

EtherCAT 转 Modbus 控制 KINCO 变频器使用手册

目录

EtherCAT 转 Modbus 模块介绍	3
参数配置说明	3
读功能	6
写功能	7
Kinco 变频器设置	8
通信控制参数组地址说明	9
应用举例	9
添加设备描述文件	11
设备组态	12
应用示例	13

EtherCAT 转 Modbus 模块介绍

本模块 RTU 主站最多可支持 10 个从站模块，每个模块可设置 64 路数字（线圈）输出、64 路离散输入、16 个输入寄存器和 16 个保持寄存器。

使用 Modbus RTU 功能时，每个从站会生成各自读写命名，每条命令以一定周期（可设）进行轮询，当周期过小时，主站会给出警告，但是仍然会在超时报后尽快将下一条指令发出。

使用 Modbus RTU 功能时，写功能调用不能小于轮询周期的 2 倍，不然会出现一直进行写功能，回读数据不刷新。

Modbus 通讯可选使用 RS422 或者 RS485 接口。

参数配置说明

本章节针对 RS485/RS422-EC 模块的对象字典进行详细介绍，本章只介绍和模块功能相关对象字，其他不作介绍。

索引	子索引	名称	数据类型	可设置模式	最大/小值	功能
4070	01~14	data error	uint32_t	RO(只读)	--	保存开机后出现的错误代码和出现的时间，单数索引为出现错误的时间，双数索引为错误，错误代码解析见第 8 章节
4071						该对象字保存 17 秒内耗时最大值
	01	Tx time	uint32_t	RO(只读)	--	发送耗时
	02	Wait time	uint32_t	RO(只读)	--	等待接收耗时+接收耗时
	03	Total time	uint32_t	RO(只读)	--	Tx time + Wait time + Modbus 标准间隔等待 (3.5T 间隔) + 处理时间
8000	1	Baudrate	enum	Pre-Op	min:1 max:16	用于选择串口波特率 1: 2400 Baud 2: 4800 Baud 3: 9600 4: 12.2K 5: 14.4k 6: 19.2k 7: 38.4k 8: 57.6K 9: 115.2K 10: 自定义波特率

						用于选择串口数据格式 第一位是数据长度，第二位是校验，第三位是停止位 1: 8E1 2: 8O1 3: 8N1 4: 8E1.5 5: 8O1.5 6: 8N1.5 7: 8E2 8: 8O2 9: 8N2 10: 7E1 11: 7O1 12: 7E1.5 13: 7O1.5 14: 7E2 15: 7O2 16: 9N1 17: 9N1.5 18: 9N2 当 8000: 1 设置为 0x10 时，波特率使用本子索引中的值，直接输入十进制数即可，如 115200 对应 115.2kbps 波特率
	2	dataframe	enum	Pre-Op	min:1 max:18	
	3	explicit baudrate	uint32_t	Pre-Op	--	轮询时间，在 modbus 模式下，每一条指令轮询设置的时间，关于轮询时间详细设置方式见第九章节
	4	polling time	uint16_t	Pre-Op	min:50 max:65535	将该子索引给 1，清除保存的用户参数，该位自复位
	5	Slave Reset	bool	Pre-Op	min:0 max:1	
	6	Error Reset	bool	Op	min:0 max:1	该子索引给 1，将清除 4070 中保存的错误代码，该位自复位
	7	Device Mode	enum	Pre-Op	min:0 max:1	选择模块工作模式， 0: modbus 模式 1: 透明传输模式
	8	Device Interface	enum	Pre-Op	min:0 max:1	选择模块工作接口 0: RS485 接口 1: RS422 接口

800x (x=1-A)	1	Slave addr	Byte	Pre-Op	min:0 max:255	从站接口地址, 1-255, 不可为 0, 为 0 时从站无效, 关闭从站
	2	coil is readable	bool	Pre-Op	min:0 max:1	线圈是否需要回读 0: 关闭回读 1: 开启回读功能
	3	hold reg readable	bool	Pre-Op	min:0 max:1	保持寄存器是否需要回读, 0: 关闭回读 1: 开启回读功能
	5	coil start address	uint6_t	Pre-Op	min:0 max:65535	从站 modbus 协议中线圈起始地址
	6	the number of coil	Byte	Pre-Op	min:0 max:64	从站 modbus 协议中线圈数量
	8	Discrete input start addr	uint6_t	Pre-Op	min:0 max:65535	从站 modbus 协议中离散输入起始地址
	9	the number of DI	Byte	Pre-Op	min:0 max:64	从站 modbus 协议中离散输入数量
	B	Input register start addr	uint6_t	Pre-Op	min:0 max:65535	从站 modbus 协议中输入寄存器起始地址
	C	the number of Input register	Byte	Pre-Op	min:0 max:16	从站 modbus 协议中输入寄存器数量
	E	Hold register start addr	uint6_t	Pre-Op	min:0 max:65535	从站 modbus 协议中保持寄存器起始地址
	F	the number of Hold register	Byte	Pre-Op	min:0 max:16	从站 modbus 协议中保持寄存器数量

