

CAT-H IO 卡使用手册

前言

非常感谢您选用广州市研为电子科技有限公司的 EtherCat IO 卡，我们将竭诚为您提供最优质的服务。本手册将主要为您介绍 IO 卡的功能、设置、以及运行所需要的参数设定等。

1、操作环境：

- 操作温度：0 ~ 60 °C
- 储存温度：-20 ~ 80 °C
- 湿度：5 ~ 95% 无凝结

2、请勿将设备置于以下环境：

- 阳光直射处
- 环境温度超过规定处
- 环境湿度超过规定处
- 温度变化剧烈及可能结露处
- 有腐蚀性或可燃性气体处
- 灰尘、尘埃过多，或含盐分、铁粉高处
- 可能有水、油污、药品等飞溅处
- 可能将振动或冲击直接传至本体处

3、应尽量避免在以下场所使用，使用时也应采取相应措施。

- 由于静电容易引起干扰的场所
- 强电磁场发生的场所
- 可能受到辐射的场所
- 强电弧产生的场所
- 附近有强电流电源线经过的场所

4、应使用合格的附件，如开关、编码器等，并进行正确的布线，否则可能引起系统运转混乱。

5、避免在产品上加压重物及蒙灰。

6、绝对禁止对产品进行改造，否则有使产品失效，设备损坏的危险。

7、产品投入设备中使用前应先编程试运行，在确保配置正确以及所有部件都有效之后才正式投入使用。

8、应在确认设备已不受任何影响后，再进行试运行，否则可能引起设备破损。

9、正式启动前，请务必确认设定参数、附属部件的正常运作，否则可能引起设备破损。

10、系统在运转中，绝对禁止身体任何部分或与身体有关联的任何物品置于设备的运转作用区域，否则有受伤危险。

11、 当发生错误停机时，应检查并排除原因，并确保安全后，再重新设定参数进行运行，否则可能引起受伤或引起设备损坏。



订购注意事项：

1. 品质保证

(1) 保证期限

本公司的保证期限为从本公司向贵公司提供产品后 1 年内。

(2) 保证范围

在上述保证期限内产品出现质量问题，本公司负责免费对故障产品进行维修或更换，用户可以在购买处进行更换或要求维修，但下列情形除外：

- a 在产品规格书规定的条件、环境、使用方法外使用而引起故障；
- b 本公司产品以外的原因引起的故障；
- c 非本公司进行的改造或者修理引起的故障；
- d 未按本公司产品固有使用方法使用引起的故障；
- e 产品投入流通时的科学技术水平尚不能发现的缺陷存在的；
- f 由于不可抗力等其他非本公司责任导致的。

另，本条所述保证是指对本公司产品本身的保证，就本公司产品的故障所引起的其他人身和财产的损害，不在本保证范围之内。

2. 责任免除

(1) 本公司所承担的责任仅限于本公司产品在保证期间中发生故障时进行更换和修理。在任何情况下，对包括因本公司产品引起的特别损失、间接损失在内的一切损失本公司均不承担责任。

(2) 使用可编程设备时，非因本公司人员进行的编程引起的全部后果，本公司不承担任何责任。

3. 适合用途、条件

(1) 将本公司产品与其他产品组合使用时，客户应事先确认适用规格及相关法律法规等。将本公司产品用于客户的系统、设备、装备时，客户应自己确认其适用性。因产品适应性而导致的问题，请恕本公司不承担任何责任。

(2) 用于下述场合时，请事先向本公司销售人员咨询，确认设计规格资料，并在选择产品时，在规格、性能方面要留有充分余地的同时，应当考虑各种安全对策，以保证即使发生故障，也能将危险减低到最小程度：

- a 用于户外、或者可能有潜在的化学污染或妨害电气情形下；
- b 用于原子控制设备、焚烧设备、铁路、航空、医用设备、安全设施以及其他必须经相关部门许可的特殊设备；
- c 用于可能危及人身、财产的系统、设备、装置；
- d 其他用于类似上述 a - c 的被要求高度安全性、高信赖性的用途。

(3) 当客户将本公司产品用于与人身、财产安全密切相关的场合时，应明确告知使用者系统整体的危险性，并确保安全性应采取特殊的附加设计。

4. 设计规格的变更

本产品目录中记载的各项设计规格，由于各种原因，可能会根据需要进行变更，购买时请与销售网点的人员确认实际的设计规格。

5. 服务范围

本公司产品价格中不包含技术人员派遣等服务费用。如客户需要提供相关技术服务，可与销售网点的人员联系。

6. 适合范围

上述内容仅限中国大陆地区的交易。其他地区和海外的交易及使用注意事项请与当地销售网点的人员咨询。

7. 关于海外使用

本产品中如包含有汇兑及国际贸易管理法中规定需办理许可的出口货物（或技术）时，对外出口（或向外国公民提供本产品时），请办理许可申请及相关手续。

9. 本公司保留对本手册的修订权，手册的修订更新，将在公司网站上发布，恕不另行通知，具体请关注我们的网站。

10. 最新修订日期：20260114



联系信息：

广州市研为电子科技有限公司

网址：www.gzyanwei.com

电话：020-39337878

传真：020-39159010

Email: info@gzyanwei.com

1 说明

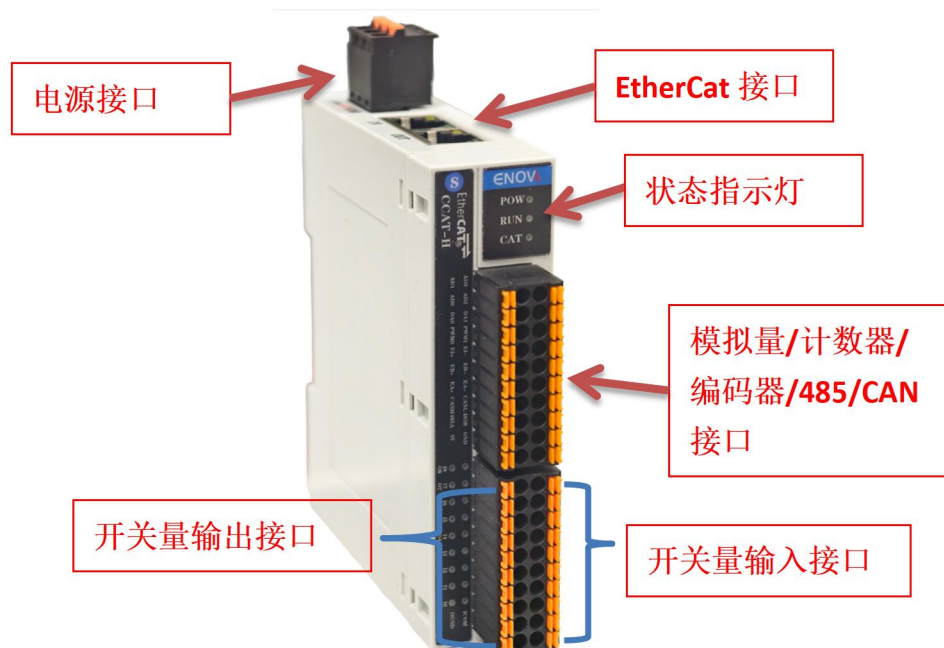
EtherCAT IO 卡是基于 EtherCAT 总线通信的 IO 卡。EtherCat 主站通过网口(IN)与 IO 卡通信，网口（OUT）可与其他 EtherCAT 从站设备连接。

1.1 硬件资源

- 1、2 个 Ecat 总线网络接口。
- 2、9 路开关量输出。
- 3、9 路开关量输入。
- 4、4 路模拟量输入。
- 5、2 路模拟量输出。
- 6、2 路 PWM 输出。
- 7、1 个编码器接口。
- 8、1 个计数器接口。
- 9、1 个 485 串口。
- 10、1 个 CAN 接口

2 接口布局

IO 卡的整体布局如图所示：

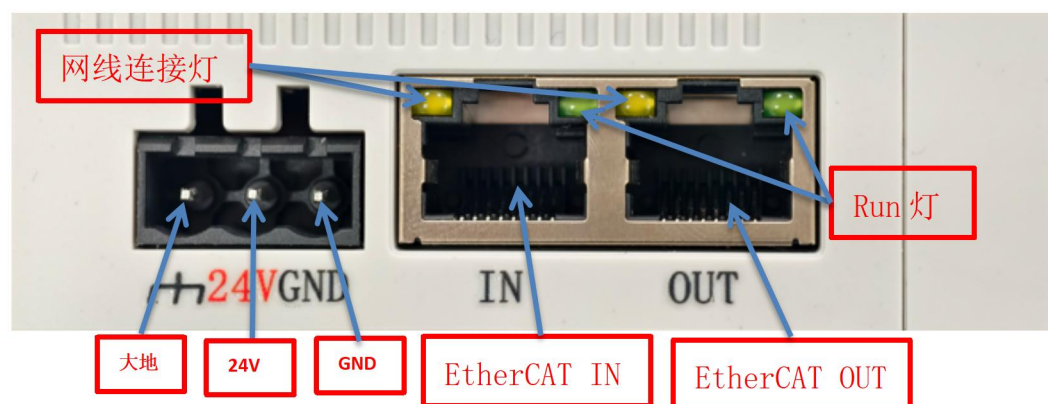


接线示意图：



2.1 电源和 EtherCat 接口

IO 卡的电源输入和 EtherCat 接口位于侧面，如下图所示，其中，从左边起分别为大地，24V，GND，EtherCat In 口，EtherCat Out 口。



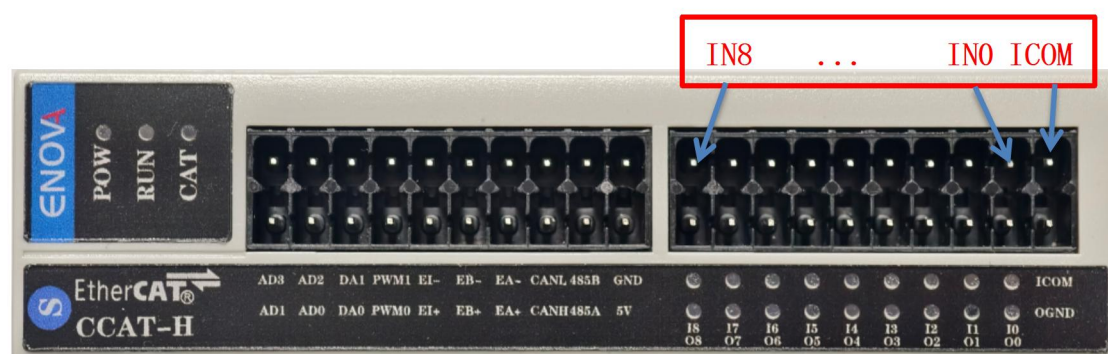
EtherCat 口上的连接指示灯可以指示网线连接状态：灯亮，网线连接正常，灯灭，网线连接断开。

