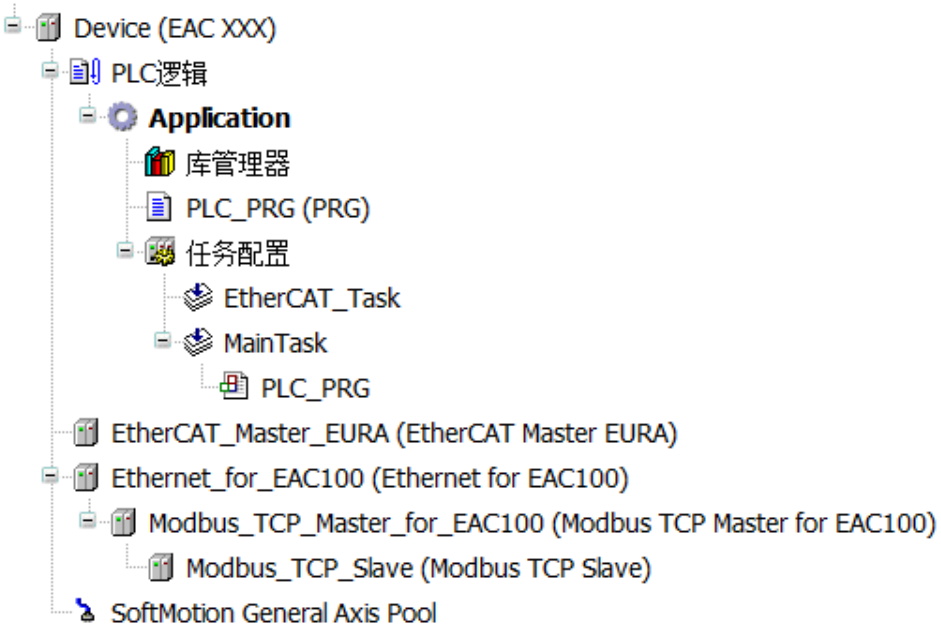
右击 Modbus_TCP_Master_for_EAC100，点击“添加设备”，如图所示。