注意：在单一对象字典无法将某一个从站数据读取完毕时，可以使用多个对象字典读取同一个从站中的值，将寄存器和线圈的起始地址进行相应偏移即可。

DataValid 功能描述:

DataValid 为自定义的功能配置标志，数据类型为 byte (8-bit), 高 4 位用来区分不同的 modbus 从站；低 4 位区分 modbus 不同功能码。

DataValid 修改说明:

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
X4	X3	X2	X1	a	b	c	d

读功能

Bits 7:4 X4X3X2X1:取对应对象字典 (0x8001~0x800a) & 0x00ff 后的值, 即 1-10。

- Bit3 a: 读输入寄存器的值
- 0: 此次显示的值是之前通过 modbus 读取的输入寄存器的值
- 1: 此次显示的值是刚通过 modbus 读取上来的输入寄存器的
- Bit2 b: 读离散输入的值
- 0: 此次显示的值是之前通过 modbus 读取的离散输入的值
- 1: 此次显示的值是刚通过 modbus 读取上来的离散输入的值
- Bit1 c: 读保持寄存器的值
- 0: 此次显示的值是之前通过 modbus 读取的保持寄存器的值
- 1: 此次显示的值是刚通过 modbus 读取上来的保持寄存器的
- Bit0 d: 读线圈的值
- 0: 此次显示的值是之前通过 modbus 读取的线圈的值
- 1: 此次显示的值是刚通过 modbus 读取上来的线圈的值

写功能

Bits 7:4 X4X3X2X1:取对应对象字典 (0x8001~0x800a) & 0x00ff 后的值, 即 1-10。

Bit3 a: 保留

Bit2 b: 保留

Bit1 c: 写保持寄存器, 0 和 1 交替写值, 开始一次写保持寄存器

Bit0 d: 写线圈, 0 和 1 交替写值, 开始一次写线圈

示例:

写一次对象字典 0x8003 所设置的从站保持寄存器, 第一次 DataValid=0x32, 第二次 DataValid=0x30, 第三次 DataValid=0x32。

写一次对象字典 0x8003 所设置的从站线圈, 第一次 DataValid=0x31, 第二次 DataValid=0x30, 第三次 DataValid=0x31。

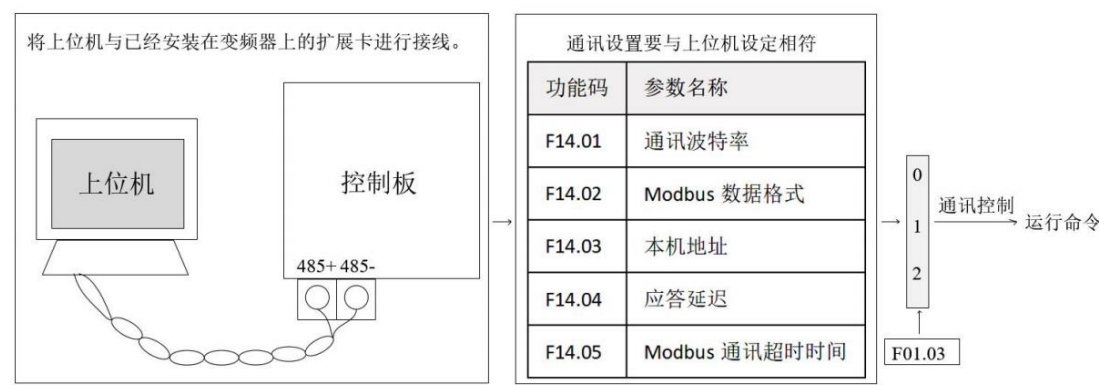
写一次对象字典 0x8003 所设置的从站线圈和保持寄存器, 第一次 DataValid=0x33, 第二次 DataValid=0x30, 第三次 DataValid=0x33 (默认从上电开始)。

