

Servo Motion

Application Note

Modbus TCP comms between AC500eco and E530PTI

AN504 Rev A (Chinese)



简介

本文档以 ABB AC500 PLC 和 E530 PTI 伺服驱动器之间通过 Modbus TCP 通信协议进行通信为例，详细介绍了基于 Automation Builder 2.7 进行硬件配置，调用功能块编写通信程序的过程。如果有更多技术细节需要咨询和支持，请联系当地的 ABB 技术服务团队。

兼容性

本例基于

- AC500 PLC 固件版本 3.7.0.0 测试
- E530PTI 伺服驱动系统固件版本 1.3.0.7 测试
- Automation builder V2.7.0.24 测试
- Servo composer V1.04.0 测试。

运行本例所需的硬件为 ABB AC500 V3 系列 PLC。AC500 和 AC500eco 两种主要型号。因为所有的 CPU 都可以支持 Modbus TCP，这个范围内的任何 CPU 都可以。所有 AC500 V3 CPU 在代码的 PM 部分之后都有 4 个数字的类型名称，例如 PM5072-T-2ETH，不像上一代 CPU 只有 3 个数字，例如 PM564-ETH。

质保责任

用户应对使用本文件中描述的产品负全部责任。ABB 不承担任何担保。无论法律依据如何，ABB 不承担与产品或提供的示例或本产品中包含的文件的应用有关的责任。责任免责不适用于故意或重大过失的情况。本声明应受瑞士法律管辖并按照瑞士法律解释，但不包括其法律冲突规则和《维也纳国际公约》。

目录

准备	3
硬件准备.....	3
软件准备.....	3
知识准备.....	3
Modbus TCP 原理.....	3
编程和验证	4
硬件配置.....	5
程序编写.....	6
伺服驱动器侧配置.....	9
读写测试.....	10
调试.....	13

准备

硬件准备:

AC500 PM5052-T-ETH, 一台
 网络交换机, 一台
 E530PTI 伺服驱动器, 一台
 调试电脑, 一台
 网线若干
 220VAC 电源 24VDC 电源。

软件准备:

Automation Builder 2.7 premium.
 Servo Composer1.4.0。

知识储备:

合格的电气电工知识
 了解 Modbus TCP 的一般原理。
 默认对 ABB 的 PLC 硬件和软件有基本了解, 对 E530PTI 伺服驱动系统有基本了解。

Modbus TCP 原理

Modbus TCP 通信协议是一项应用层报文传输协议, 物理层接口采用以太网接口, 设备交互方式为 Master/Slave 方式。

Modbus 的操作对象以下四种:

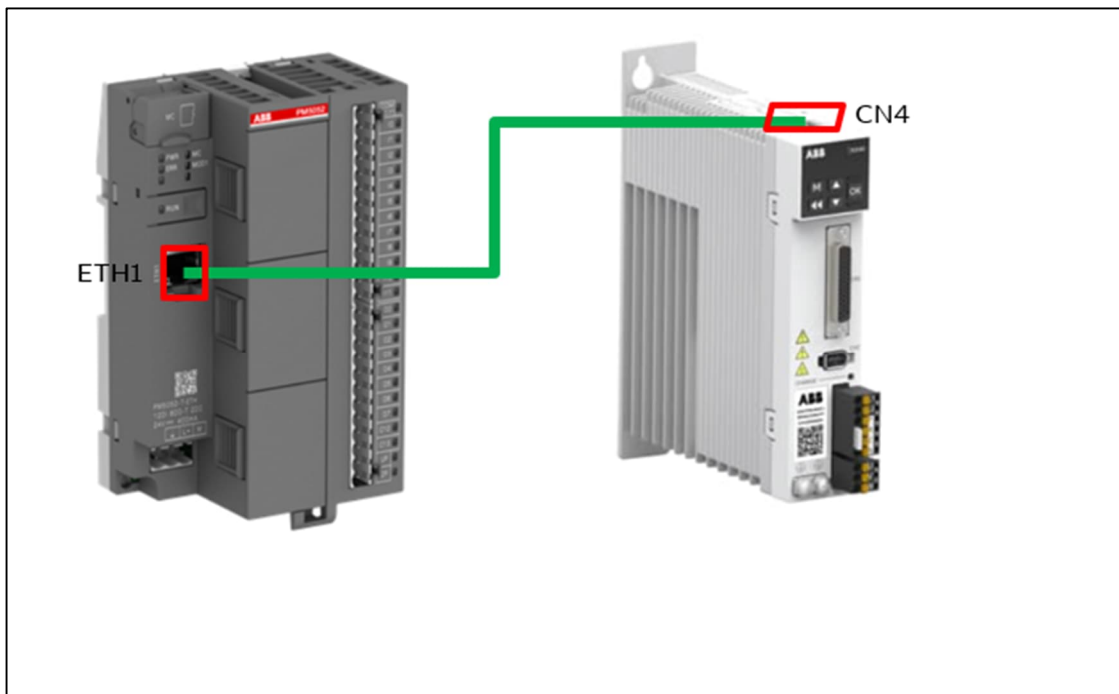
操作对象	描述
线圈	PLC 的输出位, 是开关量, 支持读/写
离散量	PLC 的输入位, 是开关量, 支持只读
输入寄存器	PLC 中只能在模拟量输入端操作的寄存器, 支持只读
保持寄存器	PLC 中用于输出模拟量的寄存器, 支持读/写

ABB E530PTI 伺服驱动器支持的 Modbus 操作功能码如下表:

功能码	名称	操作数据类型	作用描述	寄存器地址
03H	读保持寄存器	字(byte)	在一个或多个保持寄存器中取得当前的二进制值	40001-49999
04H	读输入寄存器	字(byte)	在一个或多个输入寄存器中取得当前的二进制值	30001-39999
10H	写多个保持寄存器	字(byte)	写多个保持寄存器, 将多个字节写入到寄存器	40001-49999
17H	读/写多个保持寄存器	字 (byte)	读和写多个保持寄存器的值	40001-49999

