Servo Motion

Application Note

Using CANopen for Motion Control

AN518 Rev A (CN)

介绍

本应用说明详细介绍了如何使用 AC500eco V3 以及集成到 Automation Builder 中的工具通过 CANopen 协议控制非 ABB 伺服驱动器。目前，该过程需要几个手动步骤，这些步骤以后可能会发生变化，因此请始终检查本应用说明的最新版本。如果您需要任何其他信息，请联系您当地的 ABB 支持团队。

兼容性

PLC 程序和库是使用 **Automation Builder v2.8.0** 或更高版本编写的，可以从这里下载：
https://new.abb.com/plc/automationbuilder

运行此示例所需的 PLC 硬件是 **ABB AC500eco V3 PLC 系列**。它们有两种主要类型：AC500 和 AC500eco。由于此系列的所有 CPU 都可以支持 CANopen，因此任何 CPU 都可以。在本例中，我们将重点介绍**配备 TA5146-CN 选件模块**的 AC500eco V3 PM 5092-T-2ETH。

AC500 V3 CPU 的型号在代码的 PM 部分后有四个数字，例如 PM5072-T-2ETH，而上一代 CPU 只有三个数字，例如 PM564-ETH。

保修、责任：
用户应对本文件中描述的本产品的使用负全部责任。ABB 不提供任何保证。ABB 与所提供的产品或示例或这些产品中包含的文件的应用相关的责任，无论法律依据如何，均应被排除。责任免除不适用于故意或重大过失的情况。预先发送的声明应受瑞士法律管辖并按其解释，但不包括其法律冲突规则和《维也纳国际货物销售公约》（CISG）。

目录

1. 概述

1.1 带 CANopen 的 AC500 V3

当用户需要通过 CANopen 协议连接到轴或设备时，AC500 V3 PLC 可以使用下列多种方法中的一种。

1. AC500eco V3 安装 TA5146-CN 选件模块。
2. AC500 V3 板载 CANopen
3. AC500 V3 + CM598_CAN 选件模块。

在本文中，我们将重点介绍方法 1，尽管它们中的任何一个都可用于实现相同的结果。

2. 编程和控制方法

当然，一旦设备连接，那么我们必须考虑如何控制它。包含 PLCopen 功能块的 AC500 运动库为控制这些设备提供了一种方便和标准的方式。为了实现这些功能的核心控制，需要在硬件和控制协议之间建立连接。有两种方法可以实现此目的 – 手动编写代码或使用向导。AB Motion Wizard 提供了一种快速简便的机制来自动为用户完成此连接，但它目前仅支持 ECAT 和 PTO 硬件。尽管如此，部分使用此方法是建立连接并让程序可以运行最快的方法，因此本文档将概述如何使用 Wizard 方法，以及使其与 CANopen 设备一起运行所需的必要编辑和解决方法。

请注意，本例中提到的一些功能块包含在运动控制扩展库中，可在此处下载:
https://search.abb.com/library/Download.aspx?DocumentID=9AKK108468A9993&LanguageCode=en&LanguageCode=zh&DocumentPartId=1&Action=Launch

2.1 硬件配置

如前所述，PLC 需要一个选项模块 “TA5146-CN” 来充当网络的 CANopen 主站，它有需要配置的 DIP 开关，并且它需要位于 CPU 中的一个插槽中，CPU 通常具有 2 或 3 个插槽，具体取决于 CPU 型号。

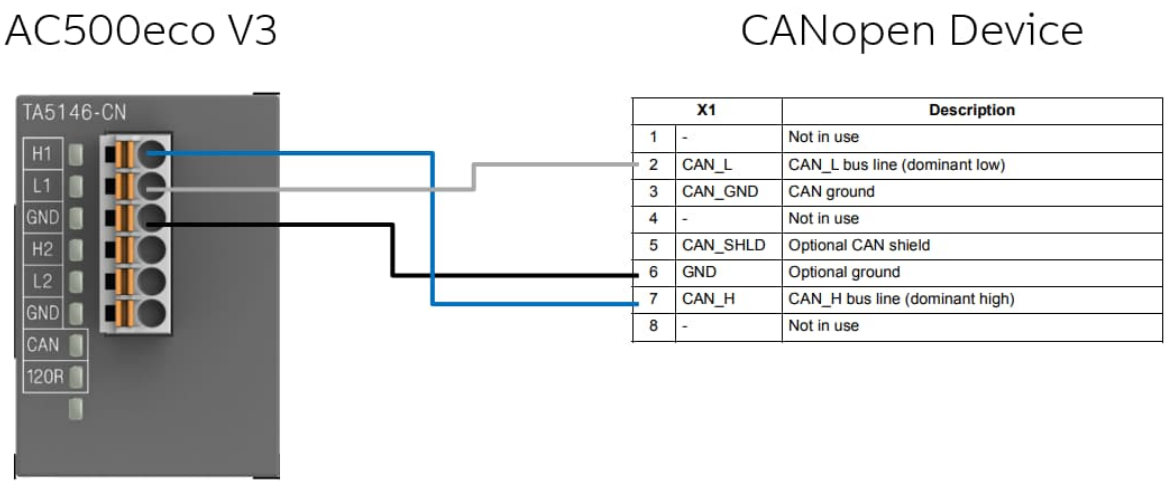
2.2 安装选件模块

TA5146-CN 的安装比较简单。推入左侧的卡舌，然后卸下占用插槽的空白模块，然后将选件模块放回原位。您现在可以启动并使用它了。

2.3 布线和终端电阻

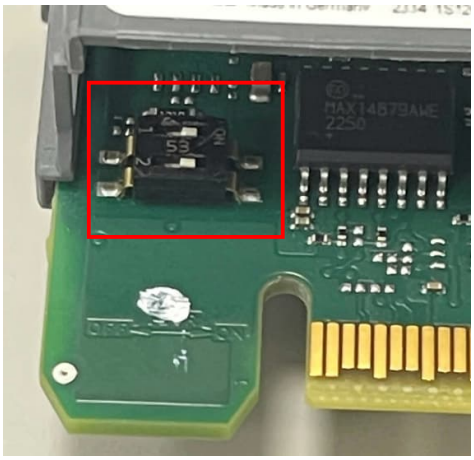
为设备布线时，应使用屏蔽双绞线电缆：

可以按如下所示连接：



X1		Description
1	-	Not in use
2	CAN_L	CAN_L bus line (dominant low)
3	CAN_GND	CAN ground
4	-	Not in use
5	CAN_SHLD	Optional CAN shield
6	GND	Optional ground
7	CAN_H	CAN_H bus line (dominant high)
8	-	Not in use

除了应在网络两端使用终端电阻外，这些电阻应为 120 欧姆，连接在传输线和接地之间。在 AC500eco TA5146-CN 中集成了这些电阻器，可以根据用户要求打开或关闭。在这种情况下，由于我们是网络的一端，因此我们必须将开关 1 和 2 选择为“On”。



此外，连接的驱动器/设备也应该具有这些连接，但它们取决于设备，因此应查阅设备用户手册以了解如何执行此操作。

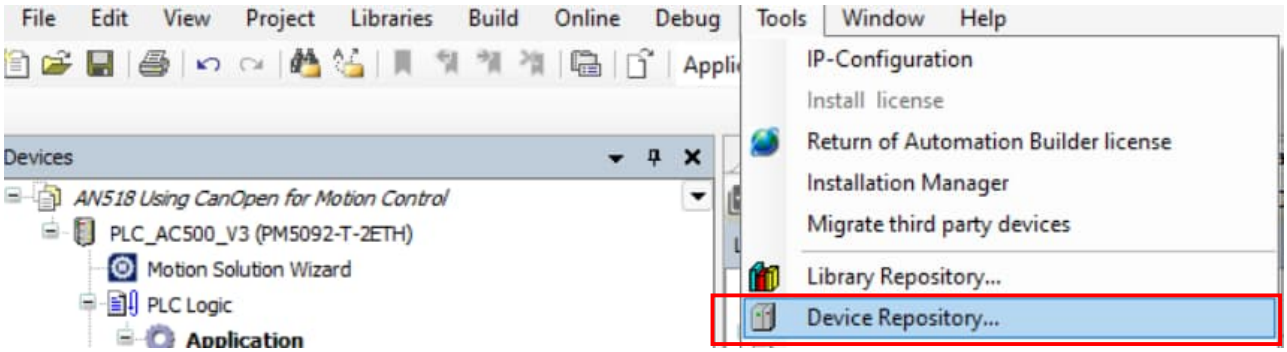
注意：终端电阻仅在网络中的第一个和最后一个节点上需要

3. 项目配置

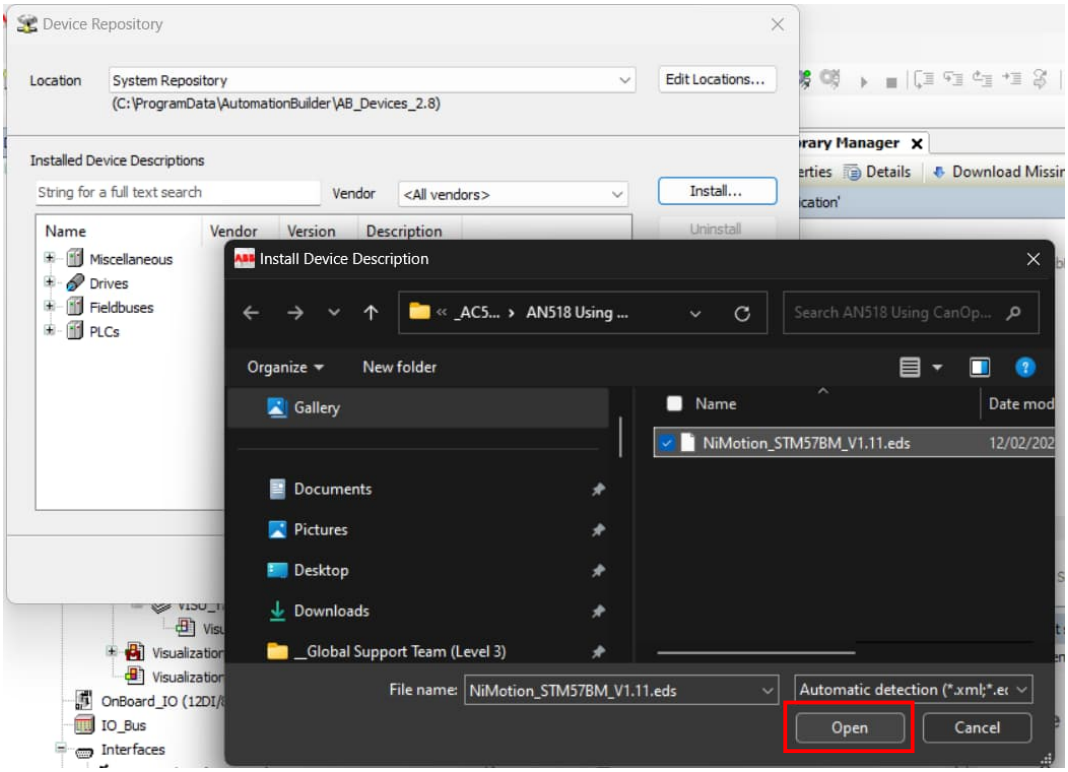
本节介绍如何配置 Automation Builder 项目，以便与连接的设备进行通信和控制。在下面的示例中，我们将使用第三方“STM57BM”设备，它是一个集成的驱动器和电机设备。