Kinco 变频器设置

设置变频器操控面板上的参数 F01.03=2，选择使用通讯方式给变频器设置运行命令，可以实现对变频器的启动、停止等相关命令控制。

设定参数 F01.04 = 5，选择通讯作为主频率指令来源。

下面以 Modbus 协议为例说明用通讯给定运行命令的过程:



进行 Modbus 通信时，需要设置 F14.01(波特率)、F14.02(数据格式)、F14.03(本机地址)。

参数码 (地址)	名称	内容	出厂值 (设定范围)	可调属性
F14. 01 (0x0E01)	MODBUS 通信波特率	0: 1200 BPS 1: 2400 BPS 2: 4800 BPS 3: 9600 BPS 4: 19200 BPS 5: 38400 BPS 6: 57600 BPS 7: 115200 BPS	5 (0~7)	实时更改
F14. 02 (0x0E02)	MODBUS 数据格式	0: (N, 8, 1)无校验, 数据位: 8, 停止位: 1 1: (E, 8, 1)偶校验, 数据位: 8, 停止位: 1 2: (O, 8, 1)奇校验, 数据位: 8, 停止位: 1 3: (N, 8, 2)无校验, 数据位: 8, 停止位: 2 4: (E, 8, 2)偶校验, 数据位: 8, 停止位: 2 5: (O, 8, 2)奇校验, 数据位: 8, 停止位: 2	0 (0~5)	实时更改
F14. 03 (0x0E03)	本机地址	设定本机通讯地址	1 (0~247)	实时更改
F14. 04 (0x0E04)	应答延迟	变频器接收数据结束到向上位机发送数据的中间间隔时间。	2ms (0ms~20ms)	实时更改
F14. 05 (0x0E05)	MODBUS 通讯超时时间	当设置为 0. 0s, Modbus 通信超时时间无效。设置非 0 时则有效, 如果本次通信与下一次通信的间隔时间超出 F14. 05 (MODBUS 通信超时时间), 系统将报通信故障错误。	0. 0s (0. 0s~60. 0s)	实时更改

通信控制参数组地址说明

功能说明	地址定义	数据意义说明		R/W 特
通信给定主频率	0x7010 (F01.04=5)	0~10000 对应 0.00%~100.00%的最大频率 (F01.11)		W/R
通讯给定辅助频率	0x7010 (F01.05=5)	0~10000 对应 0.00%~100.00%的最大频率 (F01.11)		W/R
通信命令设定	0x7000	0x0000: 无命令 0x0001: 正转运行 0x0002: 反转运行 0x0003: 正转点动 0x0004: 反转点动	0x0005: 减速停机 0x0006: 紧急停机 0x0007: 自由停机 0x0008: 故障复位	W
通信写端子	0x70XX	地址低位为: 01: 写 A01 03: 写 D0 04: 写脉冲输出		W
变频器故障码	0x6F00	变频器当前故障代码 (见故障代码表)		R
通信给定上限频率	0x7010 (F01.12=5)	0~10000 对应 0.00%~100.00%的最大频率 (F01.11)		W
VF 分离的电压设定	0x7010 (F05.20=5)	0~10000 对应 0.00%~100.00%的额定电压值		W
速度控制下转矩上限源 (电动)	0x7010 (F06.11=5)	0~10000 对应 0.00%~100.00%的上限设定值 (F06.12)		W
速度控制下转矩上限源 (发电)	0x7010 (F06.13=5)	0~10000 对应 0.00%~100.00%的上限设定值 (F06.14)		W
通信给定 PID 设定值	0x7010 (F16.00=5)	0~10000 对应 0.00%~100.00%		W
通信给定 PID 反馈值	0x7010 (F16.03=4)	0~10000 对应 0.00%~100.00%		W
多段速指令 0 频率设定	0x7010 (F17.00=5)	0~10000 对应 0.00%~100.00%的最大频率 (F01.11)		W
故障状态读取	0x61XX	变频器故障状态 (最多保存上两次故障) (见故障代码表)		R
输入端子状态	0x6010/0x6013/0x6019	0x6010: DI 端子输入状态 0x6013: AI 端子输入状态 0x6019: 键盘电位计输入状态		R
输出端子状态	0x6011/0x601A	0x6011: DO 端子输出状态 0x601A: AO 端子输出状态 (0—1000 对应输出 0V~10V, 0mA~20mA)		R