ABB AC500 eCo PLC 和 E530PT1 伺服驱动器的连接方式如下图:

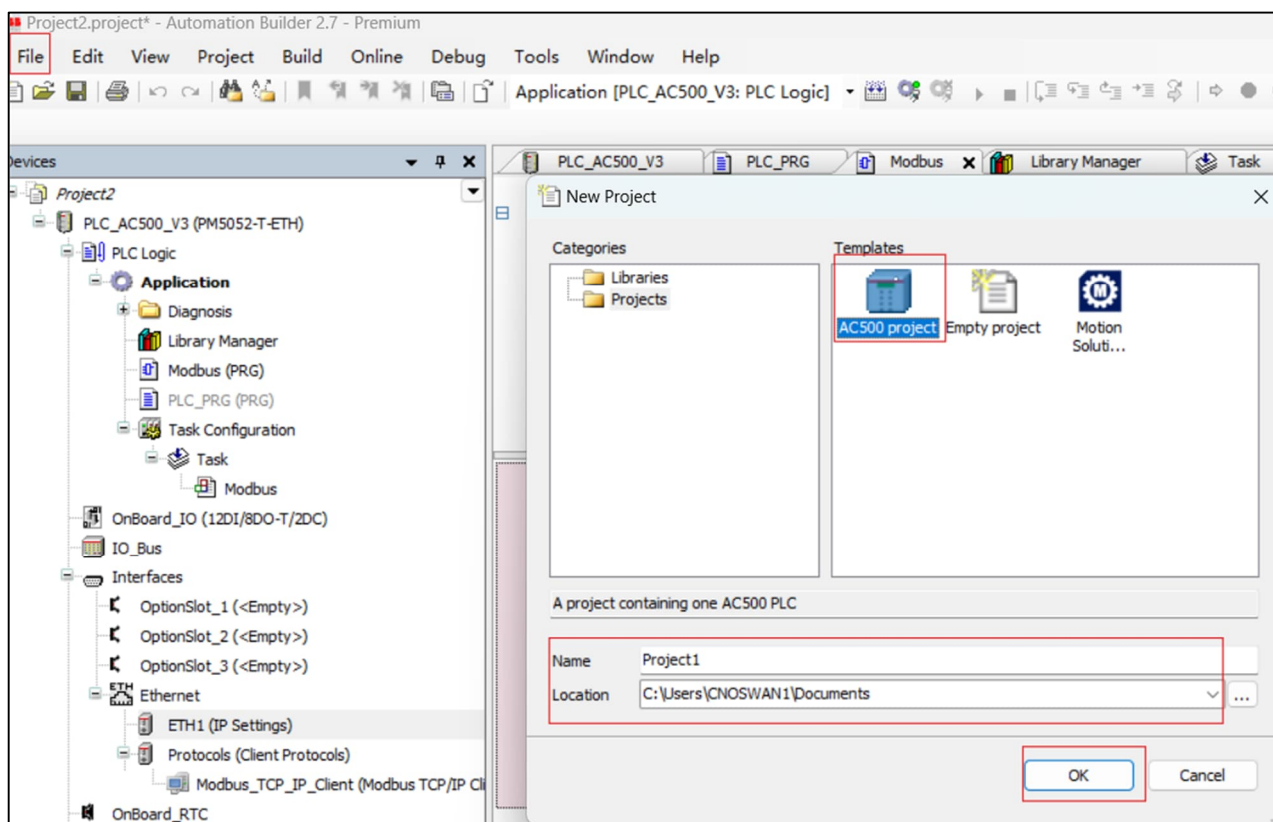
通信线缆应该选择 RJ45 接头的标准 CAT5e 屏蔽双绞线, 本例 PLC 侧连接到 ETH1 接口, 伺服器驱动器侧连接到驱动器上方的 CN4 接口。



