



应用笔记

如何使用西门子S7-1500 (CPU 1516-3 PN/DP) 启动 电机控制器/驱动器

版本1.0.1

应用笔记-示例程序-数字输入

版本1.0.0

内容

1	预期用途和受众	2
2	要求	2
3	创建新项目和通信设置	3
4	Modbus映射	6
5	定位模式运行示例	8
5.1	启用通信	8
5.2	如何开始定位动作	9
6	责任	9

1 预期用途和受众

基于两个示例项目，本应用笔记向您展示如何在TIA Portal中使用西门子S7-1500（CPU 1516-3 PN/DP）启动电机控制器/驱动器。为安全起见，请遵循有效的OEM说明。本文件中要求的所有产品仅供经过培训的专家使用。在使用产品之前，请确保所有用户充分阅读、理解并遵循本文件中的说明。

2 要求

通知

不兼容导致的故障！这里需要的产品有各种版本。提前选择/安装/配置正确的产品。

- ▶ 满足要求。
- ▶ 仅使用兼容设备。
- ▶ 遵循有效的OEM说明。

在使用我们的示例项目之前，您必须确保电机和从动驱动器的正常运行。

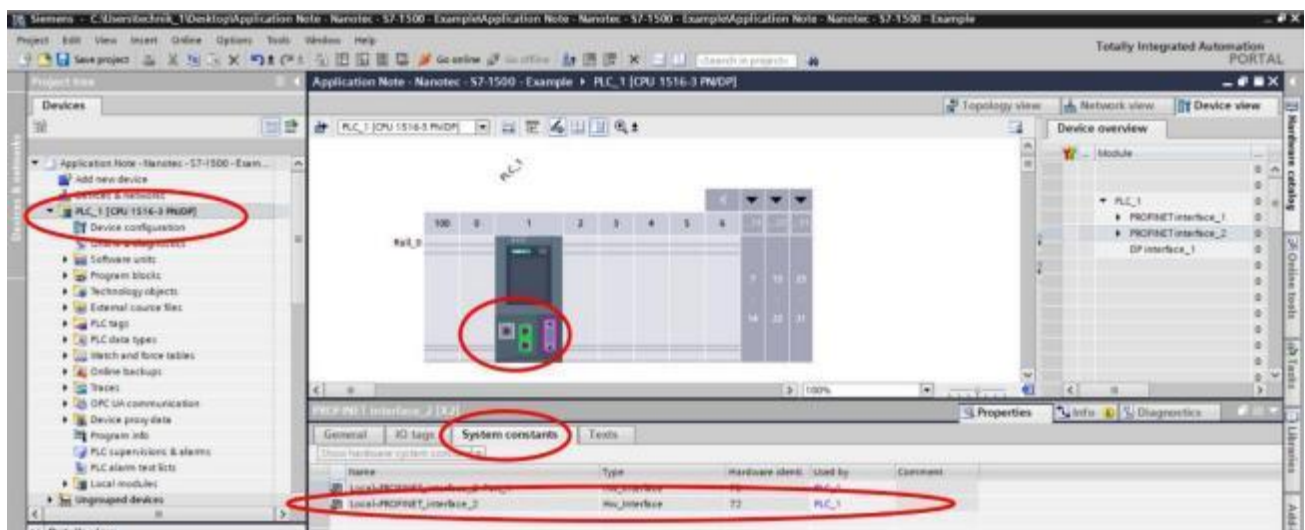
- 1 提前配置从驱动器（即控制器映射）。
- 2 保持控制器/驱动器操作能够正常执行

硬件	● 西门子S7-1500（CPU 1516-3 PN/DP），硬件版本3，引导加载程序版本V 2.2.1，固件版本V 02.06.00
	● CPBH03-TC，固件版本FIR-V2213-B577172
软件	● TIA门户v16，第7步专业版v16