3.1 初始配置和 EDS 导入

在配置硬件之前，我们必须将 EDS 文件导入到设备存储库。为此，我们必须打开 Automation Builder 并在菜单栏 Tools 中选择 Device Repository



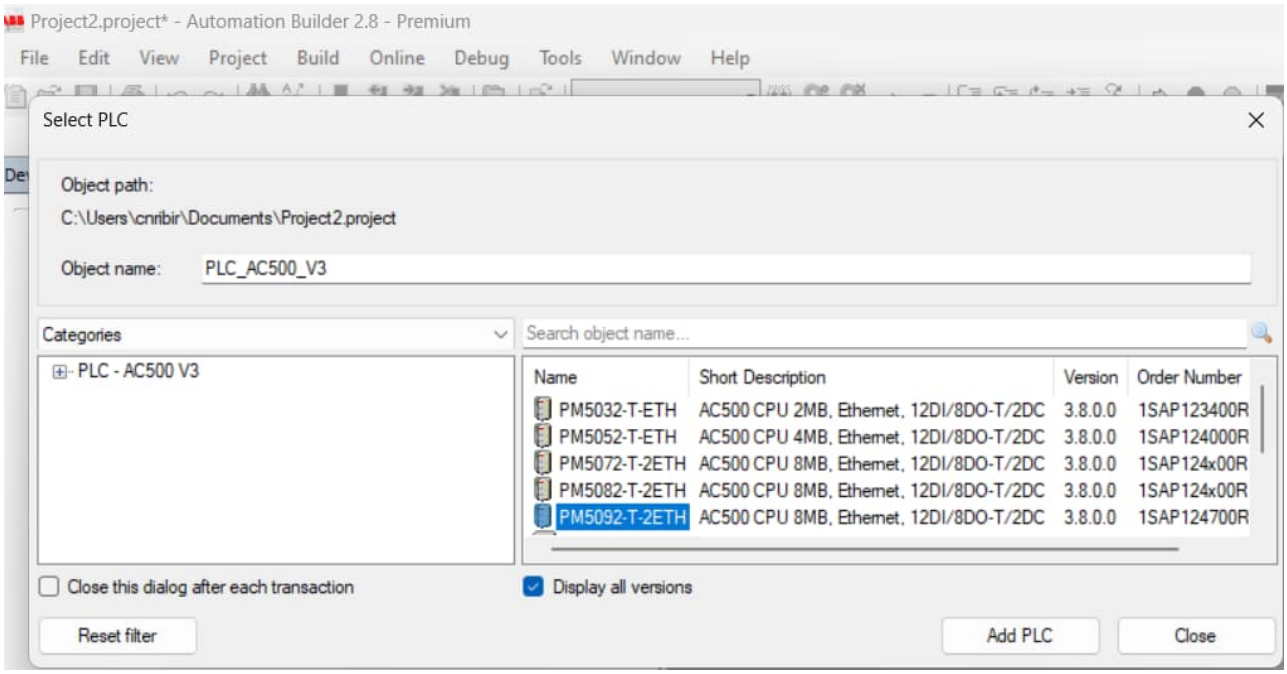
接下来选择“安装...”并导航到文件的存储位置（扩展名应为“eds”），然后单击“打开”。



现在，您将看到设备 'STM57XXB_CANopen_M' 已在您的应用程序中供选择。现在可以编写代码了。

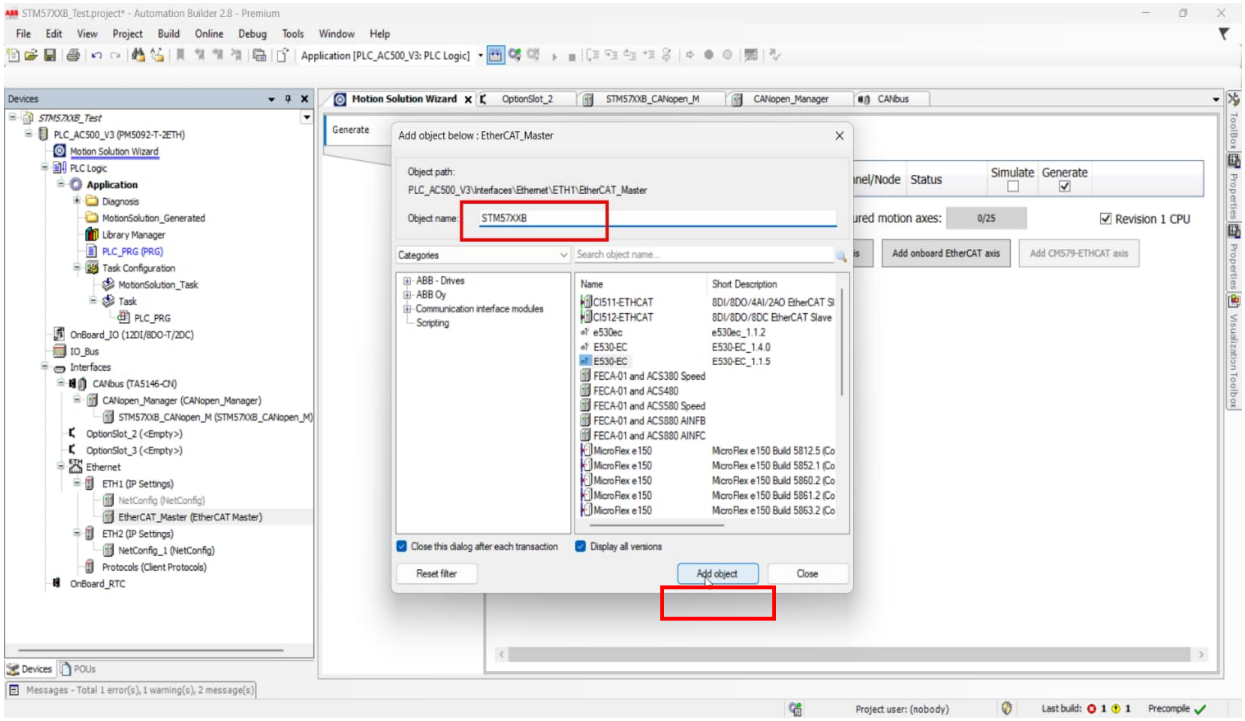
3.2 创建实轴代码

在 Automation Builder 中打开一个类型为“Motion Solution Project”的新项目，然后我们必须选择将要使用的 PLC 类型，在本例中为 PM5092-2ETH。



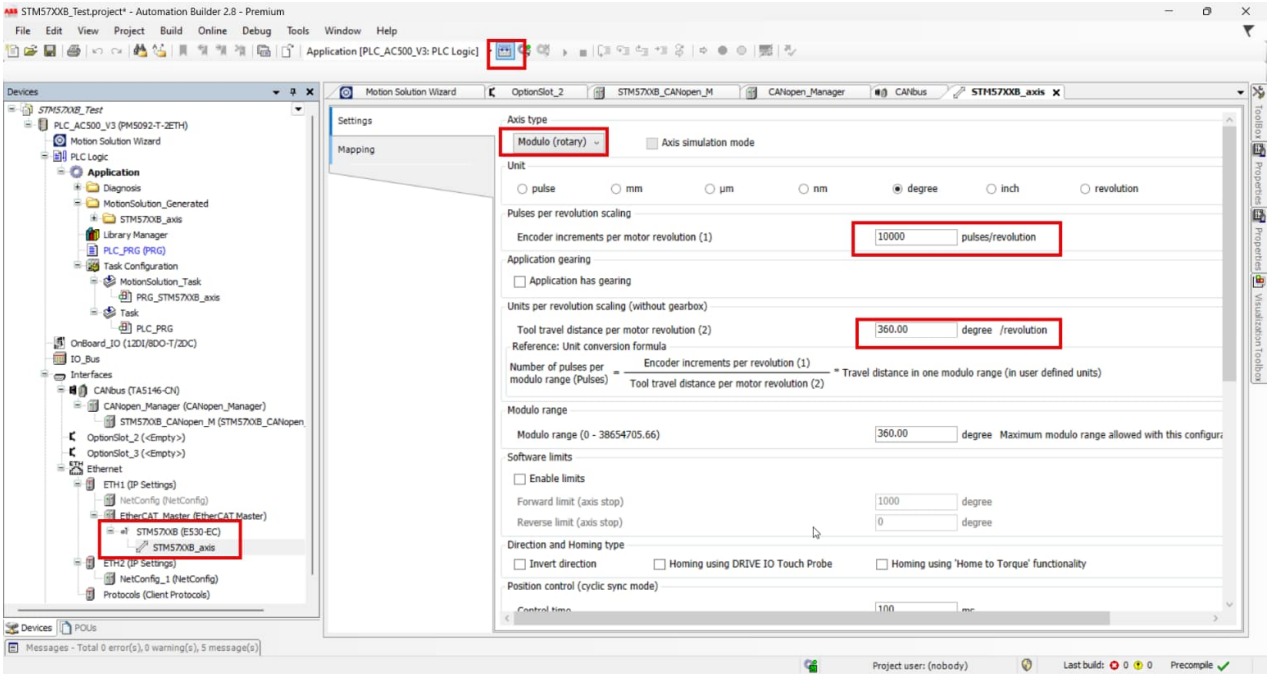
注： 如前所述，我们将使用向导为我们生成一个模板文件，用于集成 Axis。完成此步骤并生成模板后，它可以重新用于其他 CANopen 轴，无需重复向导步骤。

现在已完成硬件配置，打开 Motion Solution Wizard 并单击“添加板载 EtherCAT 轴”并选择硬件 e530-EC – 建议对象名称与您稍后将使用的实际轴名称相同，以便更容易分辨各个轴。