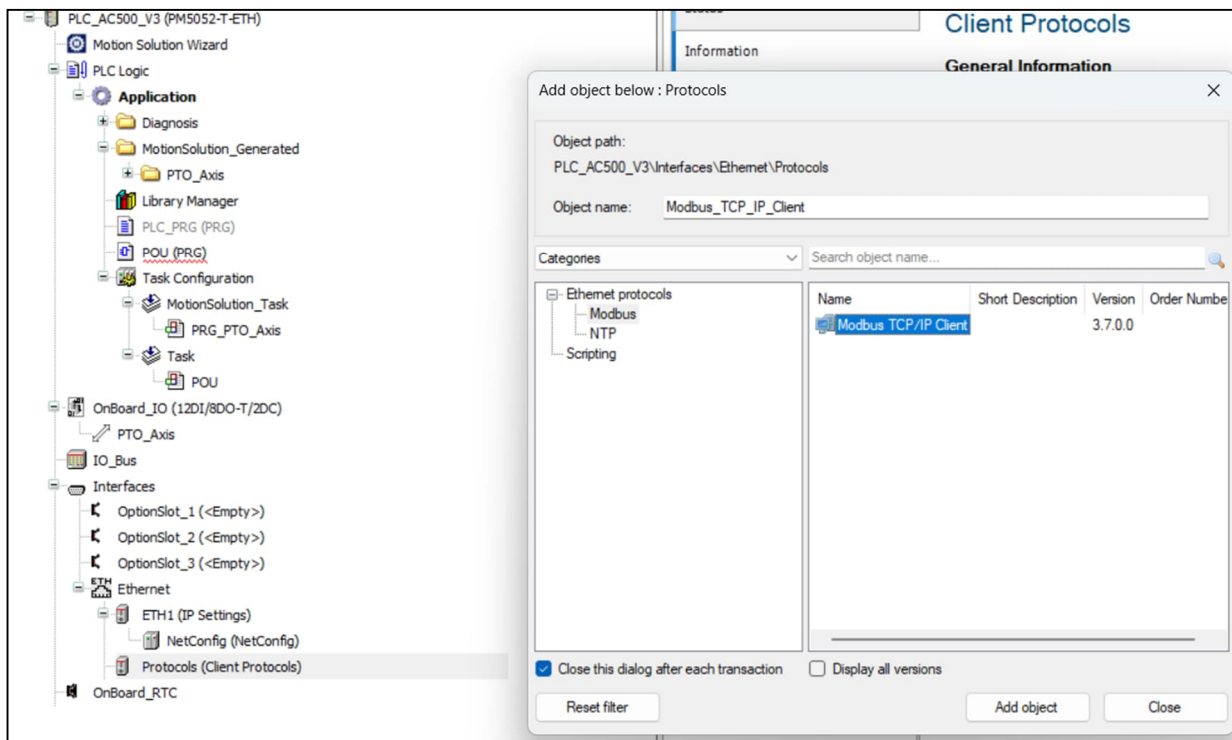
编程和验证

硬件配置:

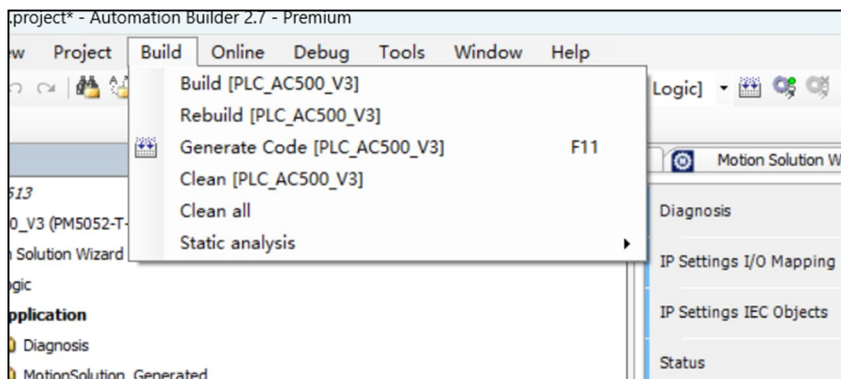
1, 打开 Automation Builder 2.7, 新建标准工程, 选择“AC500 project”模板, 并命名, 选择存放的工程路径, 路径名不要包含中文字符, 防止工程运行出现不必要的问题。



2, 点击设备树, 在“interfaces-ethernet-Protocols (Client Protocols)”下添加“Modbus TCP/IP Client”。

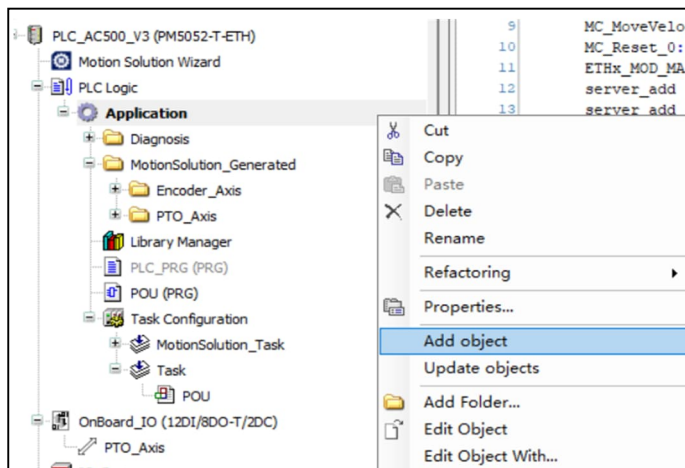


3, 菜单栏, 选择“build-Rebuild”。编译无误后, 准备进行程序编写。

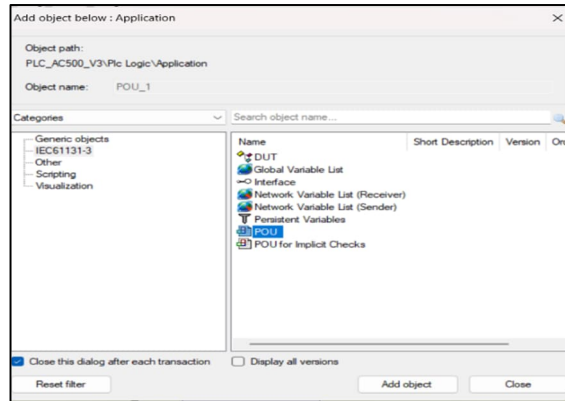


程序编写

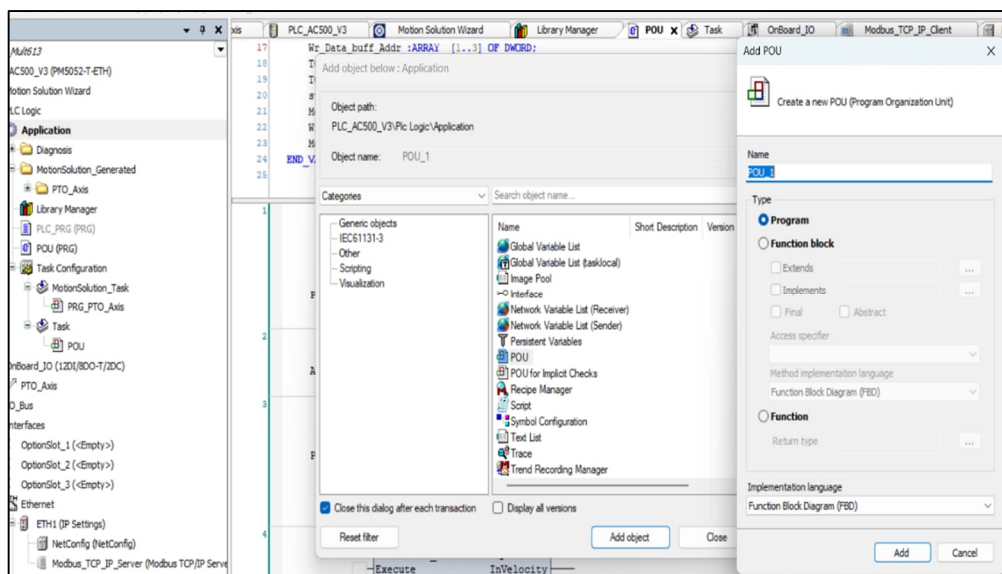
1, 在设备树窗口, 点击“PLC Logic->application”, 右击选择“Add object。”