在弹出的对话框中，选择 Modbus TCP Slave。

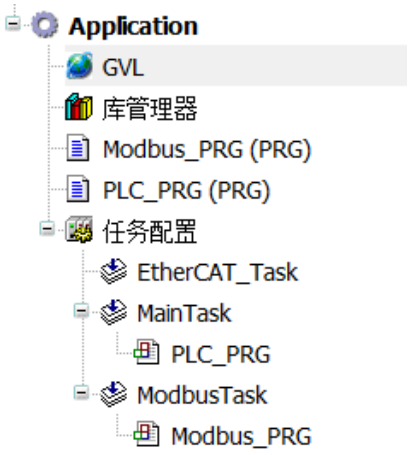


至此，设备组态完成，工程树结构如图所示。



软件开发

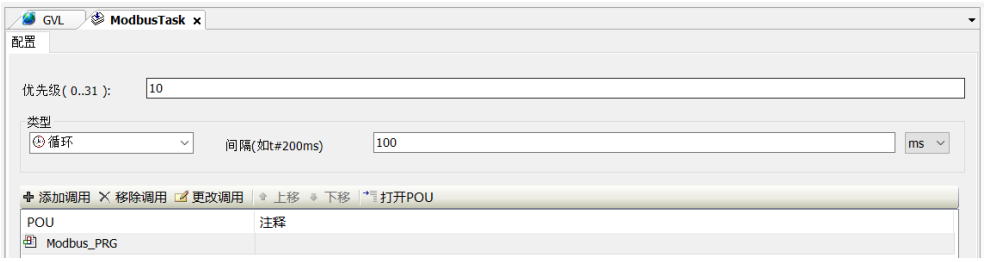
工程的任务和程序部分，实现如图所示的配置。



配置任务和程序：

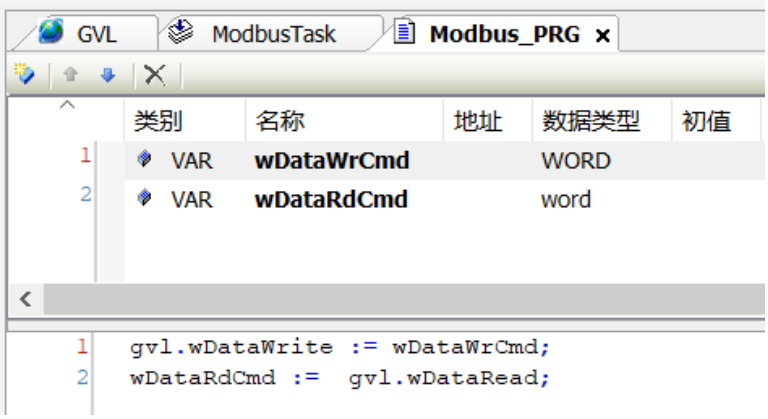
GVL 数据区间的配置，用于映射到 Modbus 通信设备中，如图所示。

新建的 Modbus 任务配置，如图所示。



编写程序代码：

Modbus_PRG 中的程序代码，如图所示。

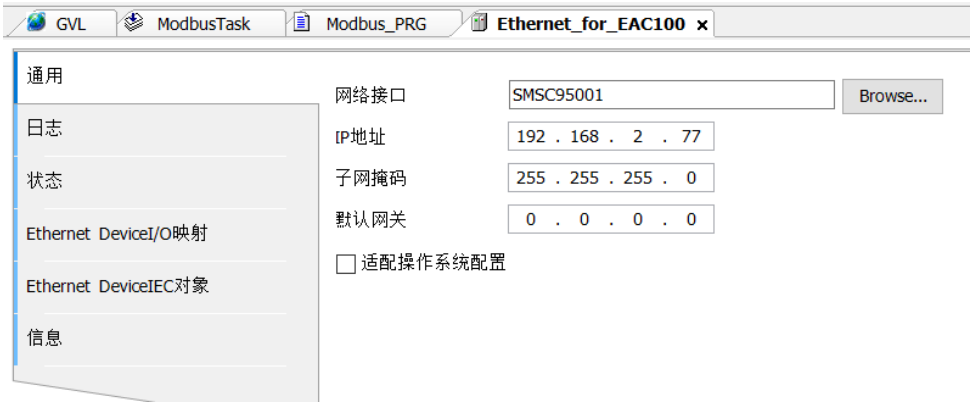


设备配置与数据映射：

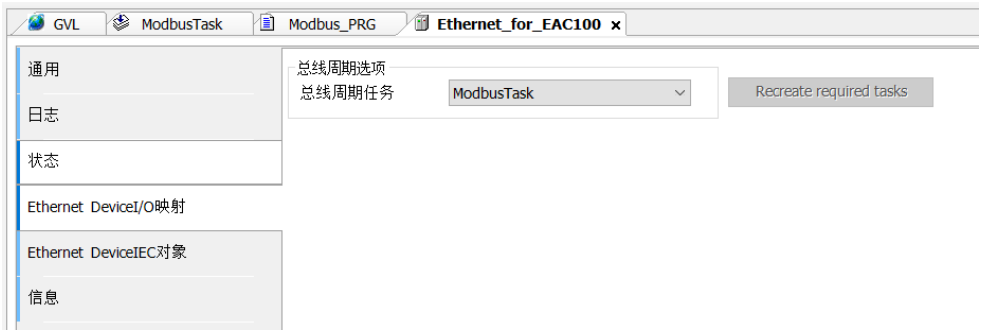
调整设备通信参数，使其符合实际网络通信需求，并将程序数据映射设备中。

配置通信参数：

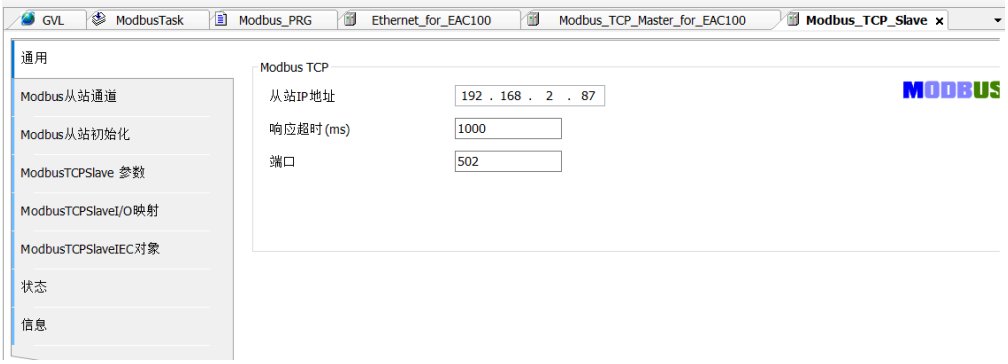
修改以太网通信参数，以满足实际通信需求，包括网卡名称、网卡 IP 地址、子网掩码和默认网关等，如图所示。

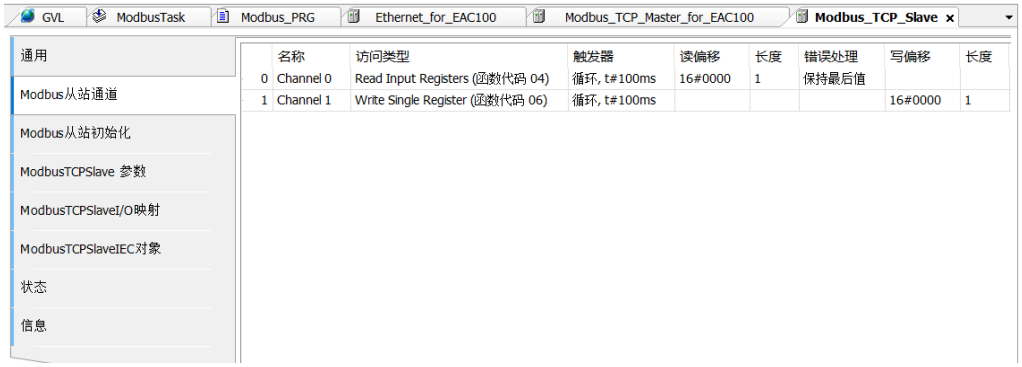


调整设备的关联任务，使设备的总线周期任务配置为 ModbusTask。



配置待访问的 Modbus 从站的 IP 地址，如图所示，并为其配置 1 个 16 位数据的输入寄存器和 1 个 16 位数据的保持寄存器区。





数据映射：

将 GVL 数据区间的数据与 Modbus 从站设备中的数据建立映射，如图所示。

至此，工程开发完毕，可以从本公司官网获得工程案例，编译成功之后下载到目标设备上运行调试，如图所示。

Modus TCP 从站的使用方法

本节将详细说明如何在 CODESYS 编程软件中开发 Modbus TCP 主站功能，操作步骤如下：

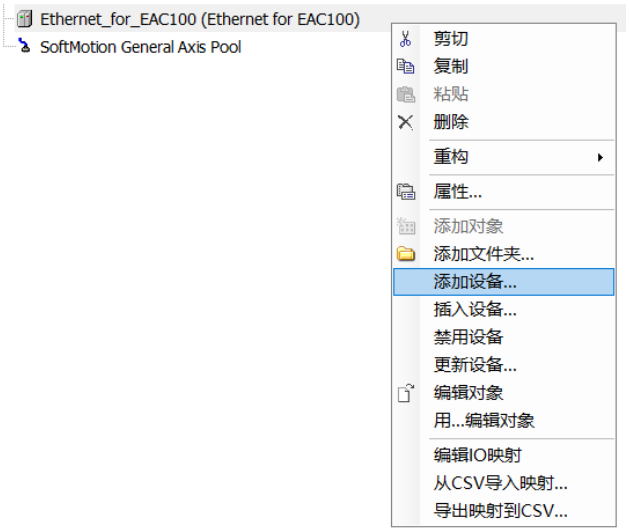
设备组态：

向工程中添加以太网设备：

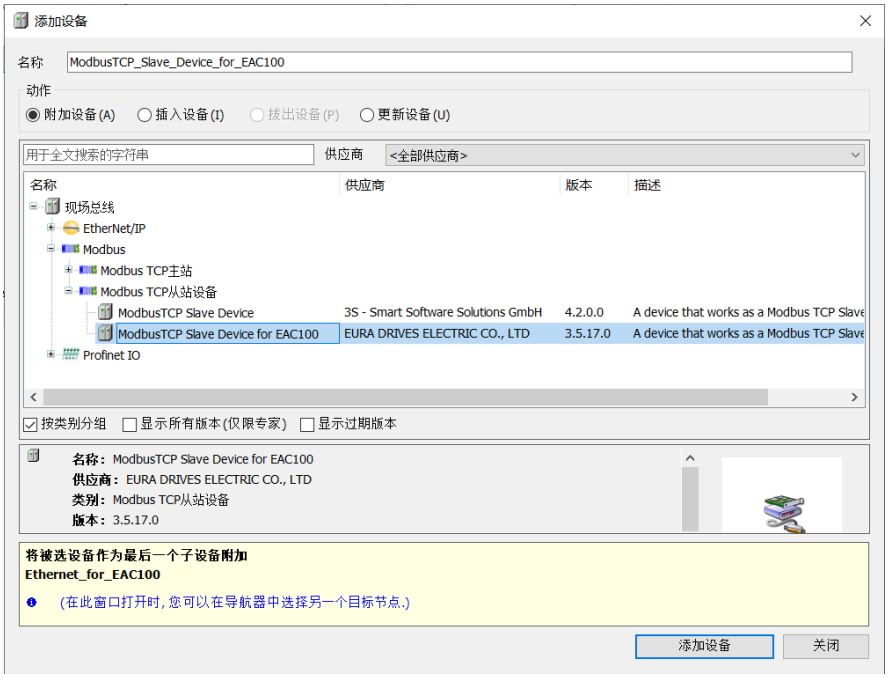
见 Modbus TCP 主站的使用方法相关章节。

添加 Modbus TCP 主站：

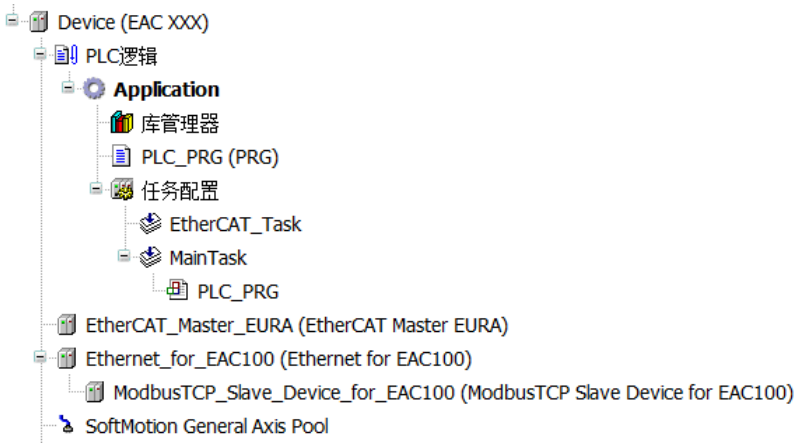
向新添加的 Ethernet_for_EAC100 设备，添加 Modbus TCP 从站，如图所示。