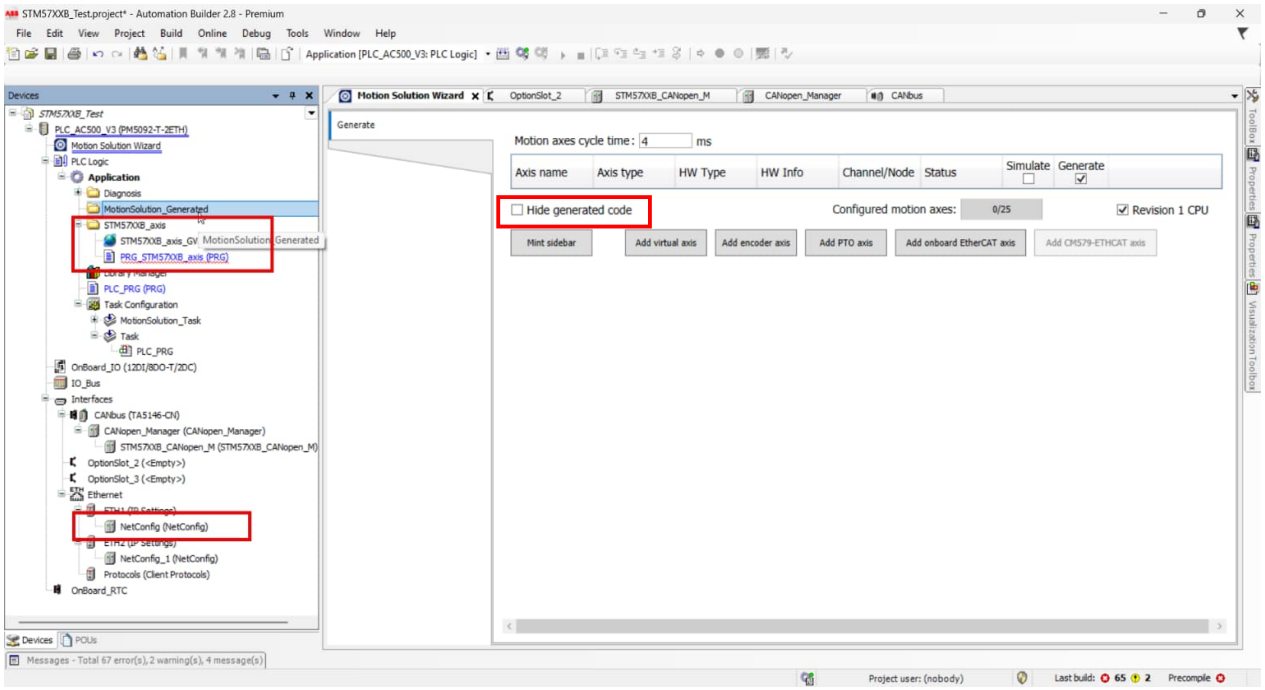
注意：选择的硬件类型在这里并不重要，但我们必须选择其中一个。

接下来，以根据实际应用要求设置参数配置参考轴，在本例中为每转 10000 个脉冲和 360 个用户单位，然后生成代码



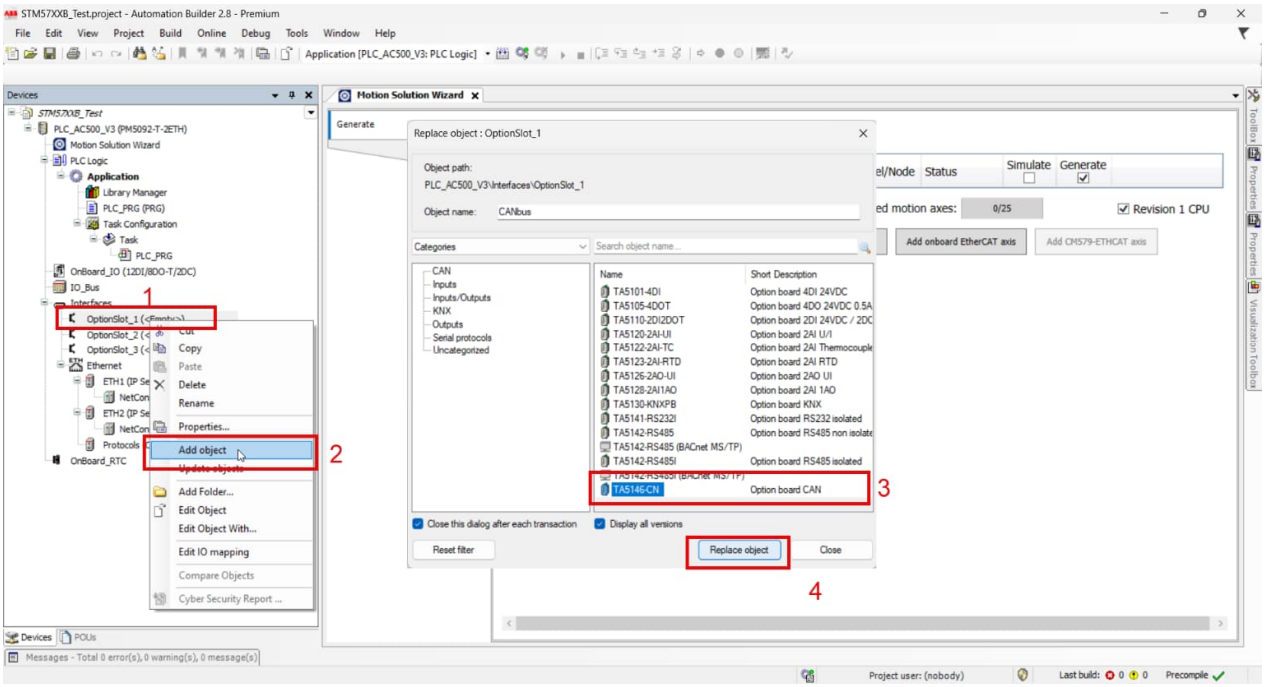
现在我们已经为标准 EtherCAT 轴**生成了代码和 GVL 列表的基本模板**，我们必须将其从'MotionSolution_Generated'文件夹（由运动控制向导控制）移动到 PLC 的'应用程序'根目录或根目录中的新文件夹。为此，请剪切名为 'STM57XXB_Axis' 的文件夹并按照说明移动它，然后删除参考轴和 EtherCAT 主站（除非应用程序的其他部分需要它们）

注意：要查看 'MotionSolution_Generated' 文件夹，请确保没有选中 '隐藏生成的代码' 复选框 。

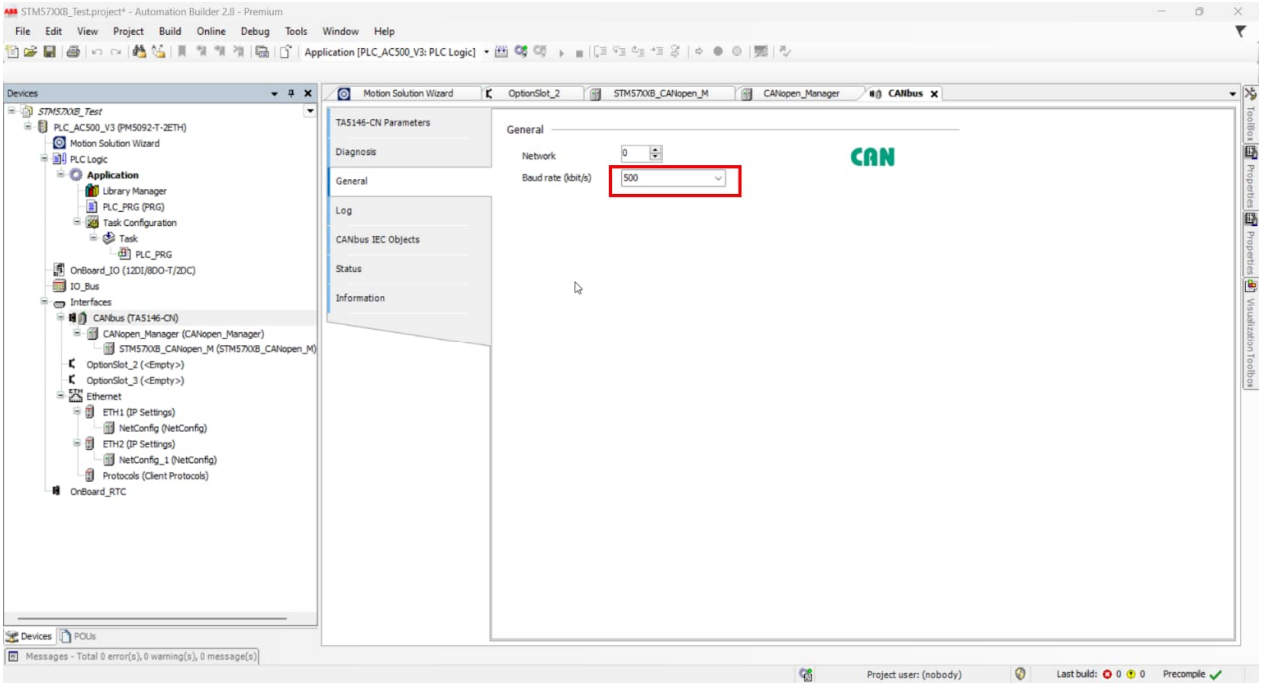


3.3 硬件和 CANopen 管理器的配置

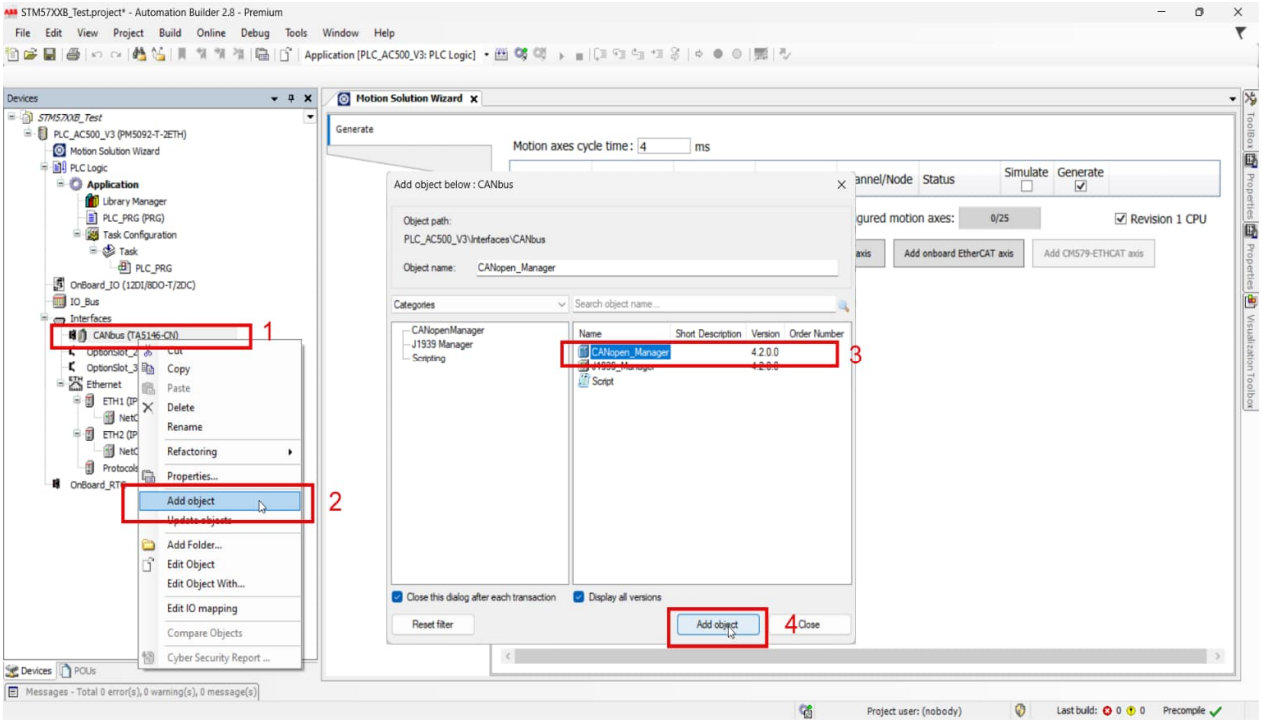
现在将“TA5146-CN”添加到配置中，为此，我们必须选择我们计划使用的插槽，在本例中为“插槽 1”，接下来右键单击并“添加对象”，选择 TA5146-CN 并按“替换对象”