Run 灯可以指示 EtherCat 通信状态：

- 长灭：INIT 状态
- 快闪：Pre-OP 状态
- 慢闪：Safe-OP 状态
- 长亮：OP 状态

2.2 开关量输入

提供 9 个开关量输入接口，一个输入公共端 ICOM。如图所示：



每个接口都有 1 个 LED 灯显示开关是断开还是闭合。

接线方案如下所示：

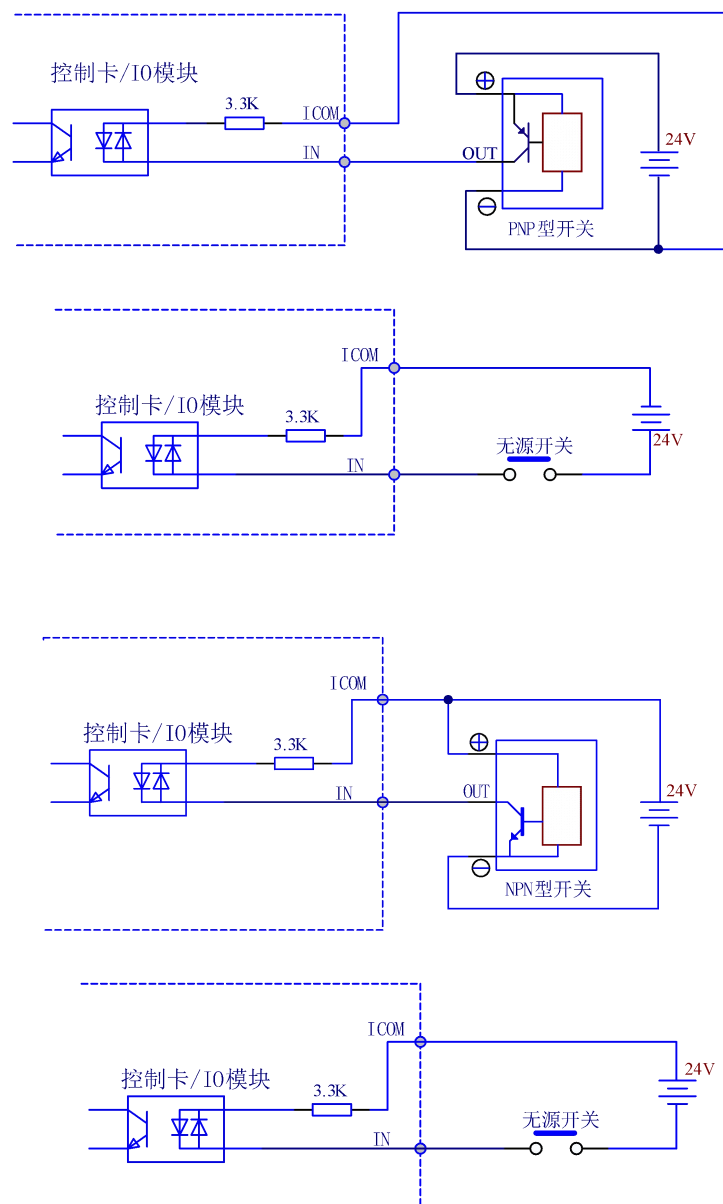
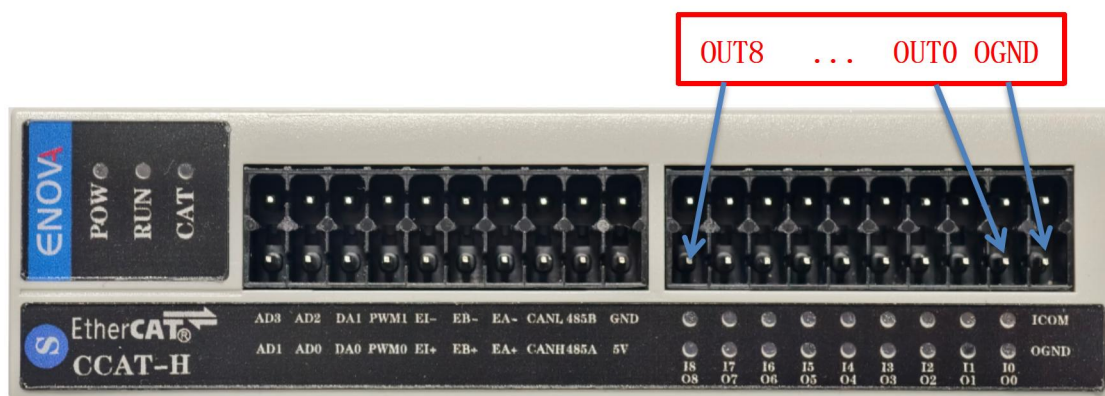


图 2-1 开关量输入接口接线方案

2.3 开关量输出

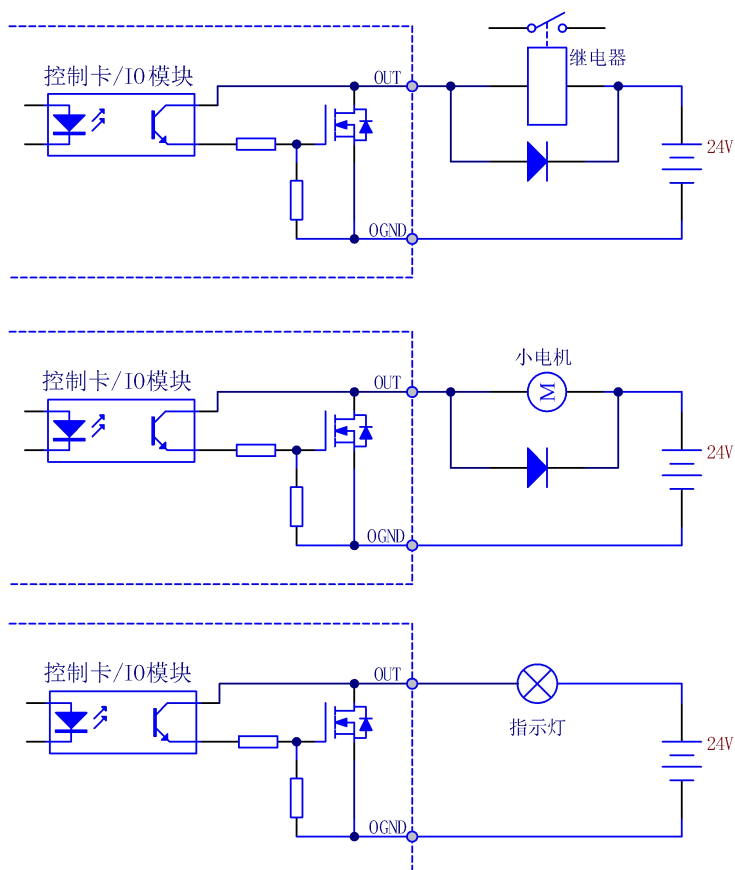
提供 9 个开关量输出接口，OUT0 ~ OUT8 共用一个输出公共端 OGND，如图所示：



每个接口都有 1 个 LED 灯显示输出是断开还是闭合。

每一路输出的灌入电流最大不能超过 500mA，为延长寿命，建议小于 300mA，若使用较大功率的负载，请通过继电器或其它转接电路。“图 2-2 开关量输出接口接线方案”中继电器等外围设备的 24V 供电，可以与控制卡共用一组电源，也可以使用独立的电源（抗干扰性更强）。

接线方案如“图 2-2 开关量输出接口接线方案”所示：



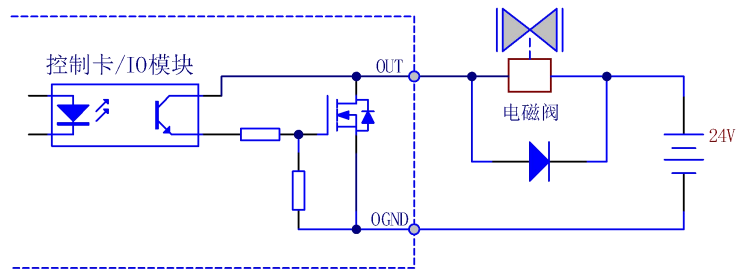
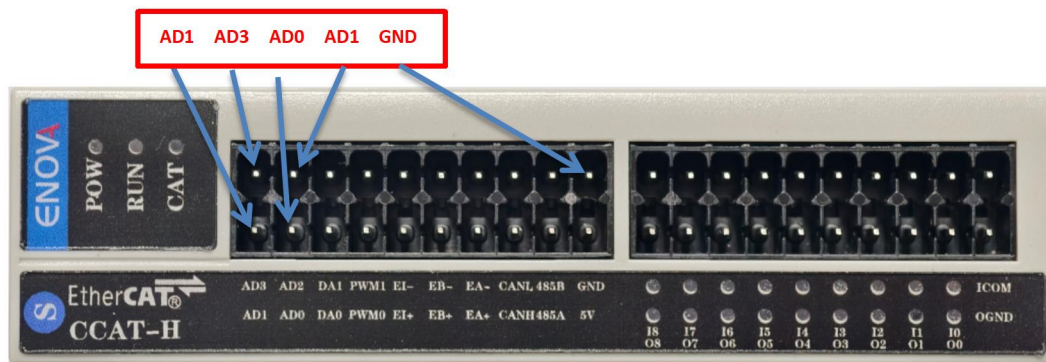