应用举例

通讯方式给定频率的范围对应的频率范围为 0% ~ 100.00%(100.00%对应最大频率)。假设 F01.11“最大频率”设为 50.00Hz。如果写命令中写入的频率值 2710H，转换 10 进制为 10000。那么实际写入的频率值 50.00*100%=50.00Hz。

下面以 Modobus 协议为例说明用通讯给定主频率的过程。例如，利用通讯给定方式设置频率为 50.00Hz 时，发送写命令为 01 06 70 10 27 10 88 F3。

字节	含义
01H (可以设置)	变频器地址
06H	写命令
7010H	通讯地址
2710H (转换为十进制为 10000，指 100%的最大频率)	目标频率值
88F3H	CRC 校验

同理，利用通讯给定方式设置频率为 25.00Hz 时，发送写命令为

01 06 70 10 13 88 9F 99。其中，13 88 为 5000 转换为十六进制，指 50%最大频率。

通过操作面板进入菜单，将参数 F01.03 设为 2 选择使用通讯方式给变频器设置运行命令。



再将参数 F01.04 设为 5，选择通讯作为主频率指令来源。



然后设置 Modbus 的通讯参数将 F14.01 设为 5，Modbus 通讯的波特率设为 38400BPS。



F14.02 设为 0，通讯的数据格式为无校验，8 位数据，1 位停止位。

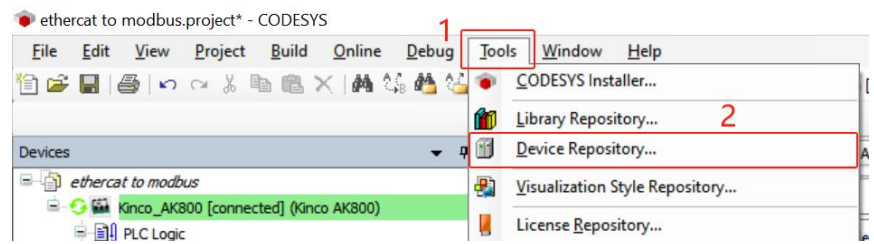


F14.03 设为 1，将变频器的通讯站号设为 1。

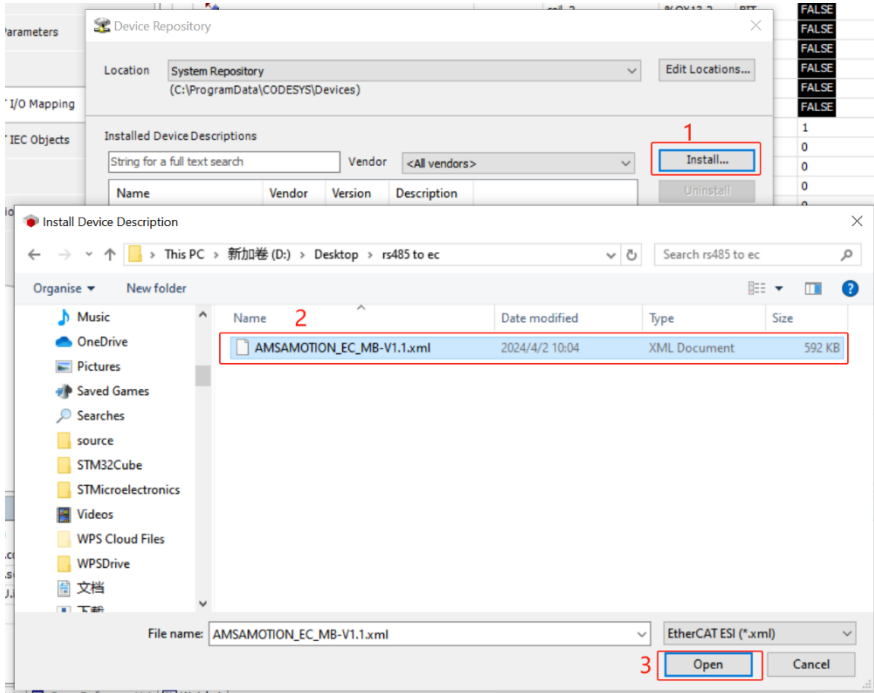


添加设备描述文件

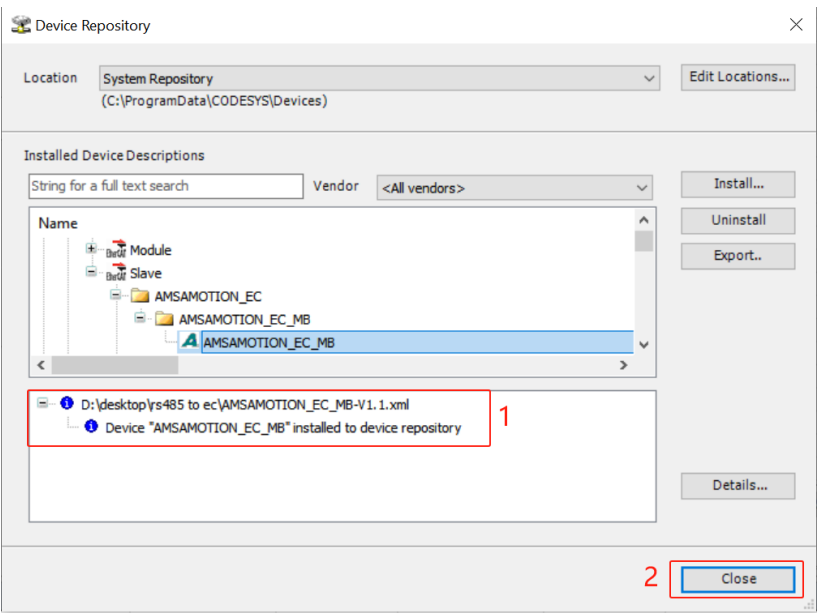
打开 Codesys 软件，在菜单栏中，点击 “Tools -> Device Repository”。