在弹出的对话框中，选择位于 Modbus 串行主站中的 ModbusTCP_Slave_Device_for_EAC100，供应商为 EURA DRIVES ELECTRIC CO.,LTD，如图所示。

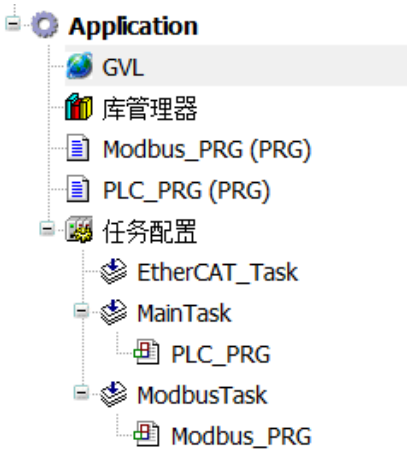


至此，设备组态完成，工程树结构如图所示。



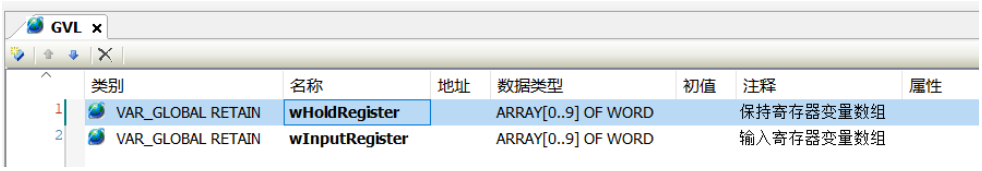
软件开发

工程的任务和程序部分，实现如图所示的配置。

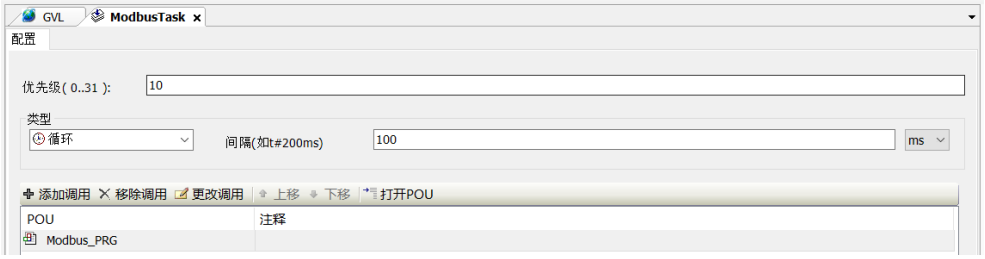


配置任务和程序：

GVL 数据区间的配置，用于映射到 Modbus 通信设备中，如图所示。

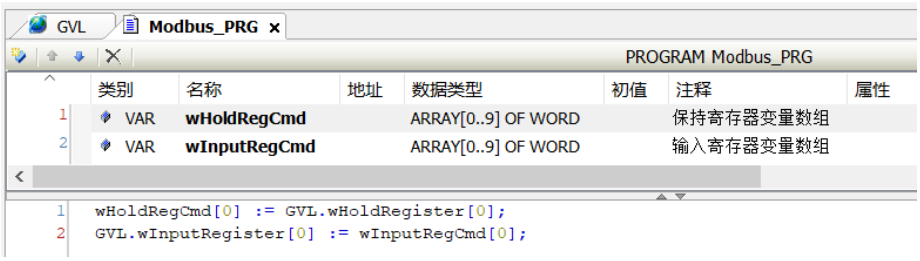


新建的 Modbus 任务配置，如图所示。



编写程序代码：

Modbus_PRG 中的程序代码，如图所示。

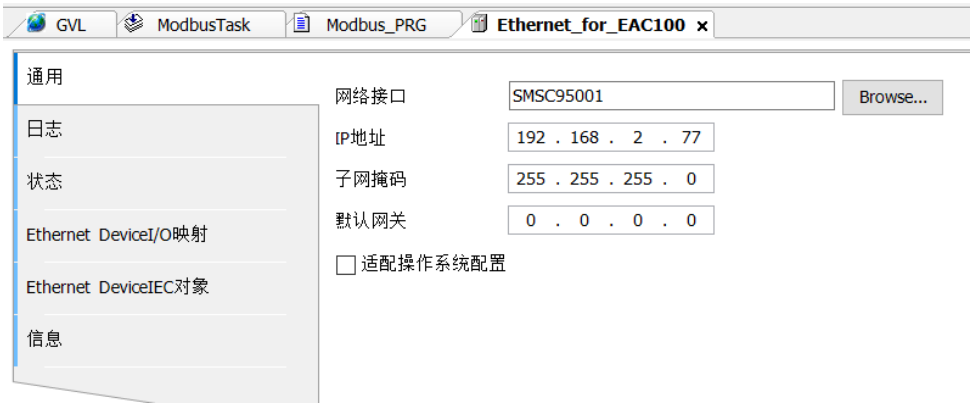


设备配置与数据映射：

调整设备通信参数，使其符合实际网络通信需求，并将程序数据映射设备中。

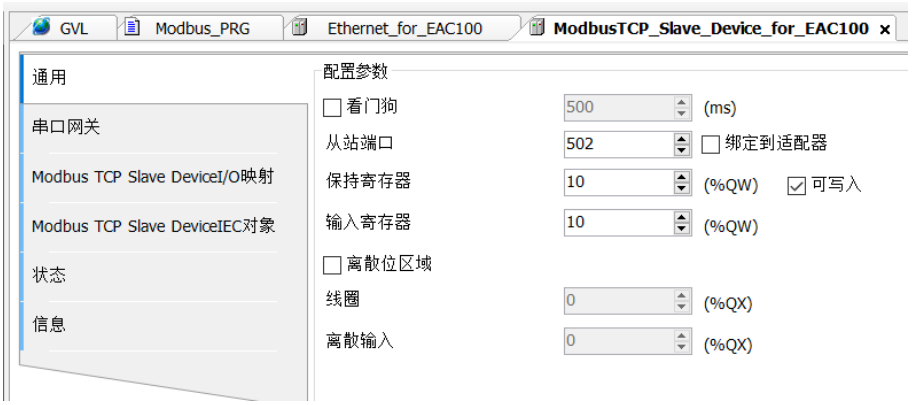
配置通信参数：

修改以太网通信参数，以满足实际通信需求，包括网卡名称、网卡 IP 地址、子网掩码和默认网关等，如图所示。



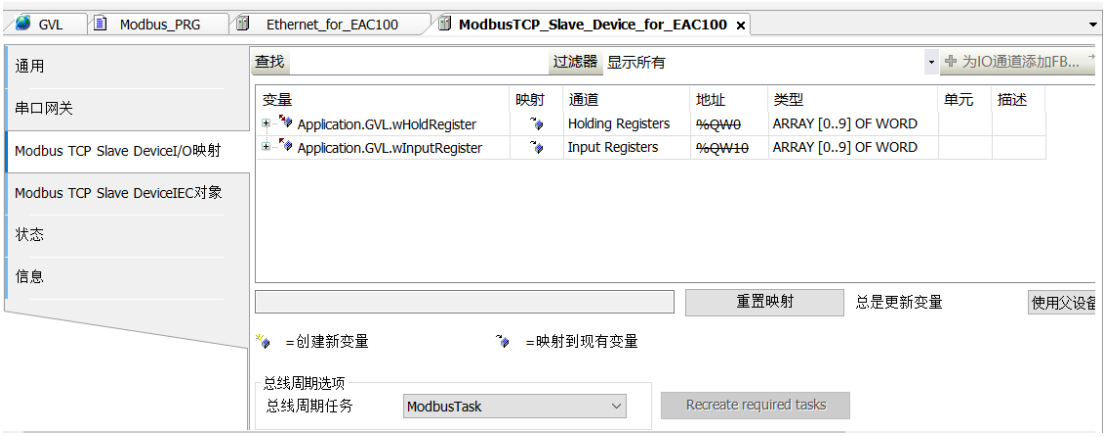
调整设备的关联任务，使设备的总线周期任务配置为 ModbusTask。

配置 Modbus 从站的寄存器数量，如图所示。



数据映射：

将 GVL 数据区间的数据与 Modbus 从站设备中的数据建立映射，如图所示。

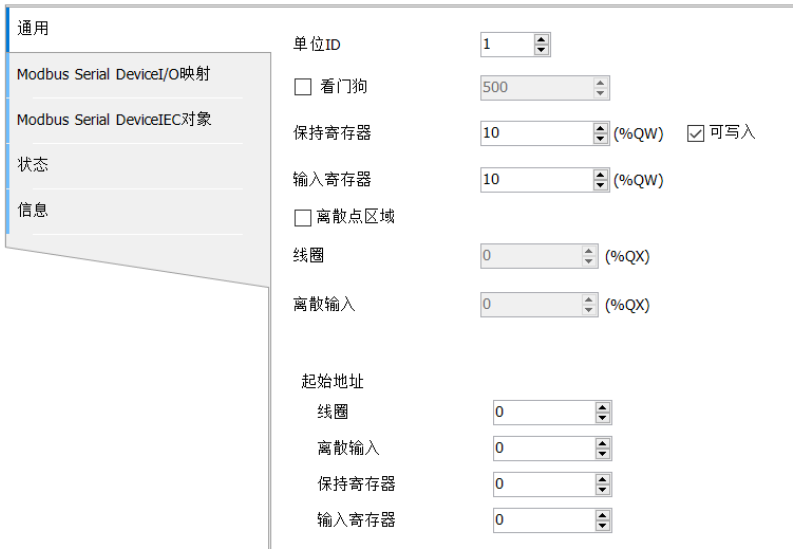