图 2-2 开关量输出接口接线方案

2.4 模拟量输入

提供 4 路模拟量输入，如图所示，输入电压范围是 0 ~ 10V。



接线方案如“图 2-3 模拟量输入接口接线方案”所示

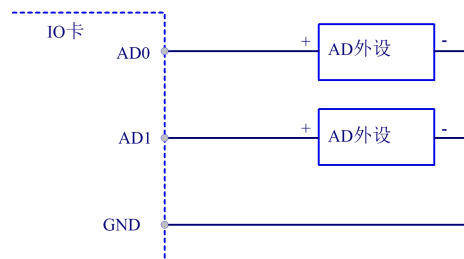


图 2-3 模拟量输入接口接线方案

2.5 模拟量输出

提供 2 路模拟量输出，如图所示，输出电压范围是 0 ~ 10V。



接线方案如“图 2-4 模拟量输出接口接线方案”所示。

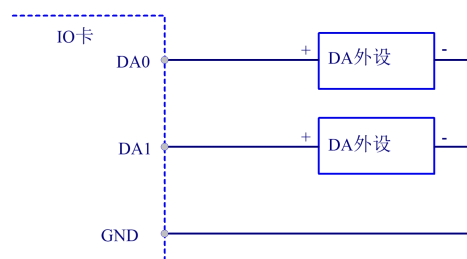
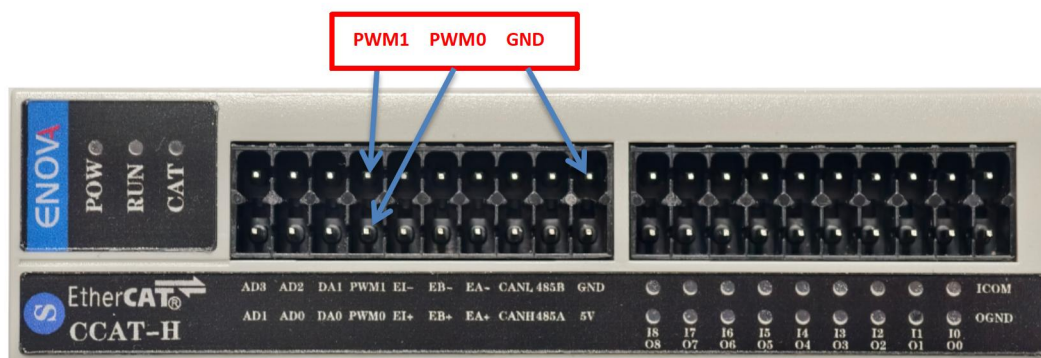


图 2-4 模拟量输出接口接线方案

2.6 PWM

提供 2 路 PWM 输出，如图所示。PWM 输出为 TTL 电平（5V），最大频率 65K，可设置 PWM 的有效电平。



接线方案如“图 2-5 PWM 输出接口接线方案”所示。

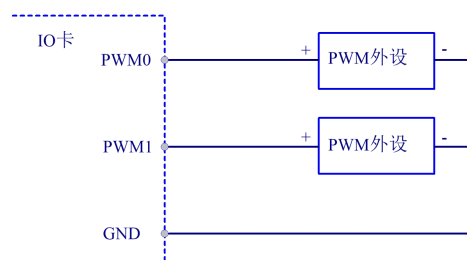


图 2-5 PWM 输出接口接线方案

2.7 编码器/计数器

提供一个编码器接口，如图所示。编码器计数的位数为 32 位。可用于连接电子手轮、光栅尺等外设。该接口可接收 EIA/TIA-422-B 或 EIA/TIA-423-B 标准差分信号，如由 MC3487 或 AM26LS31 等器件驱动的信号。EA+、EA- 分别为 A 相差分信号对的正端和负端，EB+、EB- 为 B 相的差分信号对，EI+、EI- 为 Z 相的差分信号对。若手轮输出的是单端信号，则只需将电子手轮的各相输出分别连接各差分对的负端（EA-、EB-），并将电子手轮的 VCC 连接至差分对的正端。A 相差分信号对和 B 相差分信号对是编码器计数信号，Z 相差分信号对是计数器信号。



接线方案如“图 2-6 编码器差分连接方案”和“图 2-7 编码器单端连接方案”所示：

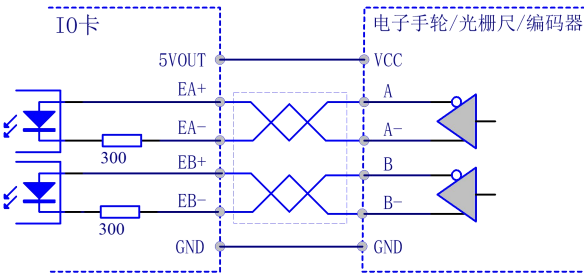


图 2-6 编码器差分连接方案

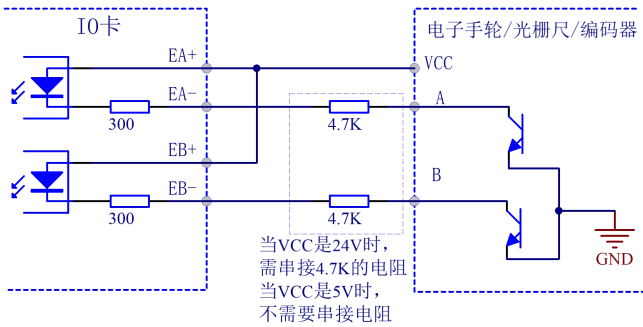


图 2-7 编码器单端连接方案

接线方案如“图 2-8 计数器接口接线方案”所示。

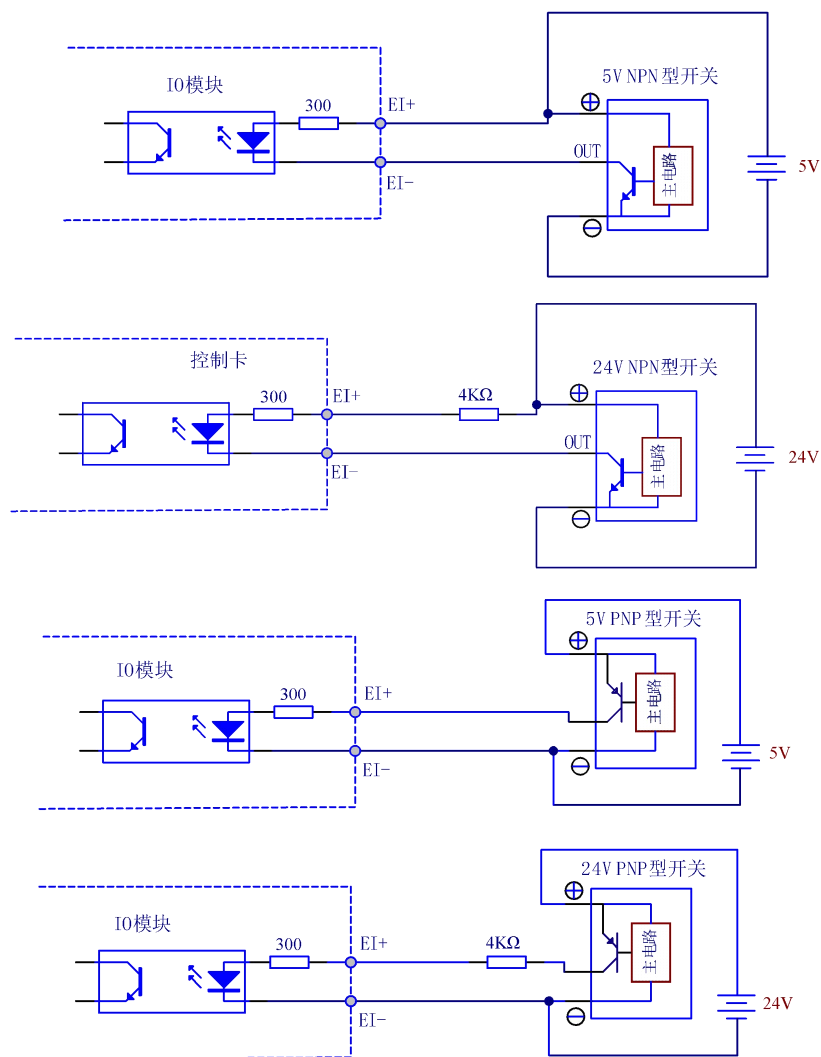
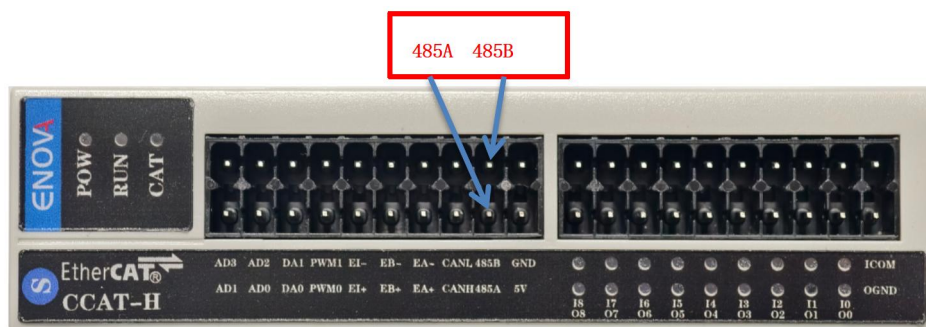


图 2-8 计数器接口接线方案

2.8 485 串口

提供 1 个 485 串口，如图所示。



2.9 CAN □



3 对象字典

3.1 对象类型

类型名称	说明
VAR	单一数值，如一个 UNSIGNED8、UNSIGNED16 或 INTEGER32 等。
ARRAY	多重数据字段对象，字段中的变量均为相同的数据型态。如 UNSIGNED16 数组等，Sub-index 0 属于 UNSIGNED8，所以不是数组数据
RECORD	多重数据字段对象，字段中的变量皆为不同的数据型态。Sub-index 0 属于 UNSIGNED8，不是 RECORD 资料。

3.2 数据类型

类型名称	说明
UNSIGNED8	无符号 8 位数据。
UNSIGNED16	无符号 16 位数据
UNSIGNED32	无符号 32 位数据。
INTEGER32	有符号 32 位数据