接下来，我们必须为连接的设备和网络功能设置正确的波特率，对于本例的 STM57B5B 驱动器，正确的波特率为 500 kbit/s

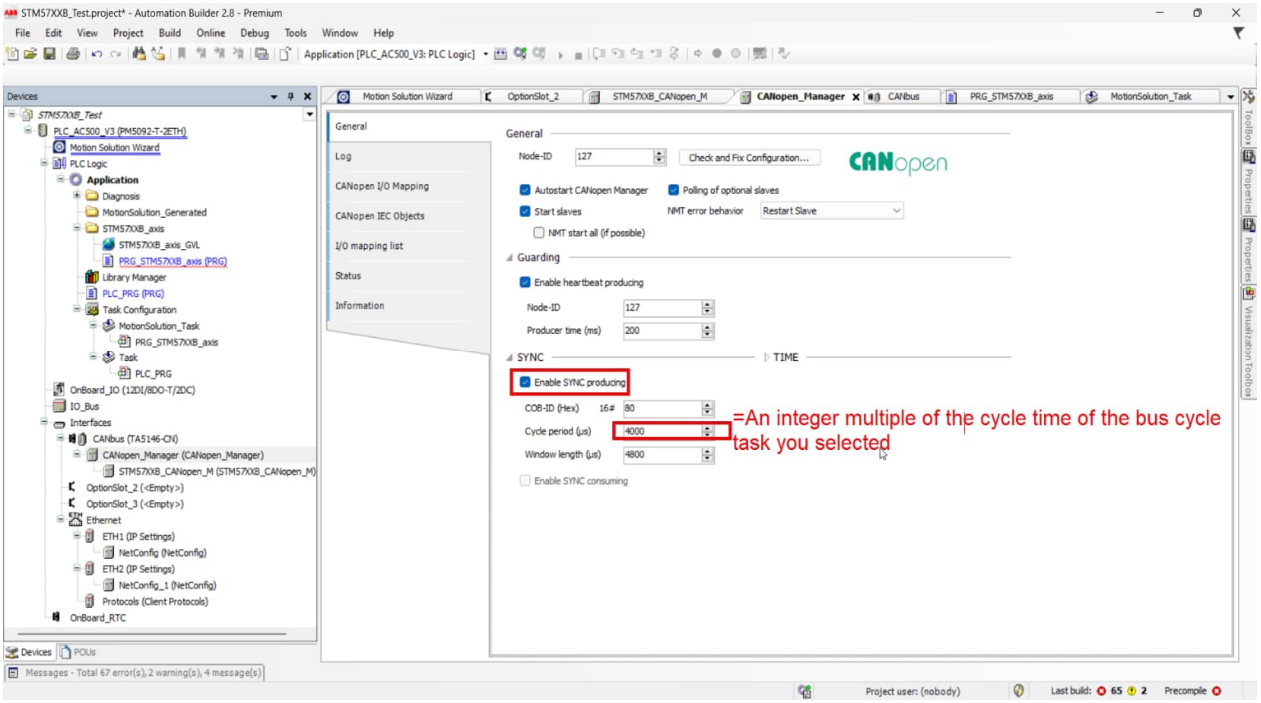


现在在 “TA5146-CN”下添加 CANopen 主站，右键单击 CANbus （TA5146-CN），“添加对象”，选择 CANopen 主站，然后按“添加对象”。

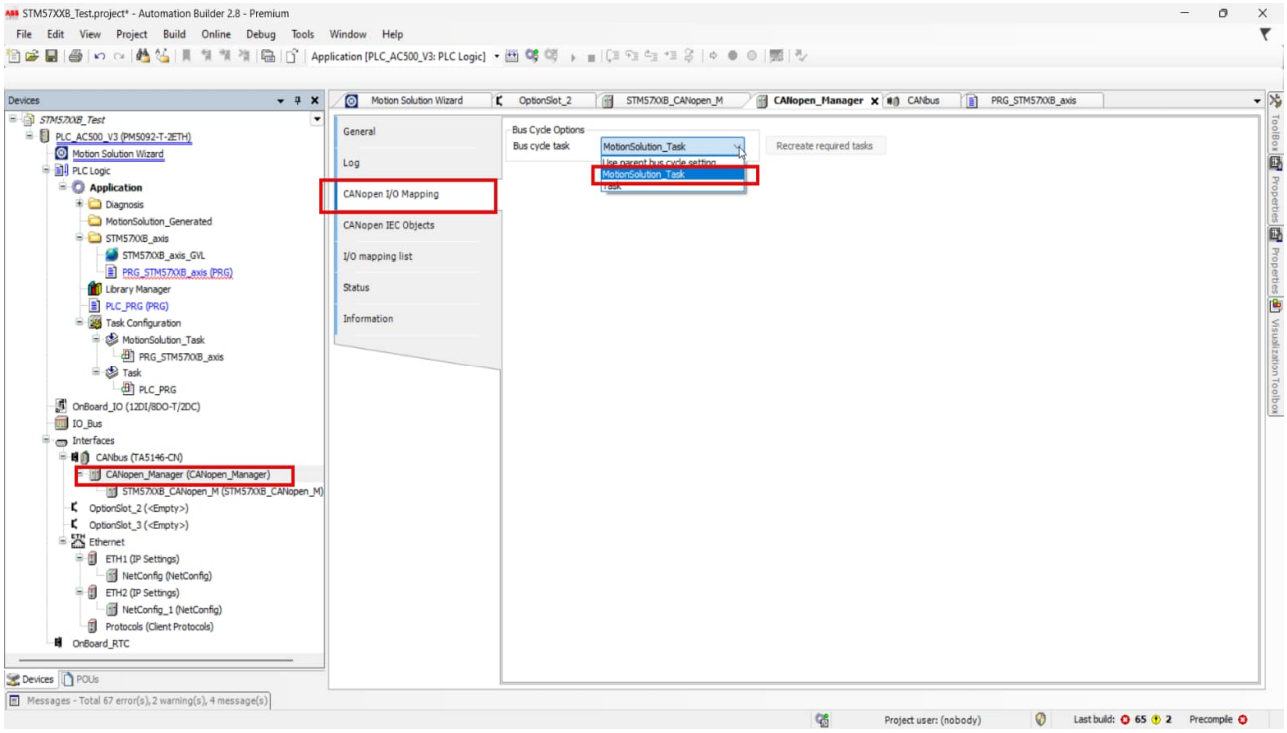


现在，CANopen 主站已添加完成，根据网络和设备要求对其进行配置。

首先配置 'General' 设置。此处选择了 'Enable SYNC producing' 以及与将调用它的任务的周期时间匹配的周期;在本例中，我们选择了 4000µs/4ms。

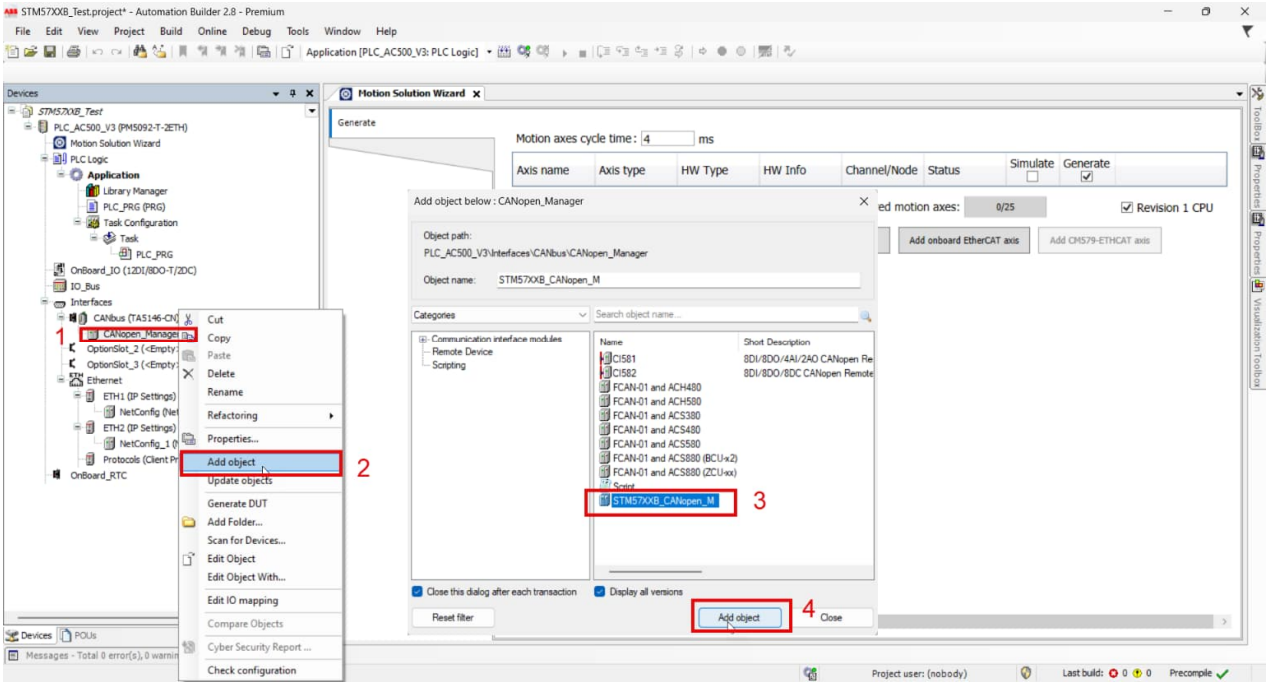


根据上述设置，我们现在可以导航到 'CANopen I/O Mapping 页面并选择我们想要用来驱动这些通信的任务。由于我们已经使用了“Motion Solution Wizard”，因此它创建了 MotionSolution_Task，我们可以在此处选择它。



3.4 添加驱动器

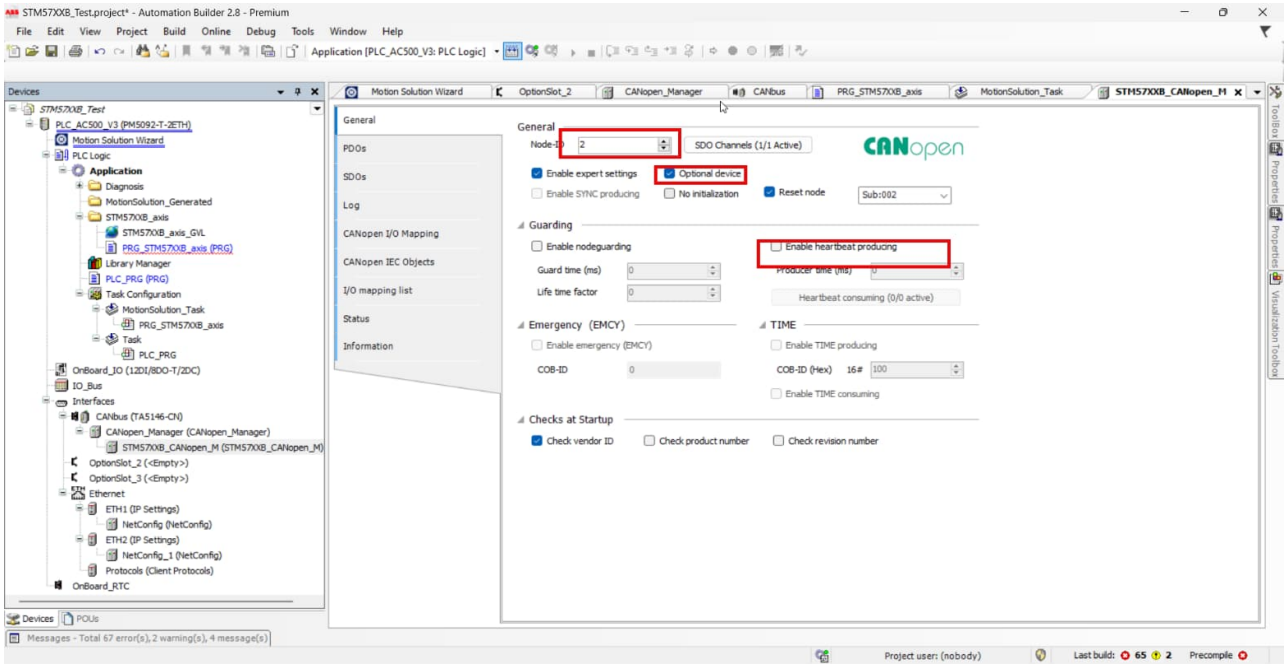
现在 CANbus 硬件和主站都已添加，我们可以将驱动器添加到配置中。为此，请右键单击 CANopen 主站，选择“添加对象”，选择“STM57XXB_CANopen_M”（我们之前导入的），然后按“添加对象”



