深圳市雷赛控制技术有限公司
SHENZHEN LEADSHINE CONTROL TECHNOLOGY CO.,LTD

EM64DX-E1 模块用户手册

Version 1.0

2020 年 3 月 20 日

修改记录

修改日期	版本	修改说明		拟制人
		原来内容	更新内容	
20200320	V1.0		初版	产品部

调试机器要注意安全！用户必须在机器中设计有效的安全保护装置，在软件中加入出错处理程序。否则所造成的损失，雷赛公司没有义务或责任负责。

目　录

第 1 章　产品概述 ---- 5
1.1　产品简介 ---- 5
1.2　产品特点 ---- 5
1.3　技术规格 ---- 5
1.4　安装使用 ---- 7
第 2 章　产品外观及硬件接线 ---- 8
2.1　产品外观 ---- 8
2.2　接口分布及针脚定义 ---- 8
2.2.1　电源接口 ---- 10
2.2.2 EtherCAT 接口定义 ---- 10
2.2.3 X1/X2 接口定义 ---- 10
2.2.4 X3 接口定义 ---- 11
2.2.5 X4　接口定义 ---- 12
2.3　接口电路 ---- 12
2.3.1　通用输入信号接口 ---- 12
2.3.2　通用输出信号接口 ---- 13
2.3.3　可选配通用输入输出接口 ---- 15
第 3 章　指示灯定义及说明 ---- 16
3.1　指示灯定义 ---- 16
3.2　指示灯闪烁规则 ---- 16
3.3　指示灯状态 ---- 17
第 4 章　功能说明 ---- 20
4.1 通用输入功能 ---- 20
4.2 输入计数功能 ---- 20
4.3 通用输出功能 ---- 20
4.4 输出延时翻转功能 ---- 20
4.5 总线掉线输出保持功能 ---- 21
4.6 总线复位输出保持功能 ---- 21
4.7 从站站号拨码功能 ---- 21
第 5 章　对象字典 ---- 22

5.1 通用参数----22
5.2 参数配置----23
5.3 TxPDO ----28
5.4 RxPDO ----29
第 6 章 使用例程 ----30
6.1 IEC 控制器示例 ----30
6.1.1 硬件连接----30
6.1.2 EtherCAT 主站的添加及配置 ----31
6.1.3 模块的添加 ----35
6.1.4 模块的配置----39
6.1.5 应用例程----40
6.2 BASIC 控制器示例----50
6.2.1 硬件连接----50
6.2.2 EtherCAT 主站的添加及配置 ----51
6.2.3 模块的添加----52
6.2.4 模块的配置----58
6.2.5 BASIC 应用例程 ----58
6.2.6 API 应用例程 ----63
6.3 控制卡示例 ----65
6.3.1 硬件连接----65
6.3.2 从站 ID 设置 ----65
6.3.3 组建 EtherCAT 网络----65
6.3.4 应用例程----66

第 1 章 产品概述

1.1 产品简介

雷赛 EM64DX-E1 模块是一款基于 ASIC 技术的高性能、高可靠性的 EtherCAT 总线 IO 扩展模块，具有 32 路通用输入接口和 16 路通用输出接口以及 16 路双通道输入输出接口。输入输出接口均采用光电隔离和滤波技术，可以有效隔离外部电路的干扰，以提高系统的可靠性。

EM64DX-E1 主要用于与雷赛公司的支持 EtherCAT 总线通讯的控制卡和控制器配套使用。

1.2 产品特点

① 32 路通用输入：提供光电隔离、抗干扰滤波。

② 16 路通用输出：提供光电隔离、抗干扰滤波。

③ 16 路双通道输入输出（可配置为 16 路输入或 16 路输出）：提供光电隔离、抗干扰滤波。

④ 内部 24V 隔离电源，具有直流滤波器。

⑤ 铁壳安装，插拔式接线端子，支持螺丝安装和导轨安装。

1.3 技术规格

EM64DX-E1 IO 扩展模块的主要规格指标如下：

表 1.1 EM64DX-E1 规格指标

输入特性		输出特性	
IO 端子排	直插按压式	IO 端子排	直插按压式
输入通道数	32~48 路	输出通道数	16~32 路
EtherCAT 指示灯	RUN 指示灯、L/A 指示灯、ERR 指示灯	EtherCAT 指示灯	RUN 指示灯、L/A 指示灯、ERR 指示灯
其他指示灯	POWER 指示灯、RUN 指示灯、ERR 指示灯	其他指示灯	POWER 指示灯、RUN 指示灯、ERR 指示灯
输入端子指示灯	1 个绿色 LED/通道	输出端子指示灯	1 个绿色 LED/通道

输入类型	低电平输入有效（NPN）	输出类型	晶体管(NMOS 漏型输出)
输入电流	5-10mA	负载电压	5~24V DC
响应频率	1Hz-4KHz	响应频率	1Hz-4KHz（通用输出）
输入阻抗	≤5.6kΩ		
输入 ON/OFF 响应时	20us/50us	输出 ON/OFF 响应时间	20us/60us
输入电压	21~27V DC	输出电流	500mA(峰值/全负载 300mA)
额定输入电压	24V DC		
最大连续电压	30V DC	漏电流	最大 8uA/通道
浪涌	35V DC，500ms	浪涌电流	2A，100ms
导通电流	3.5mA 以上/5V 以下		
关断电流	1.5mA 以下/19V 以上		
光隔离	500V AC，1 Minute	光隔离	500V AC，1 Minute
隔离组数	单独隔离/通道	隔离组数	单独隔离/通道
输入滤波时间	软件滤波		
输入保护	光电隔离、滤波	输出保护	光电隔离、滤波
拨码开关			
自适应/拨码开关 4 位，复用口默认为输出通道			
旋转开关			
2 个 16 位旋转开关设定站号			
运行环境			
环境温度	存储温度：-30 ° C ~ 70 ° C		
	工作温度：-10 ° C ~ 55 ° C		
相对湿度	95%无凝结		
跌落实验			
自由落体	0.5 m，5 次，正常工作		
认证			
CE 认证、ETG 认证			

1.4 安装使用

EM64DX-E1 IO 扩展模块采用定位孔的方式安装和导轨安装，安装尺寸如图 1.1、1.2 所示(单位均为 mm)：

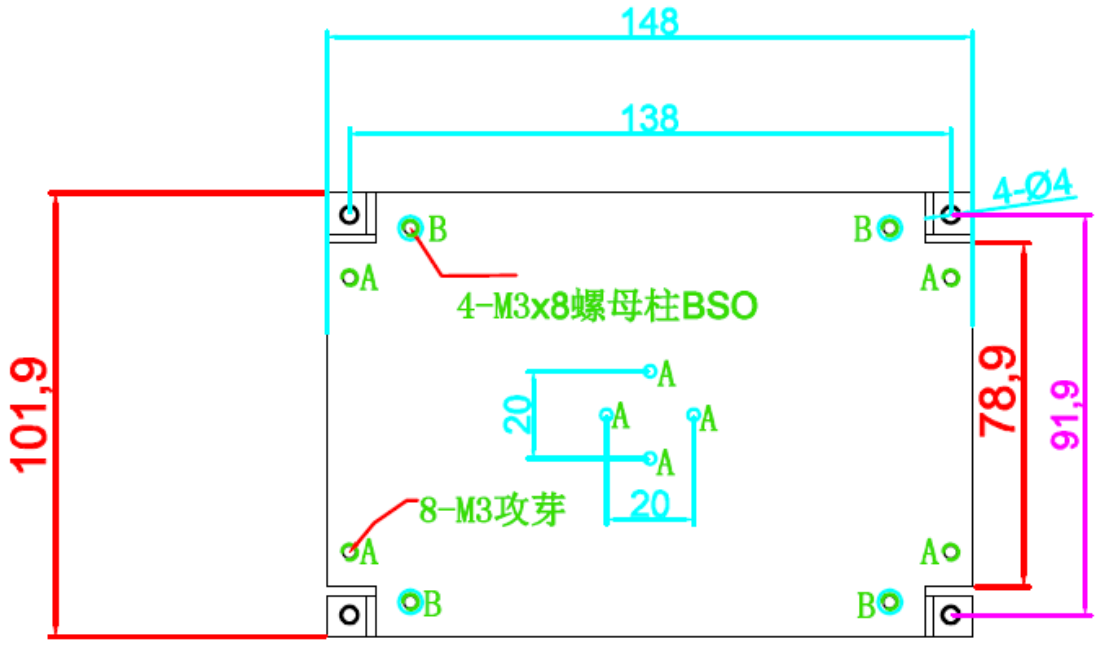


图 1.1 安装底板俯视图

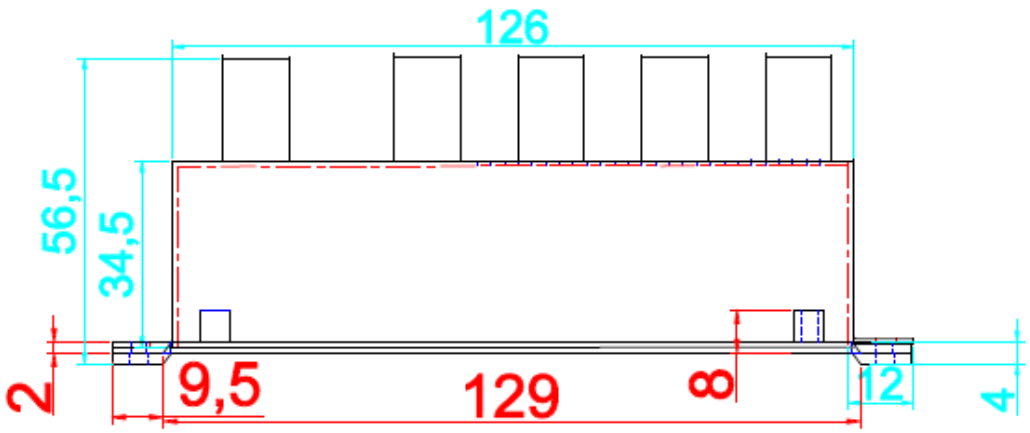


图 1.2 安装底板正视图

第 2 章　产品外观及硬件接线

2.1　产品外观

EM64DX-E1 IO 扩展模块提供 32 路输入接口、16 路输出接口以及 16 路双通道输入输出接口，带有两个立式 RJ45 型 EtherCAT 扩展口，产品外观如图 2.1 所示。

图 2.1 EM64DX-E1 产品外观图

2.2　接口分布及针脚定义

EM64DX-E1 IO 扩展模块硬件接口分布如图 2.2 所示，其接口定义表如表 2.1 所示。

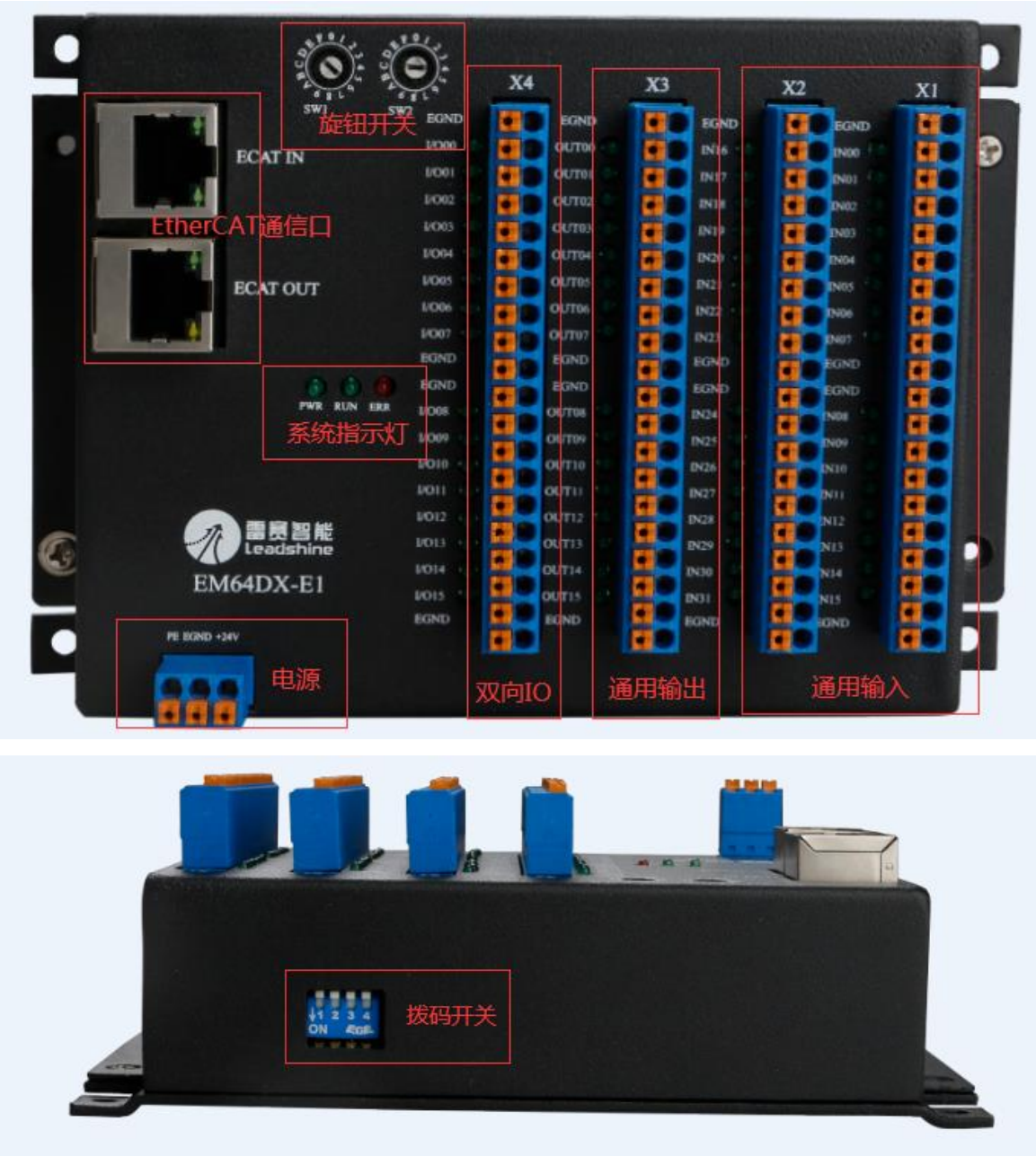


图 2.2 EM64DX-E1 硬件接口分布图

表 2.1 接口功能简述

名称	功能介绍
电源	直流 24V 电源输入
ECAT IN	EtherCAT 总线 IN 接口
ECAT OUT	EtherCAT 总线 OUT 接口
X1/X2	通用输入端口
X3	通用输出端口

X4	输入输出两用端口（由拨码开关决定）
拨码开关	输入输出两用端口的配置开关
旋钮开关	模块 ID 号

2.2.1 电源接口

电源接口为 24V 电源输入接口，标有+24V 的端子接+24V，标有 EGND 的端子接外部电源地。PE 为外壳地接口。

2.2.2 EtherCAT 接口定义

接口 ECAT IN、ECAT OUT 是 EtherCAT 总线接口，采用 RJ45 端子，其引脚号和信号对应关系见表 2.2 所示：

表 2.2 接口ECAT IN、ECAT OUT引脚号和信号关系表

ECAT-IN 信号	信号描述	ECAT-OUT 信号	信号描述	说明
1	TX+	1	TX+	发送信号+
2	TX-	2	TX-	发送信号-
3	RX+	3	RX+	接收信号+
4	NC	4	NC	保留
5	NC	5	NC	保留
6	RX-	6	RX-	接收信号-
7	NC	7	NC	保留
8	NC	8	NC	保留

2.2.3 X1/X2 接口定义

X1 接口具有 16 路通用输入（IN00-IN15），对应的引脚分布如下：

1	2	3	4	5	…	16	17	18	19	20
EGND	IN00	IN01	IN02	IN03	…	IN12	IN13	IN14	IN15	EGND

X2 接口具有 16 路通用输入（IN16-IN31），对应的引脚分布如下：

1	2	3	4	5	…	16	17	18	19	20
EGND	IN16	IN17	IN18	IN19	…	IN28	IN29	IN30	IN31	EGND

2.2.4 X3 接口定义

X3 接口具有 16 路通用输出（OUT00-OUT15），对应的引脚分布如下：

1	2	3	4	……	17	18	19	20
EGND	OUT00	OUT 01	OUT 02		OUT 13	OUT 14	OUT 15	EGND

2.2.5 X4 接口定义

X4 定义(输入或输出)根据拨码开关确定，出厂默认全为输出，四个拨码默认全 OFF。每个拨码控制四个端口，每个拨码相对独立，用户可以根据自己的习惯进行拨码。每个拨码对应控制的端口分布如下：

拨码号	控制对应端口编码	状态	控制对应 4 个端口的状态	对应端口号
1	I/000 I/001 I/002 I/003	ON	IN	当作为输入端口时，端口号顺序为 IN32-IN47； 当作为输出端口时，端口号顺序为 OUT16-OUT31；
		OFF	OUT	
2	I/004 I/005 I/006 I/007	ON	IN	
		OFF	OUT	
3	I/008 I/009 I/010 I/011	ON	IN	
		OFF	OUT	
4	I/012 I/013 I/014 I/015	ON	IN	
		OFF	OUT	

例：当把拨码 1 和 4 的状态拨到 ON 时，此时 I/O 00-I/O 03 对应的状态为 IN32-IN35，I/O 12-I/O 15 对应的状态为 IN44-IN47。I/O 04-I/O 11 对应的状态为 OUT20-OUT27。

2.3 接口电路

2.3.1 通用输入信号接口

EM64DX-E1 IO 扩展模块为用户提供 32 路通用数字输入接口，用于开关信号、传感器信号或其它信号的输入。其接口电路加有光电隔离元件，可以有效隔离外部电路的干扰，以提高系统的可靠性。其输入接口接线图如图 2.3 所示：

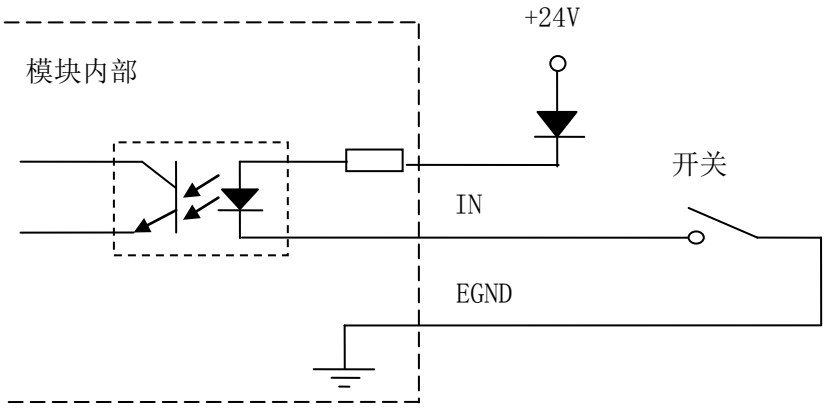


图 2.3 通用输入接线图

2.3.2 通用输出信号接口

EM64DX-E1 IO 扩展模块为用户提供了 16 路通用数字输出接口，由 MOS 管驱动，单路输出电流可达 0.3A，可用于对继电器、电磁阀、信号灯或其它设备的控制。其接口电路都加有光电隔离元件，可以有效隔离外部电路的干扰，提高了系统的可靠性。输出电路采用 OD 设计，上电默认 MOS 管关断。模块通用数字输出信号控制常用元器件的接法如下：