至此，工程开发完毕，可以从本公司官网获得工程案例，编译成功之后下载到目标设备上运行调试，如图所示。

Modbus 从站（服务端）的数据保持

在 SP17 中，Modbus 从站（服务端）的保持数据区增加“可写入功能”，将保持寄存器映射的 IEC 的%Q 地址区。实现 Modbus 从站保持数据区的掉电保持功能，需要做如下 2 步：

1. 映射到保持区的变量类型，为掉电保持类型，详见附图。
2. 开启 Modbus RTU/TCP 从站（服务端） “可写入”选项，详见附图。


通用
Modbus Serial DeviceI/O映射
Modbus Serial DeviceIEC对象
状态
信息
单位ID
1
看门狗
500
保持寄存器
10
(%QW)
可写入
输入寄存器
10
(%QW)
离散点区域
线圈
0
(%QX)
离散输入
0
(%QX)
起始地址
线圈
0
离散输入
0
保持寄存器
0
输入寄存器
0


通用
串口网关
Modbus TCP Slave DeviceI/O映射
Modbus TCP Slave DeviceIEC对象
状态
信息
配置参数
看门狗
500
(ms)
关闭TCP套接字
从站端口
502
绑定到适配器
保持寄存器
10
(%QW)
可写入
输入寄存器
10
(%QW)
离散位区域
线圈
0
(%QX)
离散输入
0
(%QX)
数据模型
起始地址
线圈
0
离散输入
0
保持寄存器
0
输入寄存器
0
保持寄存器和输入寄存器数据区域覆盖

附录 3：数字量拓展 I/O 的使用说明

控制器本体的数字量拓展 I/O，基于本地总线（DIDO）实现，可设置的最小周期任务为 10ms。

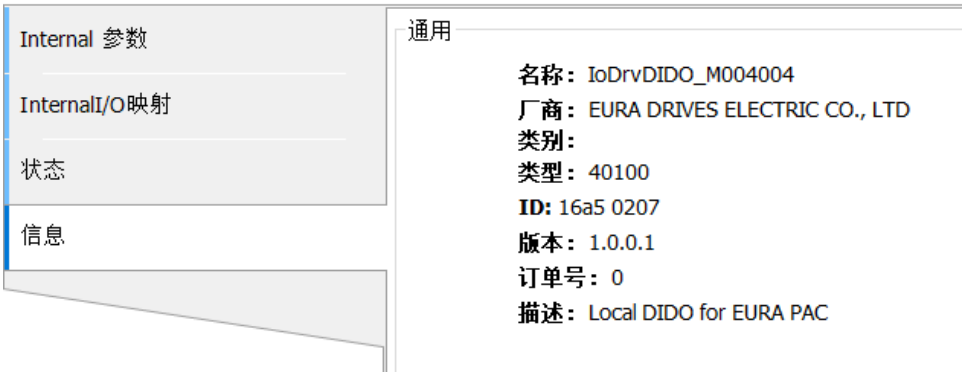
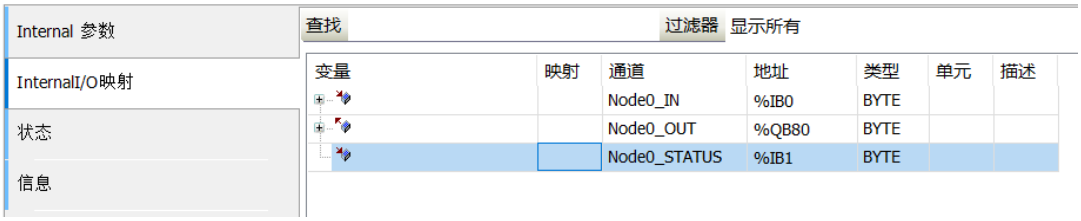