在弹出的窗口中点击 “Install”，打开设备描述文件的路径，选中并打开。

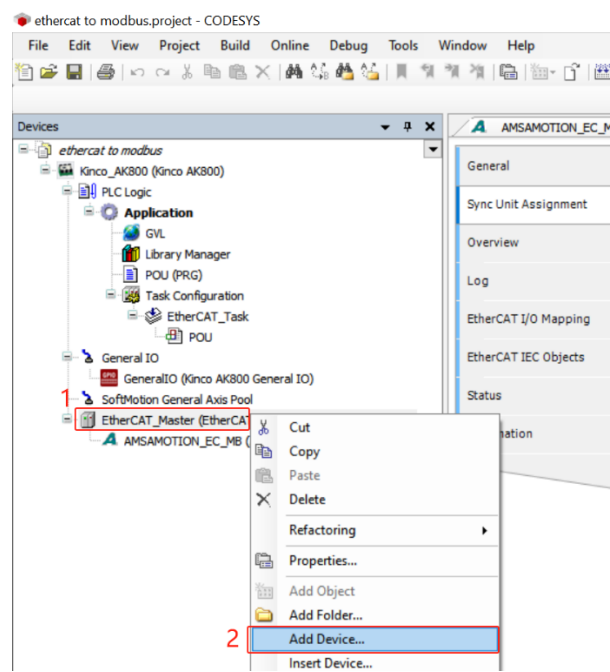


安装成功后关闭弹窗。

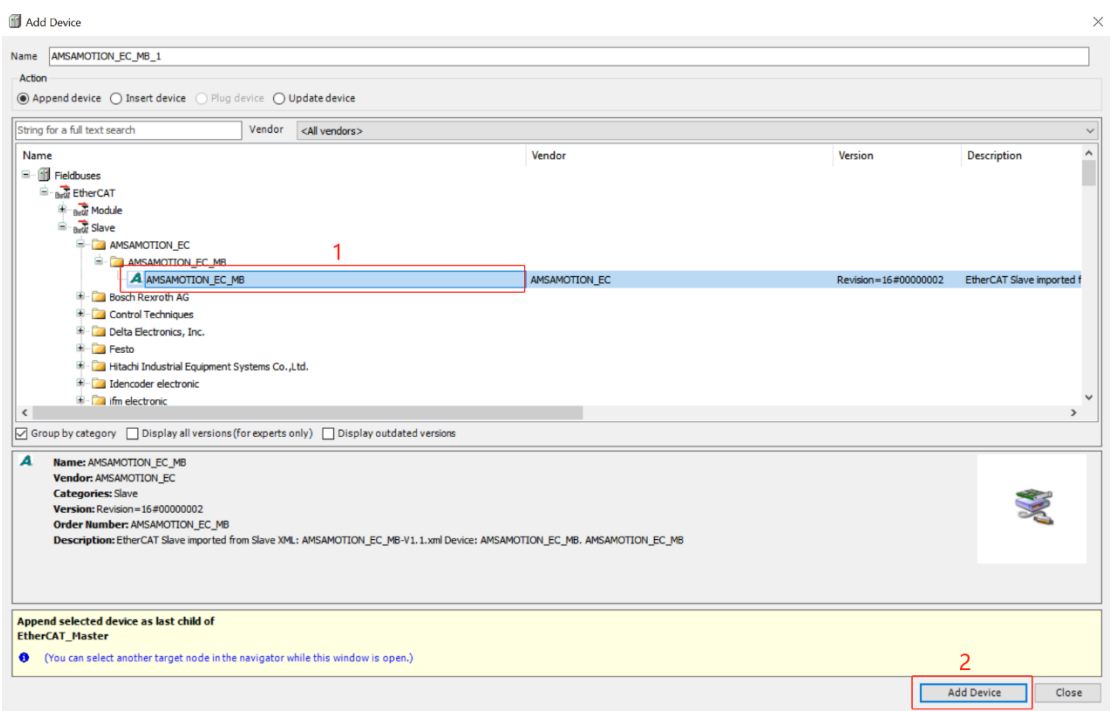


设备组态

在项目树中右击 EtherCAT 模块，单击“Add Device”添加新的模块，如下图所示。



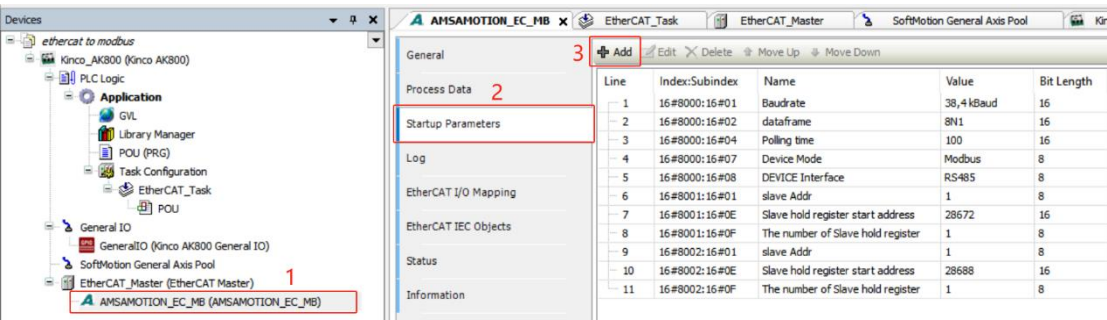
在弹出的窗口中选中 EtherCAT 转 Modbus 模块“AMSAMOTION_EC_MB”，并添加到设备中“Add Device”，如下图所示。



应用示例

接下来将以 Kinco 运动控制器 AK800 通过 EtherCAT 转 Modbus 通讯控制 Kinco 变频器 KC100 为例。

模块添加到设备后双击选中模块， 并点击“Startup Parameter”配置启动参数。



在弹出的参数配置窗口找到串口配置参数“usart configuration”， 添加到启动参数， 并进行通讯参数的配置。

16#8000:16#00	usart configuration			
:16#01	Baudrate	RW	UINT	16#0003
:16#02	dataframe	RW	UINT	16#0003
:16#03	Explicit baudrate	RW	DINT	16#000...
:16#04	Polling time	RW	UINT	16#0032
:16#05	Slave Reset	RW	BOOL	16#00
:16#06	Error Reset	RW	BOOL	16#00
:16#07	Device Mode	RW	USINT	16#01
:16#08	DEVICE Interface	RW	USINT	16#00