（1）通用发光二极管

通用数字输出接口控制发光二极管时，需要接一限流电阻 R，限制电流在 10ma 左右，电阻值大约在 2K 到 5K 左右，根据使用的电源来选择，电压越高，使用的电阻值越大些。接线图如图 2.4 所示。

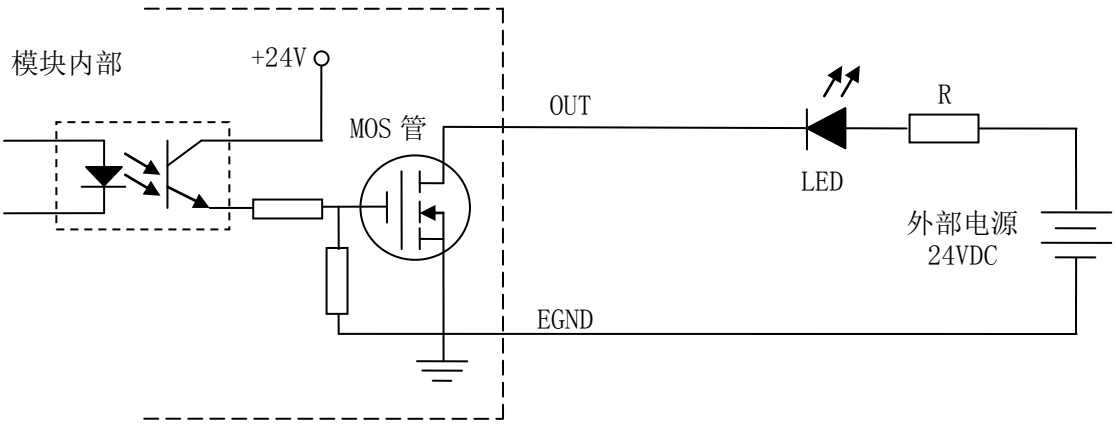


图 2.4 通用输出接线图

（2）灯丝型指示灯：

通用数字输出端口控制灯丝型指示灯时，为提高指示灯的寿命，需要接预热电阻 R，电阻值的大小，以电阻接上后输出口无输出时，灯不亮为原则。接线图如图 2.5 所示。

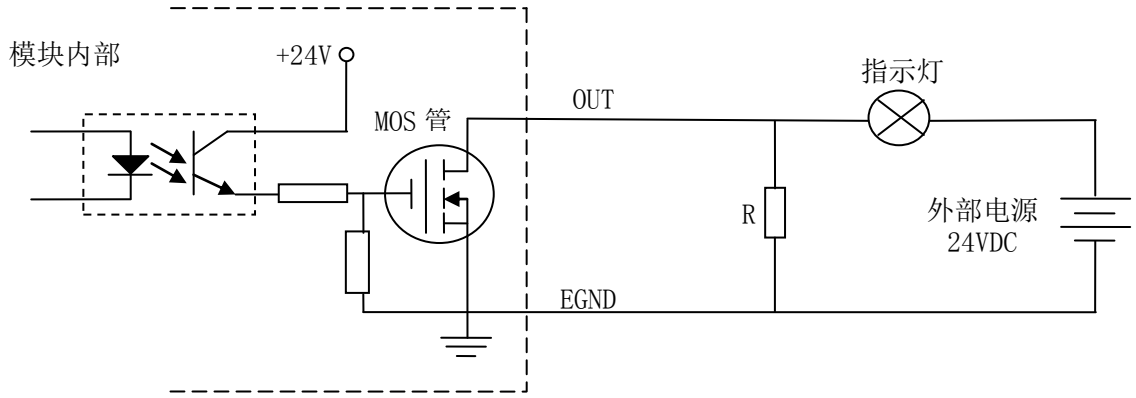


图 2.5　通用输出接线图

（3）小型继电器：

继电器为感性负载，当继电器突然关断时，其电感会产生一个很大的反向电压，有可能击穿输出 MOS 管，模块内输出口有续流二极管，以保护输出口 MOS 管。继电器接线图如图 2.6 所示。

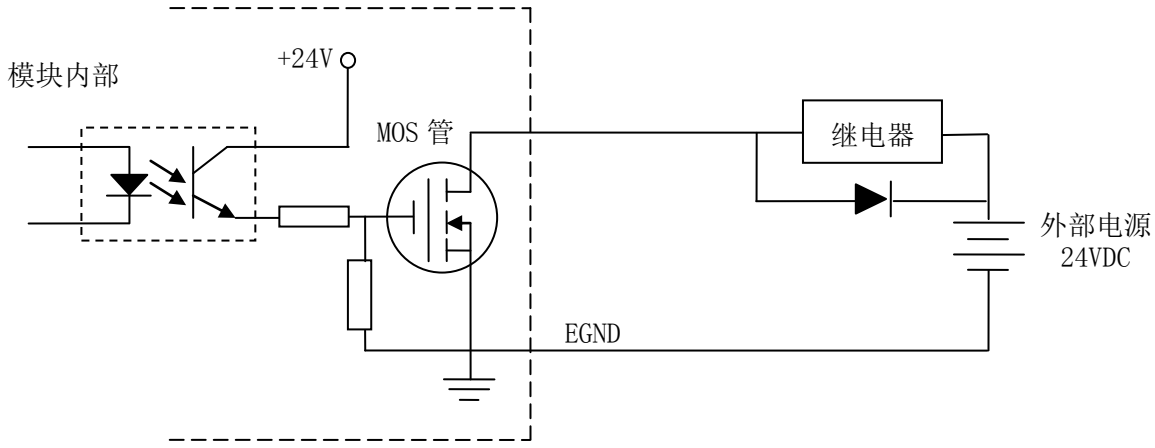


图 2.6　通用输出接线图

注　意：在使用通用数字输出端口时，切勿把外部电源直接接至通用数字输出端口上，否则会造成 MOS 管损坏。

2.3.3 可选配通用输入输出接口

可选配通用输入输出接口 X4，在默认情况下是普通输出接口，侧边拨码开关为 OFF。如果接口 X4 使用为输入接口，那么侧边拨码需要拨为 ON。此时输出电路的输出信号无效，不被输出。使用输入或者输出的接线方式和图 2.3-2.6 相同。内部电路图如图 2.7 所示：

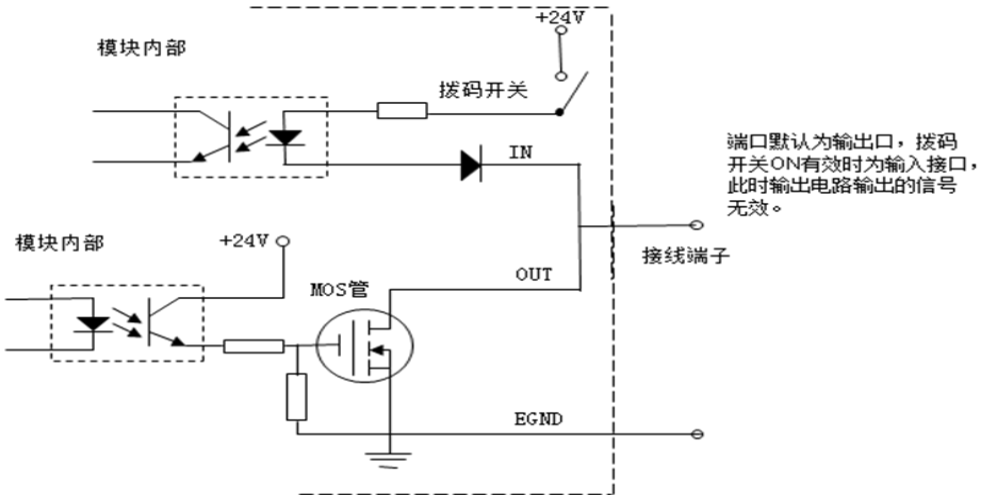


图 2.7 可选配接线电路图

第 3 章　指示灯定义及说明

3.1　指示灯定义

POWER：电源指示灯，用于指示模块+24V 的上电状态。

RUN：连接指示灯，用于指示模块当前系统的状态。

ERROR：系统错误指示灯。

RJ45 指示灯：包含绿色、红色两种指示灯，用于指示模块当前的通讯状态。

3.2　指示灯闪烁规则

EtherCAT 指示灯的闪烁遵循如图 3.1 所示的闪烁规则。

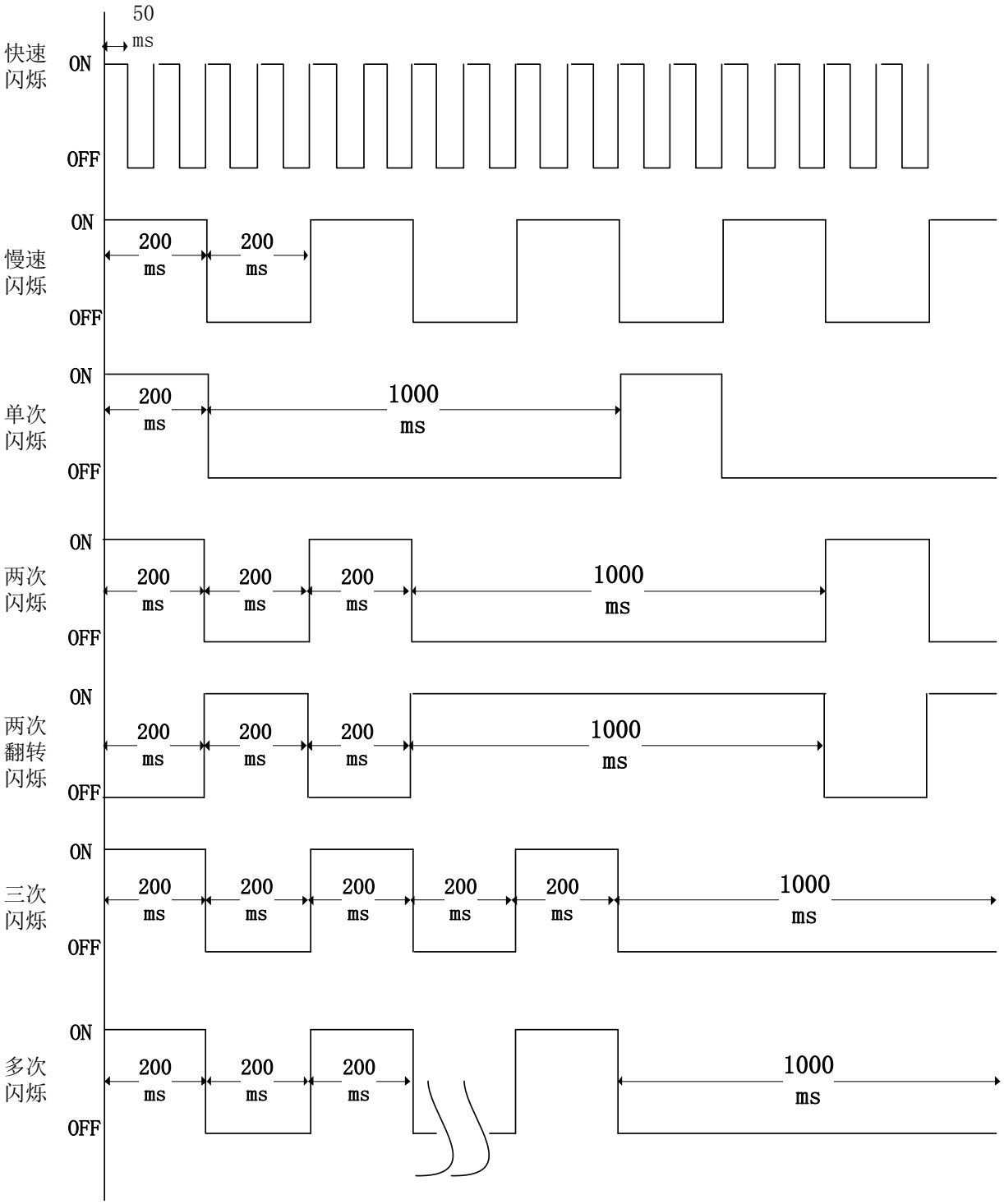


图 3.1 指示灯闪烁规则

3.3 指示灯状态

PWR 指示灯状态描述如表 3.1 所示：

表 3.1PWR 指示灯状态

PWR 指示灯	模块上电状态
常灭	模块没上电
常亮	模块已上电

RUN 指示灯状态描述如表 3.2 所示：

表 3.2RUN 指示灯状态

RUN 指示灯	模块系统状态
闪烁	系统正常运行
常亮	系统运行出现故障
	写 flash 参数时（5S 内）

ERR 指示灯状态描述如表 3.3 所示：

表 3.3ERR 指示灯状态

ERR 指示灯	模块系统状态
常灭	系统正常初始化
常亮	系统初始化出错

ECAT_RUN 指示灯状态描述如表 3.4 所示：

表 3.4 RUN 指示灯状态描述

RUN 指示灯	端口连接状态
常灭	从站处于初始化状态
慢速闪烁	设备处在预操作状态
单次闪烁	设备处在安全操作状态
常亮	设备处在操作状态

ECAT_ERROR 指示灯状态描述如表 3.5 所示：

表 3.5 ERROR 指示灯状态描述

ERROR 指示灯	错误指示
常灭	设备处于正常运行状态
慢速闪烁	通用配置错误
单次闪烁	由于本地错误，从站设备自动改变 EtherCAT 状态
两次闪烁	应用程序看门狗超时
常亮	PDI 接口看门狗超时

ECAT L/A 指示灯状态描述如表 3.6 所示：

表 3.6 L/A 绿色指示灯状态描述

RJ45 指示灯-绿灯	运行状态
常灭	无连接
常亮	连接正常但无数据交换
闪烁	连接正常并且有数据交换

第 4 章　功能说明

4.1 通用输入功能

EM64DX-E1 提供 32~48 路通用输入功能，当 X4 设置为输入时，输入端口为 IN0-IN47；当 X4 设置为输出时，输入端口为 IN0-IN31。能够检测外部信号的输入状态。

具体的映射端口号，和主站的本地输入端口数量以及从站的模块类型和数量类型有关，软件操作时请仔细核对端口号，以免出错。

4.2 输入计数功能

EM64DX-E1 提供 16 路输入计数功能，计数端口为 IN0-IN15。能够控制输入口进行计数。

4.3 通用输出功能

EM64DX-E1 提供 16~32 路通用输出功能，当 X4 设置为输入时，输出端口为 OUT0-OUT15；当 X4 设置为输出时，输出端口为 OUT0-OUT31。能够控制和读取输出信号的状态。

具体的映射端口号，和主站的本地输出端口数量以及从站的模块类型和数量类型有关，软件操作时请仔细核对端口号，以免出错。

4.4 输出延时翻转功能

EM64DX-E1 提供 8 路输出延时翻转功能，输出端口为 OUT0-OUT7。能够控制输出口进行延时翻转。

4.5 总线掉线输出保持功能

EM64DX-E1 提供 32 路总线掉线输出保持功能，输出端口为 OUT0-OUT31（当 X4 设置为输出时），可按位单独控制，由对象字典 7010H 控制。能够控制总线掉线时输出口保持。

4.6 总线复位输出保持功能

EM64DX-E1 提供 32 路总线复位输出保持功能，输出端口为 OUT0-OUT31（当 X4 设置为输出时），需结合总线掉线输出保持功能一起使用。能够控制总线掉线时输出口保持。具体输出保持结果有下表四种情况：

情况	总线复位输出保持设置	总线掉线输出保持设置	复总线位输出保持状态
1	0（不保持）	0（不保持）	0（不保持）
2	0（不保持）	1（保持）	0（不保持）
3	1（保持）	0（不保持）	0（不保持）
4	1（保持）	1（保持）	1（保持）

相关操作函数：以控制器为例

nmcs_set_slave_output_retain(ConnecNo, Enable)；当 Enable 设置为 0 时，总线复位功能无效；当 Enable 设置为 1 时，总线复位功能有效；

4.7 从站站号拨码功能

选择读取拨码作为从站站号时，需先设置对象字典 3041H 子索引 01H 为 1，拨完码后需将模块断电重启，新的拨码才能生效。（目前雷赛主站暂不支持读取模块拨码作为从站站号）

第 5 章 对象字典

5.1 通用参数

索引	子索引	名称	数据类型	访问属性	描述
1000H	00H	Device type	Unsigned32	ro	Device type and profile（设备类型） 初始值：0x0FFF0192
1008H	00H	Device name	Vis String8	ro	Manufacturer’s designation 初始值：EM64DX-E1
1009H	00H	Hardware version	Vis String8	ro	Hardware version 初始值：1.0
100AH	00H	Software version	Vis String8	ro	Software version 初始值：1.2
1018H		Identity		ro	（设备信息）
	00H	Largest sub-index	Unsigned8	ro	Largest sub-index supported » 04h
	01H	Vendor ID	Unsigned32	ro	Vendor ID 初始值：0x00004321
	02H	Product code	Unsigned32	ro	Product code 初始值：0x11100093
	03H	Revision	Unsigned32	ro	Revision number 初始值：0x19080210
	04H	Serial number	Unsigned32	ro	Serial number 初始值：0x00000001

5.2 参数配置（SDO）

输入计数参数配置					
6020H	00H	IN0 计数			
	01H	IN0_SetCountMode	Signed32	r/w	初始值默认为 0 设置 IN0 的计数方式：0 电平下降沿，1 电平上升沿， 2 电平任意沿
	02H	IN0_SetCountVal	Unsigned32	r/w	设置 IN0 的计数值，初始值默认为 0
	03H	IN0_ReadCountVal	Unsigned32	ro	读取 IN0 的计数值
6021H	00H	IN1 计数			
	01H	IN1_SetCountMode	Signed32	r/w	
	02H	IN1_SetCountVal	Unsigned32	r/w	设置 IN1 的计数值，初始值默认为 0
	03H	IN1_ReadCountVal	Unsigned32	ro	读取 IN1 的计数值
6022H	00H	IN2 计数			
	01H	IN2_SetCountMode	Signed32	r/w	
	02H	IN2_SetCountVal	Unsigned32	r/w	设置 IN2 的计数值，初始值默认为 0
	03H	IN2_ReadCountVal	Unsigned32	ro	读取 IN2 的计数值
6023H	00H	IN3 计数			
	01H	IN3_SetCountMode	Signed32	r/w	
	02H	IN3_SetCountVal	Unsigned32	r/w	设置 IN3 的计数值，初始值默认为 0
	03H	IN3_ReadCountVal	Unsigned32	ro	读取 IN3 的计数值
6024H	00H	IN4 计数			
	01H	IN4_SetCountMode	Signed32	r/w	

	02H	IN4_SetCountVal	Unsigned32	r/w	设置 IN4 的计数值，初始值默认为 0
	03H	IN4_ReadCountVal	Unsigned32	ro	读取 IN4 的计数值
6025H	00H	IN5 计数			
	01H	IN5_SetCountMode	Signed32	r/w	
	02H	IN5_SetCountVal	Unsigned32	r/w	设置 IN5 的计数值，初始值默认为 0
	03H	IN5_ReadCountVal	Unsigned32	ro	读取 IN5 的计数值
6026H	00H	IN6 计数			
	01H	IN6_SetCountMode	Signed32	r/w	
	02H	IN6_SetCountVal	Unsigned32	r/w	设置 IN6 的计数值，初始值默认为 0
	03H	IN6_ReadCountVal	Unsigned32	ro	读取 IN6 的计数值
6027H	00H	IN7 计数			
	01H	IN7_SetCountMode	Signed32	r/w	
	02H	IN7_SetCountVal	Unsigned32	r/w	设置 IN7 的计数值，初始值默认为 0
	03H	IN7_ReadCountVal	Unsigned32	ro	读取 IN7 的计数值
6028H	00H	IN8 计数			
	01H	IN8_SetCountMode	Signed32	r/w	
	02H	IN8_SetCountVal	Unsigned32	r/w	设置 IN8 的计数值，初始值默认为 0
	03H	IN8_ReadCountVal	Unsigned32	ro	读取 IN8 的计数值
6029H	00H	IN9 计数			
	01H	IN9_SetCountMode	Signed32	r/w	
	02H	IN9_SetCountVal	Unsigned32	r/w	设置 IN9 的计数值，初始值默认为 0
	03H	IN8_ReadCountVal	Unsigned32	ro	读取 IN9 的计数值