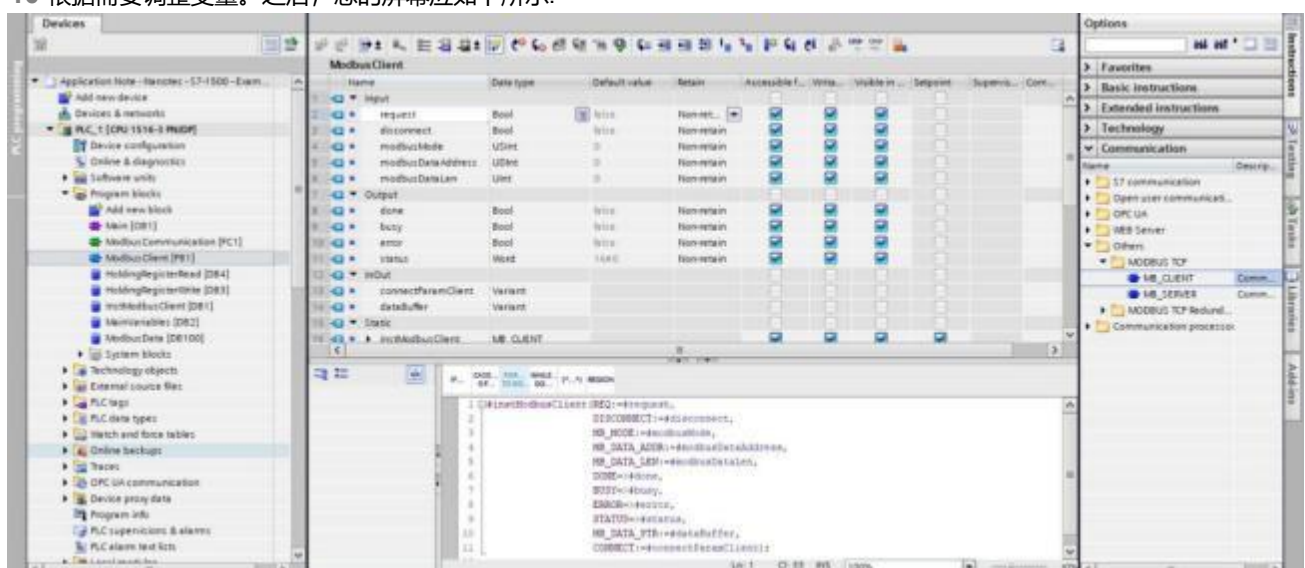
3 创建新项目 and 通信设置

在示例项目中，用于通信的功能块是Modbus通信。您可以简单地加载任何示例程序并调整设置。给定的示例显示在“项目”视图中。并使用梯形图编程。或者，您可以详细设置通信：

- 1 在TIA Portal中创建新项目。
- 2 将西门子CPU添加为新设备。
- 3 在“设备配置”中：单击驱动器所需的连接器。
- 4 对于驱动程序的端口： **检查系统常量>硬件标识符**。



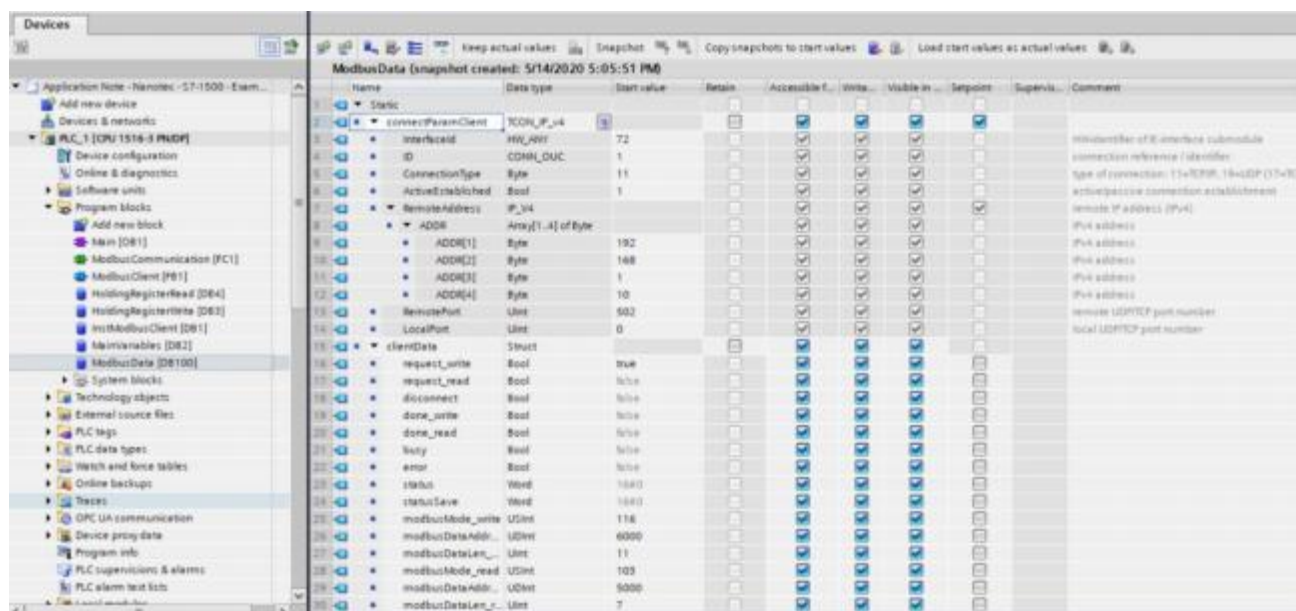
- 5 记下端口ID以备后用。
- 6 将PLC驱动器端口的IP地址（此处:192.168.0.1）设置为与驱动器的IP（此处:192.168.0.16）相同的网络。
- 7 现在，为PC端口设置PLC的IP地址（此处:192.168.1.1），与PC的以太网卡（此处:192.168.1.3）位于同一网络中。
- 8 当我们使用PLC作为Modbus客户端时，我们使用MB_客户端进行通信。因此，创建一个新的（SCL）功能块（此处：ModbusClient）。
- 9 在这个块中，您可以从指令表中拖放MB_客户端（见下图）。创建新数据块后，所需设置将显示在编程窗口中。
- 10 根据需要调整变量。之后，您的屏幕应如下所示：