配置参数如下图所示， 将波特率设为 38.4kBaud， 数据格式为 8 位， 无校验， 1 位停止位， 轮询时间设为 100ms， 模块的工作模式设为 Modbus， 通讯接口设为 RS485。

1	16#8000:16#01	Baudrate	38,4 kBaud
2	16#8000:16#02	dataframe	8N1
3	16#8000:16#04	Polling time	100
4	16#8000:16#07	Device Mode	Modbus
5	16#8000:16#08	DEVICE Interface	RS485

随后找到从模块“1st slave configuration”， 并添加到启动参数。

16#8001:16#00	1st slave configuration			
:16#01	slave Addr	RW	BYTE	16#00
:16#02	coil is readable	RW	BOOL	16#00
:16#03	Keeps the register readable	RW	BOOL	16#00
:16#05	Slave coil start address	RW	UINT	16#0000
:16#06	The number of slave coil	RW	BYTE	16#00
:16#08	slave Discrete input start ...	RW	UINT	16#0000
:16#09	The number of slave Disc...	RW	BYTE	16#00
:16#0B	Slave input register start ...	RW	UINT	16#0000
:16#0C	The number of Slave inpu...	RW	BYTE	16#00
:16#0E	Slave hold register start ...	RW	UINT	16#0000
:16#0F	The number of Slave hold...	RW	BYTE	16#00