602AH	00H	IN10 计数			
	01H	IN10_SetCountMode	Signed32	r/w	
	02H	IN10_SetCountVal	Unsigned32	r/w	设置 IN10 的计数值，初始值默认为 0
	03H	IN10_ReadCountVal	Unsigned32	ro	读取 IN10 的计数值
602BH	00H	IN11 计数			
	01H	IN11_SetCountMode	Signed32	r/w	
	02H	IN11_SetCountVal	Unsigned32	r/w	设置 IN11 的计数值，初始值默认为 0
	03H	IN11_ReadCountVal	Unsigned32	ro	读取 IN11 的计数值
602CH	00H	IN12 计数			
	01H	IN12_SetCountMode	Signed32	r/w	
	02H	IN12_SetCountVal	Unsigned32	r/w	设置 IN12 的计数值，初始值默认为 0
	03H	IN12_ReadCountVal	Unsigned32	ro	读取 IN12 的计数值
602DH	00H	IN13 计数			
	01H	IN13_SetCountMode	Signed32	r/w	
	02H	IN13_SetCountVal	Unsigned32	r/w	设置 IN13 的计数值，初始值默认为 0
	03H	IN13_ReadCountVal	Unsigned32	ro	读取 IN13 的计数值
602EH	00H	IN14 计数			
	01H	IN14_SetCountMode	Signed32	r/w	
	02H	IN14_SetCountVal	Unsigned32	r/w	设置 IN14 的计数值，初始值默认为 0
	03H	IN14_ReadCountVal	Unsigned32	ro	读取 IN14 的计数值
602FH	00H	IN15 计数			
	01H	IN15_SetCountMode	Signed32	r/w	
	02H	IN15_SetCountVal	Unsigned32	r/w	设置 IN15 的计数值，初始值默

					认为 0
	03H	IN15_ReadCountVal	Unsigned32	ro	读取 IN15 的计数值
输出延时翻转参数配置					
7020H	00H	OUT0 延时翻转设置			
	01H	OUT0_SetMod	Unsigned32	r/w	设置是否启用延时翻转： 0 不启用；1 遇低翻转；2 遇高翻转
	02H	OUT0_DelayTime	Unsigned32	r/w	设置输出端口延时翻转时间 （范围：0-100000，单位 ms）
7021H	00H	OUT1 延时翻转设置			
	01H	OUT1_SetMod	Unsigned32	r/w	设置是否启用延时翻转： 0 不启用；1 遇低翻转；2 遇高翻转
	02H	OUT1_DelayTime	Unsigned32	r/w	设置输出端口延时翻转时间 （范围：0-100000，单位 ms）
7022H	00H	OUT2 延时翻转设置			
	01H	OUT2_SetMod	Unsigned32	r/w	设置是否启用延时翻转： 0 不启用；1 遇低翻转；2 遇高翻转
	02H	OUT2_DelayTime	Unsigned32	r/w	设置输出端口延时翻转时间 （范围：0-100000，单位 ms）
7023H	00H	OUT3 延时翻转设置			
	01H	OUT3_SetMod	Unsigned32	r/w	设置是否启用延时翻转： 0 不启用；1 遇低翻转；2 遇高翻转
	02H	OUT3_DelayTime	Unsigned32	r/w	设置输出端口延时翻转时间 （范围：0-100000，单位 ms）
7024H	00H	OUT4 延时翻转设置			

	01H	OUT4_SetMod	Unsigned32	r/w	设置是否启用延时翻转： 0 不启用；1 遇低翻转；2 遇高翻转
	02H	OUT4_DelayTime	Unsigned32	r/w	设置输出端口延时翻转时间 （范围：0-100000，单位 ms）
7025H	00H	OUT5 延时翻转设置			
	01H	OUT5_SetMod	Unsigned32	r/w	设置是否启用延时翻转： 0 不启用；1 遇低翻转；2 遇高翻转
	02H	OUT5_DelayTime	Unsigned32	r/w	设置输出端口延时翻转时间 （范围：0-100000，单位 ms）
7026H	00H	OUT6 延时翻转设置			
	01H	OUT6_SetMod	Unsigned32	r/w	设置是否启用延时翻转： 0 不启用；1 遇低翻转；2 遇高翻转
	02H	OUT6_DelayTime	Unsigned32	r/w	设置输出端口延时翻转时间 （范围：0-100000，单位 ms）
7027H	00H	OUT7 延时翻转设置			
	01H	OUT7_SetMod	Unsigned32	r/w	设置是否启用延时翻转： 0 不启用；1 遇低翻转；2 遇高翻转
	02H	OUT7_DelayTime	Unsigned32	r/w	设置输出端口延时翻转时间 （范围：0-100000，单位 ms）
7010H	00H	输出口状态保持设置			
	01H	Keep_state_on_reset	Unsigned32	r/w	断线时是否保持输出口状态， 该参数的 32 个 bit 对应 32 个输出口的状态： 0：断线时该输出口状态复位，

					输出高电平； 1：断线时保持原有状态
3040H	00H	Station alias MSB			
	01H	Station_alias_set	Unsigned16	r/w	设置从站站号高 8 位(低 8 位由模块上的拨码开关决定)
3041H	00H	Station alias select			
	01H	Station_alias_select	Unsigned16	r/w	选择从站站号设置模式： 0：由主站设置，保存到从站 EEPROM(默认) 1：通过对象 0x3040（高 8 位）和从站站号拨码(低 8 位)进行设置

备注：

① 总线掉线输出口状态保持（7010H）和总线复位输出口状态保持设置时需写 FLASH，写过程中 RUN 灯常亮，RUN 灯常亮时不允许进行 SDO 操作，不允许断电。RUN 灯开始正常闪烁说明写 flash 完成，完成后参数即时生效。

② 选择从站站号设置时（3040H、3041H）需写 FLASH，写过程中 RUN 灯常亮，RUN 灯常亮时不允许进行 SDO 操作，不允许断电。RUN 灯开始正常闪烁说明写 flash 完成，完成后参数即时生效。

5.3 TxPDO

索引	子索引	名称	数据类型	访问属性	描述
TxPDO0 0x1A00 : IN					
6000H	00H	IN			
	01H	IN1	Unsigned16	ro	输入 0-15
	02H	IN2	Unsigned16	ro	输入 16-31
	03H	IN3	Unsigned16	ro	输入 32-47

5.4 RxPDO

索引	子索引	名称	数据类型	访问属性	描述
RxPDO0 0x1600 : OUT					
7000H	00H	OUT			
	01H	OUT1	Unsigned16	rw	输出 0-15
	02H	OUT2	Unsigned16	rw	输出 16-31

第 6 章　使用例程

雷赛数字 IO 模块 EM64DX-E1 符合 EtherCAT 标准，是一个标准的 EtherCAT 从站，通过 EtherCAT 总线端口可以支持 EtherCAT 总线主站的扩展使用，如雷赛 DMC-E3032 控制卡、雷赛 SMC600-IEC 系列控制器、PMC300 系列控制器、BAC300 系列控制器和 PAC 系列运动控制器。以下分别以 DMC-E3032 控制卡、SMC606-IEC 和 BAC332E 运动控制器作为主站和 EM64DX-E1 作为从站配合使用为例介绍从站的使用方法。其中 DMC-E3032 控制卡使用 C#编程，SMC606-IEC 示例使用 IEC 编程方式，BAC332E 示例使用 BASIC 和 API 编程方式。

6.1　IEC 控制器示例

6.1.1 硬件连接

雷赛 SMC606 控制器的外形如下图 6.1 所示：

图 6.1 SMC606 外形

该控制器采用 24V 直流电源供电，具有 1 路 EtherCAT。该控制器的 EtherCAT 端口信号如表 6.1 所示：

表 6.1 接口引脚号和信号关系表

EtherCAT 信号	信号描述	说明
1	TX+	发送信号+
2	TX-	发送信号-
3	RX+	接收信号+
4	NC	保留
5	NC	保留
6	RX-	接收信号-
7	NC	保留
8	NC	保留

各端口的详细描述请参考 SMC600 系列运动控制器（IEC 版）用户手册。

设备间的连接:通过超五类带屏蔽层的网线将 SMC606 的 EtherCAT 口与 EM64DX-E1 的 ECAT IN 口连接。

模块上的拨码开关，采用出厂默认配置。

6.1.2 EtherCAT 主站的添加及配置

在IEC Studio中，先创建一个使用SMC606控制器的应用工程（详细的创建过程请参考《雷赛SMC IEC Studio使用手册》)。

在已经创建好的工程中，选择设备右击，在弹出的菜单中选择“添加设备”，如图6.2所示：

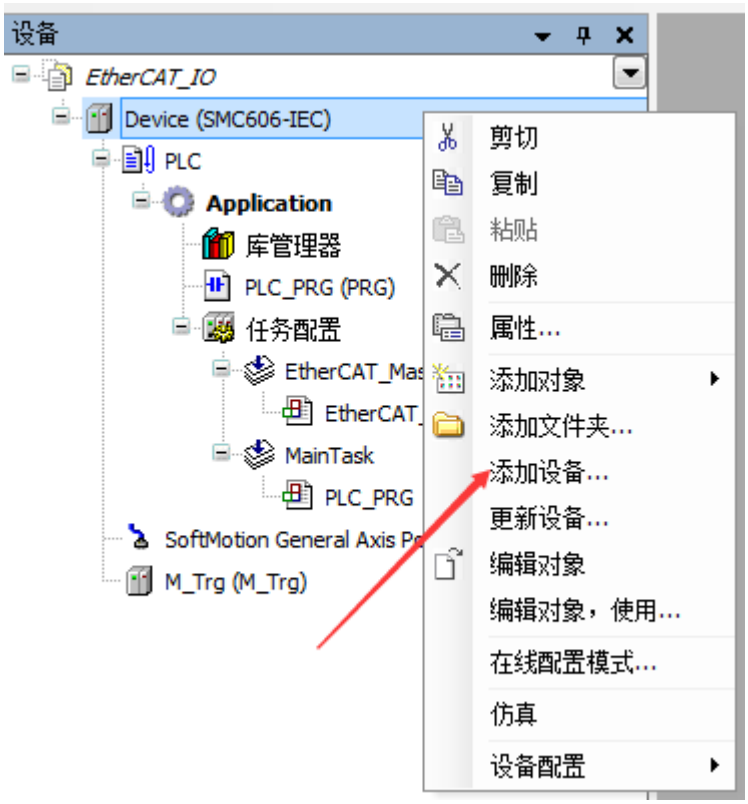


图 6.2 添加设备

在弹出的窗口中选择“现场总线”=>“EtherCAT”=>“EtherCAT Master”，然后点击添加设备，如图6.3所示：

图 6.3 添加 EtherCAT 总线

EtherCAT 任务配置：需将 EtherCAT 任务设置为最高优先级，将总线任务放在主任务中。如图 6.4 所示：

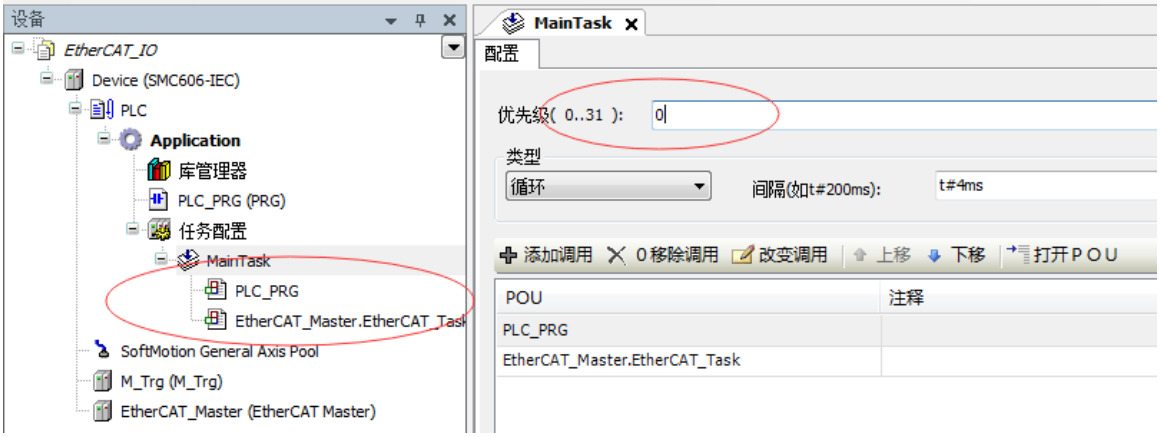


图 6.4 配置任务

注意：EtherCAT 任务与带运动模块的任务必须在同一个任务下，且为最高优先级。

主站配置：双击设备列表 EtherCAT 主站，弹出主站设置界面，如图 6.5 所示主站界面：

（1）通用界面（General）：

图 6.5 主站界面

主动配置主站/从站：主从站地址的配置方式。勾选此项，添加的主从站会自动配置地址。采用默认设置即可。

网络名称：采用默认设置，设置为 eth1。

总线周期时间（Cycle Time）：总线控制器支持 250us、500us、1ms、2ms、4ms 总线周期（根据总线控制器所带的负载而定），用户根据连接从站数量的多少选择合适的总线周期；

同步偏移（Sync Offset）：该值配置范围为 1~50，采用默认设置（默认值为 1）。该参数推荐值为 1 和 20。

诊断信息：用于实时显示主站的当前状态信息。如果显示“All slaves done！”，则表示主站配置已经完成，总线上所有从站为“操作状态”，如图 6.6 所示：

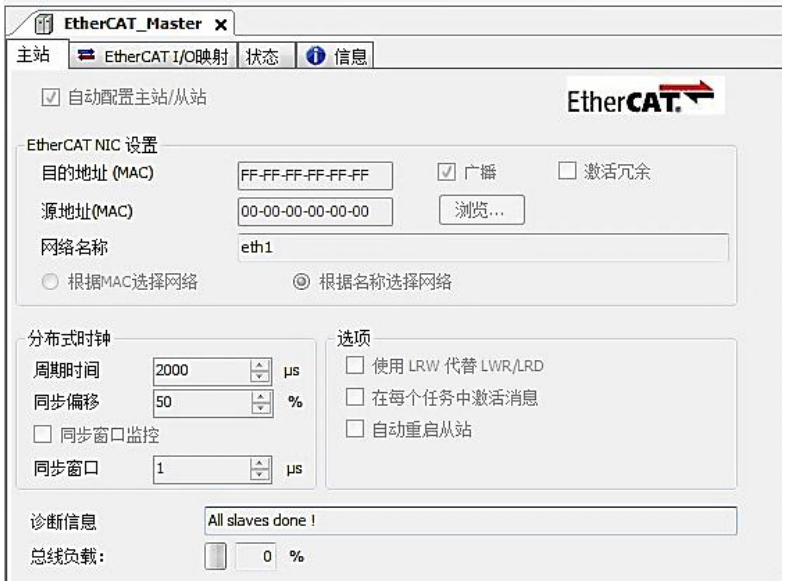


图 6.6 在线模式显示诊断信息

（2）状态界面（Status）：

在线模式下，状态界面处于观测状态，指示 EtherCAT 总线运行状态，如图 6.7 所示：

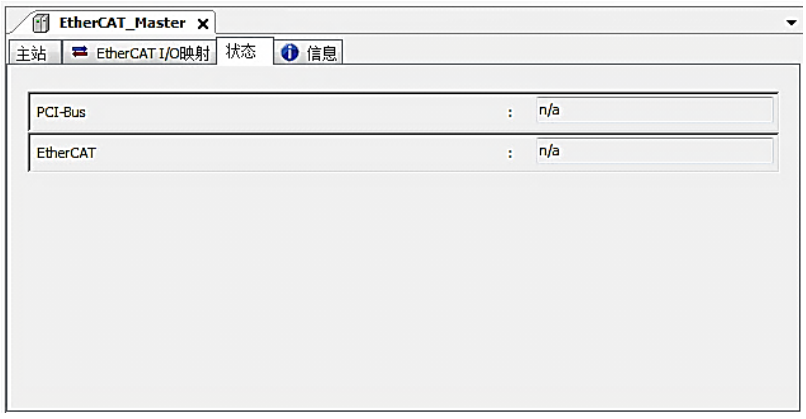


图 6.7 主站状态界面

（3）信息界面（Information）：

信息界面主要显示 EtherCAT 主站名称、厂商、类型、ID、版本及描述等信息，如图 6.8 所示：

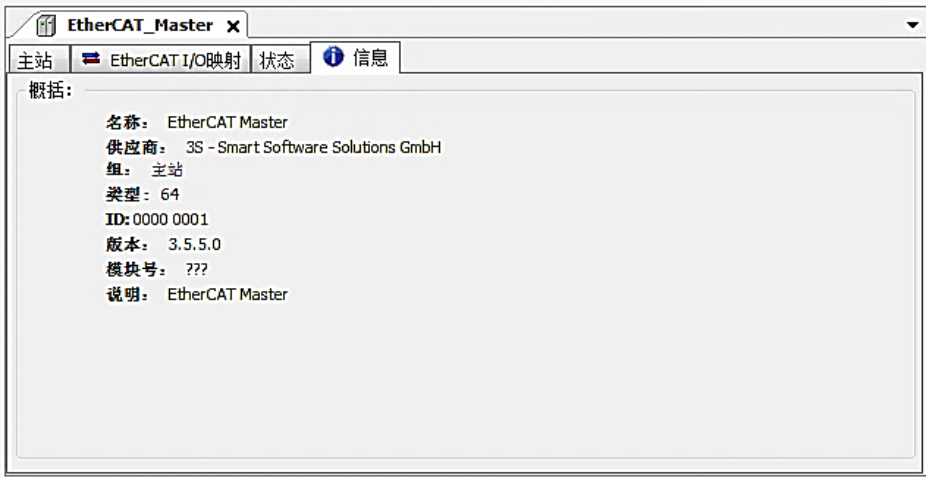


图 6.8 主站信息界面

6.1.3 模块的添加

在 Studio 中，添加 EtherCAT 从站模块有两种方式：手动添加方式和自动扫描方式。无论使用哪种方式，在添加从站之前，设备库中必须已经具有该设备（如果没有，请先添加该设备，具体的添加步骤请参考《雷赛 SMC IEC Studio 使用手册》）。

（1）手动添加模块

选择 EtherCAT_Master，右击选择“添加设备”如图 6.9 所示，在弹出的窗口选择“EtherCAT”=>“从站”=> “EM64DX-E1”然后点击添加设备。如图 6.10 所示。