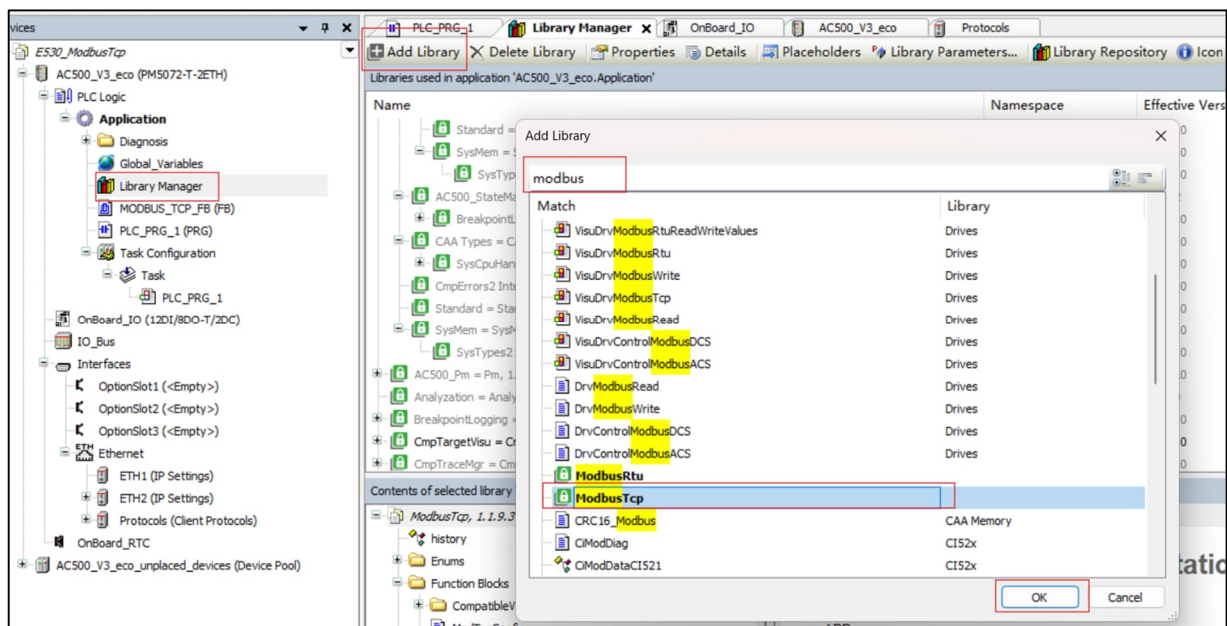
2, 选择 POU 程序组织单元。



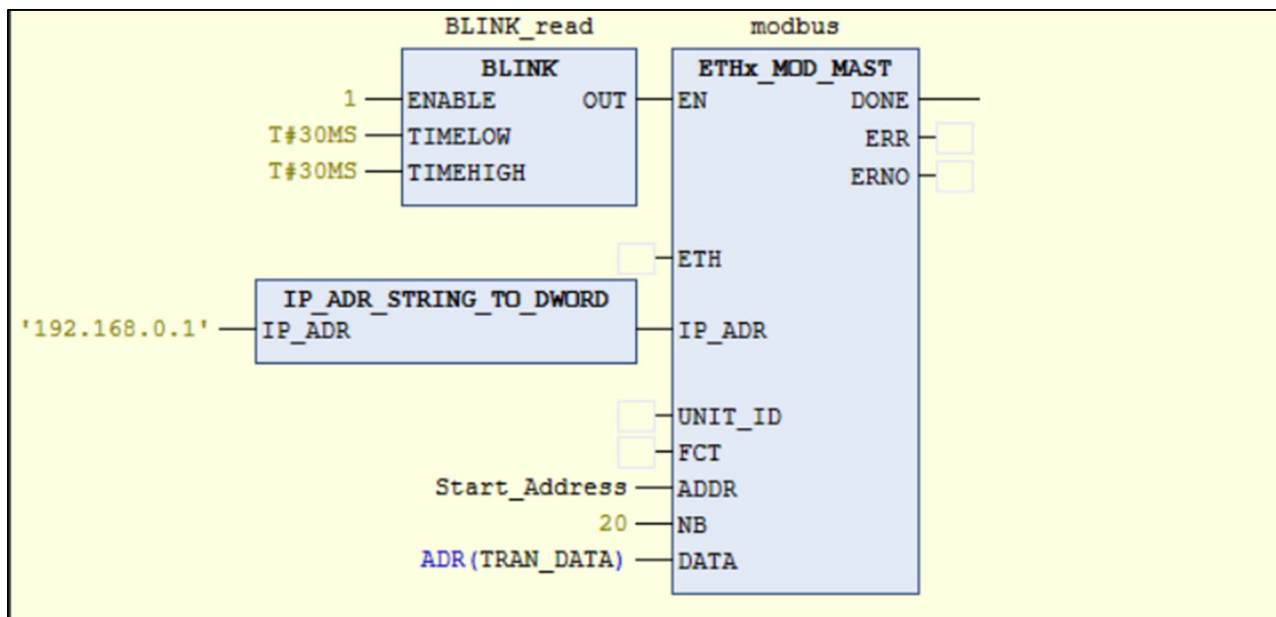
3, 选择 program, 根据个人偏好, 选择编程语言, 本例选择 FBD。