现在驱动器对象已添加，我们可以对其进行配置，为此，请完成以下步骤

首先，设置基本通信设置，首先打开驱动器对象并选择 'General'

1. 设置驱动器的实际 Node-ID
2. 选择 Optional device 选项
3. 取消选择 'Enable heartbeat producing'。



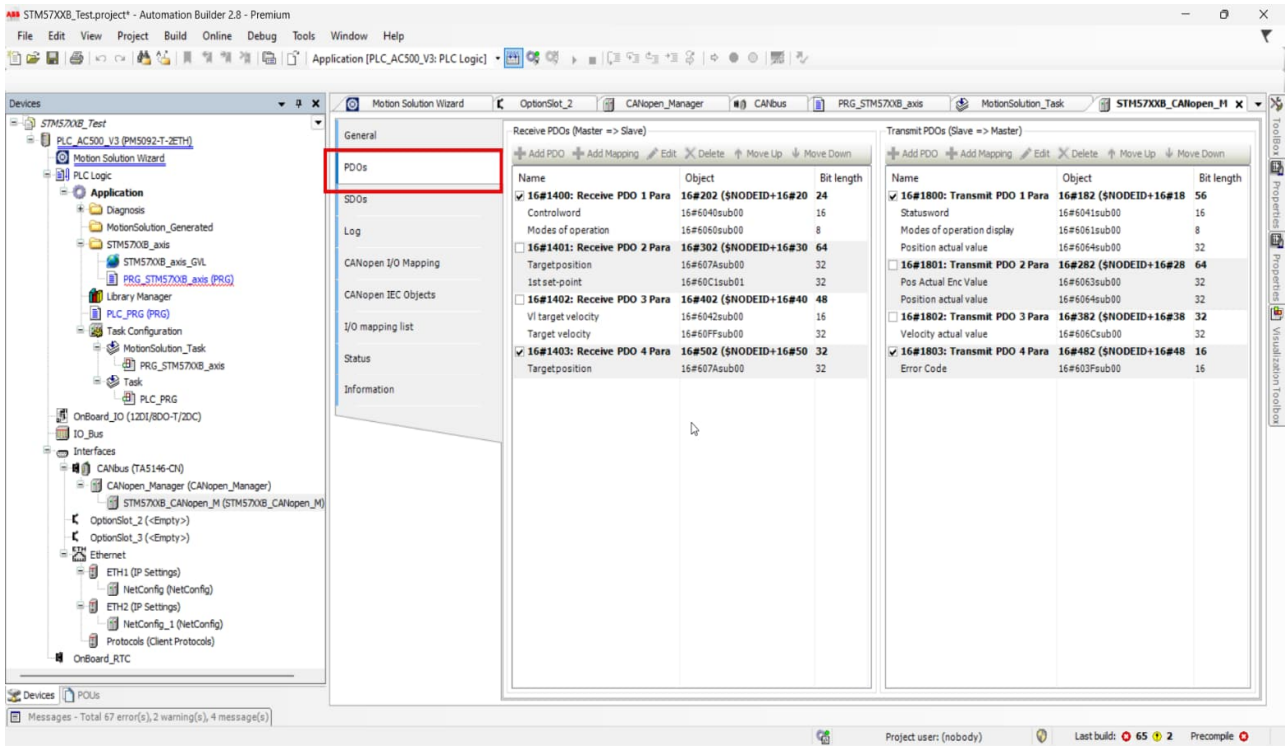
现在设置 Drives PDO 设置。为此，首先打开驱动器对象并选择 'PDO'

对于接收 PDO

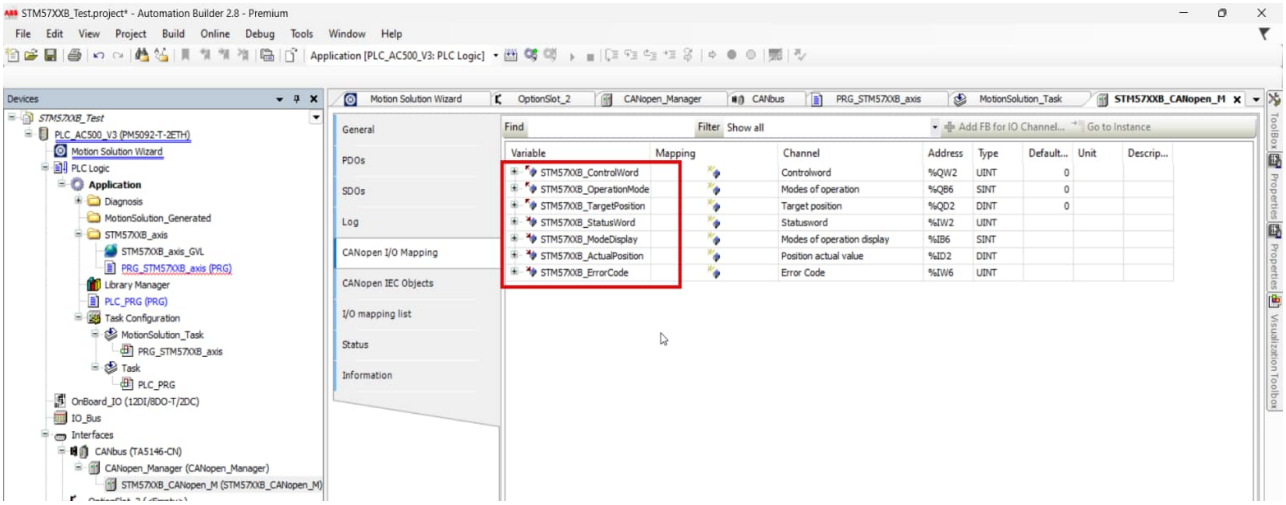
1. 如果您计划使用 IP 或 CSP 模式，请选择 16#1401
2. 如果您计划使用 CSP 模式，请选择 16#1403

对于发送 PDO

3. 在所有情况下选择 16#1800 和 16#1803。

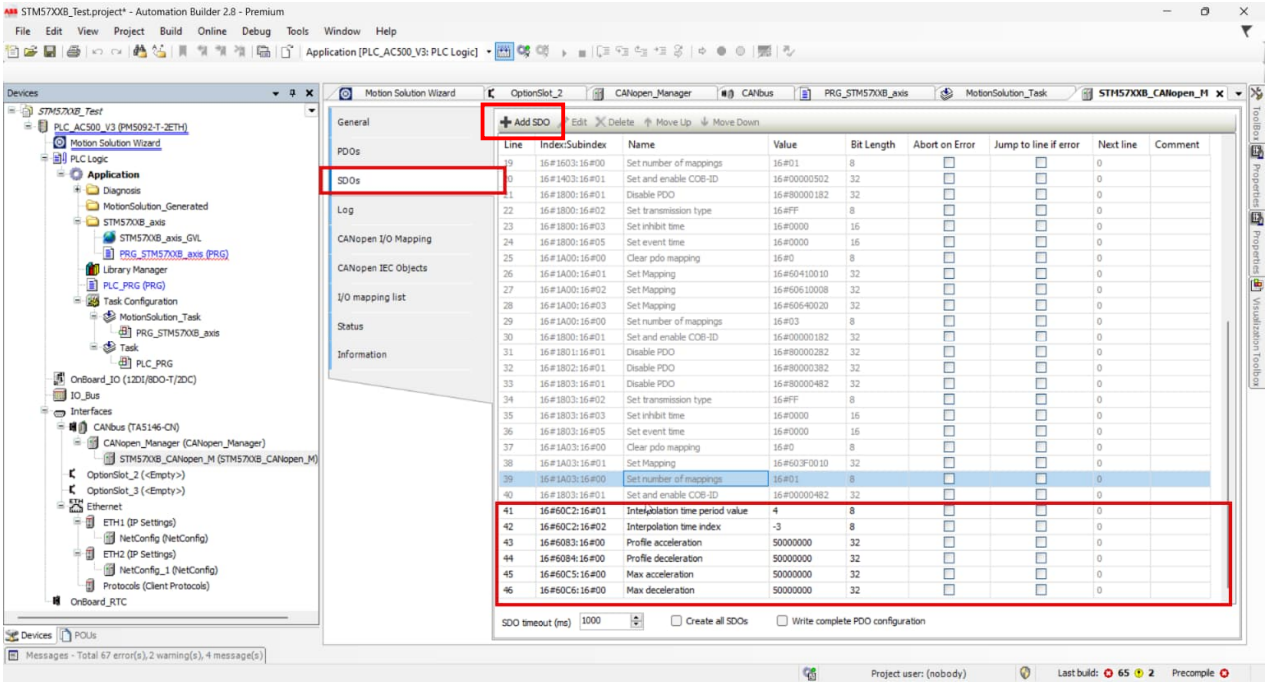


现在我们已经选择了 PDO Drive I/O 定义与它们关联的变量。为此，请转到 CANopen I/O 映射选项卡。
为了与生成的代码最紧密地匹配，我们应该使用与之前在 Section 4.2 中选择的 '*axis name*' 相同的前缀，然后是 '_'，然后是 *功能*。
在这种情况下，轴被调用 STM57XXB 因此，例如，第一个 Variable name 变为 'STM57XXB_ControlWord'，依此类推。
注意： 你也可以签入 'PRG_STM57XXB_axis' 来查看使用的变量名称，然后直接复制它们。



由于这是一个简单的驱动器，理想情况下几乎不需要用户配置，因此一些参数是通过 SDO 设置的。 设置驱动器 SDO 设置，首先打开驱动器对象并选择“SDO”。
要添加未显示为默认的其他 SDO，请使用“+ 添加 SDO”按钮，如下所示

1. 选择 16#160C2：16#01 插值时间周期值 = 4（用于插值的周期时间）
2. 选择 16#160C2：16#02 插值时间索引 = -3（根据用户手册）
3. 选择 16#16083：16#0 Profile Accel = 500000000（用户单位）
4. 选择 16#16084：16#0 Profile Decel = 500000000（用户单位）
5. 选择 16#160C5：16#0 Max Accel = 500000000（用户单位）
6. 选择 16#160C6：16#0 Max Decel = 500000000（用户单位）