图 1 DIDO 设备的版本信息 v1.0.0.1

在 v1.0.0.1 中，增加 DIDO 状态位（Node_STATUS），用于指示 I/O 是否正常运行状态，如下图所示。0：DIDO 总线未启动，1：DIDO 总线已经启动。

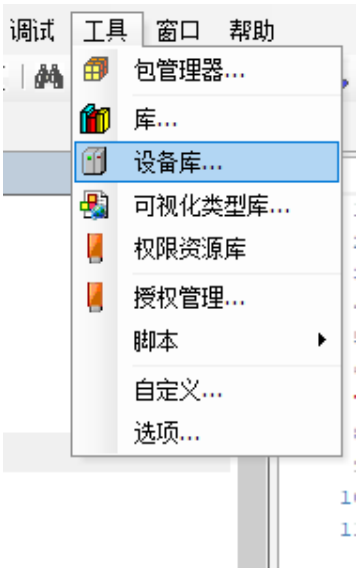


附录 4：设备库的管理

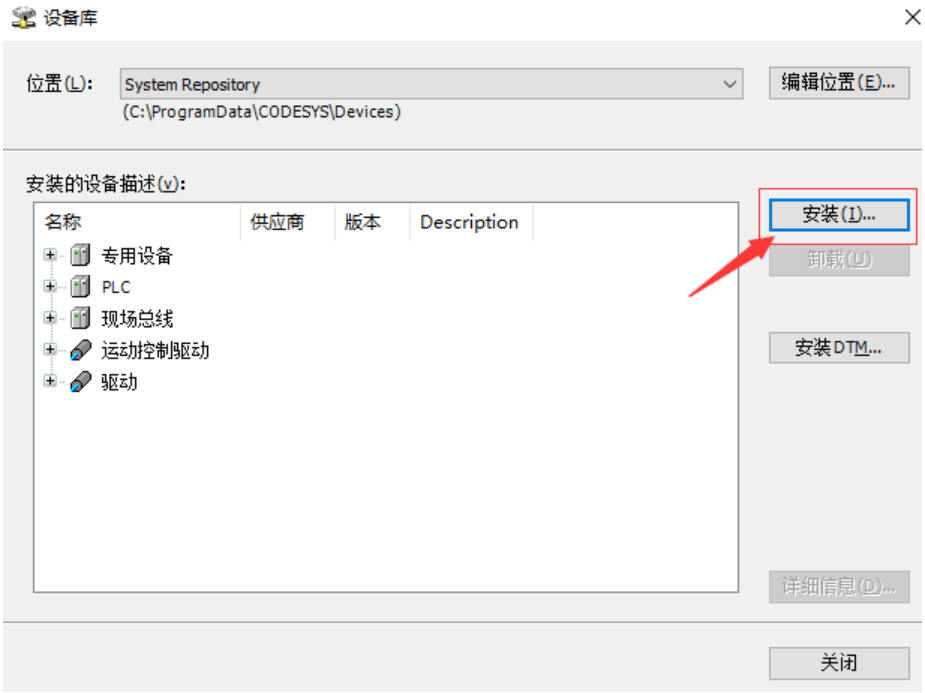
设备描述文件的安装与更新

当有新的设备需要安装或者已安装设备需要升级时，使用本节说明进行安装或更新。

运行 CODESYS，点击菜单栏中“工具”——“设备库”选项。



在弹出的对话框中点击“安装...”按钮。



在弹出的对话框筛选文件类型一栏中选择“所有支持的描述文件”，选择需要安装或者更新的文件，点击打开，设备描述文件会自动安装或更新。

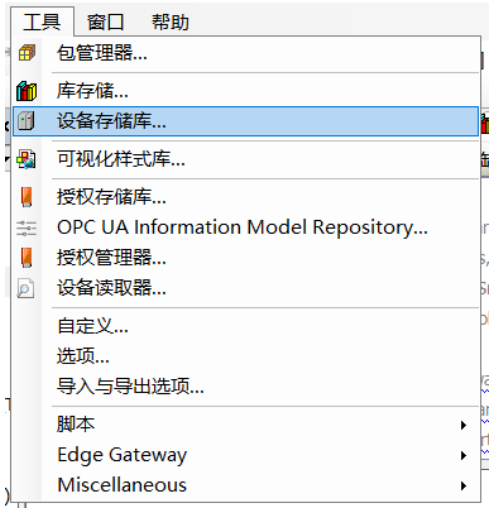
附录 5：工程版本的切换

当有新的设备需要安装或者已安装设备需要升级时，使用本节说明进行安装或更新。

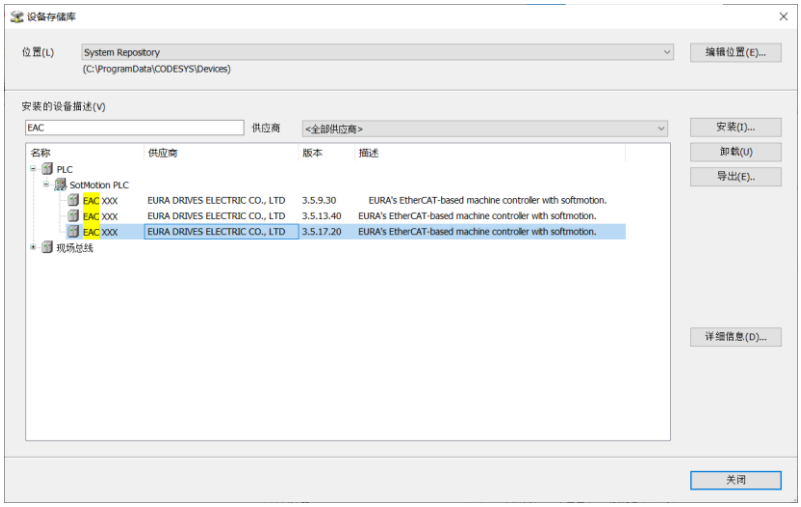
准备工作

检查已经安装的设备

打开 CoDeSys 软件，点击工具，然后选择<设备存储库>，如下图**错误!未找到引用源。**所示。



在打开的设备存储库中，输入 EAC 找到已经安装的 EAC 设备，这里可以看到有 3 个 EAC 设备，如下图所示。



删除不使用的设备

如果不再使用新版本的设备，可以将新版本的设备删除，重新启动软件并创建新工程，此时将以设备剩余版本中最新的版本创建设备。

调整新建工程的版本

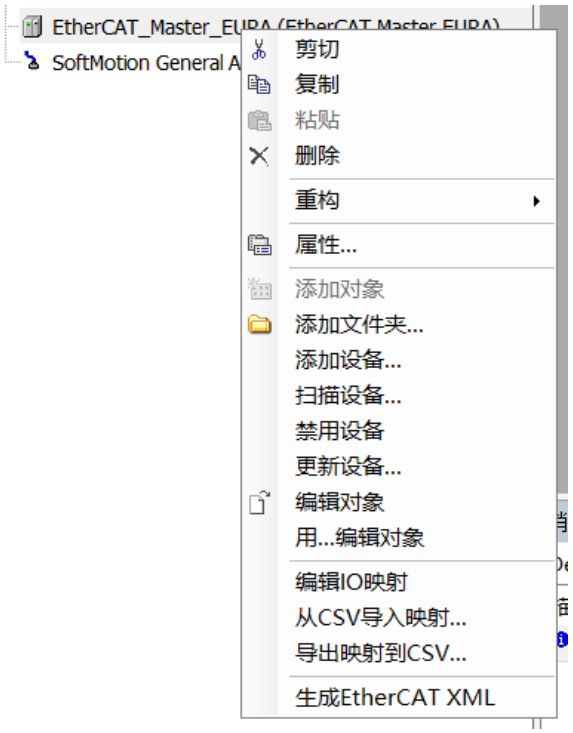
本章节主要讲述，如何在同时存在多个版本的 EAC 设备情况下，如何在新建工程中切换设备版本，以 3.5.17.20 切换到 3.5.9.30 为例。