4, 新建工程需要在“library manager”管理器中调用“Modbus TCP 1.1.9.3(ABB)”库, 加载完成后即可进行下一步。



5, 点击打开新建的 POU 块, 插入网络, 调用 “ETHx_MOD_MAST” 功能块, 该功能块可以实现 Modbus-TCP 的通信任务。



引脚名称	数据类型	描述
EN	Bool	该输入端口收到上升沿信号后触发, 执行该功能块, 通信完成后, DONE 置位, 否则功能块报错, 按照错误代码指引排查配置。
ETH	Byte	指定通信的以太网接口, 默认值为 1, 一般选择默认值
IP_ADDR	Dword	指定通信的从站的 IP 地址 (伺服驱动器的 IP 地址, 在下文中提及), 数据类型 DWORD 双字节类型
UNIT_ID	Byte	指定从站的节点号, 可以忽略
FCT Modbus TCP	Byte	通信的功能码, 常用的 Modbus TCP 通信的功能码见上文。
ADDR	Word	指定读/写操作的寄存器字节起始地址
NB	Word	从起始地址开始, 要读/写操作的寄存器地址字节个数
DATA	Dword	操作数缓存区

接下来将会用到的 Modbus TCP 通信功能码如下:

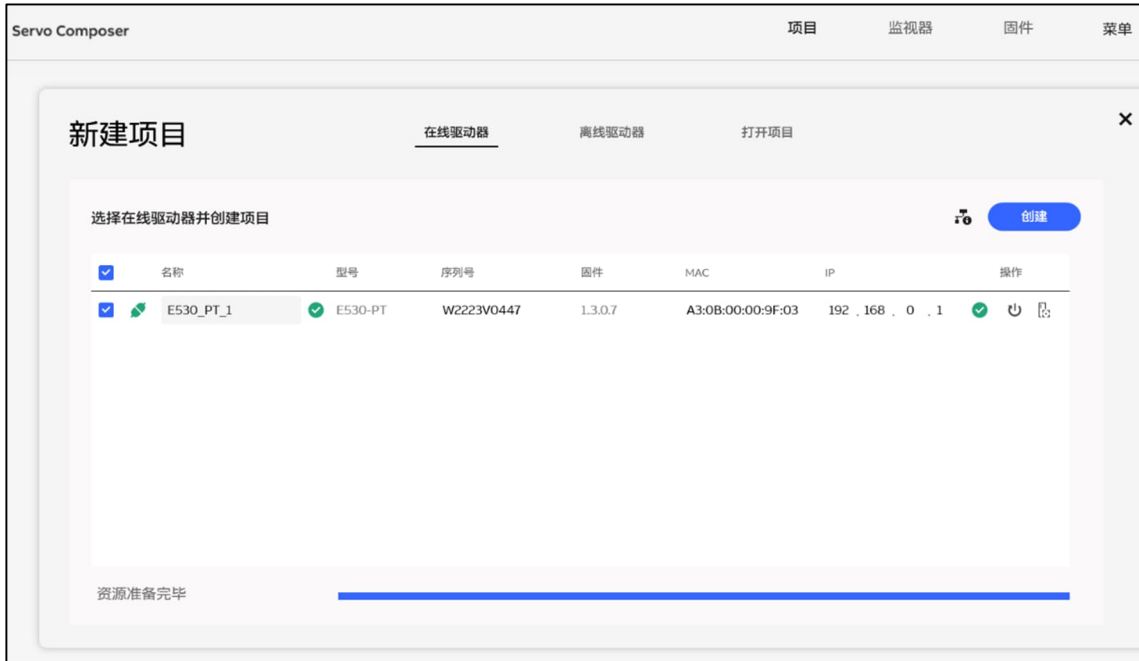
功能码	含义	数值类型	访问类型
03 (0x03)	读取多个保持寄存器	16 位	可读
06 (0x06)	写单个寄存器	16 位	可写
16 (0x10)	写多个寄存器	16 位	可写

至此，PLC 侧，简单的 Modbus TCP 通信程序建立完成，编译下载到 PLC，准备进行数据读写操作。

接下来通过 Servo Composer 伺服调试软件工具在 E530PTI 伺服驱动器侧进行配置。

伺服驱动器侧配置

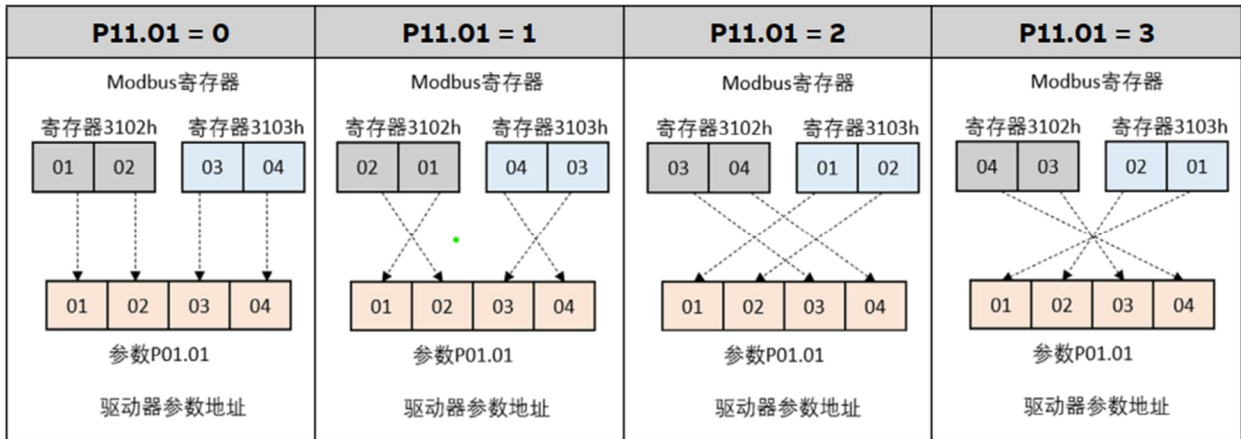
1, 打开 ServoComposer, 在线搜索到正在进行通信的驱动器, 勾选后, 并点击“创建”。