- 11 创建一个新的数据块，您可以在其中调整连接设置（此处：ModbusData）。

在这里，您必须调整几个设置，包括端口硬件ID（请参阅步骤3至5在上面）、驱动器IP地址和Modbus通信详细信息（模式、寄存器地址、写入/读取长度）。

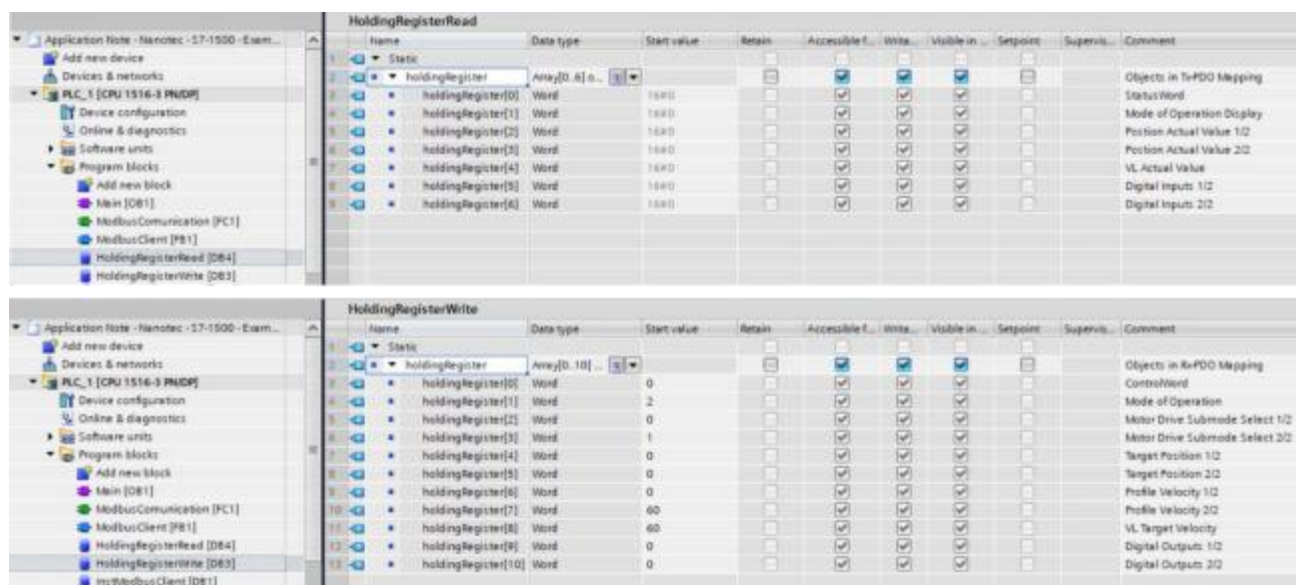
完成默认映射后，屏幕应如下所示(地址信息请参阅后续篇章详细介绍为准):