3.3 访问权限

权限名称	说明
RO	只读
RW	可读写

3.4 对象字典一览表

对象字典	对象类型	名称	数据类型	访问权限
0x2000	VAR	数字量输入	UNSIGNED16	RO
0x2010	VAR	模拟量输入 1	UNSIGNED16	RO
0x2011	VAR	模拟量输入 2	UNSIGNED16	RO
0x2012	VAR	模拟量输入 3	UNSIGNED16	RO
0x2013	VAR	模拟量输入 4	UNSIGNED16	RO
0x2020	VAR	编码器输入	INTEGER32	RO
0x2030	VAR	计数器	UNSIGNED32	RO
0x2040	VAR	计数器配置	UNSIGNED16	RW
0x2300	VAR	数字量输出	UNSIGNED16	RW
0x2310	VAR	模拟量输出 1	UNSIGNED16	RW
0x2311	VAR	模拟量输出 2	UNSIGNED16	RW
0x2320	VAR	PWM1 频率	UNSIGNED32	RW
0x2321	VAR	PWM2 频率	UNSIGNED32	RW
0x2330	VAR	PWM1 占空比	UNSIGNED16	RW
0x2331	VAR	PWM2 占空比	UNSIGNED16	RW
0x2400	VAR	清零计数器、编码器的值	UNSIGNED16	RW
0x2401	VAR	设置 PWM 极性	UNSIGNED16	RW
0x2402	RECORD	串口设置	UNSIGNED32	RW
0x2600	RECORD	串口数据设置（发送）	UNSIGNED32	RW
0x2601	RECORD	串口数据设置（接收）	UNSIGNED32	RW
0x2602	ARRAY	串口数据（发送）	UNSIGNED32	RW
0x2603	ARRAY	串口数据（接收）	UNSIGNED32	RW
0x2609	RECORD	CAN 波特率设置	UNSIGNED32	RW
0x2610	RECORD	CAN 发送设置	UNSIGNED16	RW
0x2611	RECORD	CAN 接收设置	UNSIGNED16	RW
0x2612	ARRAY	CAN 数据（发送）	UNSIGNED32	RW
0x2613	ARRAY	CAN 数据（接收）	UNSIGNED32	RW
0x3000	RECORD	设置串口 Modbus RTU 协议中的线圈、保持寄存器	UNSIGNED8	RW
0x3001	RECORD		UNSIGNED8	RW
0x3002	RECORD		UNSIGNED8	RW
0x3003	RECORD		UNSIGNED8	RW
0x3004	RECORD		UNSIGNED8	RW
0x3005	RECORD		UNSIGNED8	RW
0x3006	RECORD		UNSIGNED8	RW

0x3007	RECORD	设置串口 Modbus RTU 协议中的离散输入、线圈、输入寄存器、保持寄存器	UNSIGNED8	RW
0x3020	RECORD		UNSIGNED8	RW
0x3021	RECORD		UNSIGNED8	RW
0x3022	RECORD		UNSIGNED8	RW
0x3023	RECORD		UNSIGNED8	RW
0x3024	RECORD		UNSIGNED8	RW
0x3025	RECORD		UNSIGNED8	RW
0x3026	RECORD		UNSIGNED8	RW
0x3027	RECORD		UNSIGNED8	RW

3.5 对象数据

3.5.1 开关量输入(0x2000)

IO 卡上有 9 个开关量输入端口，每个端口的状态对应着 0x2000 对象的相应 bit。当开关连通时，对应的 bit 置 0，当开关断开时，对应的 bit 置 1。

Index	0x2000
名称	开关量输入
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RO
PDO 映射	是
值范围	0 ~ 0xFFFF
默认值	0

对象的每个 bit 与输入端口的对应关系：

BIT							9	8
输入端							IN9	IN8
BIT	7	6	5	4	3	2	1	0
输入端	IN7	IN6	IN5	IN4	IN3	IN2	IN1	IN0

3.5.2 模拟量输入(0x2010、0x2011、0x2012、0x2013)

IO 卡提供有 2 路模拟量输入，每路的转换精度为 12 位。

Index	0x2010
名称	模拟量输入 1
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RO

PDO 映射	是
值范围	0 ~ 0x0FFF
默认值	0
备注	输入电压范围是：0 ~ 10V 电压与模拟量数值的转换关系为： $V = AD * 10 / 4096$ （AD 为模拟量数值，V 为电压值）

Index	0x2011
名称	模拟量输入 2
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RO
PDO 映射	是
值范围	0 ~ 0x0FFF
默认值	0
备注	输入电压范围是：0 ~ 10V 电压与模拟量数值的转换关系为： $V = AD * 10 / 4096$ （AD 为模拟量数值，V 为电压值）

Index	0x2012
名称	模拟量输入 3
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RO
PDO 映射	是
值范围	0 ~ 0x0FFF
默认值	0
备注	输入电压范围是：0 ~ 10V 电压与模拟量数值的转换关系为： $V = AD * 10 / 4096$ （AD 为模拟量数值，V 为电压值）

Index	0x2013
名称	模拟量输入 4

对象类型	VAR
数据类型	UNSIGNED16
访问权限	RO
PDO 映射	是
值范围	0 ~ 0x0FFF
默认值	0
备注	输入电压范围是：0 ~ 10V 电压与模拟量数值的转换关系为： $V = AD * 10 / 4096$ （AD 为模拟量数值，V 为电压值）

3.5.3 编码器输入(0x2020)

Index	0x2040
名称	编码器输入
对象类型	VAR
数据类型	INTEGER32
访问权限	RO
PDO 映射	是
值范围	- 2147483648 ~ 2147483647
默认值	0

3.5.4 计数器(0x2030)

Index	0x2030
名称	计数器输入
对象类型	VAR
数据类型	UNSIGNED32
访问权限	RO
PDO 映射	是
值范围	0 ~ 0xFFFFFFFF
默认值	0

3.5.5 计数器配置(0x2040)

Index	0x2040
名称	计数器功能配置
对象类型	VAR

数据类型	UNSIGNED16
访问权限	RO
PDO 映射	是
值范围	0 ~ 0xFFFF
默认值	0

对象的每个 bit 与输出端口的对应关系：

BIT	15 ~ 3	2	1	0
输出端	Res	Only	Polar	Fun
说明	Res: 预留 Only: 当 Fun=1 时, 是否只清零一次编码器 Polar: 当 Fun=1 时, 指定边沿触发清零操作; 0: 下降沿; 1: 上升沿 Fun: 检测到输入信号的功能: 0: 计数器功能; 1: 清零编码器功能			

3.5.6 开关量输出(0x2300)

IO 卡上有 9 个开关量输出端口, 对象 0x2300 的每个 bit 对应着相应端口的状态。当对应的 bit 置 0 时开关连通, 当对应的 bit 置 1 时则开关断开。

Index	0x2300
名称	开关量输出
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFF
默认值	0xFFFF

对象的每个 bit 与输出端口的对应关系：

BIT							9	8
输出端							OUT9	OUT8
BIT	7	6	5	4	3	2	1	0
输出端	OUT7	OUT6	OUT5	OUT4	OUT3	OUT2	OUT1	OUT0

3.5.7 模拟量输出(0x2310、0x2311)

IO 卡提供有 2 路模拟量输出, 每路的转换精度为 12 位。

Index	0x2310
-------	--------

名称	模拟量输出 1
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0x0FFF
默认值	0
备注	输出电压范围是：0 ~ 10V 电压与模拟量数值的转换关系为： $DA = V * 4096 / 10$ (DA 为模拟量数值，V 为电压值)

Index	0x2311
名称	模拟量输出 2
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0x0FFF
默认值	0
备注	输出电压范围是：0 ~ 10V 电压与模拟量数值的转换关系为： $DA = V * 4096 / 10$ (DA 为模拟量数值，V 为电压值)

3.5.8 PWM 输出频率(0x2320、0x2321)

Index	0x2320
名称	PWM1 输出频率
对象类型	VAR
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFFFFFF

默认值	0x3E80000(65536000)
备注	<p>PWM 的基准时钟频率是 4M。</p> <p>PWM 输出频率是实际频率放大 65536 倍之后的数值，因此实际频率的范围是 0 ~ 65535.9999，它们的转换关系为：</p> $F_{pwm} = F_d * 65536$ <p>(Fpwm 为 PWM 输出频率，Fd 为实际频率)</p>

Index	0x2321
名称	PWM2 输出频率
对象类型	VAR
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFFFFFF
默认值	0x3E80000(65536000)
备注	<p>PWM 的基准时钟频率是 4M。</p> <p>PWM 输出频率是实际频率放大 65536 倍之后的数值，因此实际频率的范围是 0 ~ 65535.9999，它们的转换关系为：</p> $F_{pwm} = F_d * 65536$ <p>(Fpwm 为 PWM 输出频率，Fd 为实际频率)</p>

3.5.9 PWM 占空比(0x2330、0x2331)

Index	0x2330
名称	PWM1 占空比
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFF
默认值	0x8000
备注	<p>PWM 占空比是实际占空比放大 65536 倍之后的数值，因此实际占空比的范围是 0 ~ 0.9999，它们的转换关系为：</p> $P_{pwm} = P_d * 65536$ <p>(Ppwm 为 PWM 占空比，Pd 为实际占空比)</p>

Index	0x2331
-------	--------

名称	PWM2 占空比
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFF
默认值	0x8000
备注	<p>PWM 占空比是实际占空比放大 65536 倍之后的数值，因此实际占空比的范围是 0 ~ 0.9999，它们的转换关系为：</p> $P_{pwm} = P_d * 65536$ <p>(P_{pwm} 为 PWM 占空比，P_d 为实际占空比)</p>

3.5.10 清零编码器、计数器的值(0x2400)

Index	0x2400				
名称	清零编码器、计数器的值				
对象类型	VAR				
数据类型	UNSIGNED16				
访问权限	RW				
PDO 映射	否				
值范围	0 ~ 0xFFFF				
默认值	0				
备注	...	Bit 9	Bit 8	...	Bit 0
	RES	RES	CLRC1	RES	CLRE
	<p>RES: 保留，无意义</p> <p>CLRE: 由 0 变为 1 时，则清零编码器值</p> <p>CLRC1: 由 0 变为 1 时，则清零计数器 1 的值</p>				

3.5.11 设置 PWM 的有效极性 (0x2401)

Index	0x2401
名称	设置 PWM、计数器的有效极性
对象类型	VAR
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否
值范围	0 ~ 0xFFFF

默认值	0xFFFF					
备注	...	Bit 13	Bit 12	...	Bit 9	Bit 8
	RES	DCNT2	DCNT1	RES	SCNT2	SCNT1
	Bit7 – bit2				Bit 1	Bit 0
	RES				SPWM2	SPWM1
	RES: 保留, 无意义 SPWM1: 设置 PWM1 的有效极性, SPWM2: 设置 PWM2 的有效极性,					

3.5.12 串口设置(0x2402)

通过此对象设置串口的波特率和串口的通信协议，以及 Modbus 通信的超时时间、轮询间隔。

Index	0x2402
名称	串口设置
对象类型	RECORD
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否

subindex	0
描述	对象成员数量
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	4
默认值	4

subindex	1
描述	Modbus 通信超时时间
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否
值范围	2 ~ 65535
默认值	100
备注	超时时间单位为 0.1 毫秒

subindex	2
描述	串口波特率
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	否
值范围	9600、19200、38400、57600、115200
默认值	115200

subindex	3
描述	Modbus 周期
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否
值范围	1~ 65535
默认值	2
备注	周期单位为毫秒

subindex	4
描述	串口通信协议
数据类型	UNSIGNED8
访问权限	RW
PDO 映射	否
值范围	0~ 0xFF
默认值	0
备注	0: Modbus RTU 协议 其他: 用户自定义协议