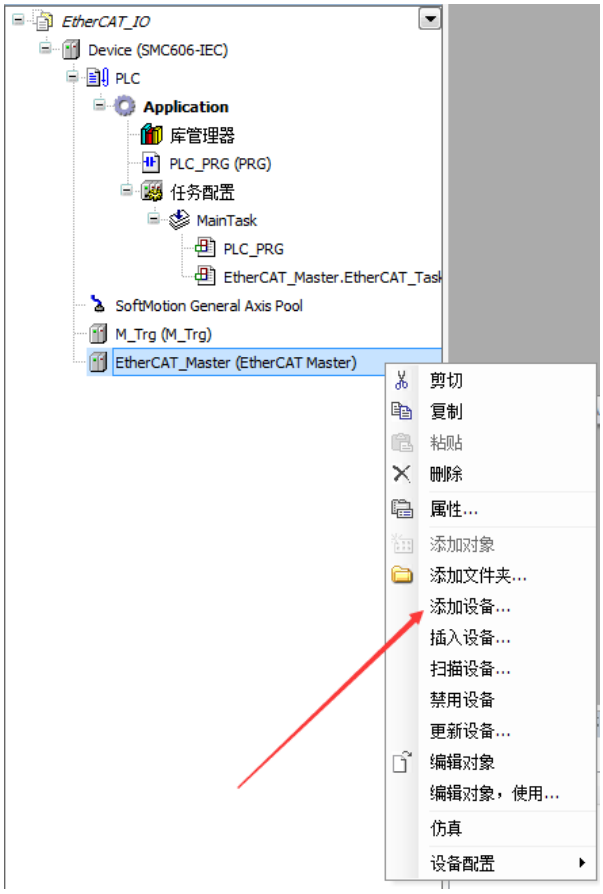


图 6.9 添加设备

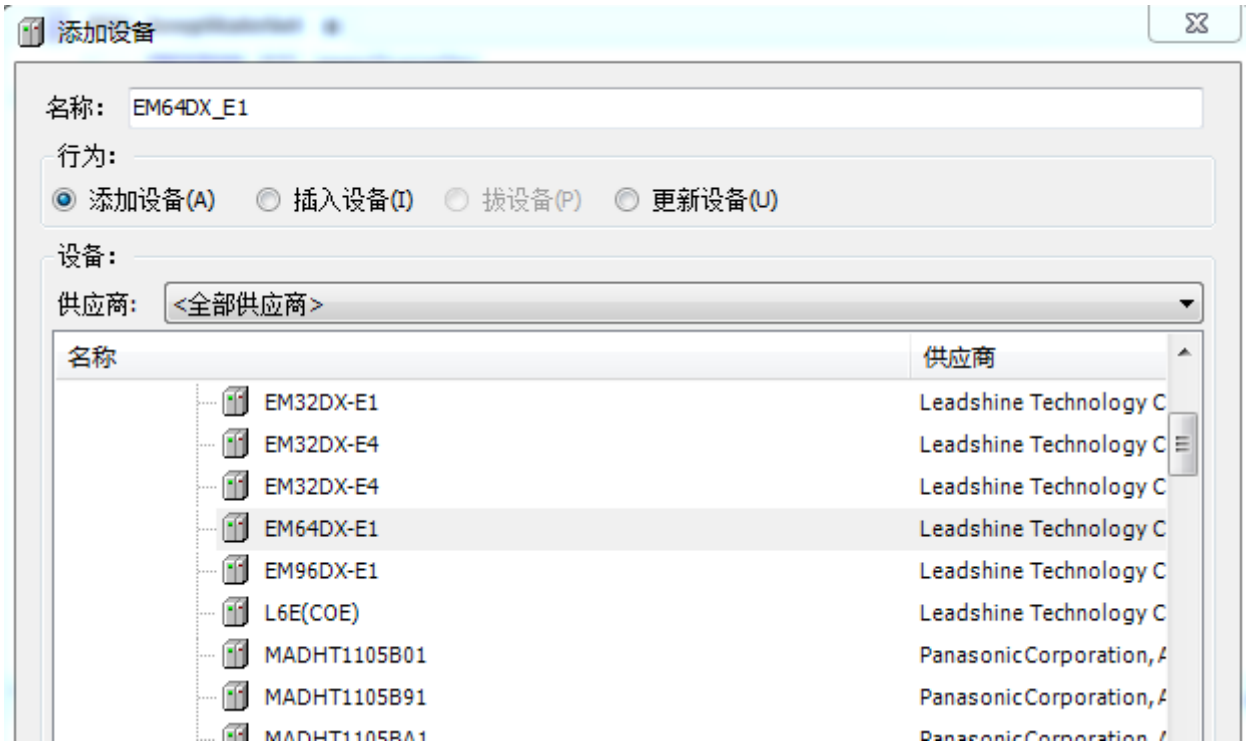


图 6.10 添加 EM64DX-E1 模块

（2）自动扫描添加设备

首先，双击“Device”，选择“扫描网络”，选择扫描出的设备后，点击“确定”，此时 Studio 已与控制器建立通讯，如图 6.11 所示：

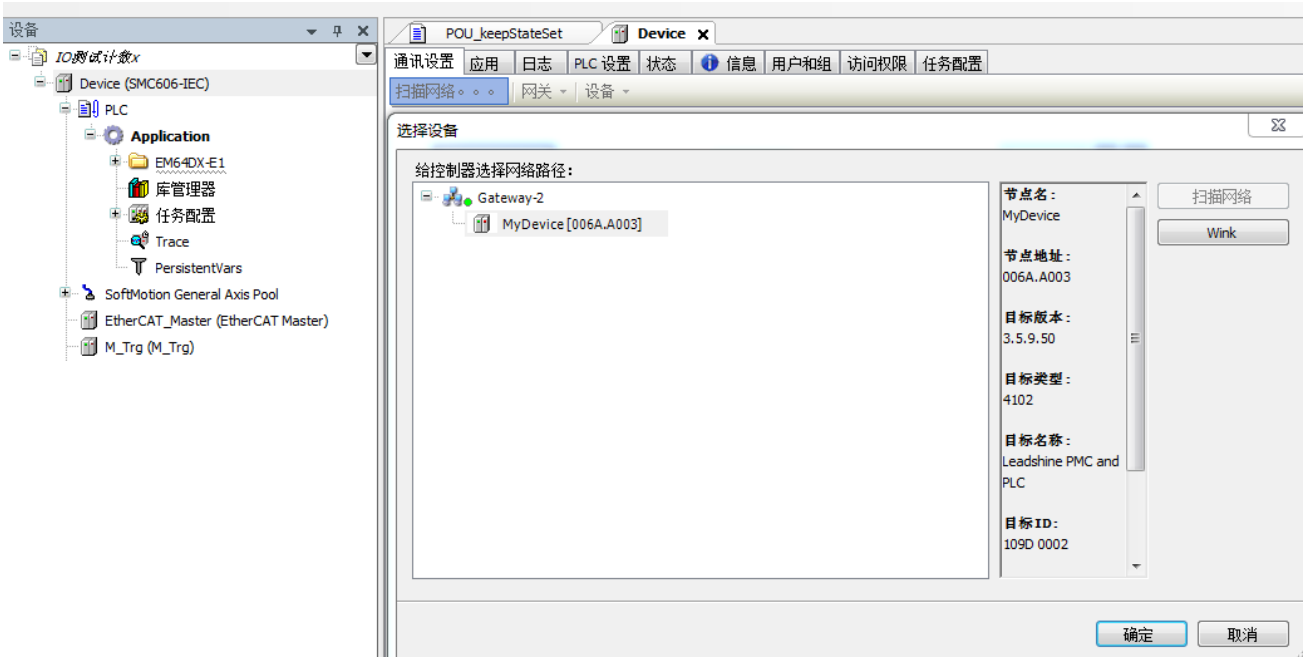


图 6.11 扫描网络

将当前应用工程下载到控制器中，然后，右击“EtherCAT_Master”选择“扫描设备”，如图 6.12 所示：

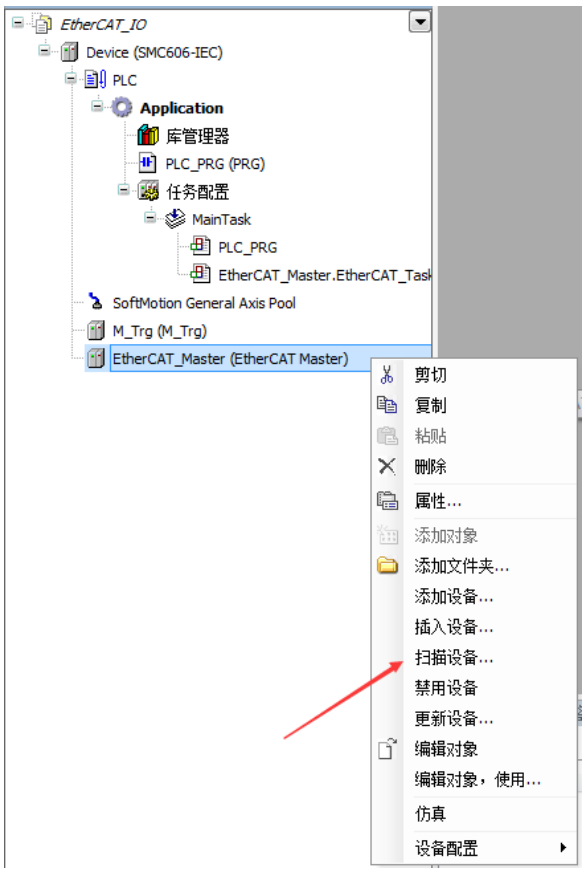


图 6.12 扫描设备

得到如图 6.13 所示设备列表，点击“复制所有设备到工程中”，左侧设备列表会自动添加扫描出来的从站，如图 6.14 所示。

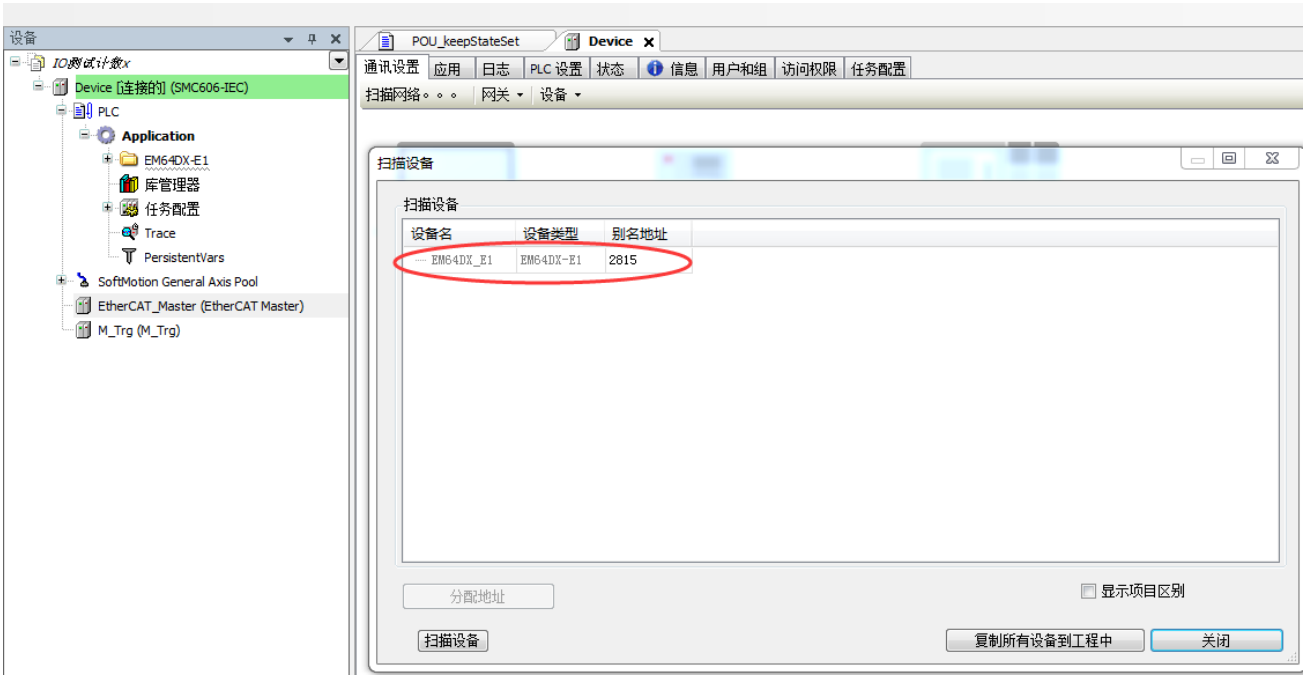


图 6.13 扫描网络

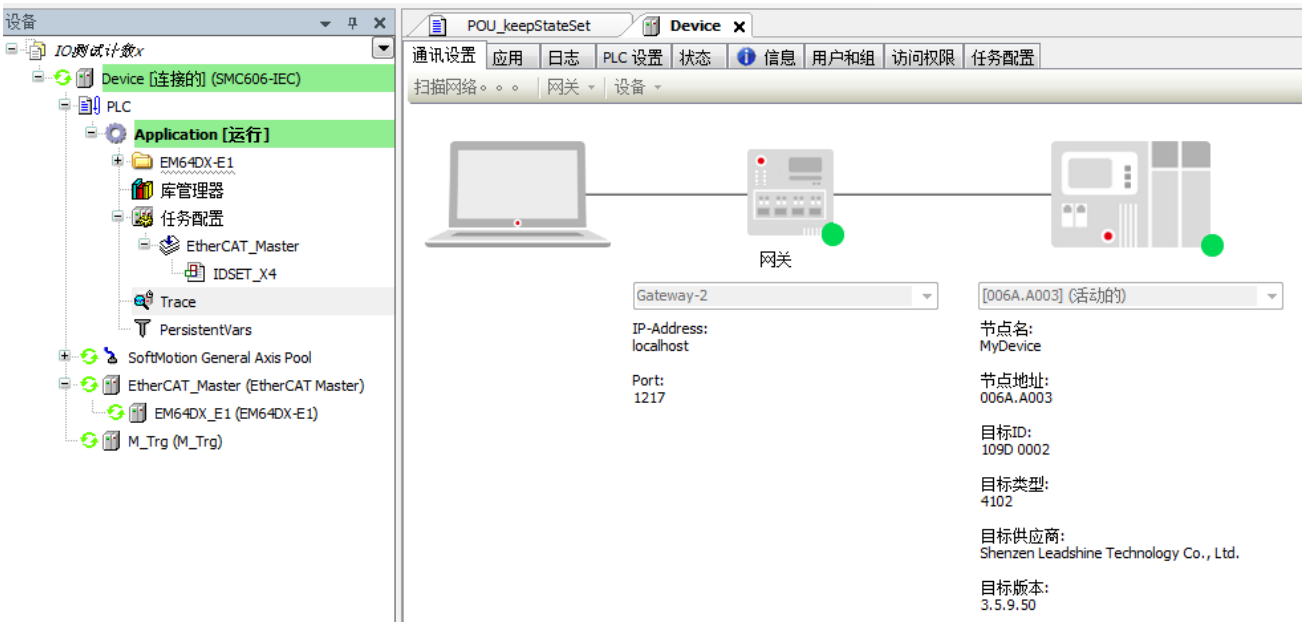


图 6.14 添加从站完成

6.1.4 模块的配置

双击左侧设备列表“EM64DX-E1”，可以看到从站的参数配置界面，如下图 6.15 所示。一般情况下，该页面参数采用默认配置。

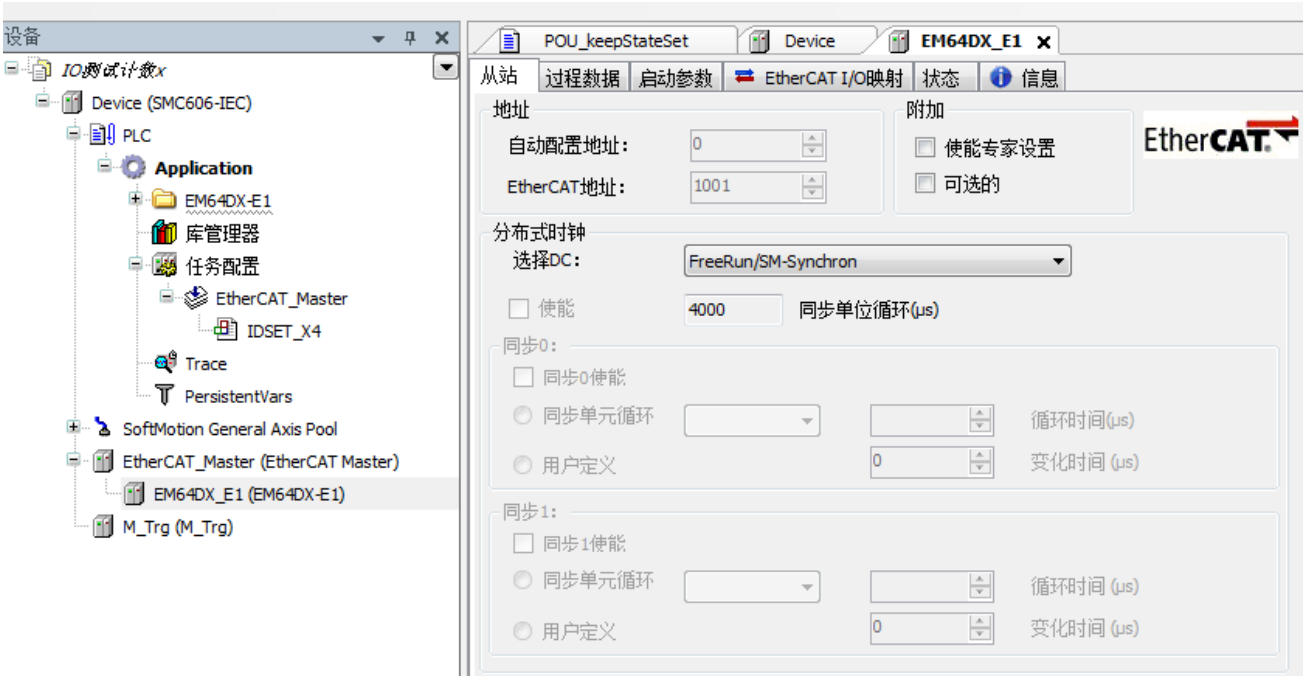


图 6.15 EM64DX_E1 参数配置界面

点击“EtherCAT I/O 映射”子页面，如下图 6.16 所示。该界面用于配置模块的输入输出参数，具体的用法请参考下一节。（注意：右下角的循环方式选择“ENABLE 2”）