新建工程

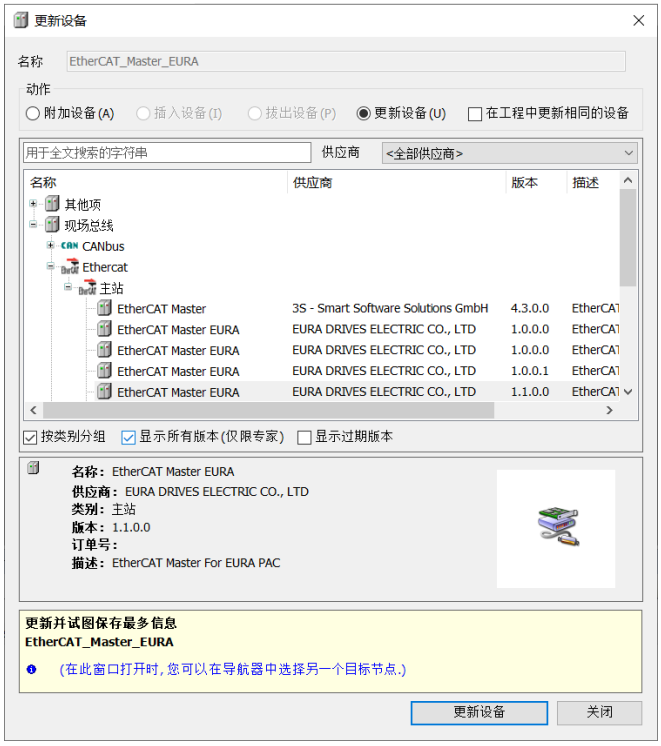
新建工程，选择 EAC 设备，如下图所示。

修改 EtherCAT 主站版本

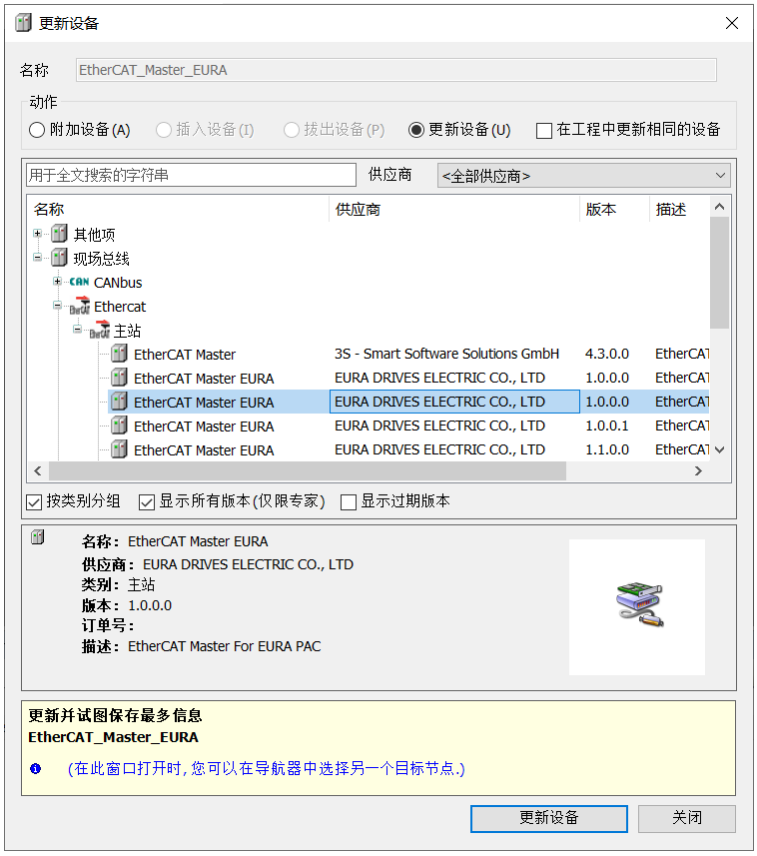
此时，使用最新版本的设备创建工程，然后修改 EtherCAT 主站版本，右击 EtherCAT_Master_EURA，选择更新设备，如下图所示。



使用 3.5.9.30 对应的 EtherCAT 主站版本（1.0.0.0），如下图所示勾选<显示所有版本>。



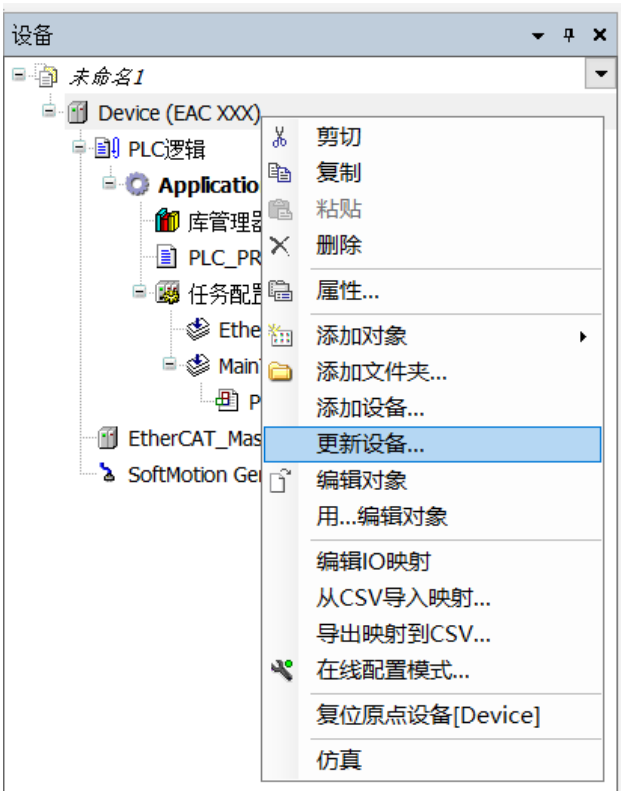
选择 EtherCAT Master EURA 的匹配版本 1.0.0.0，如下图所示。