目标设备的站号设为 1，保持寄存器的起始地址设为 28672（变频器的启停控制地址），寄存器数量设为 1。


6	16#8001:16#01	slave Addr	1
7	16#8001:16#0E	Slave hold register start address	28672
8	16#8001:16#0F	The number of Slave hold register	1

继续添加从模块“2st slave configuration”到启动参数。

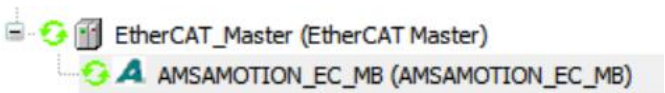
16#8002:16#00	2st slave configuration				
:16#01	slave Addr	RW	BYTE	16#00	
:16#02	coil is readable	RW	BOOL	16#00	
:16#03	Keeps the register readable	RW	BOOL	16#00	
:16#05	Slave coil start address	RW	UINT	16#0000	
:16#06	The number of slave coil	RW	BYTE	16#00	
:16#08	slave Discrete input start ...	RW	UINT	16#0000	
:16#09	The number of slave Disc...	RW	BYTE	16#00	
:16#0B	Slave input register start ...	RW	UINT	16#0000	
:16#0C	The number of Slave inpu...	RW	BYTE	16#00	
:16#0E	Slave hold register start ...	RW	UINT	16#0000	
:16#0F	The number of Slave hold...	RW	BYTE	16#00	

目标设备的站号设为 1，保持寄存器的起始地址设为 28688（变频器给定频率的地址），寄存器数量设为 1。

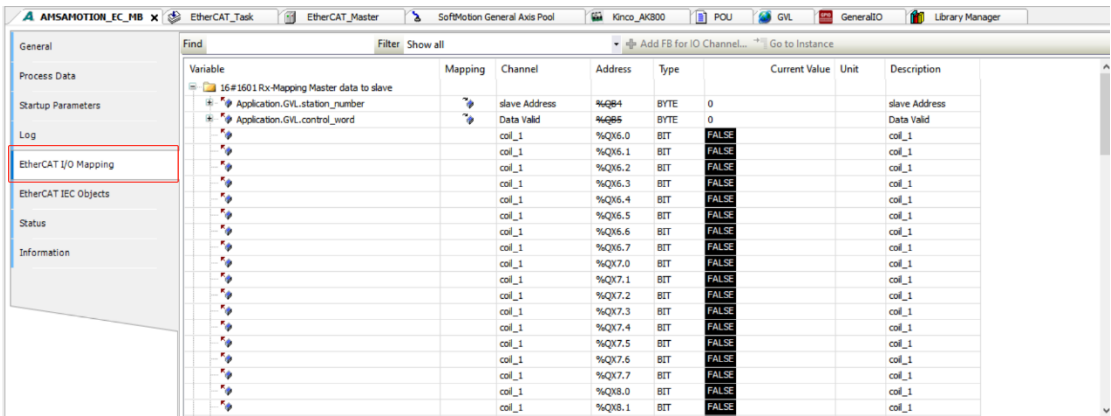
9	16#8002:16#01	slave Addr	1
10	16#8002:16#0E	Slave hold register start address	28688
11	16#8002:16#0F	The number of Slave hold register	1

在工具栏中找到扳手，单击进入在线配置模式。

若通讯无异常，设备状态如下图所示。



点击“EtherCAT I/O Mapping”查看 I/O 映射

The screenshot shows the 'EtherCAT I/O Mapping' window in a software interface. The left sidebar has a tree view with 'EtherCAT I/O Mapping' selected and highlighted with a red box. The main area displays a table of I/O mappings. The table has columns: Variable, Mapping, Channel, Address, Type, Current Value, Unit, and Description. It lists mappings for 'slave Address', 'Data Valid', and a series of coils (coil_1) mapped to various addresses and channels.