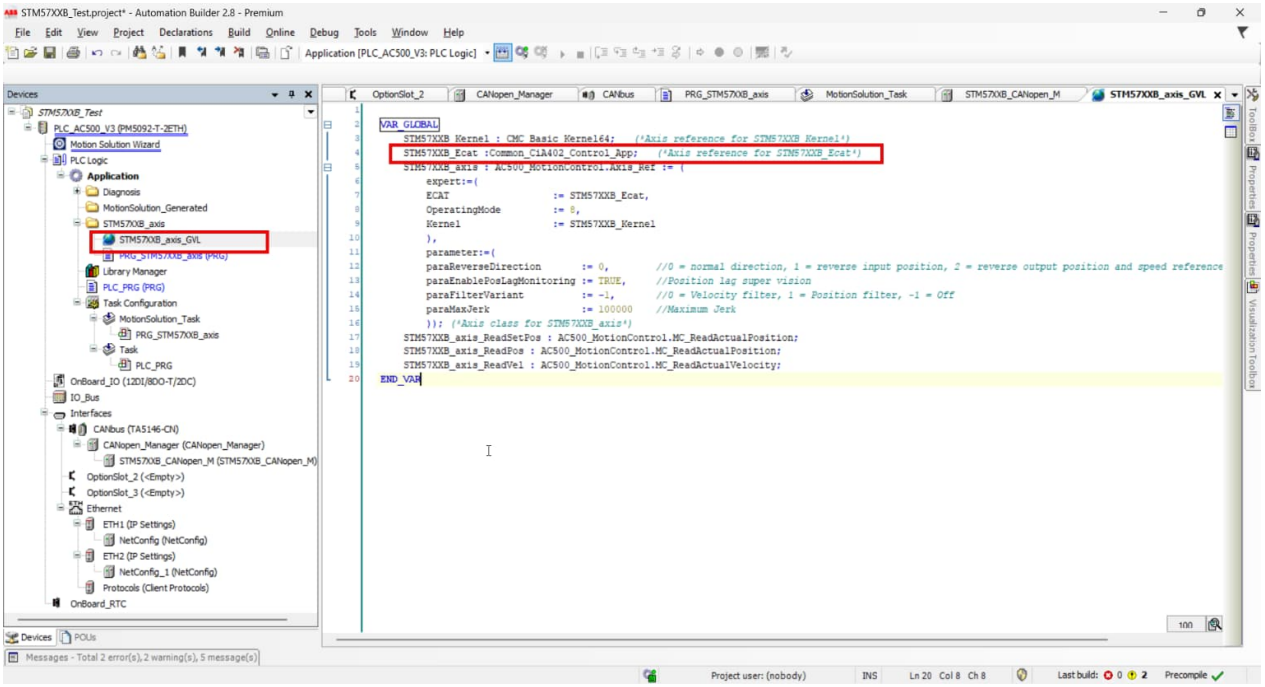


注意：上述 SDO 是此特定硬件的示例，在其他设备中可能会有所不同

3.5 编程

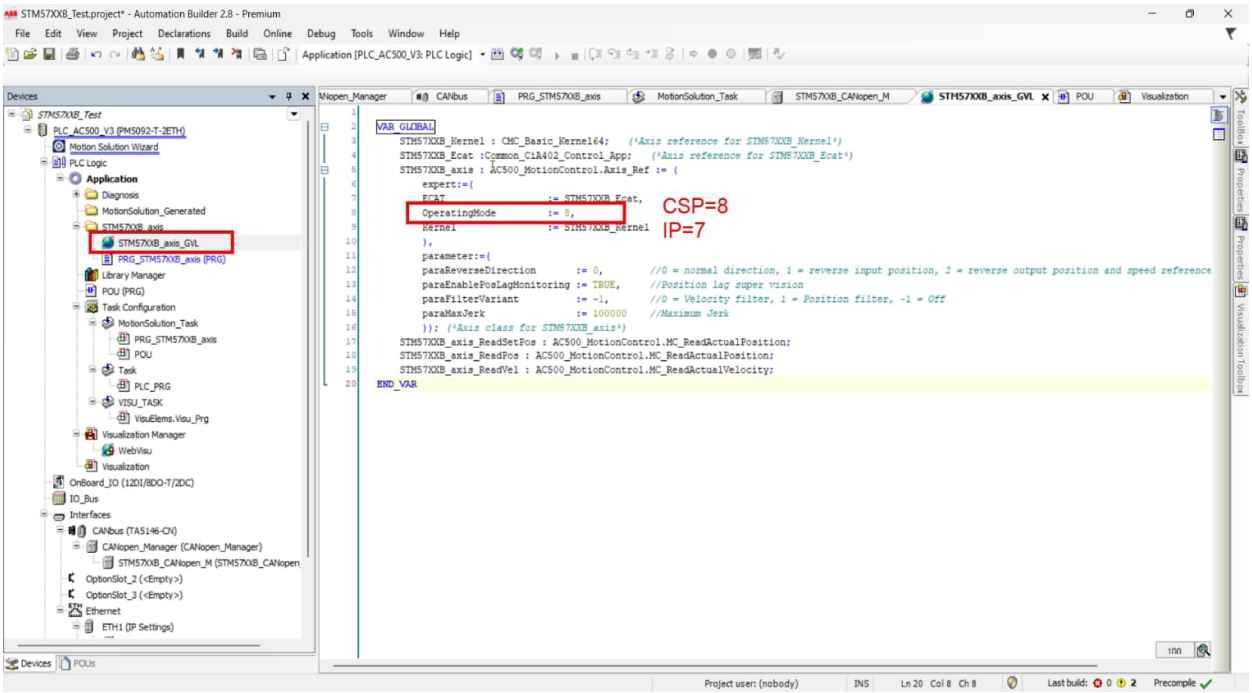
接下来可以选择 drive 接口（这是生成的代码的第一个更改）。这里的关键点是，我们将生成的代码变量声明中的 'ECAT_***CiA402_Control_App***' *的实例*更改为 '*Common*_***CiA402_Control_App*** *的实例*。这个新功能块保留了 CiA402 状态机的处理，但没有 EtherCAT 部件，这使得它非常适合 CANopen 应用。要进行正确的编辑，请打开轴 'STM57XXB_axis_GVL' 的全局变量声明，并找到引用

'**E530_EC**_Ecat： ***ECAT***_CiA402_Control_App;（*轴参考 **E530_EC_Ecat***）"，然后将其编辑为 '**STM57XXB_Ecat**：**Common**_CiA402_Control_App;（*轴参考 **STM57XXB_Ecat***）'



接下来设置所需的驱动器运行模式。要进行所需的编辑，请打开轴 'STM57XXB_axis_GVL' 的全局变量声明，找到行 'OperatingMode： = **8**'，如果需要其他控制模式，请更改常量值。

现在开始编辑不需要的（EtherCAT）代码。要进行所需的编辑，请打开轴 'PRG_STM57XXB_axis' 的程序，找到下面提到的



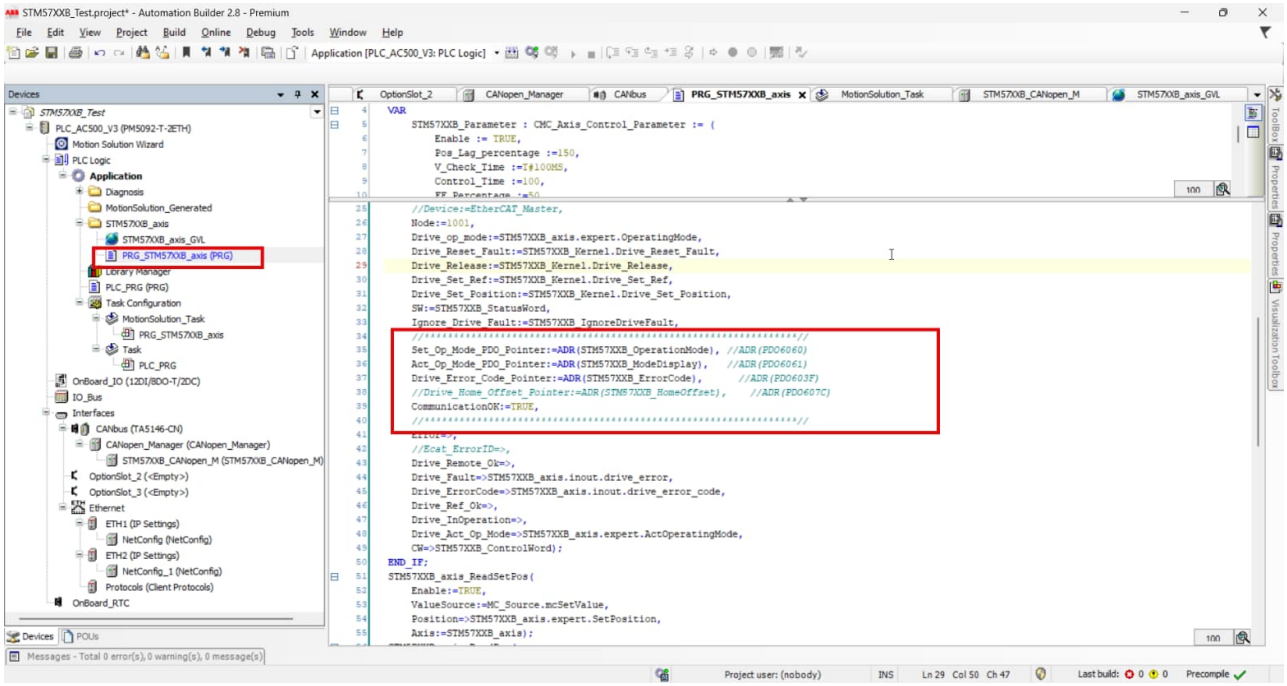
行，并在 它们前面添加 '' 或// 来 '注释掉' 不需要的代码，如下所示：

```
STM57XXB_Parameter();
//IF EtherCAT_Master.xConfigFinished THEN
    xRunCode :=TRUE;
//END_IF
IF xRunCode THEN
STM57XXB_Ecat(
    En:=TRUE,
    //Device:=EtherCAT_Master,
    Node:=1001,
    Drive_op_mode:=STM57XXB_axis.expert.OperatingMode,
    Drive_Reset_Fault:=STM57XXB_Kernel.Drive_Reset_Fault,
    Drive_Release:=STM57XXB_Kernel.Drive_Release,
    Drive_Set_Ref:=STM57XXB_Kernel.Drive_Set_Ref,
    Drive_Set_Position:=STM57XXB_Kernel.Drive_Set_Position,
    SW:=STM57XXB_StatusWord,
    Ignore_Drive_Fault:=STM57XXB_IgnoreDriveFault,
    Error=>,
    //Ecat_ErrorID=>,
    Drive_Remote_Ok=>,
//STM57XXB_axis.expert.ActualTorque :=STM57XXB_ActualTorque;

IF NOT xFirstScan AND xRunCode THEN
    STM57XXB_axis.parameter.paraMaxVelocityAppl :=6000;
    STM57XXB_axis.parameter.paraMaxAccelerationAppl :=10000;
    STM57XXB_axis.parameter.paraMaxDecelerationAppl :=10000;
    xFirstScan :=TRUE;
END_IF;
// STM57XXB_TouchProbeFunction :=STM57XXB_axis.expert.TouchProbeFunction;
// STM57XXB_axis.expert.TouchProbeStatus :=STM57XXB_TouchProbeStatus;
// STM57XXB_axis.expert.TouchProbePositionPos1 :=STM57XXB_TouchProbePositionPos1;
// STM57XXB_axis.expert.TouchProbePositionPos2 :=STM57XXB_TouchProbePositionPos2;
```