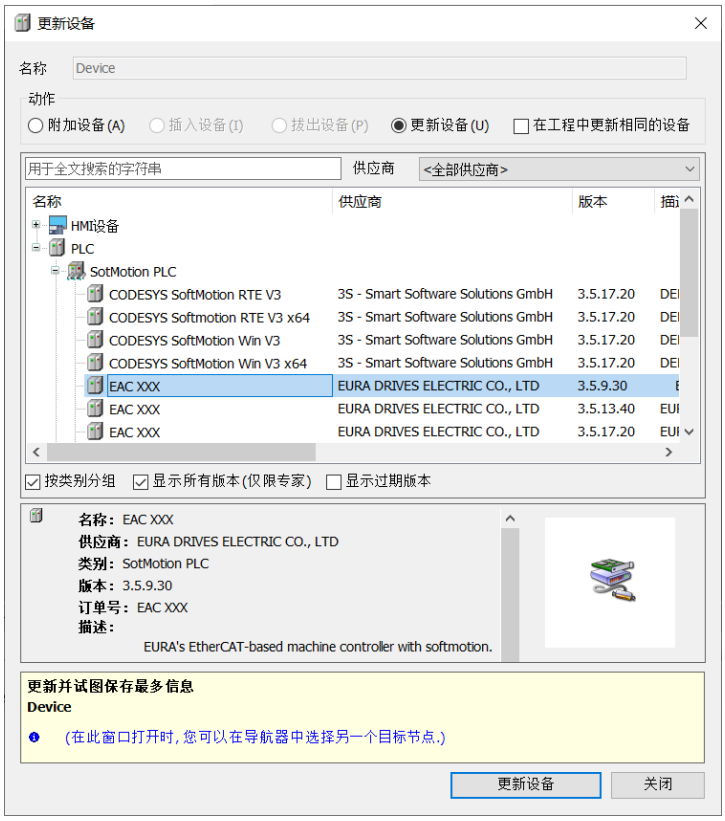
注意：SP17 之前的版本，EtherCAT 网卡为 SMSC95001，降低版本之后也应该同步切换为对应的网卡名称。

修改 EAC 设备版本

与修改 EtherCAT 主站的版本方法相近，修改 EAC 设备版本。右击 EAC 设备，如下图所示。

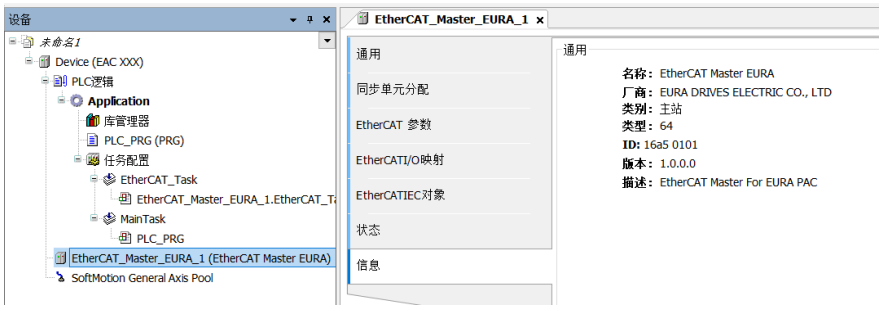


选择 3.5.9.30 版本的 EAC 设备，如下图所示。



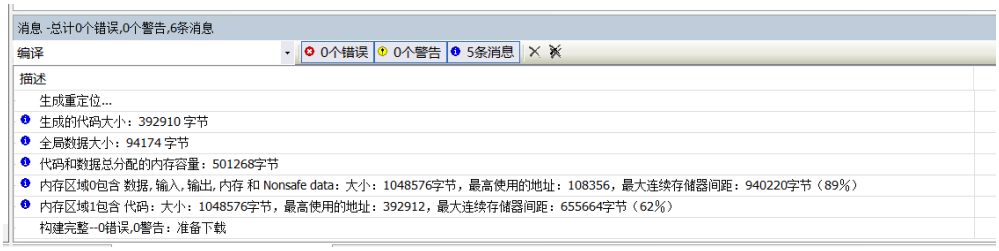
确认修改之后的设备版本信息

修改之后的 EtherCAT 主站设备信息，如下图所示。



修改之后的 EAC 设备信息，如图所示。

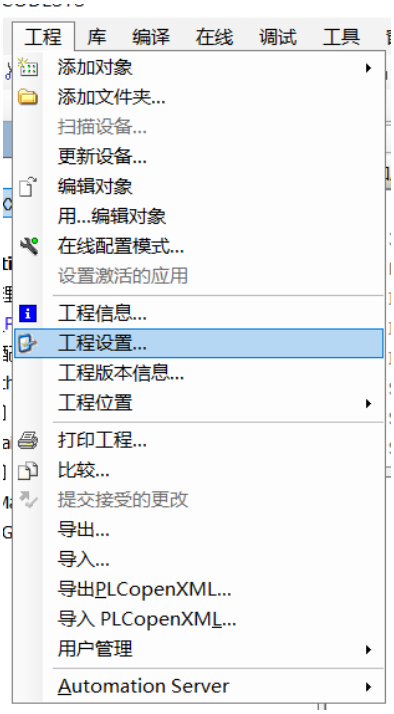
修改完成之后，重新编译设备，提示编译成功，如图所示。


消息 -总计0个错误,0个警告,6条消息
编译
0个错误
0个警告
5条消息
描述
生成重定位...
生成的代码大小：392910 字节
全局数据大小：94174 字节
代码和数据总分配的内存容量：501268字节
内存区域0包含 数据, 输入, 输出, 内存 和 Nonsafe data：大小：1048576字节，最高使用的地址：108356，最大连续存储器间距：940220字节（89%）
内存区域1包含 代码：大小：1048576字节，最高使用的地址：392912，最大连续存储器间距：655664字节（62%）
构建完整--0错误,0警告：准备下载

其他调整

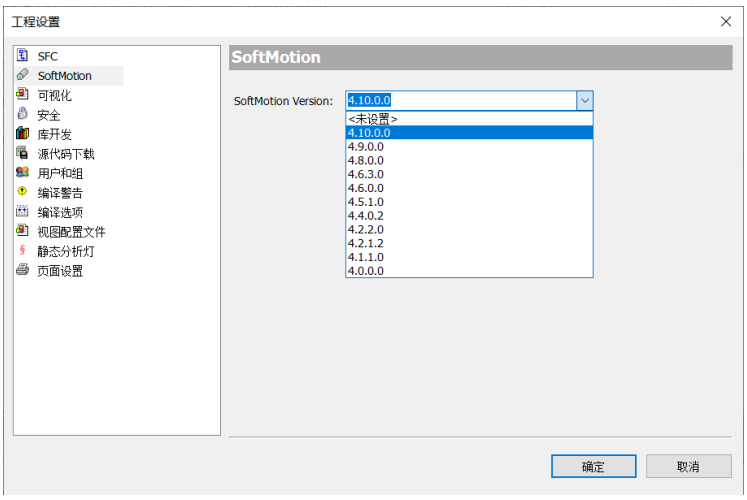
通过之前的调整，已经将设备由 3.5.17.20 切换至 3.5.9.30。为了保持最大的兼容性，还可以通过调整编译器版本，Softmotion 版本以及指定库的版本。