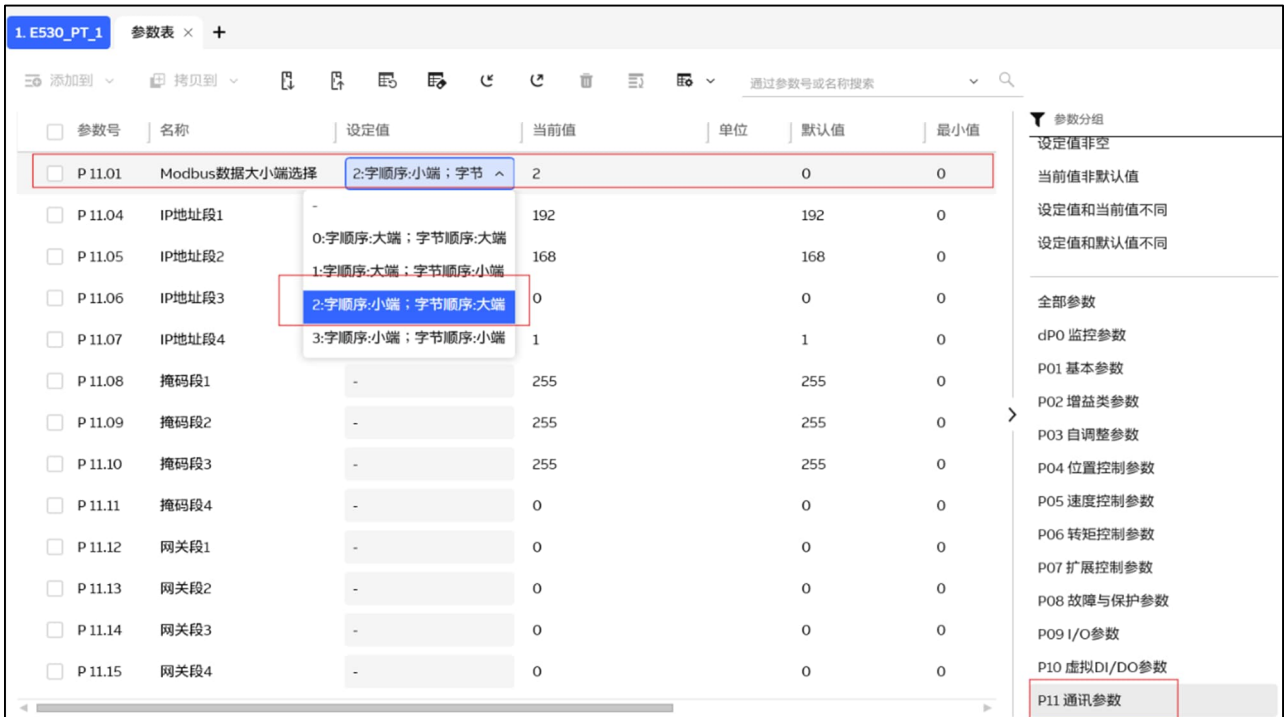
2, 点击红框里面的驱动器, 打开参数表。



3, 选择 P11 通信参数, AC500 PLC 默认的是字节大端顺序, 因此对 P11.01--Modbus 数据大小端选择参数进行设置, 设为 2: 字顺序小端, 字节顺序大端。下图展示了不同大小端配置模式下, Modbus 寄存器和驱动器参数的对应关系:



选择 P11.04-P11.07,对 IP 地址进行设置,保证伺服驱动器的地址和 PLC 的地址在同一网段, 且不冲突。保存, 断电重启, 设置生效。



1. E530_EC_1 参数表 × +

添加到 拷贝到 通过参数号或名称搜索

参数号	名称	设定值	当前值	单位	默认值	最小值
<input type="checkbox"/> P 11.04	IP地址段1	<input type="text" value="192"/>	192		192	0
<input type="checkbox"/> P 11.05	IP地址段2	-	168		168	0
<input type="checkbox"/> P 11.06	IP地址段3	-	0		0	0
<input type="checkbox"/> P 11.07	IP地址段4	-	1		1	0
<input type="checkbox"/> P 11.08	掩码段1	-	255		255	0
<input type="checkbox"/> P 11.09	掩码段2	-	255		255	0
<input type="checkbox"/> P 11.10	掩码段3	-	255		255	0

读写测试

1, 以 3026h 为读取的起始地址为例, 其十进制数值地址为 12326, 将其写入功能块的输入引脚“ADDR”, 由于 E530 PTI 伺服器驱动器侧的参数以双字节形式占用寄存器地址, 因此在 PLC 侧的功能块“ETH-Modbus TCP Master”中, 读取或者写入的地址的字节数量 NB 应该是读取寄存器的 2 倍, 因此读取十个参数, 功能块需要读取的字节数量 NB 为 20 且触发 PLC 程序里面的功能块, 通过对比 PLC 在线监控的数据, 数据一致, 至此, 说明通信成功。

PLC_AC500_V3.Application.Modbus

Expression	Type	Value	Prepared value
buf_data	ARRAY [0..10] OF D...		
buf_data[0]	DINT	10751	
buf_data[1]	DINT	2952	
buf_data[2]	DINT	0	
buf_data[3]	DINT	276870	
buf_data[4]	DINT	0	
buf_data[5]	DINT	0	
buf_data[6]	DINT	0	
buf_data[7]	DINT	410	
buf_data[8]	DINT	0	
buf_data[9]	DINT	0	
buf_data[10]	DINT	0	