我们还必须添加一些新代码来处理驱动器 PDO 数据和功能块之间的数据交换。要进行所需的编辑，请打开轴 'PRG_STM57XXB_axis' 的程序，并在第 35 行到第 39 行的 'STM57XXB_Ecat（' 的声明下添加下面提到的行：

- Set_Op_Mode_PDO_Pointer：=ADR（STM57XXB_OperationMode），//ADR（PDO6060）
- Act_Op_Mode_PDO_Pointer：=ADR（STM57XXB_ModeDisplay），//ADR（PDO6061）
- Drive_Error_Code_Pointer：=ADR（STM57XXB_ErrorCode），//ADR（PDO603F）
- communicationOK：=TRUE，



这些行将我们创建的新变量连接到我们之前创建的 'Common_CiA402_Control_App' 实例中的相关输入。

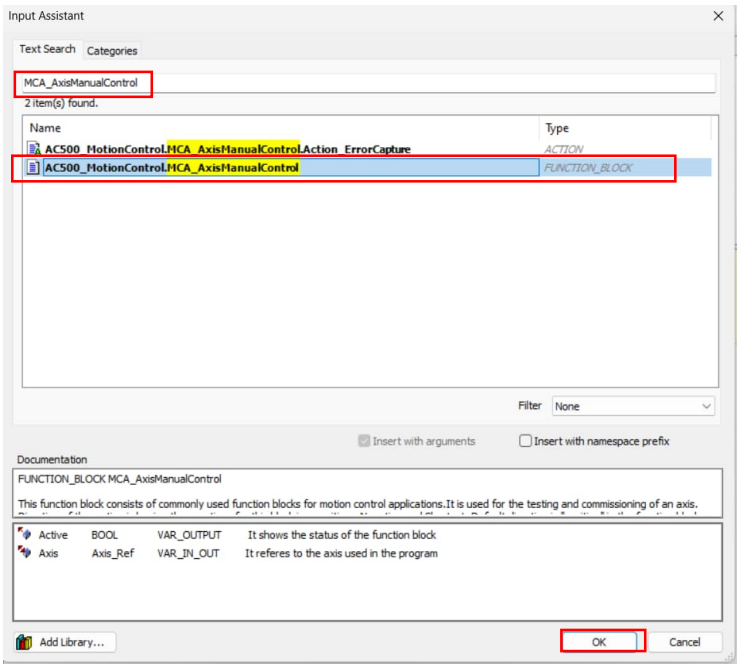
如果需要任何其他轴，则每次都可以使用新的驱动器实例名称重复上述步骤。现在，所有编辑都已完成，代码已准备好用于单个轴。

4. 交付使用

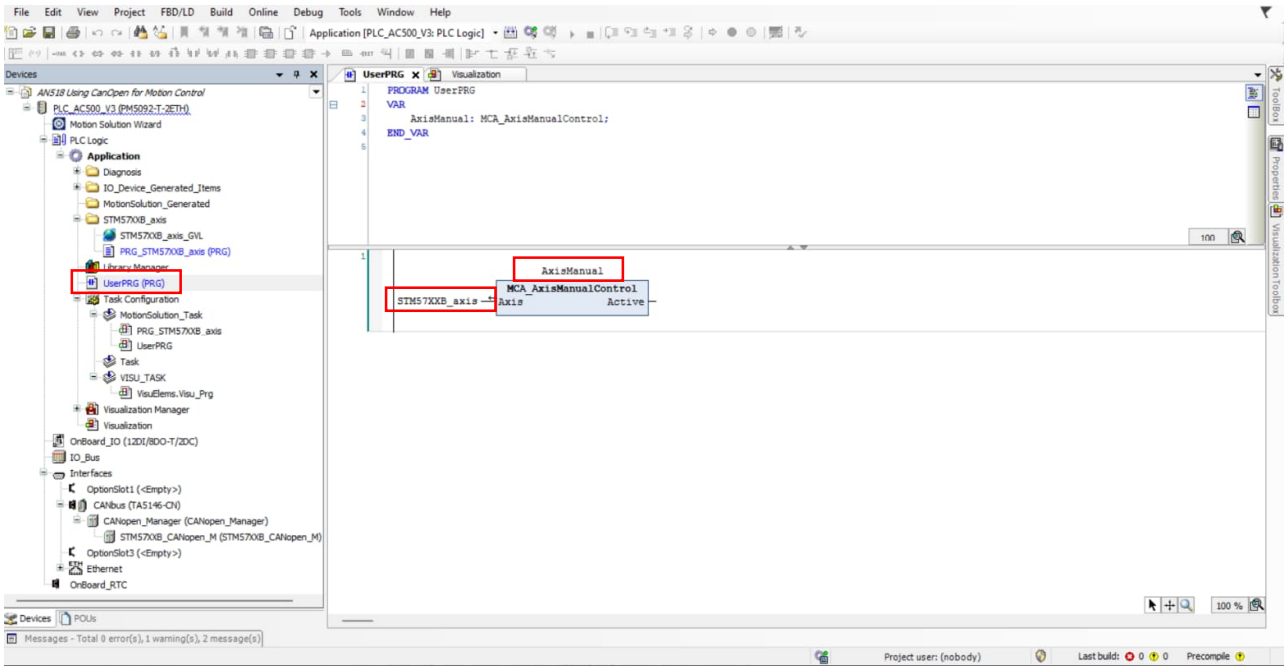
现在我们可以创建一个测试程序来检查应用程序是否正常工作，因为前面的步骤已经允许我们连接内核和硬件，我们可以使用大多数 MC 或 MCA 功能块来控制我们设置的轴（处理 EtherCAT 的轴除外）。在这种情况下，为了进行快速简单的测试，我们将使用功能块 'MCA_AxisManualControl'，它包含一个使能和点动轴的方法，并结合了一个简单的可视化界面，所有这些都在一次简单的实现中完成。

4.1 创建 Test 程序

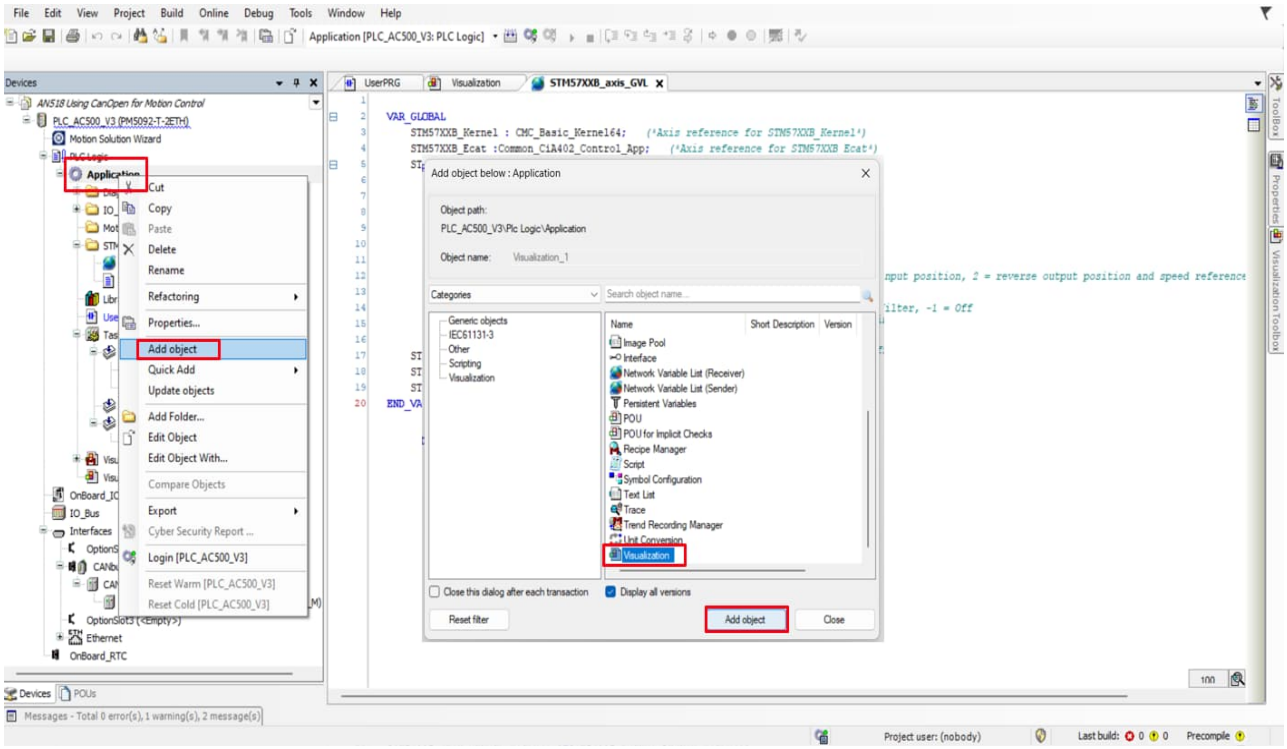
添加或创建一个 PRG，我们将使用它来运行轴程序。在这种情况下，PRG 称为 'UserPRG'。接下来，将“New Box”插入网络，选择“Test Search”并输入“MCA_AxisManualControl”，在列表中找到，然后按“确定”



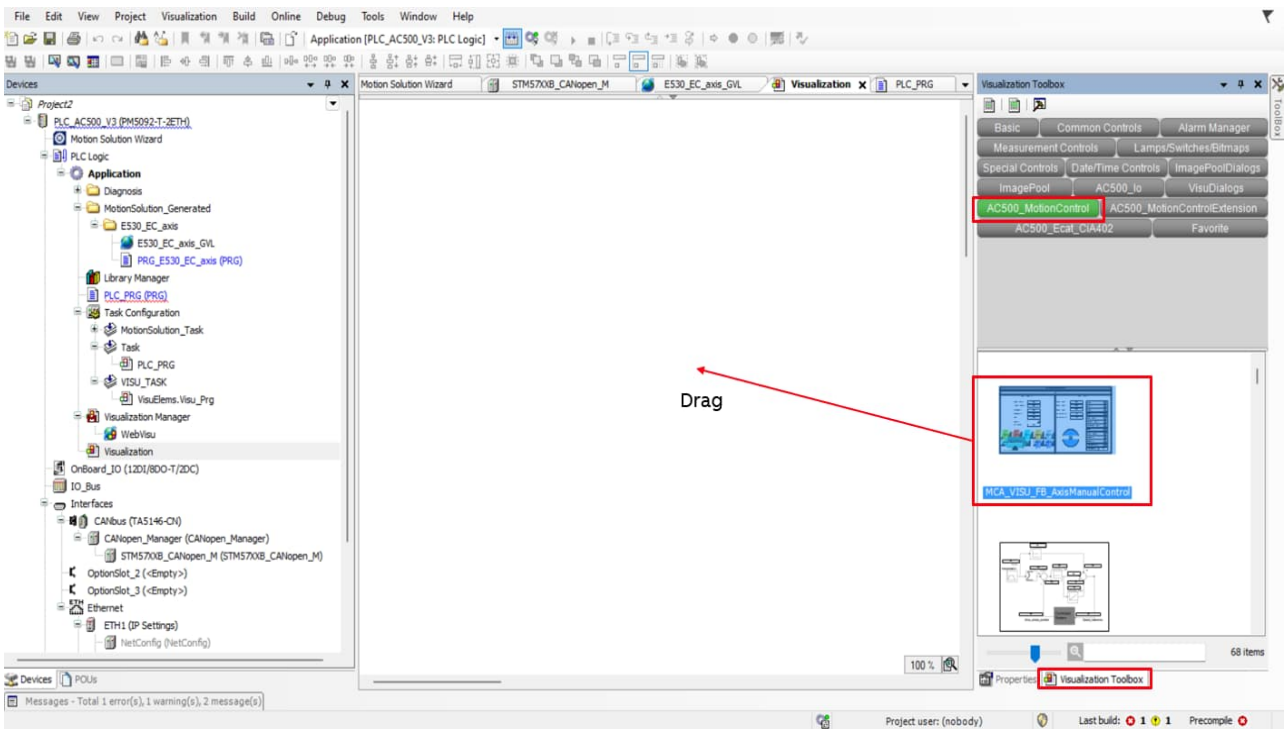
现在为其指定实例名称 'AxisManual'。最后，将输入 'Axis' 连接到 CANopen 的轴参考轴。这可以在轴 'STM57XXB_axis_GVL' 的全局变量声明中找到