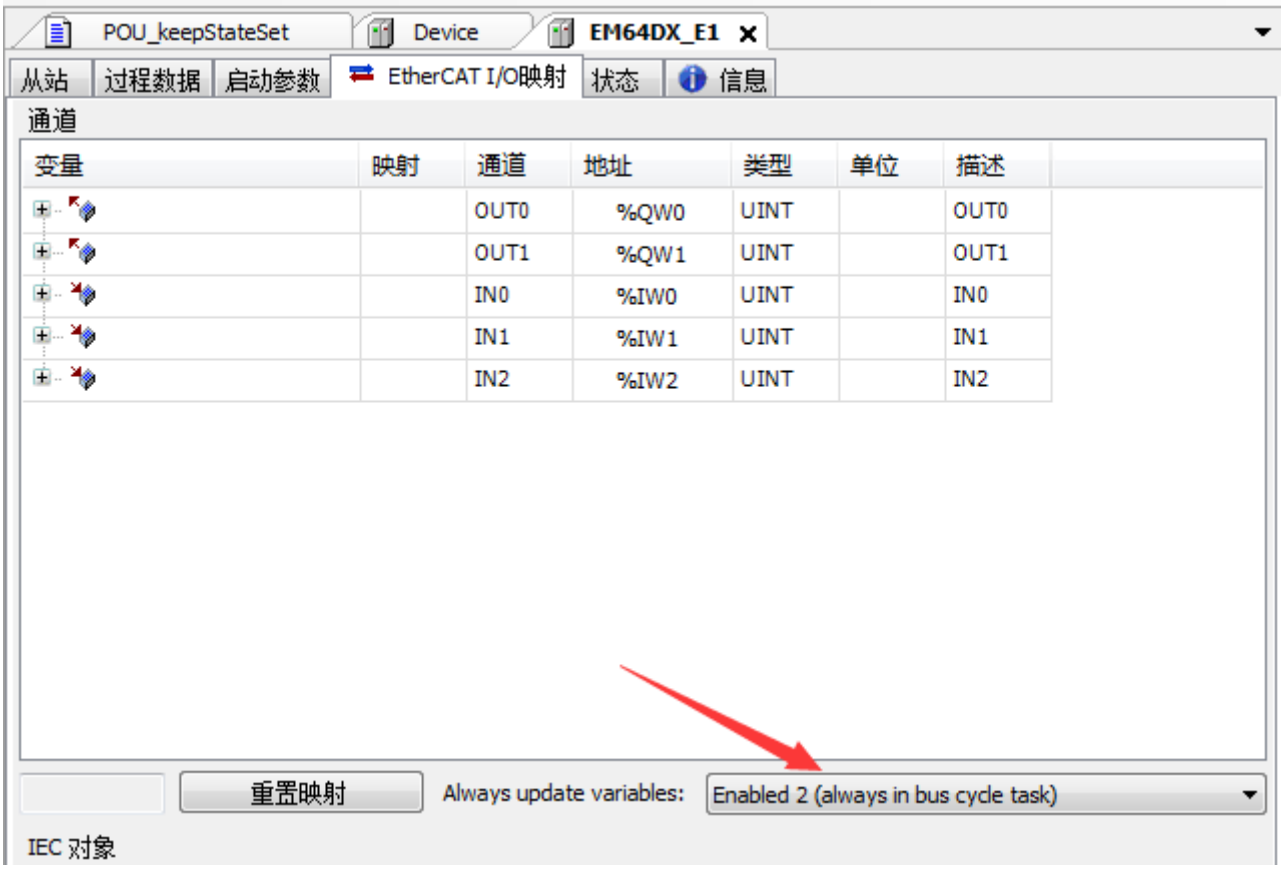


图 6.16 从站 I/O 映射配置界面

6.1.5 应用例程

（1）程序功能：

在 SMC606 控制器上实现对 EM64DX-E1 模块的输入状态读取、输出控制、输入计数、输出延时翻转、输出保持等功能。

（2）需要的资源：

“SMC606”库

（3）工程源码：

EtherCAT 扩展-“EtherCAT_IO”。

（4）编辑程序如下：

通用输入输出功能：

a. 在工程中调用总线控制器 SMC606 的 IO 数据处理模块 PD606_IO_Cmd。

b. 编写 IO 操作代码，如下图 6.17 所示。

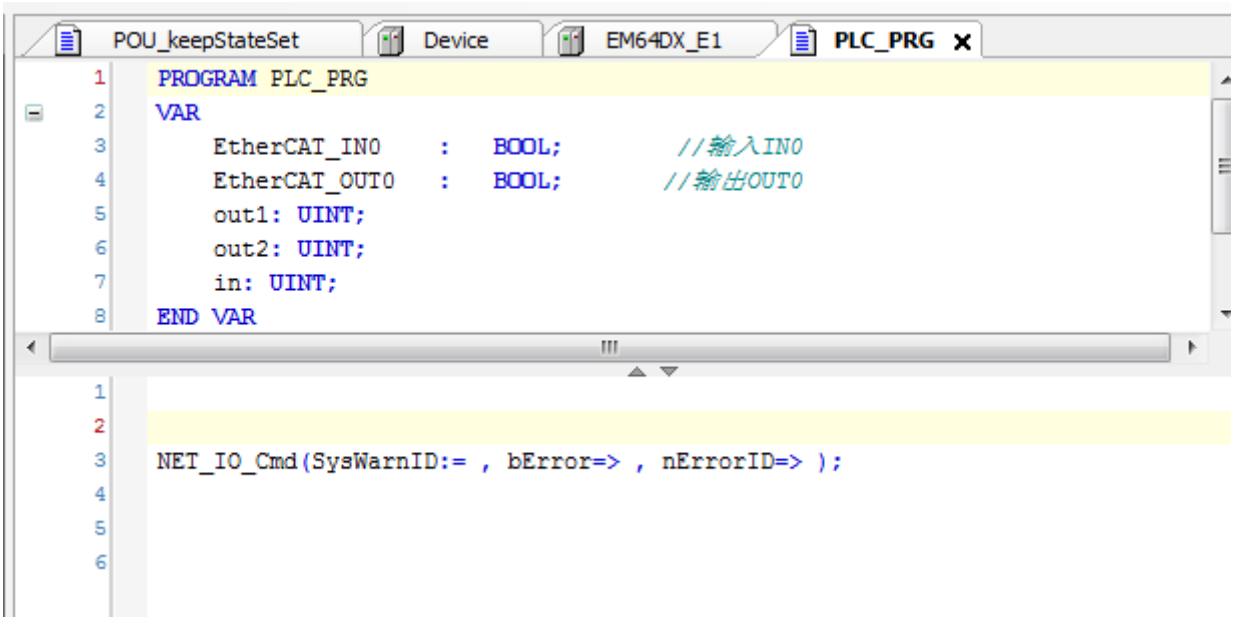
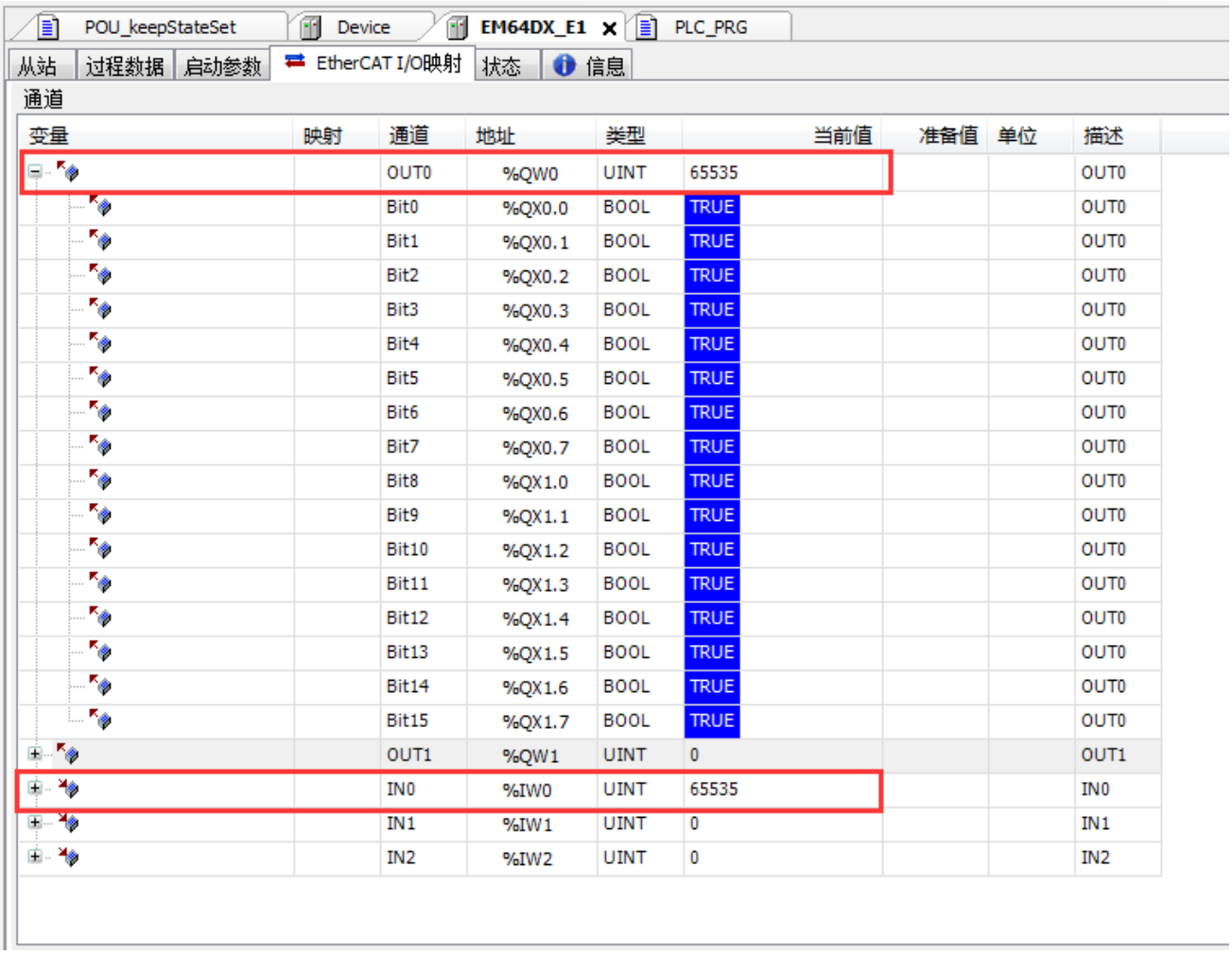


图 6.17 IO 操作代码界面

将 IN0 与 OUT0 短接，在“EtherCAT I/O 映射”界面操作 IO，给输出口有效电平（如图 6.18，输出口值：65535，OUT0-OUT15 处于有效状态），同时 IN0-IN15 也有效（值为：65535）



变量	映射	通道	地址	类型	当前值	准备值	单位	描述
		OUT0	%QW0	UINT	65535			OUT0
		Bit0	%QX0.0	BOOL	TRUE			OUT0
		Bit1	%QX0.1	BOOL	TRUE			OUT0
		Bit2	%QX0.2	BOOL	TRUE			OUT0
		Bit3	%QX0.3	BOOL	TRUE			OUT0
		Bit4	%QX0.4	BOOL	TRUE			OUT0
		Bit5	%QX0.5	BOOL	TRUE			OUT0
		Bit6	%QX0.6	BOOL	TRUE			OUT0
		Bit7	%QX0.7	BOOL	TRUE			OUT0
		Bit8	%QX1.0	BOOL	TRUE			OUT0
		Bit9	%QX1.1	BOOL	TRUE			OUT0
		Bit10	%QX1.2	BOOL	TRUE			OUT0
		Bit11	%QX1.3	BOOL	TRUE			OUT0
		Bit12	%QX1.4	BOOL	TRUE			OUT0
		Bit13	%QX1.5	BOOL	TRUE			OUT0
		Bit14	%QX1.6	BOOL	TRUE			OUT0
		Bit15	%QX1.7	BOOL	TRUE			OUT0
		OUT1	%QW1	UINT	0			OUT1
		IN0	%IW0	UINT	65535			IN0
		IN1	%IW1	UINT	0			IN1
		IN2	%IW2	UINT	0			IN2

6.18 EtherCAT I/O 映射界面

输入计数功能：

a. 选择 IN 计数器通道

计数器 0 所对应的对象字典的索引为 16#6020-计数器 15 的对象字典索引为 16#602f

示例代码如图 6.19 所示：

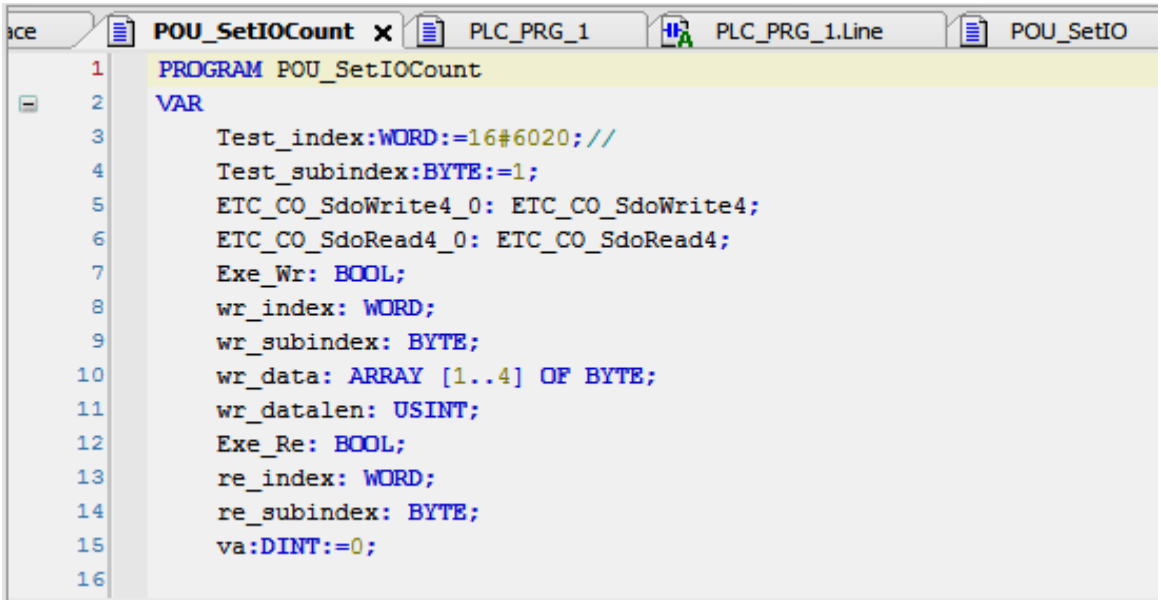


图 6.19 选择 IN 计数通道

b. 设置计数方式：

设置计数方式的对象字典子索引为 1。

计数器有三种计数方式：0 电平下降沿、1 电平上升沿 、2 电平任意沿

示例代码如图 6.20 所示：

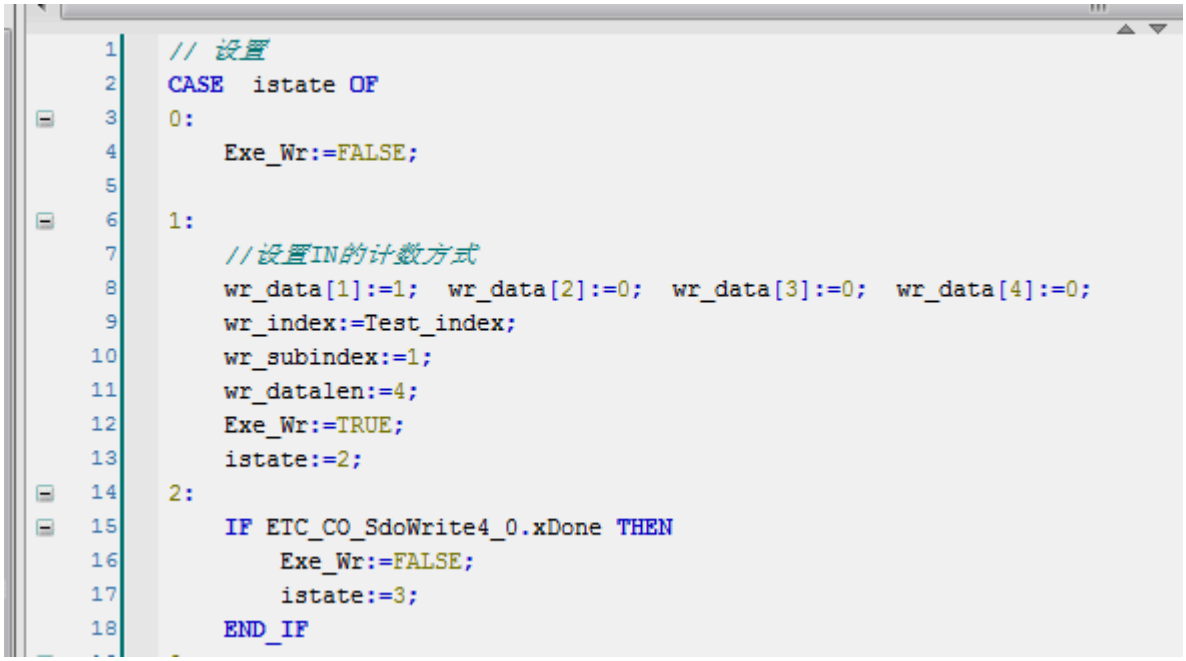


图 6.20 设置 IN 计数方式

c. 设置计数初始值：

设置计数值对象字典子索引为 2。

计数初始值有效范围：0 至 4294967295（无符号的 32 位值）

示例代码如图 6.21 所示：

```
3:
    //设置IN的计数值
    wr_data[1]:=0;  wr_data[2]:=0;  wr_data[3]:=0;  wr_data[4]:=0;
    wr_index:=Test_index;
    wr_subindex:=2;
    wr_datalen:=4;
    Exe_Wr:=TRUE;
    istate:=4;
4:
    IF ETC_CO_SdoWrite4_0.xDone THEN
        Exe_Wr:=FALSE;
        istate:=100;
    END_IF
100:
     ;
END_CASE

ACT_SDO();
```

图 6.21 设置计数值

d. 读取 IN 的计数值

读取计数值的对象字典子索引为 3。

示例代码如图 6.22 所示：

```
//读取标记
CASE istate_read OF
0:
    Exe_Re:=FALSE;

1:
    re_index:=Test_index;
    re_subindex:=3;
    Exe_Re:=TRUE;
    istate_read:=2;
2:
    IF ETC_CO_SdoRead4_0.xDone THEN
        Exe_Re:=FALSE;
        va:=Pack_ByteToDINT(Data:=RrData);
        istate_read:=100;

    END_IF
100:
        istate_read:=1;
END_CASE
```