Name	Data type	Start value	Retain	Accessible for	Write	Visible in	Setpoint	Supervision	Comment
Static									
connectParamClient	TCOR_IP_v4	72							Module ID of R interface submodule
InterfaceId	HW_INT	1							connection reference / identifier
ID	CONR_DUC	11							type of connection: 11=TCP/IP, 18=UDP (17=TCP)
ConnectionType	Byte	1							actual connection establishment
ActualEstablished	Bool								remote IP address (IPv4)
RemoteAddress	IP_v4								remote IP address (IPv4)
ADDR	Array[1..4] of Byte								remote IP address (IPv4)
ADDR[1]	Byte	192							remote IP address (IPv4)
ADDR[2]	Byte	168							remote IP address (IPv4)
ADDR[3]	Byte	1							remote IP address (IPv4)
ADDR[4]	Byte	10							remote IP address (IPv4)
RemotePort	Unit	502							remote UDP/TCP port number
LocalPort	Unit	0							local UDP/TCP port number
clientData	Struct								
request_write	Bool	true							
request_read	Bool	false							
disconnect	Bool	false							
done_write	Bool	false							
done_read	Bool	false							
busy	Bool	false							
error	Bool	false							
status	Word	16#0							
statusSave	Word	16#0							
modbusMode_write	USInt	118							
modbusDataAddr...	USInt	6000							
modbusDataLen...	Unit	11							
modbusMode_read	USInt	103							
modbusDataAddr...	USInt	5000							
modbusDataLen...	Unit	7							

注意：在我们的例子中，我们总是写和读所有映射的寄存器。对于仅读取或写入单个寄存器或特定寄存器，您必须相应地调整Modbus数据地址/长度。我们仅将示例用于标准默认映射。您还可以找到有关modbus映射的详细信息（请参阅在下面）。

现在创建两个新数据块，分别用于写入和读取数据。在本例中，我们称它们为HoldingRegisterRead 和 HoldingRegisterWrite。它们只包含一个数据类型字的数组，代表Modbus寄存器，应该如下所示：



Name	Data type	Start value	Retain	Accessible for	Write	Visible in	Setpoint	Supervision	Comment
Static									
holdingRegister	Array[0..6] of Word								Objects in TePD Mapping
holdingRegister[0]	Word	16#0							StatusWord
holdingRegister[1]	Word	16#0							Mode of Operation Display
holdingRegister[2]	Word	16#0							Position Actual Value 1/2
holdingRegister[3]	Word	16#0							Position Actual Value 2/2
holdingRegister[4]	Word	16#0							VL Actual Value
holdingRegister[5]	Word	16#0							Digital Inputs 1/2
holdingRegister[6]	Word	16#0							Digital Inputs 2/2

Name	Data type	Start value	Retain	Accessible for	Write	Visible in	Setpoint	Supervision	Comment
Static									
holdingRegister	Array[0..10] of Word								Objects in TePD Mapping
holdingRegister[0]	Word	0							ControlWord
holdingRegister[1]	Word	2							Mode of Operation
holdingRegister[2]	Word	1							Motor Drive Submode Select 1/2
holdingRegister[3]	Word	1							Motor Drive Submode Select 2/2
holdingRegister[4]	Word	0							Target Position 1/2
holdingRegister[5]	Word	0							Target Position 2/2
holdingRegister[6]	Word	0							Profile Velocity 1/2
holdingRegister[7]	Word	60							Profile Velocity 2/2
holdingRegister[8]	Word	60							VL Target Velocity
holdingRegister[9]	Word	0							Digital Outputs 1/2
holdingRegister[10]	Word	0							Digital Outputs 2/2

注：CANopen对象有8、16或32位（Modbus寄存器仅有16位）。因此，您必须在映射中对齐CANopen对象，并在读取和写入时为32位CANopen对象使用两个Modbus寄存器。

现在可以创建用于通信的梯形图程序。为此，创建一个函数类型的新块（这里是：ModbusCommunication）。注：该功能有四个网络，分别用于（1）写入数据，（2）读取数据，（3）读/写交替，以及（4）通信错误检查。