Variable	Mapping	Channel	Address	Type	Current Value	Unit	Description
16#1601 Rx-Mapping Master data to slave							
Application.GVL_station_number	slave Address		%QB4	BYTE	0		slave Address
Application.GVL_control_word	Data Valid		%QB5	BYTE	0		Data Valid
	coil_1		%QW6.0	BIT	FALSE		coil_1
	coil_1		%QW6.1	BIT	FALSE		coil_1
	coil_1		%QW6.2	BIT	FALSE		coil_1
	coil_1		%QW6.3	BIT	FALSE		coil_1
	coil_1		%QW6.4	BIT	FALSE		coil_1
	coil_1		%QW6.5	BIT	FALSE		coil_1
	coil_1		%QW6.6	BIT	FALSE		coil_1
	coil_1		%QW6.7	BIT	FALSE		coil_1
	coil_1		%QW7.0	BIT	FALSE		coil_1
	coil_1		%QW7.1	BIT	FALSE		coil_1
	coil_1		%QW7.2	BIT	FALSE		coil_1
	coil_1		%QW7.3	BIT	FALSE		coil_1
	coil_1		%QW7.4	BIT	FALSE		coil_1
	coil_1		%QW7.5	BIT	FALSE		coil_1
	coil_1		%QW7.6	BIT	FALSE		coil_1
	coil_1		%QW7.7	BIT	FALSE		coil_1
	coil_1		%QW8.0	BIT	FALSE		coil_1
	coil_1		%QW8.1	BIT	FALSE		coil_1

在通道“hold_re data_1” 写入 10000（对应 100%变频器最大设定频率 – 50Hz）。

Application.GVL.data	hold_reg_data_1	%QW7	INT	10000	hold_reg_data_1
----------------------	-----------------	------	-----	-------	-----------------

往通道“slave Address” 写 1，将目标设备站号设为 1，随后将“Data Valid”通道的 Bit5 置 TRUE，启用从模块 2 的报文参数（保持寄存器的起始地址为 28688 - 变频器给定频率的地址，寄存器数量为 1），最后翻转“Data Valid” 通道 Bit 1 的状态，发送写保持寄存器的通讯报文，通过通讯报文将变频器的给定频率设为 50Hz。

16#1601 Rx-Mapping Master data to slave					
Application.GVL.station_number	slave Address	%QB4	BYTE	1	slave Address
Application.GVL.control_word	Data Valid	%QB5	BYTE	34	Data Valid
	Bit0	%QX5:0	BOOL	FALSE	Data Valid
	Bit1	%QX5:1	BOOL	TRUE	Data Valid
	Bit2	%QX5:2	BOOL	FALSE	Data Valid
	Bit3	%QX5:3	BOOL	FALSE	Data Valid
	Bit4	%QX5:4	BOOL	FALSE	Data Valid
	Bit5	%QX5:5	BOOL	TRUE	Data Valid
	Bit6	%QX5:6	BOOL	FALSE	Data Valid
	Bit7	%QX5:7	BOOL	FALSE	Data Valid

在通道“hold_re data_1” 写入 1（对应变频器启停控制的正转控制字）。

Application.GVL.data	hold_reg_data_1	%QW7	INT	1	hold_reg_data_1
----------------------	-----------------	------	-----	---	-----------------

将“Data Valid”通道的 Bit4 置 TRUE，启用从模块 1 的报文参数（保持寄存器的起始地址为 28672 – 变频器的启停控制地址，寄存器数量为 1），然后翻转“Data Valid” 通道 Bit1 的状态，发送写保持寄存器的通讯报文，通过通讯报文控制变频器，让其输出正转频率。

Variable	Mapping	Channel	Address	Type	Current Value	Unit	Description
Application.GVL.station_number		slave Address	%QB4	BYTE	1		slave Address
Application.GVL.control_word		Data Valid	%QB5	BYTE	16		Data Valid
		Bit0	%QX5:0	BOOL	FALSE		Data Valid
		Bit1	%QX5:1	BOOL	FALSE		Data Valid
		Bit2	%QX5:2	BOOL	FALSE		Data Valid
		Bit3	%QX5:3	BOOL	FALSE		Data Valid
		Bit4	%QX5:4	BOOL	TRUE		Data Valid
		Bit5	%QX5:5	BOOL	FALSE		Data Valid
		Bit6	%QX5:6	BOOL	FALSE		Data Valid
		Bit7	%QX5:7	BOOL	FALSE		Data Valid

变频器的控制字和频率的换算关系请查阅 [通信控制参数组地址说明](#) 和 [应用举例](#)。

更多变频器详细内容请查阅“KC100 系列精易型高性能矢量变频器用户手册”。

更多 EtherCAT 转 Modbus 模块详细内容请查阅手册“RS485RS422-ECV1.2”。