值	当前值	单位	默认值	最小值	最大值	MODBUS地址
	10751	pulse	0	0	4294967295	3026h
	295.2	degree	0.0	-2147483648.0	2147483647.0	3028h
	0	rev	-32768	-32768	32767	302Ah
	276866	pulse	0	0	4294967295	302Ch
	0.0	%	0.0	-2147483648.0	2147483647.0	302Eh
	0.0	%	0.0	-2147483648.0	2147483647.0	3030h
	0.0	%	0.0	-2147483648.0	2147483647.0	3032h
	41.0	degree	0.0	-2147483648.0	2147483647.0	3034h
	0.0	degree	0.0	-2147483648.0	2147483647.0	3036h
	0.0	%	0.0	-2147483648.0	2147483647.0	3038h
	2627883.9	s	0.0	-2147483648.0	2147483647.0	303Ah

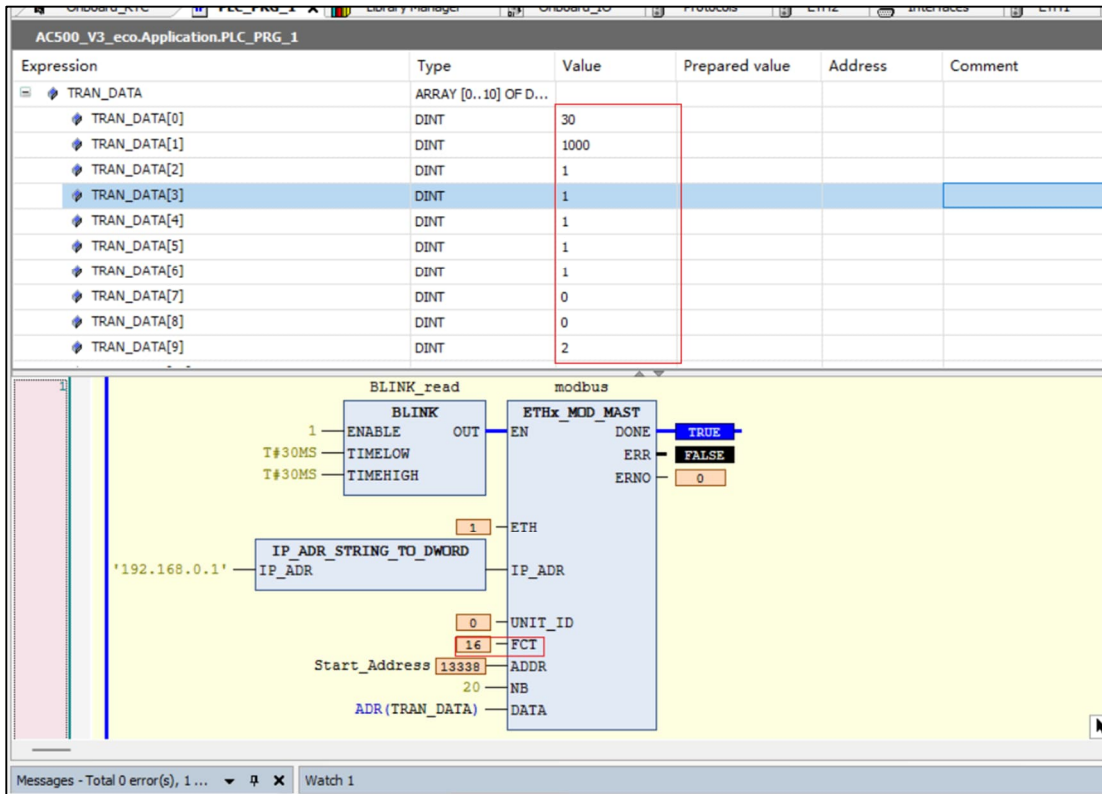
2, 需要注意的是的, Servocomposer 在线监控的数据, 例如 3028h, 当前值为 295.2, 而 PLC 侧读取的数据为 2952, 这是因为 Servo Composer 侧读到的当前值被按照比例 0.1 进行了缩放处理, 方便 Modbus TCP 通信时时读取和存放。

缩放比例默认被隐藏, 可以在 ServoComposer 软件的菜单栏进行勾选, 勾选过后, 即可展示对应参数的缩放比例。因此, 在 PLC 侧, 读取的对应寄存器的值, 要依照缩放量纲进行换算。例如, 3028h 的实际值应该是: $2952 \times 0.1 = 295.2$ 。3026h 的实际值应该是 $10751 \times 1 = 10751$ 。

当前值	比例	单位	默认值	最小值	最大值
0.55	0.01	volt		-2147483648.00	2147483647.00
0.00	0.01	ampere		-2147483648.00	2147483647.00
10751	1	pulse		0	4294967295
295.2	0.1	degree		-2147483648.0	2147483647.0
0	1	rev		-32768	32767
276872	1	pulse		0	4294967295
0.0	0.1	%	0.0	-2147483648.0	2147483647.0
0.0	0.1	%	0.0	-2147483648.0	2147483647.0
0.0	0.1	%	0.0	-2147483648.0	2147483647.0
41.2	0.1	degree	0.0	-2147483648.0	2147483647.0
-	0.1	degree	0.0	-2147483648.0	2147483647.0
-	0.1	%	0.0	-2147483648.0	2147483647.0

3, 通过以上两张图图片里面的 “buf_data” 数组的实时值和 Servo Composer 在线工具监控的伺服驱动器对应的 Modbus 地址 “3025h” 起始的地址的实时值一致。说明此次 ModbusTCP 通信成功。

4, 与此同时, 写操作, 只要将功能块的功能码 “FCT”, 修改成 “10h”, 对应的十进制数值为 16, 以起始写入地址 341Ah 为例, 其对应的十进制数值为 13338, 写入对应的 “FCT” 引脚。将写入的数组准备值写入 TRAN_DATA, 再次触发功能块后, 即可将准备的数值写入伺服驱动器相应的寄存器地址。