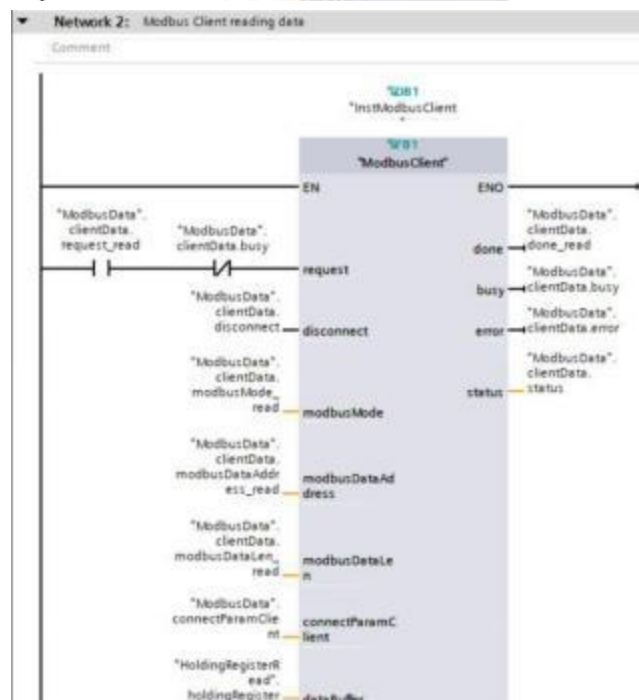
网络1:拖放

ModbusClient功能块。在这里，我们将所需的数据块称为InstModBusClient。

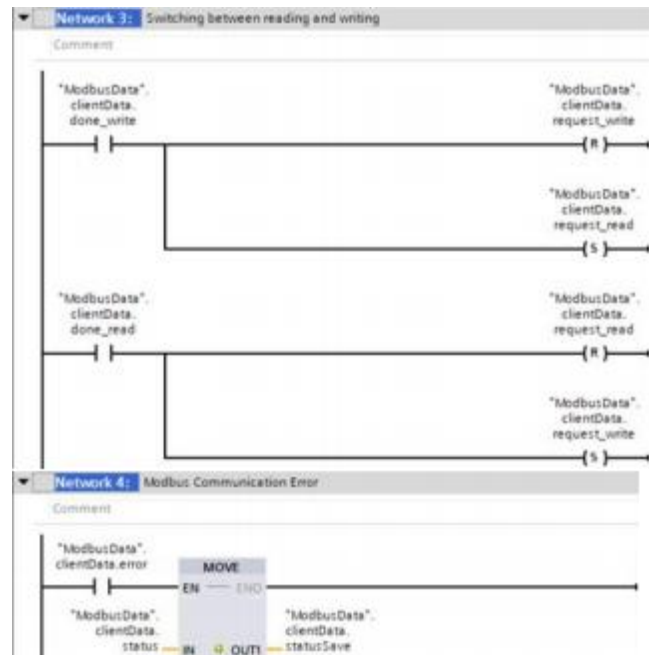
各增加一个常开/常闭输入。用正确的值填充I/O变量以进行写入。



网络2:重复步骤16至17在上面，这次是读取变量。



19网络3在写入和读取数据之间切换。它应该看起来像这样:



20网络4保存通信错误的状态, 应如下所示:

4 Modbus映射

对于标准PDO对象, 我们的驱动器预配置的默认映射:

寄存器地址	数据类型	功能
6000	UNSIGNED16	Bit0: 绝对定位模式 Bit1: 相对定位模式 Bit2: JOG点动模式 Bit3: 归零模式 Bit4: 力矩模式 Bit5: 急停/停止 Bit6&1: 绝对定位立即执行 Bit7: 复位错误 Bit8: 轴释放 Bit9: 轴使能 Bit10: 存储 存储和急停无需使能位 其它运动功能需要启用使能位
6001+6002	INTEGER32	JOG转速设定, 单位转/分
6003+6004	INTEGER32	定位转速设定, 单位转/分
6005+6006	INTEGER32	目标位置, 默认3600/圈

6007+6008	UNSIGNED32	加速度设定, 单位增加转速/秒
6009+6010	UNSIGNED32	减速度设定, 单位减少转速/秒
6011+6012	UNSIGNED32	Begin Acceleration Jerk
6013+6014	UNSIGNED32	Begin Deceleration Jerk
6015+6016	UNSIGNED32	End Acceleration Jerk
6017+6018	UNSIGNED32	End Deceleration Jerk
6018+6019	UNSIGNED32	归零方式: 35 当前位置置零 19 或 21 : 外部传感器零点 -17 或 -18 : 堵转回零
6020+6021	UNSIGNED32	归零寻找感应器的速度
6022+6023	UNSIGNED32	调零速度
6024+6025	INTEGER32	堵转归零电流
6026+6027	INTEGER32	力矩模式, 目标力矩
6028+6029	UNSIGNED32	力矩模式, 力矩斜率

注意: 32位数据, 6028 (高16位) +6029 (低16位)

状态指令表 (电机—PLC)