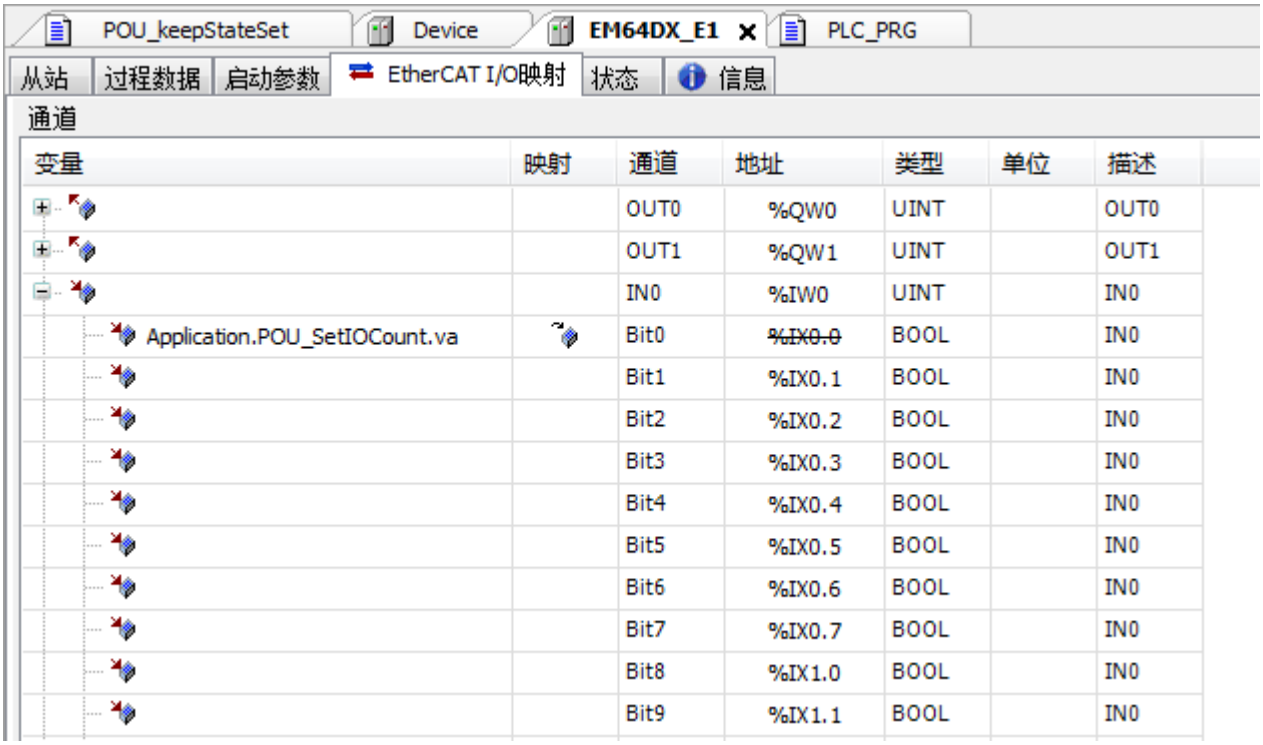
图6.22 读取计数值

完成上述步骤后，即完成对计数器参数配置的代码编写。

上述步骤的计数通道为计数器 0，计数模式设置为 1 电平上升沿，计数初始值为 0。

e.配置模块的 EtherCAT I/O 映射：

将程序中申明的变量 EtherCAT_IN0 配置到 IO 模块的映射表，配置完成后显示的界面如图 6.23 所示：



变量	映射	通道	地址	类型	单位	描述
		OUT0	%QW0	UINT		OUT0
		OUT1	%QW1	UINT		OUT1
		IN0	%IW0	UINT		IN0
Application.POU_SetIOCount.va		Bit0	%IX0.0	BOOL		IN0
		Bit1	%IX0.1	BOOL		IN0
		Bit2	%IX0.2	BOOL		IN0
		Bit3	%IX0.3	BOOL		IN0
		Bit4	%IX0.4	BOOL		IN0
		Bit5	%IX0.5	BOOL		IN0
		Bit6	%IX0.6	BOOL		IN0
		Bit7	%IX0.7	BOOL		IN0
		Bit8	%IX1.0	BOOL		IN0
		Bit9	%IX1.1	BOOL		IN0

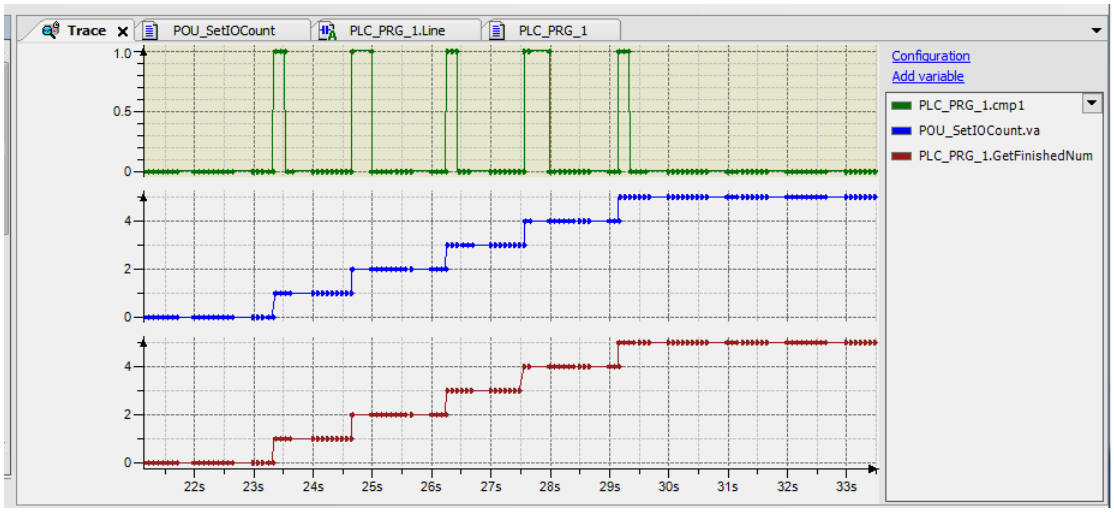
6.23 配置 IO 映射

f.运行程序：

完成上述步骤后，程序运行结果如下：

计数器 0 以计数模式 1 电平上升沿计数，计数初始值为 0。

POU_SetIOCount.va 为实际读取值，PLC_PRG_1.GetFinishedNum 为比较值（调用具有高速比较功能的例程产生）。



6.24trace读取值与比较值

输出延时翻转功能：

a. 选择输出延时翻转口：

OUT0-OUT7 所对应的对象字典的索引为 16#7020-16#7027

示例代码如图 6.25 所示：

EM64DX_E1 | PLC_PRG | POU_reverseSet_1

```
1  PROGRAM POU_reverseSet_1
2  VAR
3      Test_index:WORD:=16#7020;//
4      Test_subindex:BYTE:=2;
5      ETC_CO_SdoWrite4_0: ETC_CO_SdoWrite4;
6      ETC_CO_SdoRead4_0: ETC_CO_SdoRead4;
7      Exe_Wr: BOOL;
8      wr_index: WORD;
9      wr_subindex: BYTE;
10     wr_data: ARRAY [1..4] OF BYTE;
11     wr_datalen: USINT;
12     Exe_Re: BOOL;
13     re_index: WORD;
14     re_subindex: BYTE;
15     RrData: ARRAY [1..4] OF BYTE;
16     ReDataLeng: USINT;
17     istate: INT:=0;
```

图 6.25 延时翻转输出口选择

b. 延时翻转模式设置：

设置延时翻转模式的对象字典子索引为 1

参数有：0 不启用；1 遇低翻转；2 遇高翻转

示例代码如图 6.26 所示：

```
// 设置
CASE  istate OF
0:
    Exe_Wr:=FALSE;
1:
    //设置是否启用延时翻转：0不启用；1遇低翻转；2遇高翻转
    wr_data[1]:=1;  wr_data[2]:=0;  wr_data[3]:=0;  wr_data[4]:=0;
    wr_index:=Test_index;
    wr_subindex:=1;
    wr_datalen:=4;
    Exe_Wr:=TRUE;
    istate:=2;
2:
 IF ETC_CO_SdoWrite4_0.xDone THEN
        Exe_Wr:=FALSE;
        istate:=3;
    END_IF
```

图 6.26 延时翻转启用设置

c.输出延时翻转时间设置：

输出延时翻转时间设置的对象字典子索引为 2

该参数值的范围：0-100000（单位：毫秒，0 表示没有延时），如果超出该范围，系统强行将值设置为 100000。

```
3:  //设置输出端口延时翻转时间（范围：0-100000，单位ms）

    wr_data[1]:=0;  wr_data[2]:=10; wr_data[3]:=0;  wr_data[4]:=0;
    wr_index:=Test_index;
    wr_subindex:=2;
    wr_datalen:=4;
    Exe_Wr:=TRUE;
    istate:=4;
4:
    IF ETC_CO_SdoWrite4_0.xDone THEN
        Exe_Wr:=FALSE;
        istate:=41;
    END_IF
    ACT_SDO();
```

图 6.27 延时翻转时间设置

完成上述步骤后，即完成对 EM64DX-E1 模块输出延时翻转参数配置的代码编写。

d.运行程序：

上述延时翻转输出口为 OUT0，延时翻转模式为：遇低翻转，输出延时翻转时间为：2560ms

其输出如图 6.28 所示：

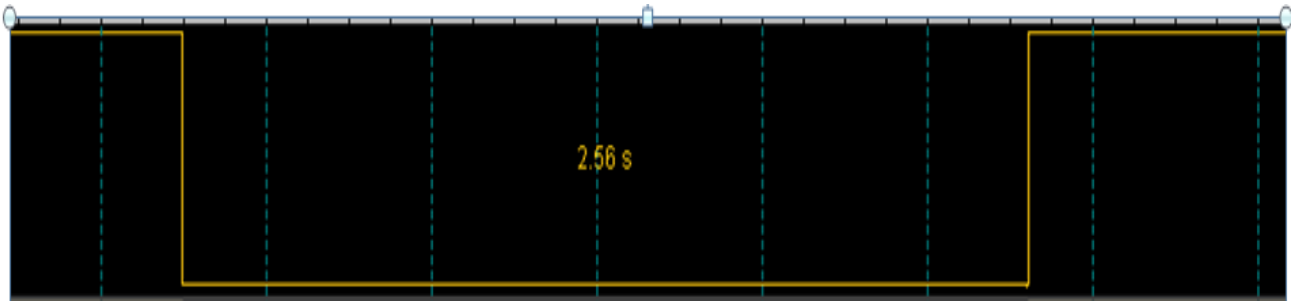


图 6.28 输出延时翻转采集

输出保持功能：

输出保持功能的对象字典的索引为：16#7011

示例代码如图6.29所示：

EM64DX_E1 | PLC_PRG | POU_reverseSet_1 | POU_keepStateSet

```
PROGRAM POU_keepStateSet
VAR
    Test_index:WORD:=16#7011;//
    Test_subindex:BYTE:=1;
    ETC_CO_SdoWrite4_0: ETC_CO_SdoWrite4;
    ETC_CO_SdoRead4_0: ETC_CO_SdoRead4;
    Exe_Wr: BOOL;
    wr_index: WORD;
    wr_subindex: BYTE;
    wr_data: ARRAY [1..4] OF BYTE;
    wr_datalen: USINT;
    Exe_Re: BOOL;
    re_index: WORD;
    re_subindex: BYTE;
    istate: INT:=0;
    istate_read: INT;
    RrData: ARRAY [1..4] OF BYTE;
    ReDataLeng: USINT;
```

图 6.29 程序

设置输出保持口：

断线时是否保持输出口状态的对象字典子索引为 1

参数的 32 个 bit 对应 32 个输出口的状态：0：不保持，1：保持

```
// 设置
CASE  istate OF
0:
    Exe_Wr:=FALSE;
1:
    //断线时是否保持输出口状态，该参数的32个bit对应32个输出口的状态：0：输出不包吃 1：输出保持

    wr_data[1]:=255;  wr_data[2]:=0;    wr_data[3]:=0;  wr_data[4]:=0;
    wr_index:=Test_index;
    wr_subindex:=1;
    wr_datalen:=4;
    Exe_Wr:=TRUE;
    istate:=2;
2:
    IF ETC_CO_SdoWrite4_0.xDone THEN
        //Exe_Wr:=FALSE;
        istate:=100;
    END_IF
```

6.30 程序

完成上述步骤后，运行例程，总线断开或报错时，OUT0-OUT7 的输出状态保持总线断开或报错前的状态。

6.2 BASIC 控制器示例

6.2.1 硬件连接

雷赛 BAC332E 控制器的外形如下图 6.31 所示：

图 6.31　BAC332E 外形

该控制器采用 24V 直流电源供电，具有 1 路 EtherCAT。

该控制器的 EtherCAT 端口信号如表 6.2 所示：

表 6.2 接口引脚号和信号关系表

EtherCAT 信号	信号描述	说明
1	TX+	发送信号+
2	TX-	发送信号-
3	RX+	接收信号+
4	NC	保留
5	NC	保留
6	RX-	接收信号-
7	NC	保留
8	NC	保留

各端口的详细描述请参考 BAC332E 系列运动控制器用户手册。

设备间的连接：通过超五类带屏蔽层的网线将 BAC332E 的 EtherCAT 口与 EM64DX-E1 的 EtherCAT IN 口连接。

模块上的拨码开关，采用出厂默认配置。

6.2.2 EtherCAT 主站的添加及配置

打开 SMC BASIC STUDIO 编程软件之后，需要新建一个工程（详细建立工程过程请参考《BAC332E 用户使用手册》）。在该工程中会自动添加 EtherCAT 主站。主站的参数除了通讯周期时间之外，其他的参数不需要用户配置，保持默认即可。连接上控制器之后，在左侧“设备”栏，双击“EtherCAT_0”即可以看到主站的相关信息，如图 6.32 所示：

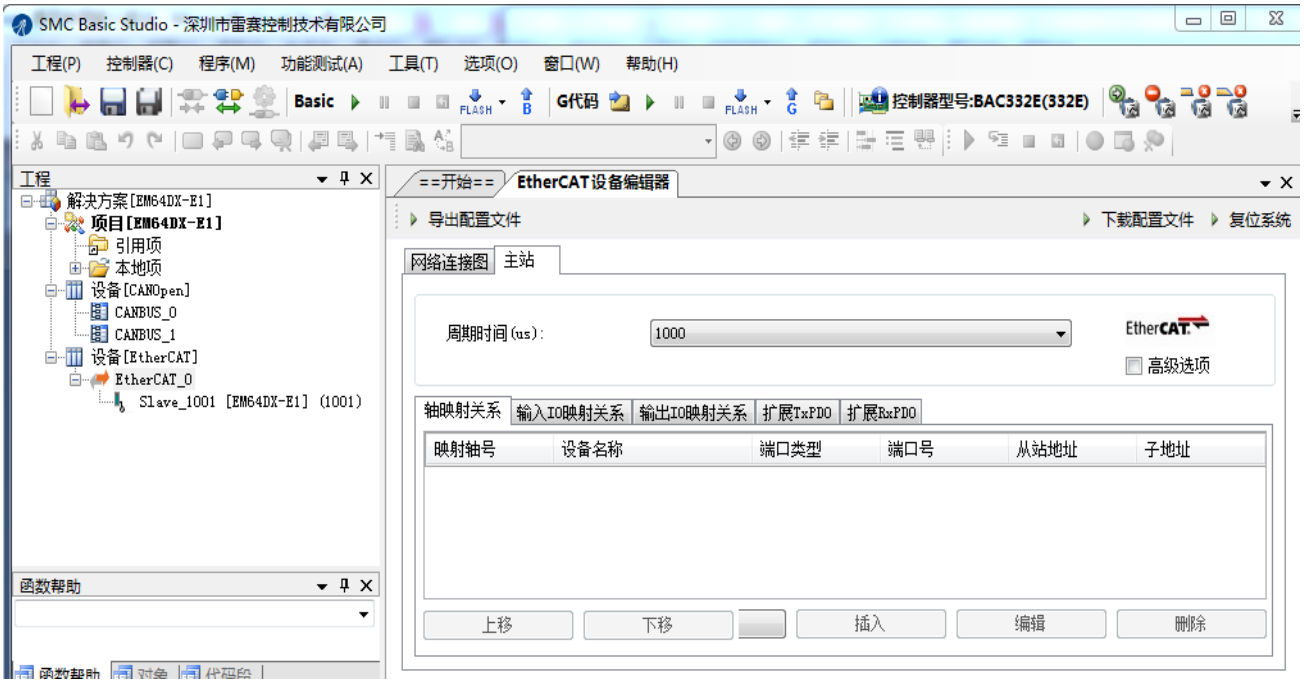


图 6.32 BAC332E 主站界面

6.2.3 模块的添加

在 SMC BASIC STUDIO 编程软件中，可以手动添加从站模块和自动扫描从站模块。在添加从站之前，必须保证设备库中有对应的模块设备描述文件，具体操作请参考《BAC332E 用户使用手册》里“安装设备描述文件”章节。

1） 手动添加

在“工程”栏的目录里，选中主站“EtherCAT_0”，然后点击鼠标右键，选择“添加从站”在弹出的窗口中找到对应的设备描述文件，如图 6.33 所示：

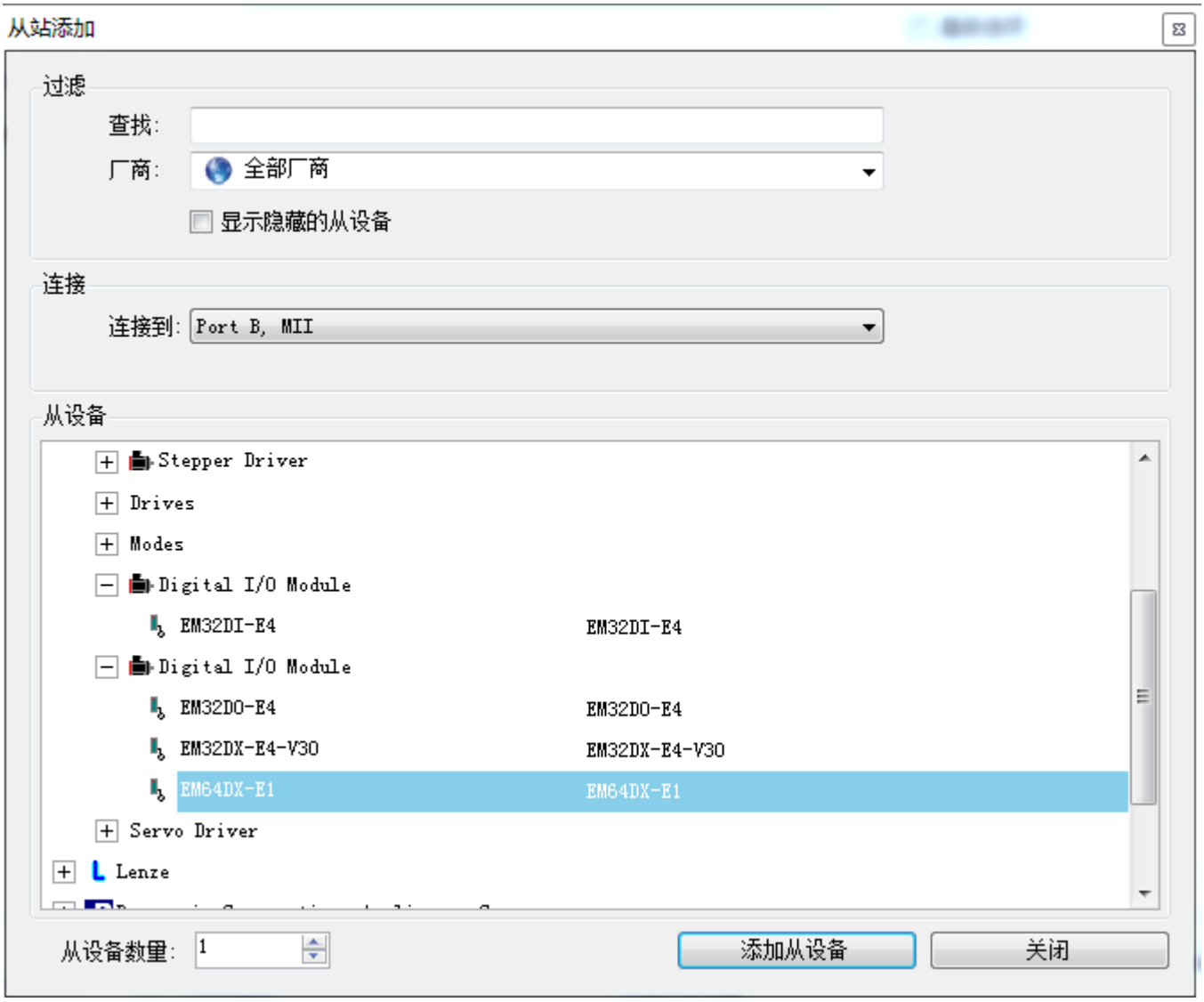


图 6.33 添加从站模块

然后选择“添加从设备”，在左侧“工程”目录下可以找到添加成功的模块。

2） 自动扫描

在“工程”栏的目录里，选中主站“EtherCAT_0”，然后点击鼠标右键，选择“扫描设备”，扫描成功后会提示是否下载对应的配置文件，同时主站目录下会出现扫描到的从站模块，如图 6.34 所示