3.5.13 串口数据（发送） (0x2602)

Index	0x2602
名称	串口数据
对象类型	ARRAY
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	是

subindex	0
描述	数据个数
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	8
默认值	8

subindex	1 ~ 8
描述	通过串口发送的数据值
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFFFFFF
默认值	0

备注

当 0x2402.4 设置为 Modbus 协议时，其值是写到从站中的线圈或保持寄存器的值。由 0x3000 ~ 0x3007 对象设置对应的 0x2602 对象属性：

对应关系		对应关系	
0x2602.1	< - > 0x3000	0x2602.5	< - > 0x3004
0x2602.2	< - > 0x3001	0x2602.6	< - > 0x3005
0x2602.3	< - > 0x3002	0x2602.7	< - > 0x3006
0x2602.4	< - > 0x3003	0x2602.8	< - > 0x3007

当 0x2402.4 设置为用户自定义协议时，Data 1 ~ 32 用于传输需要通过串口发送的数据：

bit	0 - 7	8 - 15	16 - 23	24 - 31
0x2602.1	Data 1	Data 2	Data 3	Data 4
0x2602.2	Data 5	Data 6	Data 7	Data 8
0x2602.3	Data 9	Data 10	Data 11	Data 12
0x2602.4	Data 13	Data 14	Data 15	Data 16
0x2602.5	Data 17	Data 18	Data 19	Data 20
0x2602.6	Data 21	Data 22	Data 23	Data 24
0x2602.7	Data 25	Data 26	Data 27	Data 28
0x2602.8	Data 29	Data 30	Data 31	Data 32

3.5.14 串口数据（接收） (0x2603)

Index	0x2603
名称	串口数据
对象类型	ARRAY
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	是

subindex	0
描述	数据个数
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	8
默认值	8

subindex	1 ~ 8																														
描述	串口接收到的数据值																														
数据类型	UNSIGNED32																														
访问权限	RW																														
PDO 映射	是																														
值范围	0 ~ 0xFFFFFFFF																														
默认值	0																														
备注	<div>当 0x2402.4 设置为 Modbus 协议时,是从站中读回的线圈、离散输入、输入寄存器、保持寄存器的值。由 0x3020 ~ 0x3027 对象设置对应的 0x2603 对象属性:</div> <table><tr><th colspan="2">对应关系</th><th colspan="2">对应关系</th></tr><tr><td>0x2603.1</td><td>< - > 0x3020</td><td>0x2603.5</td><td>< - > 0x3024</td></tr><tr><td>0x2603.2</td><td>< - > 0x3021</td><td>0x2603.6</td><td>< - > 0x3025</td></tr><tr><td>0x2603.3</td><td>< - > 0x3022</td><td>0x2603.7</td><td>< - > 0x3026</td></tr><tr><td>0x2603.4</td><td>< - > 0x3023</td><td>0x2603.8</td><td>< - > 0x3027</td></tr></table> <div>当 0x2402.4 设置为用户自定义协议时,0x2601.2 由其他值改为 0xFF 时,IO 卡将串口接收到的数据更新到 data 1~32,更新的个数由 0x2601.4 表明:</div> <table><tr><th>bit</th><th>0 – 7</th><th>8 - 15</th><th>16 - 23</th><th>24 - 31</th></tr><tr><td>0x2603.1</td><td>Data 1</td><td>Data 2</td><td>Data 3</td><td>Data 4</td></tr></table>	对应关系		对应关系		0x2603.1	< - > 0x3020	0x2603.5	< - > 0x3024	0x2603.2	< - > 0x3021	0x2603.6	< - > 0x3025	0x2603.3	< - > 0x3022	0x2603.7	< - > 0x3026	0x2603.4	< - > 0x3023	0x2603.8	< - > 0x3027	bit	0 – 7	8 - 15	16 - 23	24 - 31	0x2603.1	Data 1	Data 2	Data 3	Data 4
对应关系		对应关系																													
0x2603.1	< - > 0x3020	0x2603.5	< - > 0x3024																												
0x2603.2	< - > 0x3021	0x2603.6	< - > 0x3025																												
0x2603.3	< - > 0x3022	0x2603.7	< - > 0x3026																												
0x2603.4	< - > 0x3023	0x2603.8	< - > 0x3027																												
bit	0 – 7	8 - 15	16 - 23	24 - 31																											
0x2603.1	Data 1	Data 2	Data 3	Data 4																											

	0x2603.2	Data 5	Data 6	Data 7	Data 8
	0x2603.3	Data 9	Data 10	Data 11	Data 12
	0x2603.4	Data 13	Data 14	Data 15	Data 16
	0x2603.5	Data 17	Data 18	Data 19	Data 20
	0x2603.6	Data 21	Data 22	Data 23	Data 24
	0x2603.7	Data 25	Data 26	Data 27	Data 28
	0x2603.8	Data 29	Data 30	Data 31	Data 32

3.5.15 自定义协议（发送） (0x2600)

当 0x2402.4 设置为自定义协议时，可通过 0x2602 对象将串口数据传递到 IO 卡，再通过 IO 卡的串口发送。

要使用串口，则必须先将数据发送到 0x2602.1~8，然后将 0x2602.1~8 中的有效数据个数（BYTE）设置到 0x2600.4，再修改 0x2600.3 中的 ID，IO 卡通过判断 ID 的不同后，会将 0x2602.1~8 中的数据保存到缓存区，因此每发送一次数据，0x2600.3 中的 ID 必须与上一次的不同。将所有数据发送到 IO 卡后，再将 0x2600.1 由其他值改为 0xFF，IO 将缓存区的数据通过串口发送。0x2600.2 表明数据发送的状态码。

Index	0x2600
名称	串口数据
对象类型	RECORD
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	是

subindex	0
描述	成员个数
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	4
默认值	4

subindex	1
描述	发送使能
数据类型	UNSIGNED16

访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFF
默认值	0
备注	使能串口数据传输。当需要发送的数据都传输到 IO 卡之后，由其他值变为 0xFF 时可使 IO 卡将缓存的数据通过串口发送。

subindex	2
描述	数据发送的状态
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFF
默认值	0
备注	当 bit0 置 1 时，串口缓存区中的数据长度超出最大值(256) 当 bit1 置 1 时，表明串口正在发送数据。

subindex	3
描述	0x2602.1~8 中的串口数据 ID
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFF
默认值	0
备注	当 ID 与上次不同，且 subindex4 不为 0 时，0x2602.1~8 中的数据被保存到缓存区。

subindex	4
描述	0x2602.1~8 中的串口数据个数，单位 byte
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFF
默认值	0
备注	0x2602.1~8 中传输数据的有效个数 (byte)，其值不能超

	过 32。这些数据将被缓存到缓存区中，缓存区可缓存的总数据长度为 256 byte。
--	--

3.5.16 自定义协议（接收） (0x2601)

当 0x2402.4 设置为自定义协议时，串口接收到数据，会将数据保存到缓存区，并在 0x2601.1 中声明缓存区中的数据个数（最大 256）。当 0x2601.2 由其他值被改为 0xFF 时，串口会将缓存区中的数据更新到 0x2603.1~8，并在 0x2601.4 表明数据的有效个数，在 0x2601.3 中表明此数据的索引。串口每更新一次数据，数据的索引加 1。

Index	0x2601
名称	串口数据
对象类型	RECORD
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	是

subindex	0
描述	成员个数
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	4
默认值	4

subindex	1
描述	串口接收到缓存区中的数据个数
数据类型	UNSIGNED16
访问权限	RO
PDO 映射	是
值范围	0 ~ 256
默认值	0
备注	当 subindex2 由其他值改为 0xFF 时，IO 卡将串口接收到的数据更新到 0x2603.1~8，则未被更新的数据个数保存在 subindex1。

subindex	2
----------	---

描述	更新 0x2603.1~8 中的数据
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFF
默认值	0
备注	当 subindex2 由其他值改为 0xFF 时, IO 卡将串口接收到的数据更新到 0x2603.1~8, 更新的个数由 subindex4 表明。

subindex	3
描述	0x2603.1~8 中的数据 ID
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 0xFFFF
默认值	0
备注	当 ID 与上次不同时, 表明 0x2603.1~8 中的数据已经被更新。更新的数据个数由 subindex 4 决定。

subindex	4
描述	0x2603.1~8 中的数据有效数据个数 (单位 byte)
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	是
值范围	0 ~ 32
默认值	0
备注	

3.5.17 Modbus 寄存器设置 (写) (0x3000 ~ 0x3007)

IO 卡有 8 个对象用来设置 Modbus 写寄存器的通信, 每个对象可设置一个从站对象。

Index	0x3000 ~ 0x3007
名称	Modbus 寄存器设置
对象类型	RECORD
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否

subindex	0
描述	成员数量
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	7
默认值	7

subindex	1						
描述	Modbus 状态值						
数据类型	UNSIGNED16						
访问权限	RO						
PDO 映射	是						
值范围	0 ~ 0xFFFF						
默认值	0						
备注	...	Bit6	Bit5	Bit4	...	Bit1	Bit0
	RES	TYPE	SID	BIT	RES	TO	RES
	当对应的 bit 置 1 时：						
	RES：保留位，无意义						
	TO：通信超时						
	BIT：设置 Modbus 寄存器的有效位数不正确						
	SID：设置 Modbus 从站的 ID 不正确						
	TYPE：设置 Modbus 寄存器的类型不正确						

subindex	2
描述	设置 Modbus 寄存器地址
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否
值范围	0 ~ 0xFFFF
默认值	0

subindex	3
描述	设置 Modbus 寄存器的有效位数
数据类型	UNSIGNED8
访问权限	RW
PDO 映射	否
值范围	0 ~ 32
默认值	0
备注	0: 无效值, 使用时必须改写此值 当设置 Modbus 寄存器的类型为线圈时, 取值范围是 1 ~ 32, 表示需控制的线圈数量。 当设置 Modbus 寄存器的类型为保持寄存器时, 只能取值 16 或 32, 表示需控制的保持寄存器的位数

subindex	4
描述	设置 Modbus 从站的 ID
数据类型	UNSIGNED8
访问权限	RW
PDO 映射	否
值范围	1 ~ 247
默认值	0
备注	0: 无效值, 使用时必须改写此值

subindex	5
描述	设置 Modbus 寄存器的类型
数据类型	UNSIGNED8
访问权限	RW
PDO 映射	否
值范围	1、4
默认值	0
备注	0: 无效值, 使用时必须改写此值 1: 线圈 4: 保持寄存器

subindex	6
描述	使能
数据类型	UNSIGNED8

访问权限	RW
PDO 映射	否
值范围	0、1
默认值	0
备注	0: 不对此设置的寄存器进行写操作 1: 定时对设置的从站寄存器进行写操作

subindex	7
描述	通信错误次数
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否
值范围	0~65535
默认值	0
备注	