通过工程设置修改编译器和 Softmotion 的版本，如图所示。



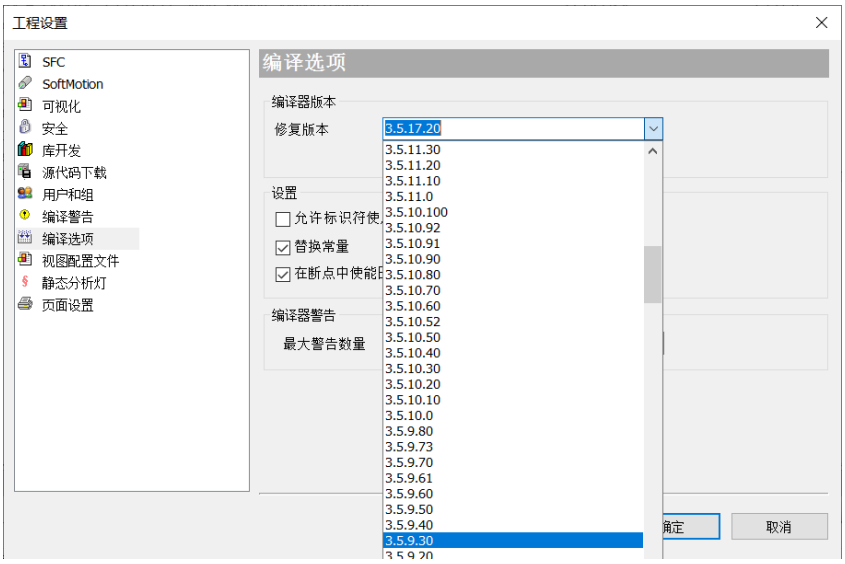
调整 softmotion 版本

修改 softmotion 的版本，可以间接修改部分库的版本，比如以 SM 开头的库，如图所示。



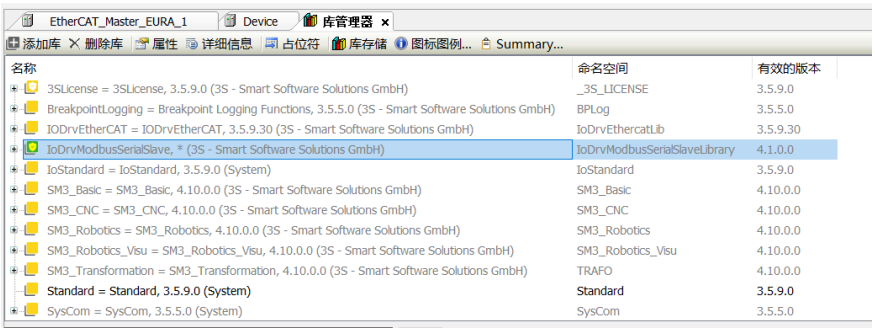
调整编译器版本

通过修改编译器，可以间接修改部分库的版本，如图所示。



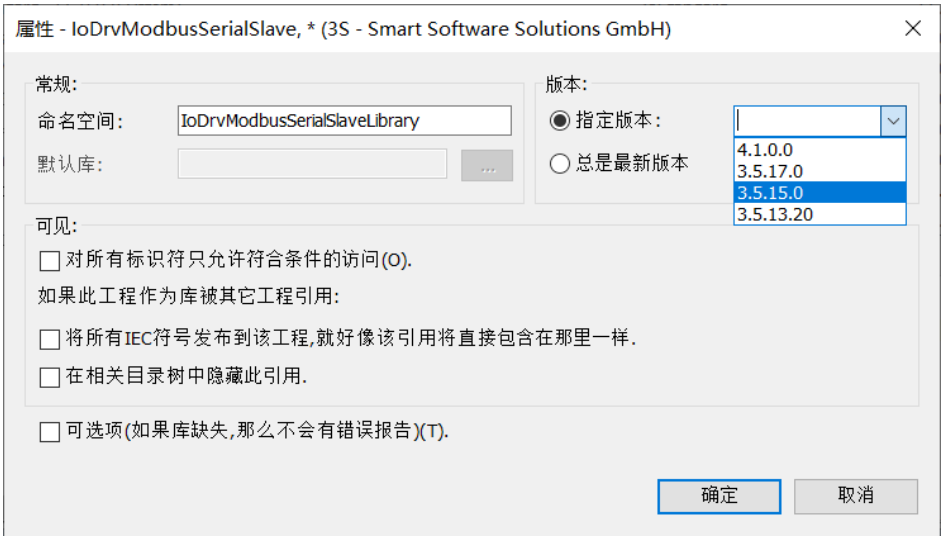
调整其他库的版本

工程中的部分库的版本与设计的不一致，可以通过库管理器修改为指定的版本，如图**错误!未找到引用源。**所示。



右击要修改的库，以 IoDrvModubSerialSlave 为例，点击属性，如图所示。

选择指定的版本，然后点击确定，如图所示。



下载缺少的库文件

如果提示缺少库文件，可以通过点击下载缺失的库，解决该问题，如图所示。

敬告用户：

感谢您选用我司产品，为保证您正确使用本产品及得到我司最佳售后服务，请认真阅读下述条款，并做好相关事宜。

只有具备一定的电气知识的操作人员才能够对本产品进行接线、上电操作；手册中示例程序仅供参考，不保证其实用性。

本公司致力于产品的不断改善和升级，手册提供资料如有变更，恕不另行通知，请自行访问本公司网站获取。

产品保修范围：按使用要求正常使用情况下，所产生的故障。

产品保修期限：本公司产品的保修期为自出厂之日起，十二个月以内。保修期实行长期技术服务。

非保修范围：任何违反使用要求的认为意外、自然灾害等原因导致的损坏，以及未经许可而擅自对产品拆卸、改装及修理的行为，视为自动放弃保修服务。

从中间商处购入产品：凡从经销代理商处购买产品的用户，在产品发生故障时，请与经销商、代理商联系。

免责条款：因下列原因造成的产品故障不在厂家 12 个月免费保修服务范围之内：

（1）、厂家不依照《产品手册》中所列程序进行正确的操作；

（2）、用户未经与厂家沟通自行修理产品或擅自改造产品；

（3）、因用户环境不良导致产品器件异常老化或引发故障；

（4）、因用户超过产品的标准范围使用产品；

（5）、由于地震、火灾、风水灾害、雷击、异常电压或其他自然灾害等不可抗力的原因造成的产品损坏；

（6）、因购买后由于人为摔落及运输导致硬件损坏。

责任：无论从合同、保修期、疏忽、民事侵权行为、严格的责任、或其他任何角度讲，EURA 和他的供货商及分销商都不承担以下由于设备所造成的特殊的、间接的、继发的损失责任。其中包括但不仅仅局限于利润和收入的损失，使用供货设备和相关设备的损失，资金的花费，代用设备的花费，工具费和服务费，停机时间的花费，延误，及购买者的客户或任何第三方的损失。另外，除非用户能够提供有力的证据，否则公司及它的供货商将不对某些指控如：因使用不合格原材料、错误设计、或不规范生产所引发的问题责任。

解释权归欧瑞传动电气股份有限公司。

如果您对 EURA 的产品还有疑问，请与 EURA 公司或其办事处联系。技术数据、信息、规范均为出版时的最新资料，EURA 公司保留部事先通知而更改的权利，并对由此造成的损失不承担任何责任。解释权归 EURA 公司。

20230914

欧瑞传动电气股份有限公司
EURA DRIVES ELECTRIC CO.,LTD
24小时服务热线： 4006-866-333
公司网址： www.euradrives.com