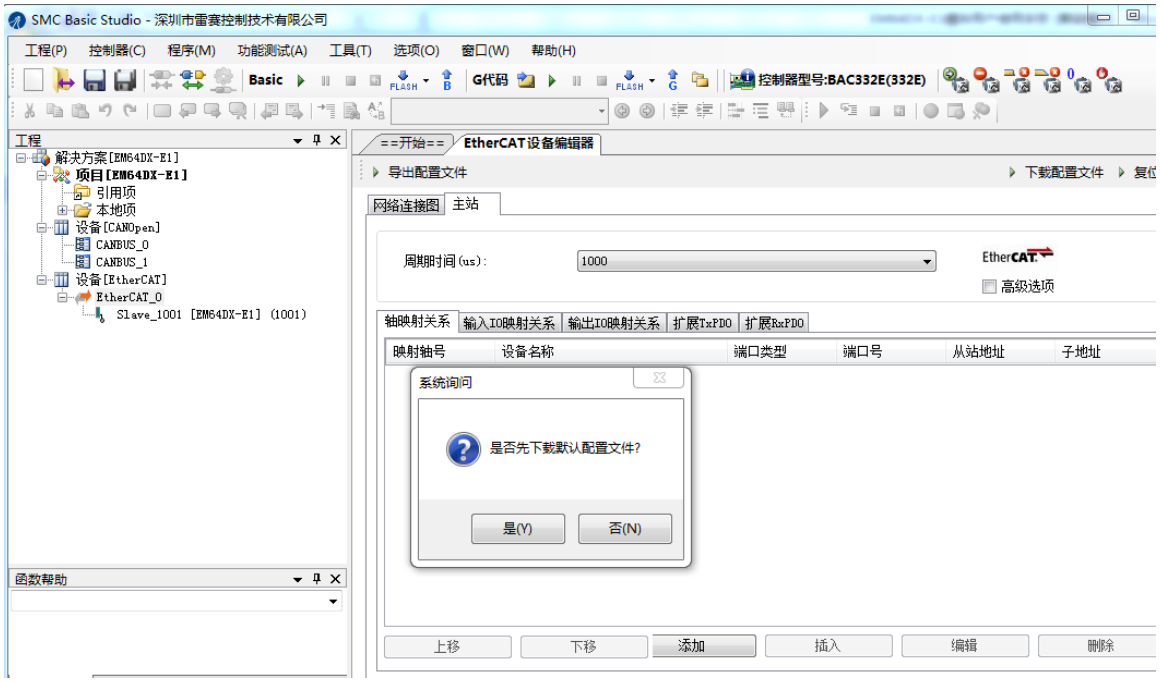


图 6.34 自动扫描设备

选择“是”;

下载成功后会重启系统，双击从站 “Slave_1001[EM32DX-E4](1001)”，可以看到从站模块的信息，如图6.35所示

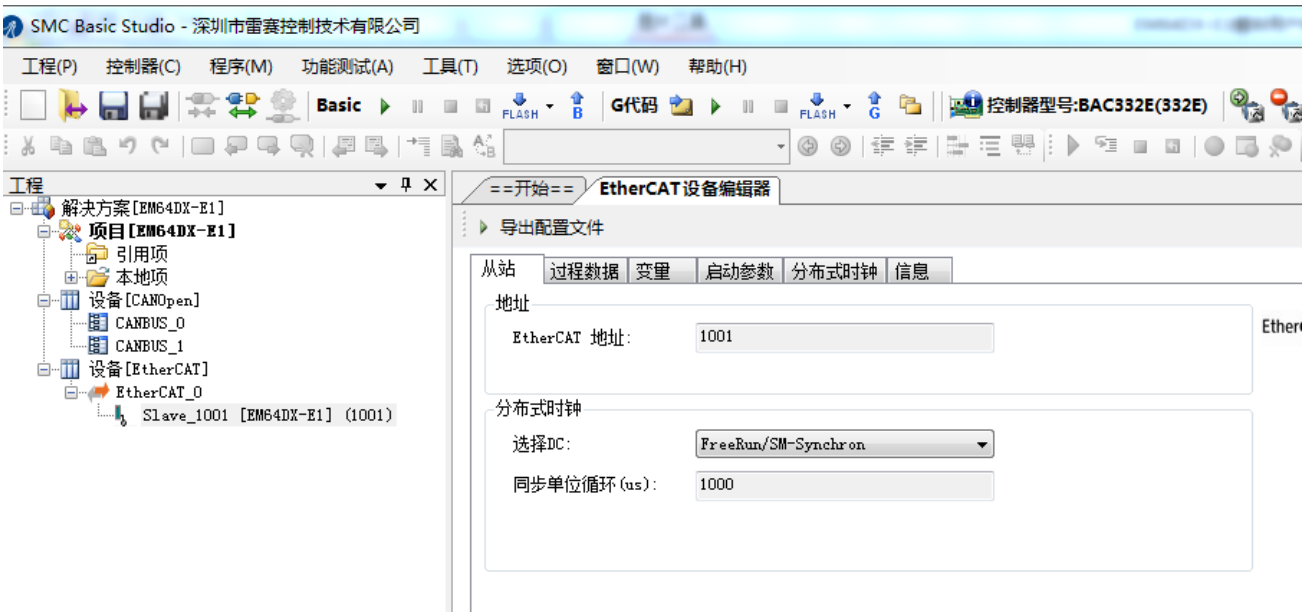


图 6.35 从站模块信息

在EtherCAT设备编辑器中，可以看到从站模块的所有信息，包括从站地址、同步时间周期、PDO、时钟、模块信息等。从站的参数都是系统默认匹配的，不需要用户修改。如下图所示：

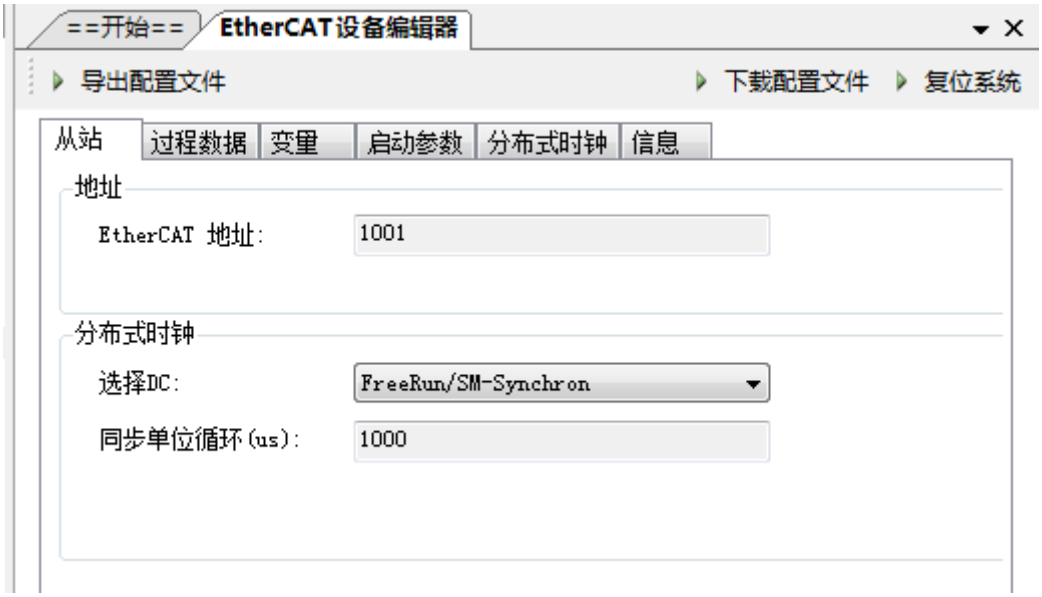


图 6.36 从站模块信息

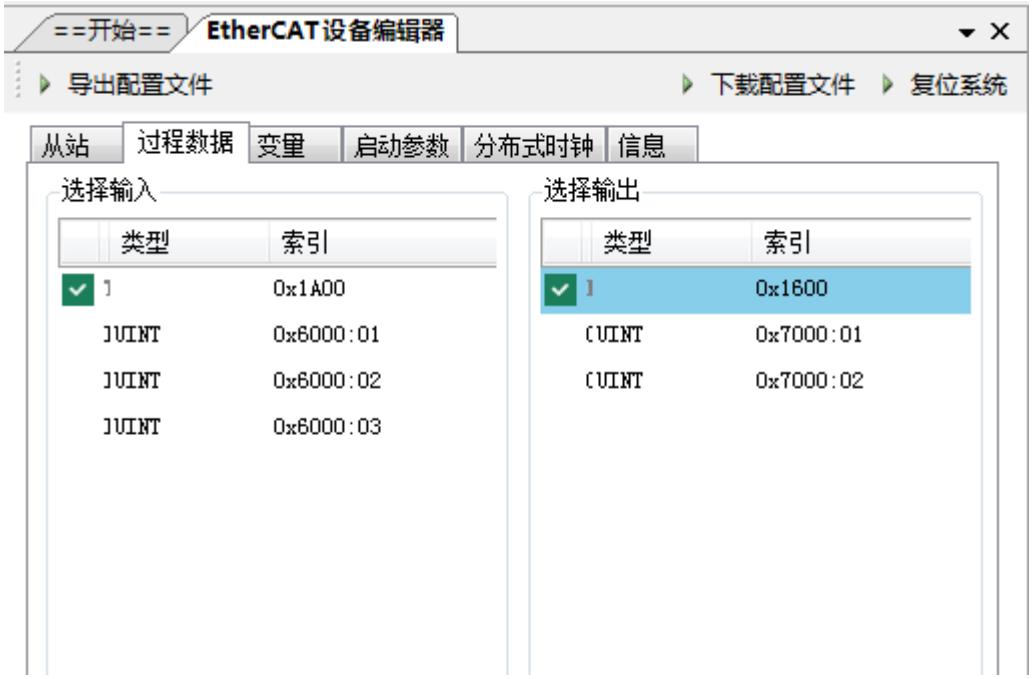


图 6.37 从站模块信息

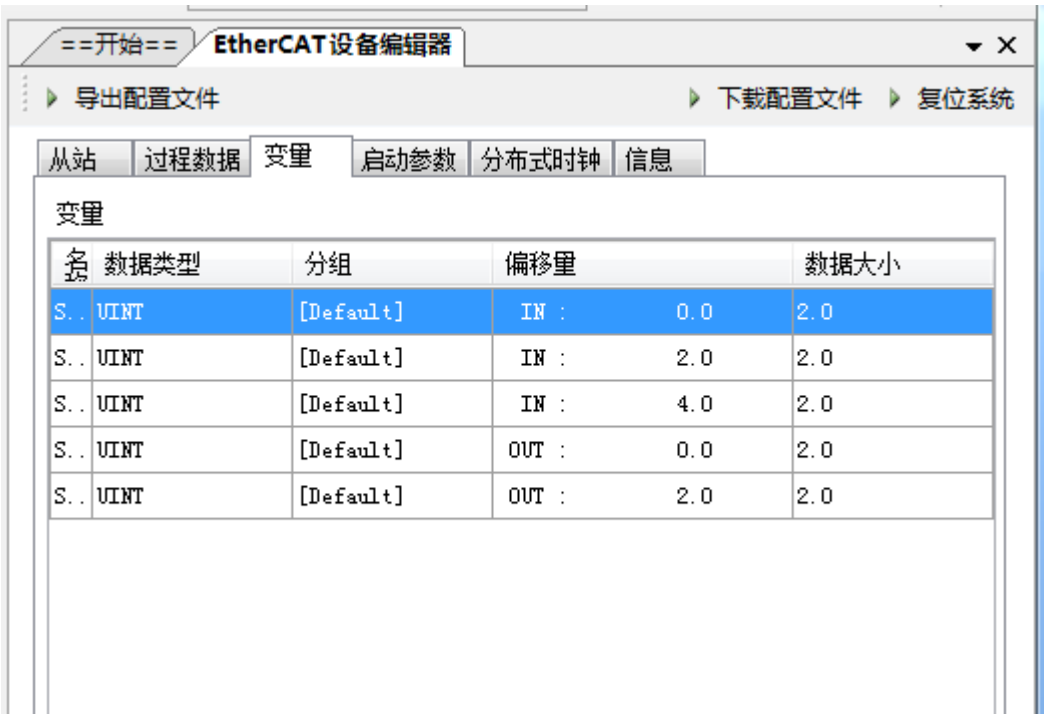


图 6.38 从站模块信息

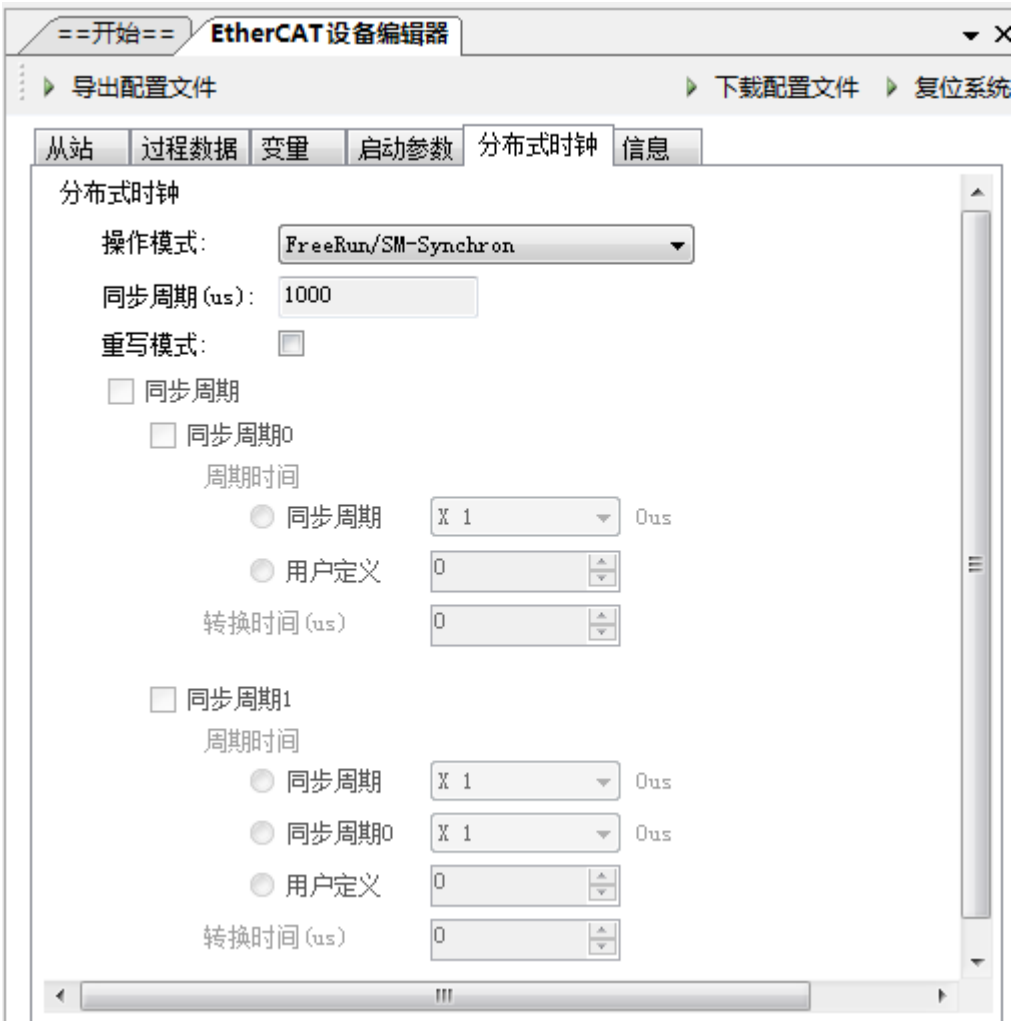


图 6.39 从站模块信息

图6.40 从站模块信息

至此，从站模块的添加已经完成。

6.2.4 模块的配置

双击“工具”栏中的EtherCAT主站“EtherCAT_0”，可以看到EtherCAT主站的包含信息。在此处将轴映射关系以及IO映射关系显示在此界面，后续程序中使用的轴号以及IO号都以此做为参考

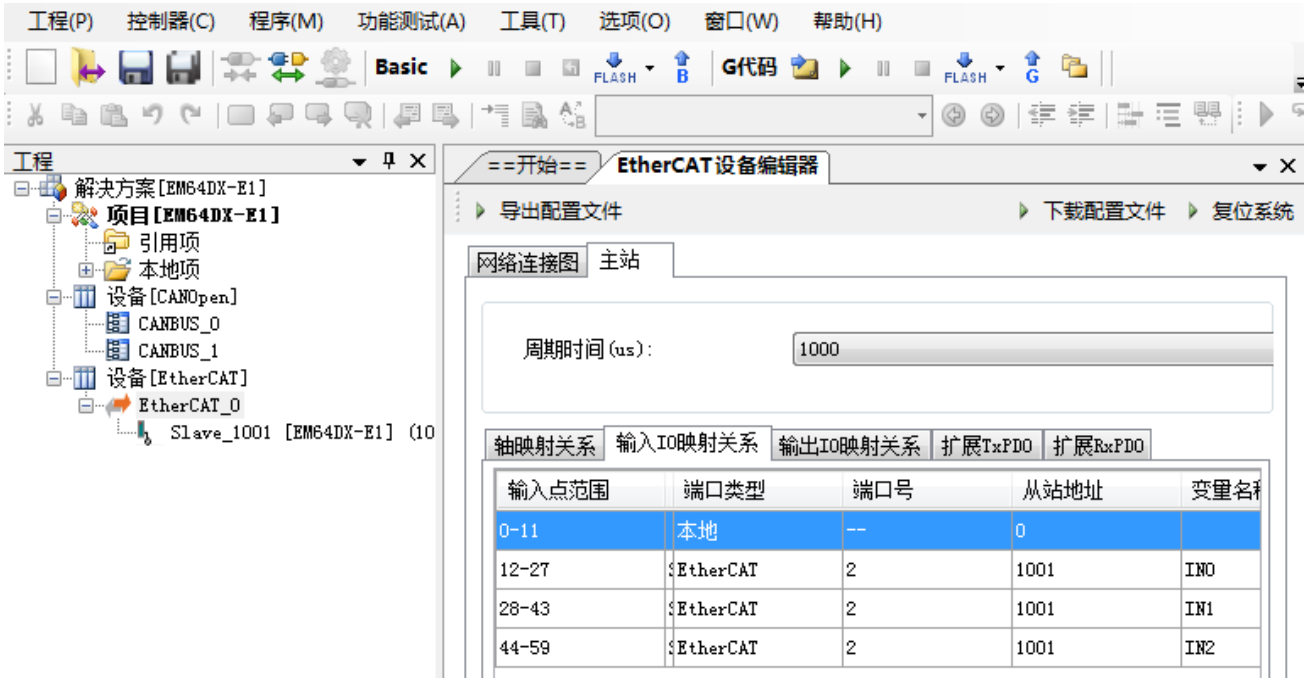


图 6.41 主站设备信息

由于BAC332E本地有12路输入和12路输出，所以输入端口0-11是BAC332E控制器上的本地输入端口，扩展模块上的输入端口IN0-IN15映射为软件端IN12-IN27.同理输出端口OUT0-OUT11映射为软件端OUT12-OUT27。

6.2.5 BASIC 应用例程

（1）程序功能：

在BAC332E控制器上控制扩展模块EM64DX-E1的输入状态读取、输出控制、输入计数、输出延时翻转、输出保持等功能。

（2）函数说明

1.SMCReadInbit(WORD bitno)

功能：读取某个输入端口的电平

参数：bitno输入端口号，取值范围：0-控制器本机输入口数目-1

返回值：指定的输入端口电平：0：低电平，导通状态；1

2.SMCWriteOutbit(WORD bitno,WORD on_off)