3.5.18 Modbus 寄存器设置（读）(0x3020 ~ 0x3027)

IO 卡有 8 个对象用来设置 Modbus 读寄存器的通信，每个对象可读取一个从站对象值。

Index	0x3020 ~ 0x3027
名称	Modbus 寄存器设置
对象类型	RECORD
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否

subindex	0
描述	Modbus 寄存器设置数量
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	7
默认值	7

subindex	1
描述	Modbus 状态值

数据类型	UNSIGNED16							
访问权限	RO							
PDO 映射	是							
值范围	0 ~ 0xFFFF							
默认值	0							
备注	...	Bit6	Bit5	Bit4	...	Bit1	Bit0	
	RES	TYPE	SID	BIT	RES	TO	RES	
<p>当对应的 bit 置 1 时：</p> <p>RES：保留位，无意义</p> <p>TO：通信超时</p> <p>BIT：设置 Modbus 寄存器的有效位数不正确</p> <p>SID：设置 Modbus 从站的 ID 不正确</p> <p>TYPE：设置 Modbus 寄存器的类型不正确</p>								

subindex	2
描述	设置 Modbus 寄存器地址
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否
值范围	0 ~ 0xFFFF
默认值	0

subindex	3
描述	设置 Modbus 寄存器的有效位数
数据类型	UNSIGNED8
访问权限	RW
PDO 映射	否
值范围	0 ~ 32
默认值	0
备注	<p>0：表示无效值。使用时必须改写此值。</p> <p>当设置 Modbus 寄存器的类型为离散输入或线圈时，取值范围是 1 ~ 32，表示需读取的离散输入或线圈数量。</p> <p>当设置 Modbus 寄存器的类型为输入寄存器或保持寄存器时，只能取值 16 或 32，表示输入寄存器或保持寄存器的位数</p>

subindex	4
描述	设置 Modbus 从站的 ID
数据类型	UNSIGNED8
访问权限	RW
PDO 映射	否
值范围	1 ~ 247
默认值	0
备注	0: 无效值, 使用时必须改写此值

subindex	5
描述	设置 Modbus 寄存器的类型
数据类型	UNSIGNED8
访问权限	RW
PDO 映射	否
值范围	0 ~ 4
默认值	0
备注	0: 无效值, 使用时必须改写此值 1: 线圈 2: 离散输入 3: 输入寄存器 4: 保持寄存器

subindex	6
描述	使能
数据类型	UNSIGNED8
访问权限	RW
PDO 映射	否
值范围	0、1
默认值	0
备注	0: 不对此设置的寄存器进行读 1: 对此设置的寄存器进行读操作

subindex	7
描述	通信错误次数
数据类型	UNSIGNED16
访问权限	RW

PDO 映射	否
值范围	0~65535
默认值	0
备注	

3.5.19 CAN 波特率设置(0x2609)

Index	0x2609
名称	波特率设置
对象类型	RECORD
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	否

subindex	0
描述	数据个数
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	2
默认值	2

subindex	1
描述	经典 CAN 的波特率
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	否
值范围	0 ~ 0xFFFFFFFF
默认值	500000
备注	

subindex	2
描述	FDCAN 的波特率
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	否
值范围	0 ~ 0xFFFFFFFF
默认值	1000000

备注	
----	--

3.5.20 CAN 发送帧信息(0x2610)

通过此对象设置串口的波特率和串口的通信协议，以及 Modbus 通信的超时时间、轮询间隔。

Index	0x2610
名称	CAN 发送的数据信息
对象类型	RECORD
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否

subindex	0
描述	对象成员数量
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	6
默认值	6

subindex	1
描述	发送标志
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	RxPDO
值范围	0 ~ 65535
默认值	0
备注	当需要发送的数据都传输到 0x2612 对象后，改变此对象的值，则 0x2612 对象中的数据将通过 CAN 口发送。

subindex	2
描述	CAN ID
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	RxPDO
值范围	0 ~ 0x1FFFFFFF

默认值	0
-----	---

subindex	3
描述	发送状态
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	TxPDO
值范围	0~ 65535
默认值	0
备注	当 subindex1 的值改变时，此对象返回发送状态

subindex	4
描述	需要发送的数据大小
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	RxPDO
值范围	0~ 64
默认值	0
备注	发送缓存区中需要发送的数据大小，单位 byte

subindex	5
描述	帧格式
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	RxPDO
值范围	0~ 65535
默认值	0
备注	0：经典 CAN； 非零：FDCAN

subindex	6
描述	远程帧标志
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	RxPDO
值范围	0~ 65535
默认值	0

备注	0：数据帧； 非零：远程帧
----	---------------

3.5.21 CAN 接收帧信息(0x2611)

通过此对象设置串口的波特率和串口的通信协议，以及 Modbus 通信的超时时间、轮询间隔。

Index	0x2611
名称	CAN 接收到的帧信息
对象类型	RECORD
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	否

subindex	0
描述	对象成员数量
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	6
默认值	6

subindex	1
描述	接收数据更新标志
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	RxPDO
值范围	0 ~ 65535
默认值	0
备注	改变对象的值，则将接收到的数据更新到 0x2613 对象中。

subindex	2
描述	CAN ID
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	TxPDO
值范围	0 ~ 0x1FFFFFFF
默认值	0

subindex	3
描述	接收到的帧数量
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	TxPDO
值范围	0~ 65535
默认值	0
备注	当 subindex1 的值改变时，此对象返回发送状态

subindex	4
描述	0x2613 对象中有效数据的大小
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	TxPDO
值范围	0~ 32
默认值	0
备注	单位 byte

subindex	5
描述	0x2613 对象中的帧格式
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	TxPDO
值范围	0~ 65535
默认值	0
备注	0：经典 CAN； 非零：FDCAN

subindex	6
描述	0x2613 对象中的远程帧标志
数据类型	UNSIGNED16
访问权限	RW
PDO 映射	TxPDO
值范围	0~ 65535
默认值	0
备注	0：数据帧； 非零：远程帧

3.5.22 CAN 数据（发送） (0x2612)

Index	0x2602
名称	CAN 发送的数据
对象类型	ARRAY
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	是

subindex	0
描述	数据个数
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	8
默认值	8

subindex	1 ~ 8				
描述	通过 CAN 发送的数据值				
数据类型	UNSIGNED32				
访问权限	RW				
PDO 映射	是				
值范围	0 ~ 0xFFFFFFFF				
默认值	0				
备注	Data 1 ~ 32 用于传输需要通过串口发送的数据：				
	bit	0 – 7	8 - 15	16 - 23	24 - 31
	0x2612.1	Data 1	Data 2	Data 3	Data 4
	0x2612.2	Data 5	Data 6	Data 7	Data 8
	0x2612.3	Data 9	Data 10	Data 11	Data 12
	0x2612.4	Data 13	Data 14	Data 15	Data 16
	0x2612.5	Data 17	Data 18	Data 19	Data 20
	0x2612.6	Data 21	Data 22	Data 23	Data 24
	0x2612.7	Data 25	Data 26	Data 27	Data 28
	0x2612.8	Data 29	Data 30	Data 31	Data 32

3.5.23 CAN 数据（接收） (0x2613)

Index	0x2602
名称	CAN 接收的数据
对象类型	ARRAY
数据类型	UNSIGNED32
访问权限	RW
PDO 映射	是

subindex	0
描述	数据个数
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	8
默认值	8

subindex	1 ~ 8				
描述	CAN 接收的数据值				
数据类型	UNSIGNED32				
访问权限	RW				
PDO 映射	是				
值范围	0 ~ 0xFFFFFFFF				
默认值	0				
备注	当 0x2611.1 的值改变时，Data 1 ~ 32 将更新为 CAN 接收到的数据：				
	bit	0 – 7	8 - 15	16 - 23	24 - 31
	0x2612.1	Data 1	Data 2	Data 3	Data 4
	0x2612.2	Data 5	Data 6	Data 7	Data 8
	0x2612.3	Data 9	Data 10	Data 11	Data 12
	0x2612.4	Data 13	Data 14	Data 15	Data 16
	0x2612.5	Data 17	Data 18	Data 19	Data 20
	0x2612.6	Data 21	Data 22	Data 23	Data 24
	0x2612.7	Data 25	Data 26	Data 27	Data 28
	0x2612.8	Data 29	Data 30	Data 31	Data 32

3.5.24 固件日期(0x4000)

Index	0x4000
名称	固件日期
对象类型	VAR
数据类型	UNSIGNED32
访问权限	RO
PDO 映射	否
值范围	0 ~ 0xFFFFFFFF
默认值	0x20251121
备注	以十六进制字符表示年、月、日

3.5.25 序列号 (0x4001)

Index	0x4001
名称	序列号
对象类型	ARRAY
数据类型	UNSIGNED32
访问权限	RO
PDO 映射	否

subindex	0
描述	数据个数
数据类型	UNSIGNED8
访问权限	RO
PDO 映射	否
值范围	2
默认值	2

subindex	1 ~ 2
描述	序列号
数据类型	UNSIGNED32
访问权限	RO
PDO 映射	否
值范围	0 ~ 0xFFFFFFFF
默认值	0
备注	每一张 IO 卡都有一个唯一的序列号，此序列号有 8byte。用 2 个 32 位的数据存储。

4 快速测试工具

EcatIO_Test.exe 程序提供了快速测试 IO 卡的所有功能。

4.1 驱动程序

要使用此程序测试 IO 卡，必须先安装驱动。驱动安装程序在配套的光盘中，或者从官网下载的光盘资料中获得。驱动安装程序名称为 IMCdrv_Ins.exe。**win7 及 win10 系统安装驱动需要管理员权限**，因此需右键单击该 exe，选择“以管理员身份运行”运行安装程序。运行安装程序出现如“图 4-1 驱动安装程序”的对话框：