当前值	比例	单位	默认值	最小值	最大值	MODBUS地址
3.0	0.1	ms	0.0	0.0	64.0	341Ah
1000	1	pulse	0	0	1073741824	341Ch
1	1		1	1	1073741824	341Eh
1	1		1	1	1073741824	3420h
1	1		1	1	1073741824	3422h
1	1		1	1	1073741824	3424h
1	1		1	1	1073741824	3426h
0	1		0	0	2	3428h
0	1		0	0	1	342Ah
2	1		2	0	2	342Ch
0	1		0	0	3	342Eh
0	1		0	0	1	3430h
1000	1	pulse or user unit 1000		1	1073741824	3432h

5, 同时需要注意的是, 写入的数据, 要注意对应数值的取值范围和缩放比例, 否则, 在写入过程中, 功能块将会报错, 由此, 在 PLC 侧的 Modbus TCP 通信的写操作完成。

调试

在调试过程中, 您可能会发现有一些由 ETHx_MOD_TCP 函数块给出的错误。如果看到任何可以快速参考的错误代码, 下面是可能生成的错误代码列表(这也可以通过 AC500_ModbusTCP 库条目中的集成帮助找到)。



错误条目	值 (16进制)	值 (十进制)	注释
NO_ERROR	16#0	0	执行成功完成
ERR_NOT_CONFIGURED	16#1	1	Modbus TCP 未配置
ERR_INVALID_INTERFACE	16#2	2	无效的以太网接口或以太网接口上没有配置服务器
ERR_ENDIANESS_NOT_SUPPORTED	16#3	3	字节序不支持
ERR_FCT_NOT_SUPPORTED	16#4	4	功能码不支持
ERR_ILLEGAL_REGISTER_ADDRESS	16#5	5	非法寄存器地址
ERR_DATA_SIZE	16#6	6	读写数据量超出系统容量
ERR_INVALID_ACCESS	16#7	7	要读/写的数据的本地存储地址无效 当使用功能码 FCT22/FCT23 时, 请使用结构体: ETH_MOD_FC22_TYPE or ETH_MOD_FC23_TYPE.
ERR_INVALID_TIMEOUT_VALUE	16#8	8	响应超时的值无效
ERR_ALREADY_IN_STATE	16#10	16	服务器已处于请求状态
ERR_FAILED_SET_STATE	16#11	17	未能将服务器设置为请求状态
ERR_SIMULTANEOUS_ACCESS	16#12	18	服务器已被其他 FB 实例访问, 请稍后再试
ERR_FAILED_CONNECT	16#100	256	连接服务器失败
ERR_EXCEPT_01_ILLEGAL_FUNCTION	16#101	257	非法功能异常 服务器的异常响应包含错误代码 01 / 16#01
ERR_EXCEPT_02_ILLEGAL_DATA_ADDRESS	16#102	258	非法数据地址 服务器的异常响应包含错误代码 02 / 16#02
ERR_EXCEPT_03_ILLEGAL_DATA_VALUE	16#103	259	非法数据值 服务器的异常响应包含错误代码 03 / 16#03
ERR_EXCEPT_04_SERVER_DEVICE_FAILURE	16#104	260	服务器设备故障 服务器的异常响应包含错误代码 04 / 16#04
ERR_EXCEPT_05_SERVER_ACKNOWLEDGE	16#105	261	服务器确认 服务器的异常响应包含错误代码 05 / 16#05
ERR_EXCEPT_06_SERVER_DEVICE_BUSY	16#106	262	服务器设备忙 服务器的异常响应包含错误代码 06 / 16#06
ERR_EXCEPT_07_SERVER_NO_ACKNOWLEDGE	16#107	263	服务器没有确认 服务器的异常响应包含错误代码 07 / 16#07
ERR_EXCEPT_08_MEMORY_PARITY_ERROR	16#108	264	内存奇偶校验错误 服务器的异常响应包含错误代码 08 / 16#08
ERR_EXCEPT_09_SERVER_PASSIVE	16#109	265	可能是被动服务器 服务器的异常响应包含错误代码 09 / 16#09
ERR_EXCEPT_0A_GATEWAY_PATH_UNAVAILABLE	16#10A	266	网关路径不可用 服务器的异常响应包含错误码 10 / 16#0A
ERR_EXCEPT_0B_GATEWAY_TARGET_DEVICE	16#10B	267	网关目标设备响应失败 服务器的异常响应包含错误码 11 / 16#0B
ERR_TIMEOUT	16#120	288	超时过期 服务器没有在指定的时间内响应
ERR_CONNECTION_CLOSED	16#121	289	连接关闭或堆栈报告意外结果
BUSY	16#FFF	4095	功能块忙碌, 请再次调用触发
FATAL_ERROR	16#5FFF	4095	致命错误状态机
ERR_NO_MEMORY	16#8001	32769	内存不足
ERR_INTERNAL_IO	16#8002	32770	I/O 层内部错误
ERR_INTERNAL_INVALID_STATE	16#8003	32771	无效状态, 致命的内部错误
ERR_INTERNAL_ILLEGAL_STATE	16#8004	32772	协议栈处于非法状态 致命内部错误

ERR_INTERNAL_UNEXPECTED_STATE	16#8005	32773	意外的状态，致命的内部错误
ERR_INTERNAL_IO_RETRY	16#8006	32774	重试 I/O 操作。操作尚未完成，需要再次调用功能块
ERR_INTERNAL_ILLEGAL_ARGUMENT	16#8007	32775	堆栈报告非法参数(内部错误)
ERR_INTERNAL_INVALID_HANDLE	16#8008	32776	非法参数，未知实例' dwHandle ' 内部错误
ERR_INTERNAL_PORTING	16#8009	32777	移植层错误 操作系统信号错误
ERR_UNKNOWN	16#800A	32778	未定义的错误

联系我们

欲了解更多信息，请联系您的

当地 ABB 代表或以下人员之一：

new.abb.com/drives/low-voltage-ac/servo-products

new.abb.com/drives

new.abb.com/drivespartners

new.abb.com/PLC

© 版权所有 2022 ABB。

保留所有权利。

规格如有更改，恕不另行通知。