功能：设置指定控制器的某个输出端口的电平

参数： bitno 输出端口号，取值范围：0-控制器本机输出口数目-1

on_off 输出电平，0：低电平，1：高电平

返回值：错误代码

3.NMCSSetNodeOD(WORD PortNum,WORD NodeNum, WORD Index,WORD SubIndex,WORD ValLength,DWORD Value)

功能：设置从站对象字典

参数：PortNum EtherCAT 端口号，固定为 2

NodeNum 从站 EtherCAT 地址，第 i 个 EtherCAT 从站地址为 1000+i

Index 对象字典索引

SubIndex 对象字典子索引

ValLength 对象字典索引长度(单位：bit)

Value 对象字典索引参数值

返回值：错误代码

4.NMCSGetNodeOD(WORD PortNum,WORD NodeNum, WORD Index,WORD SubIndex,WORD ValLength,DWORD* Value)

功能：获取从站对象字典

参数：PortNum EtherCAT 端口号，固定为 2

NodeNum 从站 EtherCAT 地址，第 i 个 EtherCAT 从站地址为 1000+i

Index 对象字典索引

SubIndex 对象字典子索引

ValLength 对象字典索引长度(单位：bit)

Value 对象字典索引参数值

返回值：错误代码

5. smc_read_inbit(WORD ConnectNo, WORD bitno)

功 能：读取指定控制器的某个输入端口的电平

参 数：ConnectNo 指定链接号：0-254,默认值 0

bitno 输入端口号，取值范围： 0~控制器本机输入口数-1

返回值：指定的输入端口电平：0：低电平，1：高电平

6. smc_write_outbit(WORD ConnectNo, WORD bitno,WORD on_off)

功 能：设置指定控制器的某个输出端口的电平

参 数：ConnectNo 指定链接号：0-254,默认值 0

bitno 输出端口号，取值范围： 0~控制器本机输出口数-1

on_off 输出电平，0：低电平，1：高电平

返回值：错误代码

7. nmcs_set_node_od(WORD ConnectNo,WORD PortNum,WORD NodeNum, WORD Index,WORD SubIndex,WORD ValLength,DWORD Value)

功 能：设置从站对象字典参数值

参 数：ConnectNo 控制器号

PortNum EtherCAT 端口号，固定为 2

NodeNum 从站 EtherCAT 地址，第 i 个 EtherCAT 从站地址为 1000+i

Index 对象字典索引

SubIndex 对象字典子索引

ValLength 对象字典索引长度(单位：bit)

Value 对象字典索引参数值

返回值：错误代码

8. nmcs_get_node_od(WORD ConnectNo, WORD PortNum, WORD NodeNum, WORD Index, WORD SubIndex, WORD ValLength, DWORD* Value)

功　能：读取从站对象字典参数值

参　数：ConnectNo　　控制器号

PortNum　　EtherCAT 端口号，固定为 2

NodeNum　　从站 EtherCAT 地址，第 i 个 EtherCAT 从站为 1000+i

Index　　对象字典索引

SubIndex　　对象字典子索引

ValLength　　对象字典索引长度(单位：bit)

Value　　对象字典索引参数值

返回值：错误代码

（3）工程源码：

通用输入输出：

```
dim  bitno =0   '输入/输出口：0
dim  on_off=0  '输出电平：0
dim  on_off1  '读取输入电平
smcwriteoutbit(bitno, on_off) '设置OUT0的输出电平为：低
on_off1 = smcreadinbit (bitno) '读取 IN0 的电平: on_off1
```

运行结果：OUT0有效时，IN0有效（指示灯亮），OUT0无效时，IN0无效（指示灯灭）

输入计数：

```
dim  nodenum=1001 '从站号
dim  mvalue =0 '输入计数模式，设置IN0的计数方式：0电平下降沿，1电平上升沿， 2电平任意沿
dim  vvalue =0  '输入计数初始值
dim  index=hex("0x6020") '设置输入计数口，IN0-IN15，对应索引值：0x6020-0x602f
dim  value  '输入计数值
nmcssetnodeod(2,nodenum,index,1,32, mvalue) '设置IN0的计数模式为：0,下降沿计数
```

```
nmcssetnodeod(2,nodenum,index,2,32, vvalue) '设置IN0的计数值的初始值为：0
nmcsgetnodeod(2,nodenum,index,3,32, value) '读取IN0的计数值
```

运行结果：OUT0输出信号有下降沿变化时，IN0计数：1

输出延时翻转：

```
dim  nodenum=1001'从站号
dim  Rindex =hex("0x7020") '设置输出延时翻转口，OUT0-OUT7，对应索引值：0x7020-0x7027
dim  RMvalue =1 '设置是否启用延时翻转：0不启用；1遇低翻转；2遇高翻转
dim  RTvalue = 5000 '设置输出端口延时翻转时间（范围：0-100000，单位ms）
nmcssetnodeod(2,nodenum, Rindex,1,32, RMvalue) '设置延时翻转： 1,遇低翻转
nmcssetnodeod(2,nodenum, Rindex,2,32, RTvalue)  '设置输出延时翻转时间:5000ms
```

运行结果：设置参数后,OUT0处于高电平状态，给OUT0低电平，OUT输出5000ms的低电平

总线掉线输出保持功能：

```
dim  nodenum=1001'从站号
dim  bitvalue =65535' 断线时是否保持输出口状态， 32个bit对应32个输出口的状态，0：不保持，1：保持
nmcssetnodeod(2,nodenum, hex("0x7011"),1,32, bitvalue) '设置OUT0-OUT15,输出保持
```

运行结果：总线断开或报错时，OUT0-OUT15保持总线断开或报错前的状态

总线复位输出保持功能：

```
dim  nodenum=1001'从站号
dim ENvalue
dim  bitvalue =65535' 断线时是否保持输出口状态， 32个bit对应32个输出口的状态，0：不保持，1：保持
nmcssetnodeod(2,nodenum, hex("0x7011"),1,32, bitvalue) '设置OUT0-OUT15,输出保持
nmcssetslaveoutputretain(1) '设置总线复位输出保持
nmcsgetslaveoutputretain(ENvalue)
```

运行结果：总线复位时，OUT0-OUT15 保持总线复位前的状态

6.2.6 API 应用例程

通用输入输出：

```
ushort  _ConnecNo=0;  //指定链接号（0-7），默认值0
ushort  bitno =0 ;  //输入/输出口：0
ushort  on_off=0  ;  //输出电平：0
short   on_off1=0;  //读取输入电平
LTSMC.dmc_write_outbit(_ConnecNo, bitno, on_off) ;  //设置OUT0的输出电平为：低
on_off1 = LTSMC. smc_read_inbit(_ConnecNo, bitno) ;  //读取IN0的电平
```

运行结果：OUT0有效时，IN0有效（指示灯亮），OUT0无效时，IN0无效（指示灯灭）

输入计数：

```
ushort  _ConnecNo=0;  //指定链接号（0-7），默认值0
ushort  nodenum=1001;  //从站号
uint  mvalue =0 ;  //输入计数模式，设置IN0的计数方式：0电平下降沿，1电平上升沿， 2电平任意沿
uint  vvalue =0  ;  //输入计数初始值
ushort  index=0x6020;  //设置输入计数口，IN0-IN15，对应索引值：0x6020-0x602f
uint  value =0 ;  //输入计数值
LTSMC.nmcs_set_node_od(_ConnecNo, 2, nodenum, index, 1, 32, mvalue) ;  //设置IN0的计数模式为：0,下降沿计数
LTSMC.nmcs_set_node_od(_ConnecNo, 2, nodenum, index, 2, 32, vvalue) ;  //设置IN0的计数值的初始值为：0
LTSMC.nmcs_get_node_od(_ConnecNo, 2, nodenum, index, 3, 32, ref  value) ;  //读取IN0的计数值
```

运行结果：OUT0输出信号有下降沿变化时，IN0计数：1

输出延时翻转：

```
ushort  _ConnecNo=0;  //指定链接号（0-7），默认值0
ushort  nodenum=1001;  //从站号
```

```
ushort  Rindex = 0x7020 ;  //设置输出延时翻转口，OUT0-OUT7，对应索引值：0x7020-0x7027
uint  RMvalue =1 ;  //设置是否启用延时翻转：0不启用；1遇低翻转；2遇高翻转
uint  RTvalue = 5000 ;  //设置输出端口延时翻转时间（范围：0-100000，单位ms）
LTSMC.nmcs_set_node_od(_ConnecNo, 2, nodenum, Rindex, 1, 32, RMvalue) ;  //设置延时翻转： 1,遇低翻转
LTSMC.nmcs_set_node_od(_ConnecNo, 2, nodenum, Rindex, 2, 32, RTvalue)  ;  //设置输出延时翻转时间:5000ms
```

运行结果：设置参数后,OUT0处于高电平状态，给OUT0低电平，OUT输出5000ms的低电平

总线掉线输出保持功能：

```
ushort  _ConnecNo=0;  //指定链接号（0-7），默认值0
ushort  nodenum=1001;  //从站号
ushort  bitvalue =65535;  // 断线时是否保持输出口状态， 32个bit对应32个输出口的状态，0：不保持，1：保持
LTSMC.nmcs_set_node_od(_ConnecNo, 2, nodenum, 0x7011, 1, 32, bitvalue) ;  //设置OUT0-OUT15,输出保持
```

运行结果：总线断开或报错时，OUT0-OUT15保持总线断开或报错前的状态

总线复位输出保持功能：

```
ushort Enable = 0;
LTSMC.nmcs_set_node_od(_ConnecNo, 2, 1001, 0x7011, 1, 32, 65535); //断线时保持输出口0-15状态，该参数的32个bit对应32个输出口的状态
LTSMC.nmcs_set_slave_output_retain(_ConnecNo, 1);//设置总线复位输出保持
LTSMC.nmcs_get_slave_output_retain(_ConnecNo, ref Enable);//读取总线复位输出保持状态
```

运行结果：总线复位时，OUT0-OUT15保持总线复位前的状态

6.3 控制卡示例

6.3.1 硬件连接

此处主站为 DMC-E3032 控制卡，从站为 EM64DX-E1。需要将 DMC-E3032 的 EtherCAT 口和 EM64DX-E1 的 ECAT IN 接口连接起来。

推荐使用超五类屏蔽网线，抗干扰，稳定，可以有效的减少异常错误。

6.3.2 从站 ID 设置

EtherCAT 从站的 ID 可由软件自动分配或者手动拨码设置。

6.3.3 组建 EtherCAT 网络

建立 EtherCAT 网络是将主站和从站建立连接，便于后期的应用程序控制。在这个过程中，将使用雷赛控制卡调试软件 DMC Motion。具体步骤如下

1）扫描从站

在 Motion 界面点击“总线配置”，在左侧设备目录树中找到 EtherCAT 主站，右键执行“扫描设备”功能。扫描后，总线网络中的所有从站都将排列到总线结构树中。如图 6.42 所示：

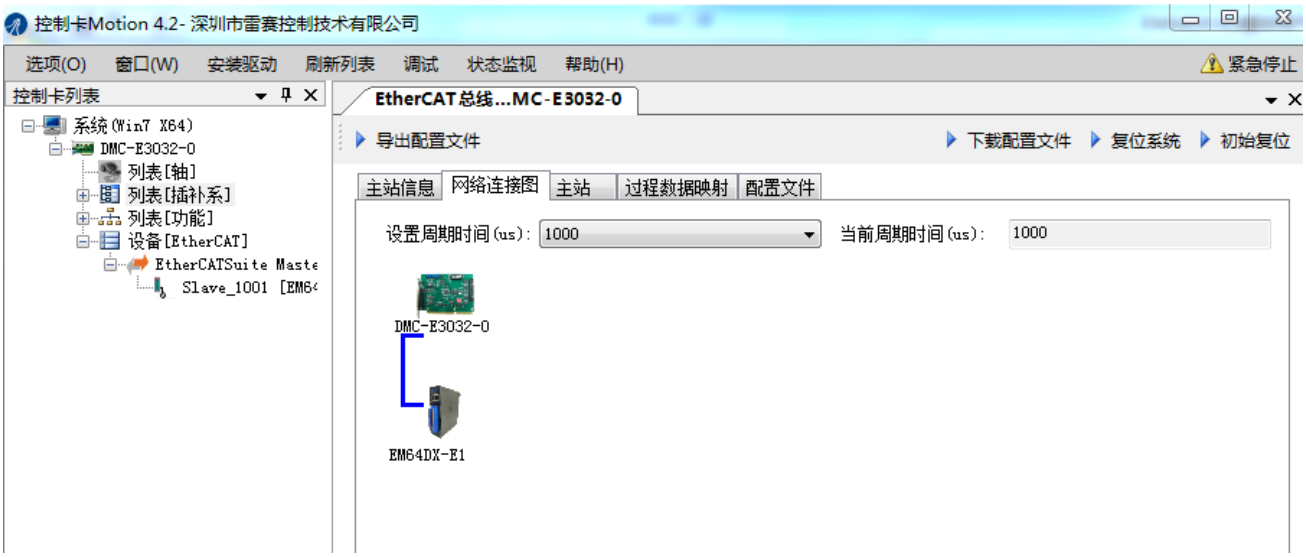


图 6.42 扫描从站

2）设置总线周期，下载配置文件，步骤如下：

①填写通信周期/指令更新周期。

②点击“下载配置文件”。等待配置文件下载成功。

3）至此，EtherCAT 网络已经建立完成，EM64DX-E1 模块已经成功添加进 EtherCAT 网络。用户可以编写应用程序来控制模块的 IO。

6.3.4 应用例程

1）程序功能

在 DMC-E3032 控制卡上实现对 EM64DX-E1 模块的输入状态读取、输出控制、输入计数、输出延时翻转、输出保持等功能。

2）函数说明

1.short dmc_read_inbit(WORD CardNo, WORD bitno)

功 能：读取指定控制卡的某个输入端口的电平

参 数：CardNo 控制卡卡号 0-7

bitno 输入端口号 0-2~31

返回值：指定的输入端口电平：0：低电平，1：高电平

2.short dmc_write_outbit(WORD CardNo, WORD bitno, WORD on_off)

功 能：设置指定控制卡的某个输出端口的电平

参 数：CardNo 控制卡卡号 0-7

bitno 输出端口号 0-2~31

on_off 输出电平，0：低电平，1：高电平

返回值：错误代码

3.short nmc_get_node_od(WORD CardNo,WORD PortNum,WORD NodeNum, WORD Index,WORD SubIndex,WORD ValLength,DWORD* Value)

功 能：读取从站对象字典参数值

参　数：CardNo　　控制卡卡号

PortNum　　EtherCAT 端口号，固定为 2

NodeNum　　从站 EtherCAT 地址，第 i 个 EtherCAT 从站为 1000+i

Index　　对象字典索引

SubIndex　　对象字典子索引

ValLength　　对象字典索引长度(单位：bit)

Value　　对象字典索引参数值

返回值：错误代码

4. short nmc_set_node_od(WORD CardNo,WORD PortNum,WORD NodeNum, WORD Index,WORD SubIndex,WORD ValLength,DWORD Value)

功　能：设置从站对象字典参数值

参　数：CardNo　　控制卡卡号

PortNum　　EtherCAT 端口号，固定为 2

NodeNum　　从站 EtherCAT 地址，第 i 个 EtherCAT 从站地址为 1000+i

Index　　对象字典索引

SubIndex　　对象字典子索引

ValLength　　对象字典索引长度(单位：bit)

Value　　对象字典索引参数值

返回值：错误代码

3）工程源码

通用输入输出功能：

```
ushort  _CardID=0;  //卡号
ushort  bitno =0 ;  //输入/输出口：0
ushort  on_off=0  ;  //输出电平：0
short  on_off1=0;  //读取输入电平
LTDMC.dmc_write_outbit(_CardID, bitno, on_off) ;  //设置OUT0的输出电平为：低
```

```
on_off1 = LTDMC.dmc_read_inbit(_CardID, bitno) ;  //读取IN0的电平
```

运行结果：OUT0有效时，IN0有效（指示灯亮），OUT0无效时，IN0无效（指示灯灭）。

输入计数功能：

```
ushort  _CardID=0;  //卡号
ushort  nodenum=1001;  //从站号
int  mvalue =0 ;  //输入计数模式，设置IN0的计数方式：0电平下降沿，1电平上升沿， 2电平任意沿
int  vvalue =0  ;  //输入计数初始值
ushort  index=0x6020;  //设置输入计数口，IN0-IN15，对应索引值：0x6020-0x602f
int  value =0 ;  //输入计数值
LTDMC.nmc_set_node_od(_CardID, 2, nodenum, index, 1, 32, mvalue) ;  //设置IN0的计数模式为：0,下降沿计数
LTDMC.nmc_set_node_od(_CardID, 2, nodenum, index, 2, 32, vvalue) ;  //设置IN0的计数值的初始值为：0
LTDMC.nmc_get_node_od(_CardID, 3, nodenum, index, 3, 32, ref  value) ;  //读取IN0的计数值
```

运行结果：OUT0输出信号有下降沿变化时，IN0计数：1

输出延时翻转功能：

```
ushort  _CardID=0;  //卡号
ushort  nodenum=1001;  //从站号
ushort  Rindex = 0x7020 ;  //设置输出延时翻转口，OUT0-OUT7，对应索引值：0x7020-0x7027
int  RMvalue =1 ;  //设置是否启用延时翻转：0不启用；1遇低翻转；2遇高翻转
int  RTvalue = 5000 ;  //设置输出端口延时翻转时间（范围：0-100000，单位ms）
LTDMC.nmc_set_node_od(_CardID, 2, nodenum, Rindex, 1, 32, RMvalue) ;  //设置延时翻转： 1,遇低翻转
LTDMC.nmc_set_node_od(_CardID, 2, nodenum, Rindex, 2, 32, RTvalue)  ;  //设置输出延时翻转时间:5000ms
```

运行结果：设置参数后,OUT0处于高电平状态，给OUT0低电平，OUT输出5000ms的低电平

总线掉线输出保持功能：

```
ushort  _CardID=0; //卡号
```

```
ushort  nodenum=1001;  //从站号
int  bitvalue =65535;  // 断线时是否保持输出口状态， 32个bit对应32个输出口的状态，0：不保持，1：保持
LTDMC.nmc_set_node_od(_CardID, 2, nodenum, 0x7011, 1, 32, bitvalue) ;  //设置OUT0-OUT15,输出保持
```

运行结果：总线断开或报错时，OUT0-OUT15保持总线断开或报错前的状态

总线复位输出保持功能：

```
ushort Enable = 0;
LTDMC.nmc_set_node_od(_CardID, 2, 1001, 0x7011, 1, 32, 65535); //断线时保持输出口0-15状态，该参数的32个bit对应32个输出口的状态
LTDMC.nmc_set_slave_output_retain(_CardID, 1);//设置总线复位输出保持
LTDMC.nmc_get_slave_output_retain(_CardID, ref Enable);//读取总线复位输出保持状态
```

运行结果：总线复位时，OUT0-OUT15 保持总线复位前的状态

深圳市雷赛控制技术有限公司
SHENZHEN LEADSHINE CONTROL TECHNOLOGY CO.,LTD

深圳市雷赛控制技术有限公司

地　址：深圳市南山区学苑大道 1001 号南山智园 A3 栋 9 楼

邮　编：518052

电　话：0755-26415968

传　真：0755-26417609

Email：　info@szleadtech.com.cn

网　址：http：；　//www.szleadtech.com.cn