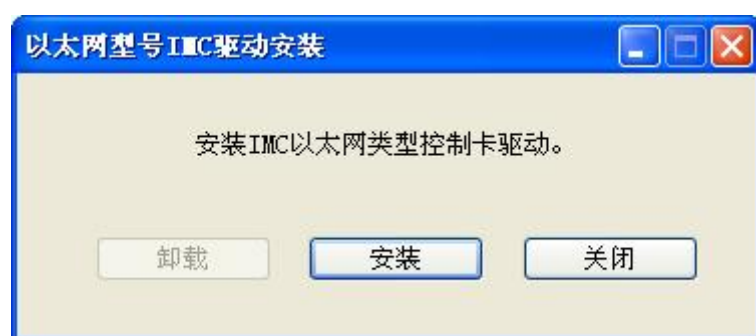


图 4-1 驱动安装程序

点击【安装】按钮，直接进行安装。几秒钟之后驱动安装完成，如“图 4-2 驱动安装完成对话框”所示：



图 4-2 驱动安装完成对话框

4.2 测试工具说明

测试工具程序启动的界面如下图所示：

设备

网卡选择: Intel(R) 82574L Gigabit Network Connection 搜索从站 IO卡名称:

硬件版本: 00000 软件版本: 0000 日期: 0000.00.00 序列号: 0000000000000000

IO测试

串口(自定义协议)测试

串口(Modbus协议)测试

设置

输入端	输入值	输入端	输入值	输出端	输出值	输出端	输出值	模拟量	模拟值	模拟值(换算)	计数器	计数值
IN0:	0	IN8:	0	OUT0:	0	OUT8:	0	AD1:	0	0.000	CNT1:	0
IN1:	0	IN9:	0	OUT1:	0	OUT9:	0	AD2:	0	0.000	CNT2:	0
IN2:	0	IN10:	0	OUT2:	0	OUT10:	0	DA1:	0	0.000	ENC:	0
IN3:	0	IN11:	0	OUT3:	0	OUT11:	0	DA2:	0	0.000		
IN4:	0	IN12:	0	OUT4:	0	OUT12:	0	PWM1freq:	0	0.000		
IN5:	0	IN13:	0	OUT5:	0	OUT13:	0	PWM2freq:	0	0.000		
IN6:	0	IN14:	0	OUT6:	0	OUT14:	0	PWM1prop:	0	0.000%		
IN7:	0	IN15:	0	OUT7:	0	OUT15:	0	PWM2prop:	0	0.000%		

所有输入输出监控

模拟量输出

模拟量输出1: 5 V 输出 测试模拟量输出 5 V 输出

PWM输出

频率1: 1000.0 Hz/s 频率2: 1000.0 Hz/s 占比1: 50 % 占比2: 50 % 设置 测试 PWM 输出 设置

开关量输出

OUT0 OUT1 OUT2 OUT3 OUT4 OUT5 OUT6 OUT7 全1

OUT8 OUT9 OUT10 OUT11 OUT12 OUT13 OUT14 OUT15 全0

测试开关量输出

清零计数器值

一键测试

DA输出: 每 2 秒增加电压 1 V

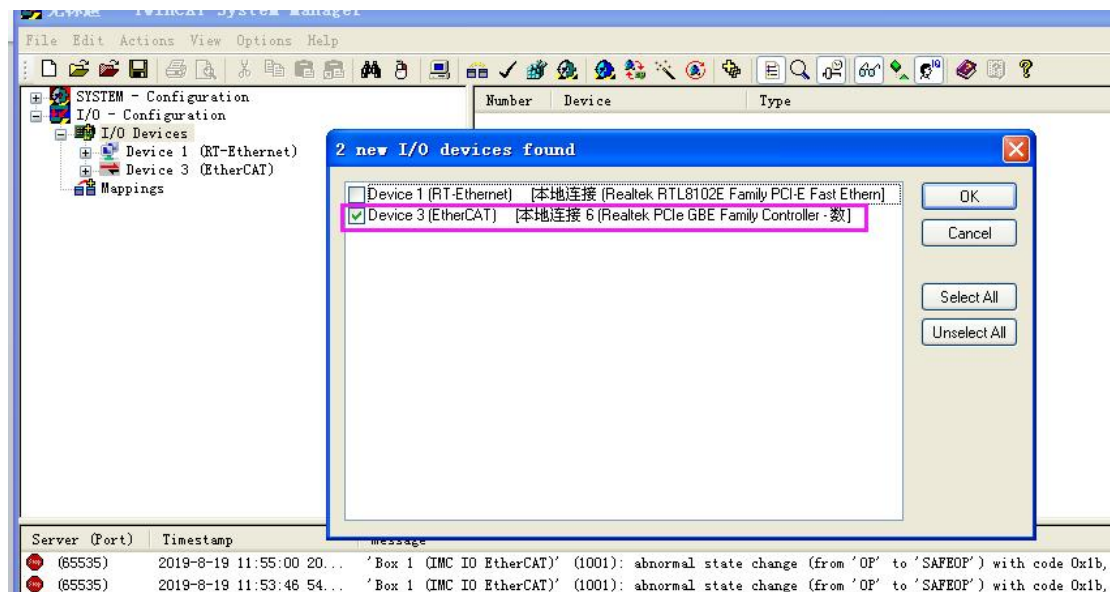
开关量输出: 每隔 0.5 秒改变一次状态

5 twincat 测试

首先，在测试之前，请预先把 xml 放入 twincat 目录的 “\Io\EtherCAT” 文件夹下；然后打开 twincat，进入 system Manager(注：此 twincat 的版本为 v2.11.2301，其它的版本测试方法类似)。

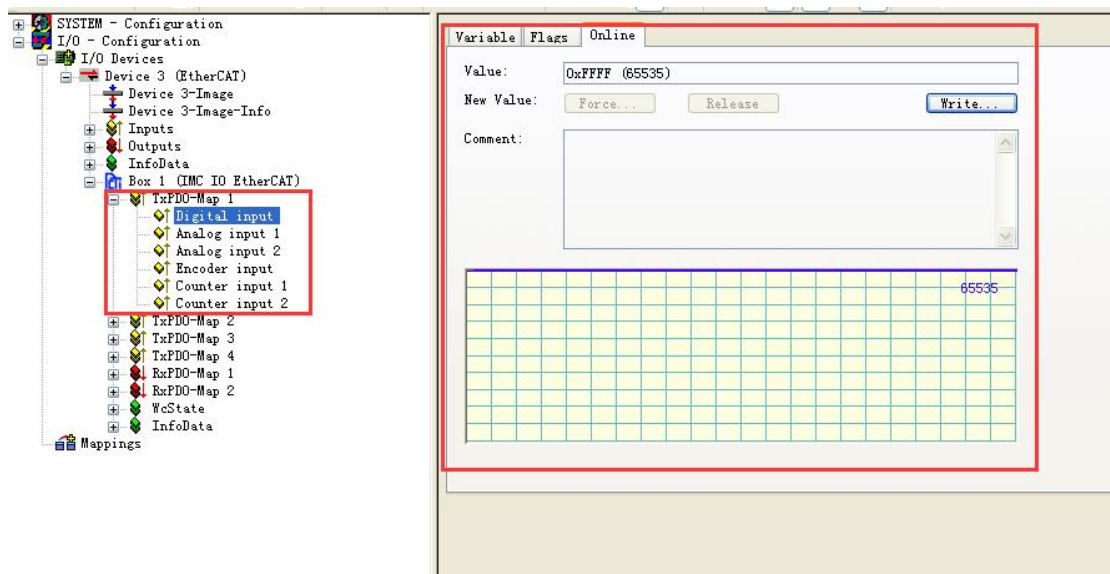


右键点击 I/O Devices，然后点击 Scan Devices，可以找到对应网络设备，选择正确的连接并点击 OK 即可运行。



5.1 输入测试

输入部分在 TxPDO-Map 1 部分，如图，其中对应的 Online 中的 Value 部分可以读到实时的值：



6.1.1 开关量输入测试

具体的输入的实时值需在 Box 1 中的 TxPDO-Map 1 才会显示出来。接入开关后，接通和断开开关均可以看到上图的 Digital input 1 的值变化。单纯的测试可以先将 24V+ 接到 COM 口，然后再用 24V- 去触碰输入口，即可看到输入的变化。（在 Digital input 1 中的 Online 可以看到输入的实时值）；

6.1.2 模拟量输入测试

在 TxPDO-Map 1 中有 Analog input 1、Analog input 2、Analog input 3 和 Analog input 4 四路模拟量输入，测试的时候四路的测试步骤都一样，故只选择一路做详细的测试。模拟量输入的范围是 0~10V，相应的，会将伏特数线性转化为 0~0x0FFFh 的值。测试时，可以将 DA0 端口和 DA0、DA1、DA2、DA3 端口直连，当改变 DA0 时，即可看到模拟量输入的变化（其值的变化可以在 Analog input 1、Analog input 2、Analog input 3 或 Analog input 4 的 Online 中看到）；

5.1.3 编码器的测试

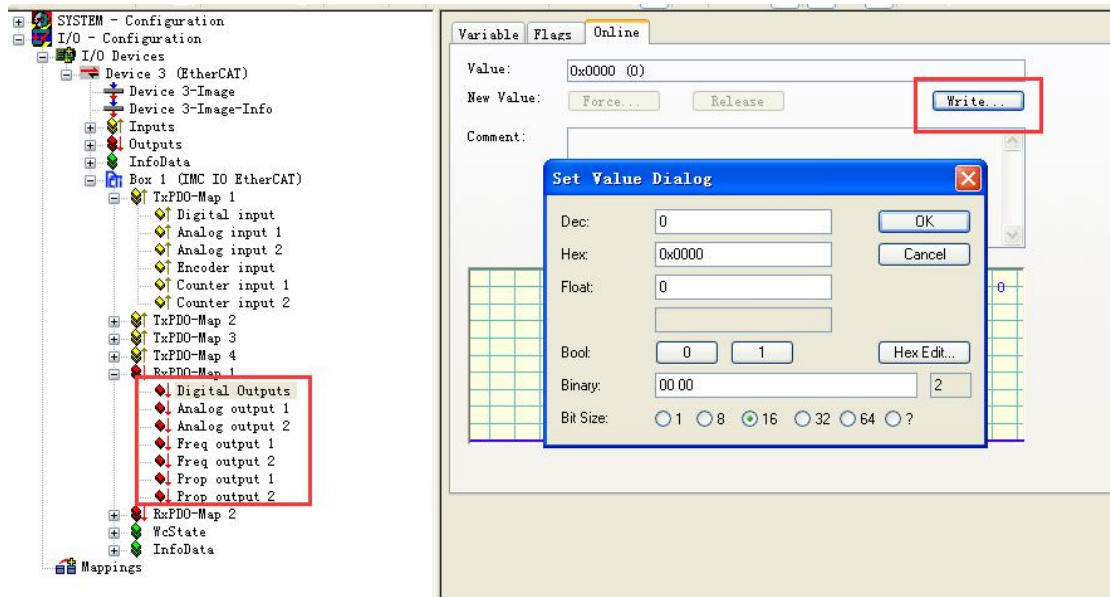
在 TxPDO-Map 1 中可以看到 Encoder input，此对象是编码器的实时值，在接入手轮等设备之后，摇动手轮即可看到相应的读数变化（在 Encoder input 中的 Online 里面）；