现在，我们必须添加可视化效果来运行测试。右键单击应用程序、添加对象，从列表中选择“可视化”，然后从之后出现的任何弹出窗口中选择“添加对象”和“添加”。

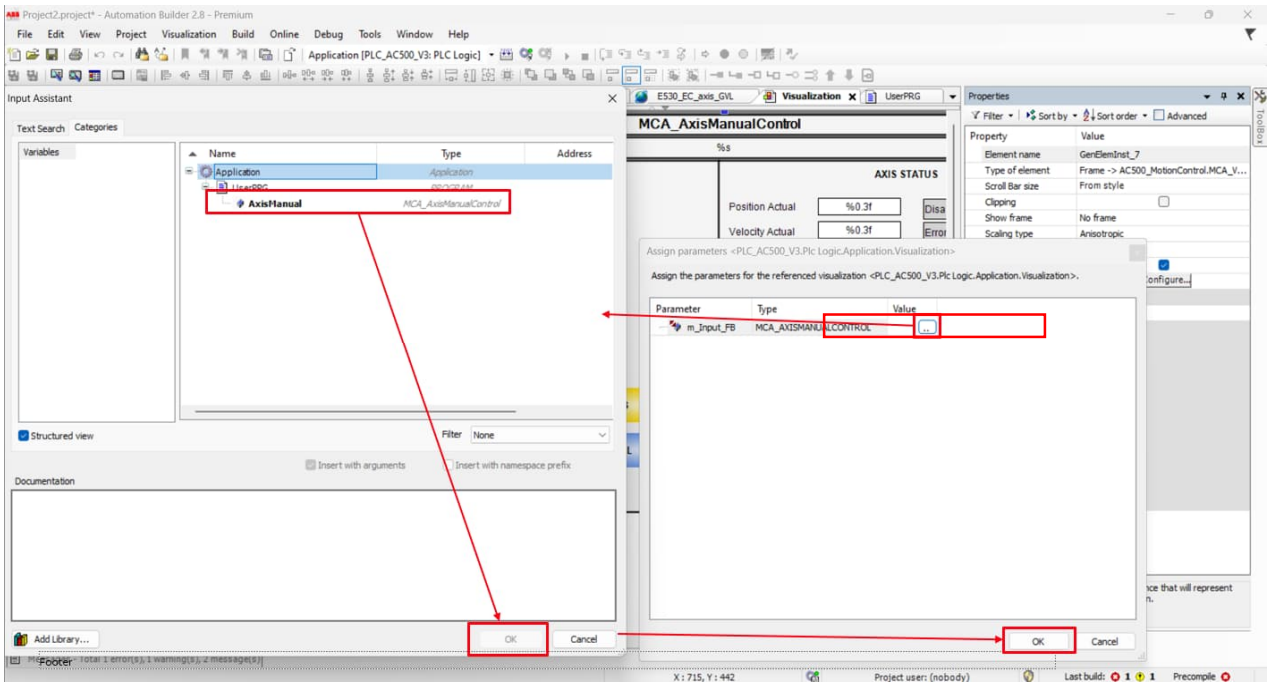


接下来，我们必须添加所需的 Visualization 实例。打开 Visualization（可视化），选择 Visualization Toolbox（可视化工具箱），AC500_MotionControl 选项卡，找到我们需要的 Visualization（可视化），在本例中为“MCA_VISU_FB_AxisManualControl”，然后我们可以将其拖到工作区中。

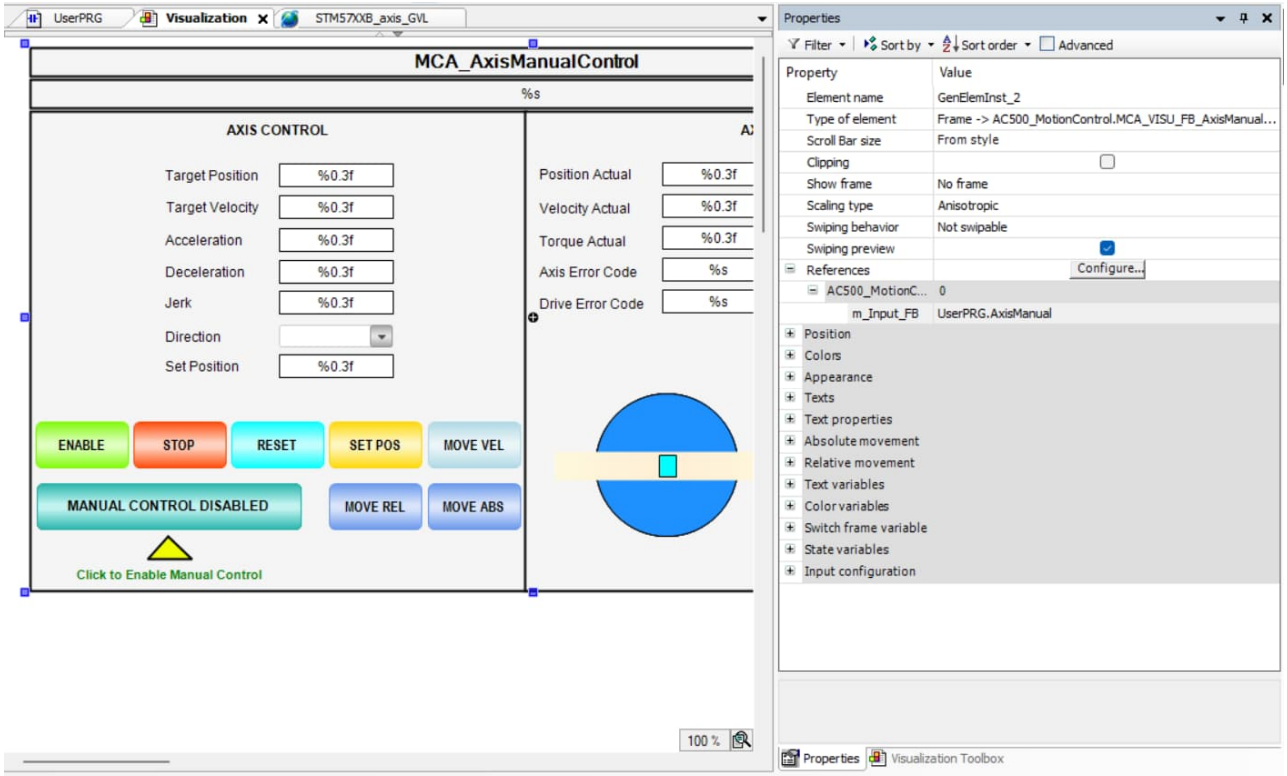


这将自动生成一个弹出窗口，提示用户将可视化实例连接到功能块的实例。

要添加连接，首先单击“...”，然后在“UserPRG （PRG）”中添加此实例时，用户应在此处查找该实例，然后单击“确定”，然后单击“确定”以关闭此过程的这一部分。



您可以看到，这些步骤已在 Visualization 中配置了引用。



现在测试程序已完成并可以使用了

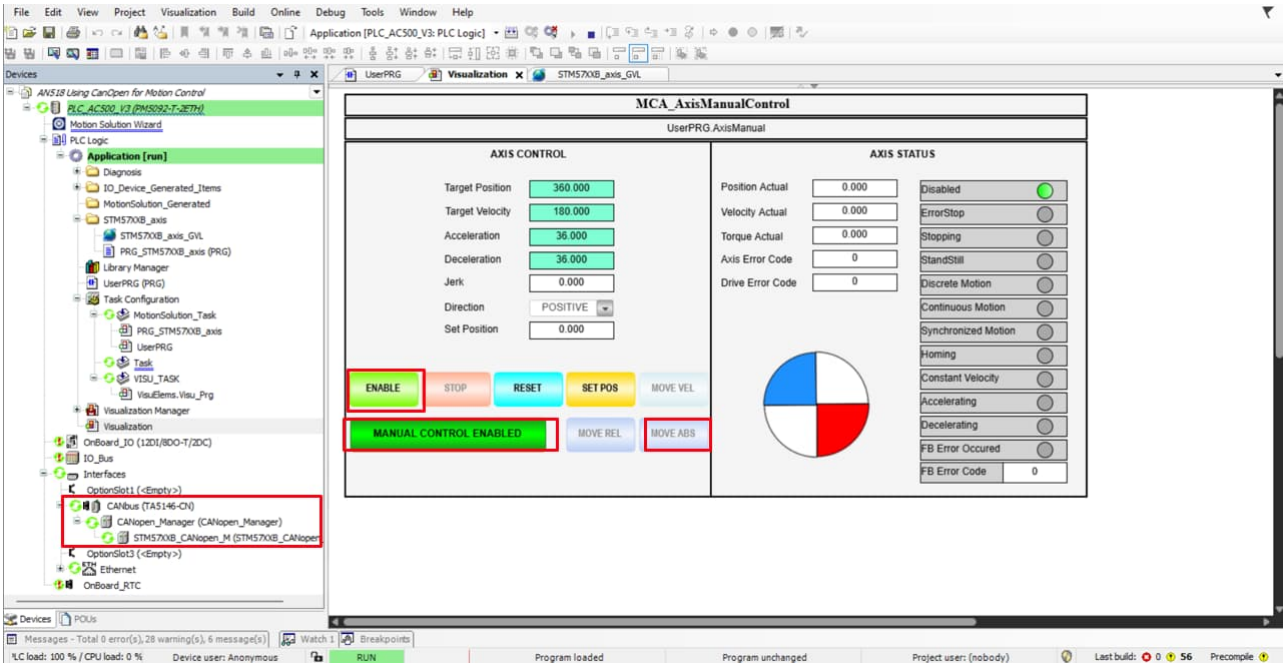
4.2 测试解决方案

现在，只要没有下载错误，就可以通过 Build > Generate Code 编译代码。要 关闭 Online > Login（使用用户首选方法），然后选择 'Create Boot Project'，然后将程序投入运行。

程序成功运行后，RUN LED 应亮起为绿色，“TA5146-CN”上的“CAN”LED 应亮起“橙色”，这表明硬件和程序运行成功。

此外，设备树中还内置了硬件诊断，用户可以在其中看到绿色圆圈（ ）来显示特定分支的运行状况，如下图所示。

如果不存在错误， 则用户可以自由地“启用手动控制”，启用驱动器并输入所需的移动配置文件参数并开始发布移动。
与此示例匹配的程序与本文档一起提供。



联系我们。

有关更多信息，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