寄存器地址	数据类型	功能
5000+5001	INTEGER32	实时速度 单位 转/分
5002+5003	INTEGER32	实时位置 默认 3600脉冲/转
5004	UNSIGNED16	当前错误信息
5005	UNSIGNED16	历史错误信息
5006	UNSIGNED16	历史错误信息 (1个字节)
5007	UNSIGNED16	系统状态字 bit3:错误 bit10:到位 bit10&bit12: 回零到位 bit7: 触发限位警告
5008+5009	UNSIGNED32	配置日期

5010+5011	UNSIGNED32	自定义状态 bit0 0 轴释放 bit0 0x01 轴使能 bit1 0x02 运行中 bit2 0x04 运行到达 bit3 0x08 错误 bit5 0x20 急停 bit8 0x100 JOG模式 bit9 0x200 单点定位模式 bit10 0x400 回零模式 bit11 0x800 力矩模式 bit12 0x1000 连续定位模式 bit13 0x2000 存储完成
5012+5013	INTEGER32	实时电流
5014+5015	UNSIGNED32	输入点状态 bit16~bit18 代表Input1~4

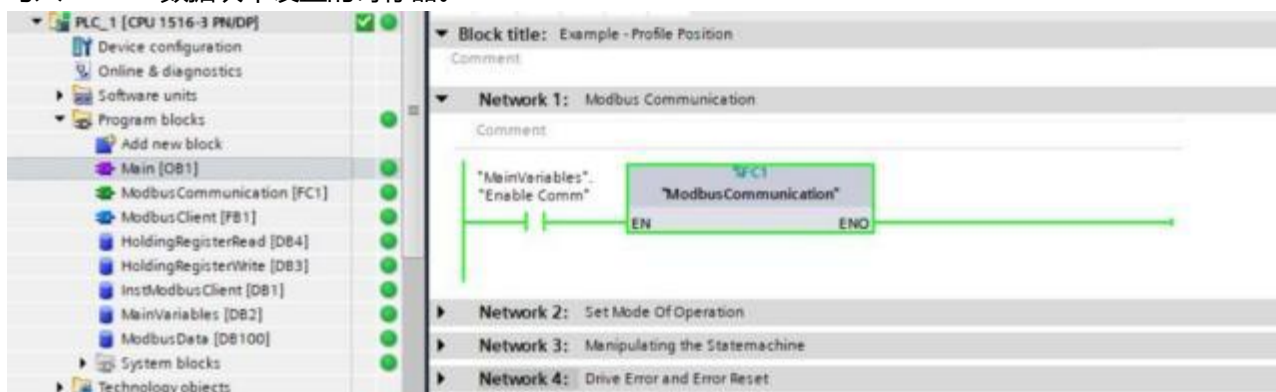
注意: 32位数据, 低位在前, 高位在后, 如5010 (高16位) +5011 (低16位)

5 位置模式运行的说明

此篇章简要说明如何开始定位模式运行

5.1 主梯形图程序

网络1仅启用Modbus通信。要开始通信, 只需将ENABLE COMM变量设置为1 (=true)。系统应自动读取/写入Modbus数据块中设置的寄存器。



5.2 如何开始定位动作

- 1 要开始通信，请设置Enable Comm输入。
- 2 设置Enable Drive Input (启用驱动输入) 以启动电机。
- 3 在两个数据块HOLDINGREGISTERWRITE.HOLDINGREGISTER[5]和[6]中设置目标位置。
- 4 **注：目标位置需要两个寄存器，因为Modbus只接受16位值，而对象目录需要32位目标位置。要达到32位，请组合两个16位值，例如-2000（十进制）或FFFF F830（十六进制）：**
 - a 保持寄存器WRITE.保持寄存器[5]==FFFF（十六进制）
 - b 保持寄存器WRITE.保持寄存器[6]==F830（十六进制）
- 5 设置目标速度保持寄存器WRITE.保持寄存器[4]=FA0（十六进制）
- 6 启动绝对定位运行保持寄存器WRITE.保持寄存器[0]=201(十六进制)

6 免责声明

应用说明在特定客户应用中的适用性和使用的责任完全由客户负责。客户有责任评估、调查和决定应用说明是否有效，是否适用于各自的客户应用。

因设备和模块处理不当而导致的缺陷不在保修范围内。在任何情况下，NANOTEC均不对因所提供的信息而产生的任何直接、间接、附带或后果性损害承担任何责任。

此外，关于我们的责任的规定销售和交付条款和条件 应适用。