5.1.4 计数器的测试

在 TxPDO-Map 1 中可以看到 Counter input1，它们是计数器的实时值，在对其中一个输入口接入对应的 5V 开关之后，反复断开，接合开关，即可看到计数器的值在实时的变化；如果只是做简单的测试，用编码器的 5VOUT 接到 C+，并用 C- 触碰 GND，并松开，反复该过程即可看到读数变化（可在 Counter input 的 Online 中读到变化）。

5.2 输出测试

在 RxPDO-Map1 中可操作部分输出，其中在对应的项目(如开关量输出)中的 Online 可以写入设置的值，点击 Write 即可，如图：



5.2.1 开关量输出测试

输出的实时显示的值在 RxPDO-Map 1 中，由于输出口在 RxPDO-Map 1 中是可读可写的，所以亦可以在 RxPDO-Map 1 中对输出口进行控制。测试时可将 24V- (即电源的地) 接入 OGND，然后打开某个输出口，用小型的继电器连接 24V+ 和输出口即可。当输出改变时，可观察到 twincat 中的值变化以及继电器的变化；

5.2.2 模拟量输出测试

在 RxPDO-Map 1 中可以设置模拟量输出，范围是从 0~10V，对应的数值为 0~0xFFFFh，在 Analog output 中的 Online 的 Values 中写值即可改变，测试时可用万用表一端接 DA，另一端接 GND，可测量到电压。或者直接将 DA 接到 AD，即可从模拟量输入那里读到数值；

5.2.3 PWM 的测试

在 RxPDO-Map 1 中可以看到 Freq output 1/2 以及 Prop output 1/2，这两个分别对应设置 PWM 的频率以及 PWM 的占空比。设置好之后，PWM 输出口连接示波器，既可看到波形的变化(比如 Freq output 写入 655360，Prop output 写入 32768 可看到频率为 10，占空比为 50%的波形)。

5.3 其它测试

I/O 卡上有一个串口接口，可对应的接 modbus 协议或者非标准协议的设备(如果是非标准的，需要预先配置，默认的是标准的 modbus 协议)。而 modbus 协议

中的配置可通过地址 0x3000~0x3007 以及 0x3020~0x3027，如图：

+ 3000:0	set modbus send Reg 1	RW	> 6 <
+ 3001:0	set modbus send Reg 2	RW	> 6 <
+ 3002:0	set modbus send Reg 3	RW	> 6 <
+ 3003:0	set modbus send Reg 4	RW	> 6 <
+ 3004:0	set modbus send Reg 5	RW	> 6 <
+ 3005:0	set modbus send Reg 6	RW	> 6 <
+ 3006:0	set modbus send Reg 7	RW	> 6 <
+ 3007:0	set modbus send Reg 8	RW	> 6 <
+ 3020:0	set modbus receive Reg 1	RW	> 6 <
+ 3021:0	set modbus receive Reg 2	RW	> 6 <
+ 3022:0	set modbus receive Reg 3	RW	> 6 <
+ 3023:0	set modbus receive Reg 4	RW	> 6 <
+ 3024:0	set modbus receive Reg 5	RW	> 6 <
+ 3025:0	set modbus receive Reg 6	RW	> 6 <
+ 3026:0	set modbus receive Reg 7	RW	> 6 <
+ 3027:0	set modbus receive Reg 8	RW	> 6 <

5.3.1 485，modbus 通信

在 IO 卡上，有专门为串口通信的 485 接口，可通过该接口与其它的 485 串口设备通信。通信前，需要对该串口进行一些设置，包括从站的 ID，寄存器地址，寄存器的有效位数，并设置是否开启该地址的读写，如果不开启，则无法对此地址的寄存器进行读写(注：所有的配置均要在未开启地址读写之前写入，开启之后则无法再配置)。配置之后，即可通过写入 0x2600 来控制寄存器，或者通过读取 0x2601 来获得寄存器的状态。

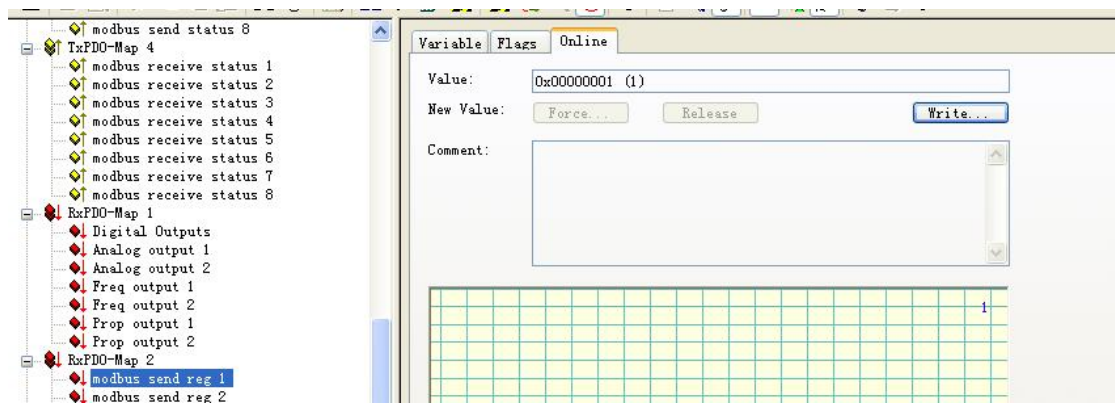
示例：往从站地址为 1，地址为 0 的线圈寄存器写入 1 的值

首先，设置读写使能关闭，即 0x3000.7 为 0；

其次，往 0x3000.3 写入该寄存器的地址 0；0x3000.4 写入该寄存器的有效位数 1；0x3000.5 写入从站的 ID，值为 1；0x3000.6 写入寄存器的类型为线圈(值为 1)；0x3000.7 写入对该寄存器的读写使能(值为 1)；如图：

+ 2402:0	set modbus attribute	RW	> 4 <
+ 2600:0	Modbus send data reg	RW	> 8 <
+ 2601:0	Modbus receive data reg	RW	> 8 <
+ 3000:0	set modbus send Reg 1	RW	> 6 <
3000:01	op status	RW P	0x0002 (2)
3000:02	reg address	RW	0x0001 (1)
3000:03	reg bit	RW	0x01 (1)
3000:04	slave id	RW	0x01 (1)
3000:05	reg type	RW	0x01 (1)
3000:06	run	RW	0x01 (1)
+ 3001:0	set modbus send Reg 2	RW	> 6 <
+ 3002:0	set modbus send Reg 3	RW	> 6 <

最后，往已做好 PDO 映射的 0x2600.1 (即 RxPDO-Map 2 中的 modbus send reg1) 中写入值 1 即可，如图：



6 结合使用 iMC6 系列测试 IO 卡

如果您拥有我们公司的 iMC6 系列的运动控制卡，可以用这张卡来设置，读取 IO 卡的参数。准备工作和连接普通的 IO 卡一样，先把 xml 放入我们 iMC6 系列的软件的 xml 文件夹下，然后用我们的 iMCAT-6xxE 软件搜索，并连接，配置即可，选择所有的 PDO 映射，如下图(注：如要进行以下实验，请勿勾选“映射 IMC 参数”，此功能为将 PDO 参数映射到我们控制卡的寄存器上，会使得 PDO 写功能无效)：由于所有的读，写功能测试方法类似，故只挑选数字量输出和数字量输入进行测试，其它测试不再赘述。

In/Out	id	sm	pdo	index	subindex	bitlen	映射IMC参数	IMC参数类型	默认值 (HEX)	描述
out 1	1	2	0x1600	0x2300	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Digital Outputs
out 2	2	2	0x1600	0x2310	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Analog output 1
out 3	3	2	0x1600	0x2311	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Analog output 2
out 4	4	2	0x1600	0x2320	0	32	<input type="checkbox"/>	由系统分配	0x00000000	Freq output 1
out 5	5	2	0x1600	0x2321	0	32	<input type="checkbox"/>	由系统分配	0x00000000	Freq output 2
out 6	6	2	0x1600	0x2330	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Prop output 1
out 7	7	2	0x1600	0x2331	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Prop output 2
out 8	8	2	0x1601	0x2600	1	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 1
out 9	9	2	0x1601	0x2600	2	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 2
out 10	10	2	0x1601	0x2600	3	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 3
out 11	11	2	0x1601	0x2600	4	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 4
out 12	12	2	0x1601	0x2600	5	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 5
out 13	13	2	0x1601	0x2600	6	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 6
out 14	14	2	0x1601	0x2600	7	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 7
out 15	15	2	0x1601	0x2600	8	32	<input type="checkbox"/>	由系统分配	0x00000000	modbus reg 8
in 1	1	3	0x1A00	0x2000	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Digital input 1
in 2	2	3	0x1A00	0x2010	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Analog input 1
in 3	3	3	0x1A00	0x2011	0	16	<input type="checkbox"/>	由系统分配	0x00000000	Analog input 2

6.1 测试数字量输出

在功能测试模块中选择“总线输入输出”部分，选择对应的从站，并设置其对应的 PDO 映射 ID 为 1，写入输出值，并设置，即可看到 IO 卡上的指示灯的变化，如图：

参数配置 零点 点动 点到点 直线插补 圆弧插补 龙门驱动 电子齿轮/手轮 开关量输入输出 间隙补偿 总线输入输出 SDO请求

从站Position: 0 : IMC IO MODEL with EtherCAT

寄存器id: 1 (只能输入从站配置过的输入输出ID)

寄存器值: 0000AAAA 十进制 十六进制 单精度

设置输出 获取输入

注意：此处的功能需要根据您配置到控制卡中的从站配置来操作。

6.2 测试数字量输入

选择对应的从站，并设置其对应的 PDO 映射 ID 为 1，当输入开关状态时，读取输入值，即可看到寄存器值的变化，如图：

参数配置

零点

点动

点到点

直线插补

圆弧插补

龙门驱动

电子齿轮/手轮

开关量输入输出

间隙补偿

总线输入输出

SDO请求

从站Position:

0 : IMC IO MODEL with EtherCAT

寄存器id:

1

(只能输入从站配置过的输入输出ID)

寄存器值:

00000001

十进制

●

十六进制

单精度

设置输出

获取输入

注意：此处的功能需要根据您配置到控制卡中的从站配置来操作。