运动控制产品

应用说明
通用驱动器接口：通过 Profinet 使用西门子 S7™ 系列 PLC

AN00263

B 版中文

编写好的 PLC 功能块与预先编写的 Mint 应用程序结合使用，

通过 PROFINET IO 轻松控制 MicroFlex e190 和 MotiFlex e180 驱动器

引言

本应用说明详细介绍了如何使用 TIA Portal（版本 12 或更高版本）项目和任何使用 FW 3.3.12 或更高版本的 SIMATIC S7 CPU（S7-1200, S7-300, S7-400）导入和配置 ABB 通用驱动器接口（GDI）和相关的 TIA 博图库“ABB Motion GDI Library”。相同的原则可应用于较旧的 Simatic Step 7 项目但固件版本必须高于 V3.xx。该库提供预先编写的数据结构和功能块，可与基于 Mint 的 GDI 无缝集成，允许西门子 PLC 控制运行支持 PROFINET IO 的 Mint 程序的 ABB 驱动器（MicroFlex e190 和 MotiFlex e180）。注：必须为 MicroFlex e190 和 MotiFlex e180 驱动器提供可编程 Mint 存储卡（选件代码+ N8020）。

本说明可以提高所有项目的一致性，并大大简化西门子 PLC 简单点到点运动控制应用的开发。

对读者要求：需具备西门子 PLC、SIMATIC S7、PROFINET IO 配置、Mint Workbench 和 Mint GDI 的基本工作知识。建议读者参考文档 AN00204，以了解有关 Mint GDI 操作和配置的详细信息。

前提条件

软硬件要求：

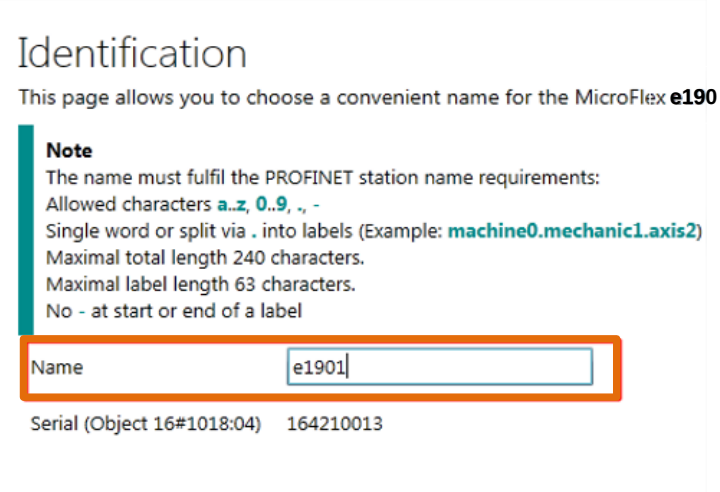
- Mint Workbench 5852 或更高版本（有关最新下载和支持信息，请参阅 www.abb.com/motion 或 new.abb.com/drives/low-voltage-ac/motion）
- ABB 伺服驱动器（如 MotiFlex e180 或 MicroFlex e190 驱动器），固件版本 5900 或更新，（仅需要驱动器，不需要电机来测试与 PLC 的数据交换）
- 最新版的 Mint GDI 程序（2.26 或更高版本）
- 运行 TIA portal 12 版本（或更高版本）的 PC 或笔记本电脑
- 使用 FW 3.3.12 或更高版本的 SIMATIC S7 CPU（S7-1200，S7-300，S7-400）
- 四端口（或更多）以太网交换机
- 连接 PROFINET 连接器、PLC 处理器、 ABB 运动驱动器和 PC 到以太网交换机的以太网电缆

假设读者具备西门子 TIA Portal、ABB 运动驱动器和 Mint Workbench 的基本工作知识。在本文档中，凡是提及 Microflex e190 的，都可用 Motiflex e180 替代。

配置设备（站）名

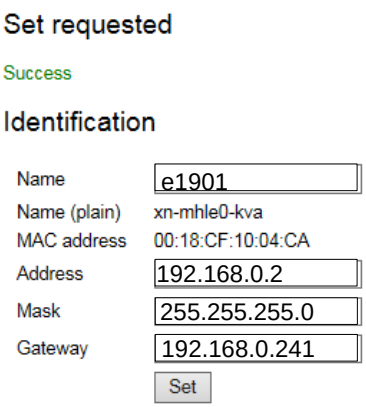
PROFINET 站名与用户指定的设备名相同。可用以下方法设置：

使用 Workbench 中的配置向导的 Identification 页面（这是推荐的最简单的方法）。页面包含详细的编码信息，站名称需满足命名规则。配置向导不支持域名代码：



使用驱动器的 PROFINET 网页（可通过 Mint Sidebar 访问）。网页允许用户输入包含 Unicode 字符（比如ü），支持 Punycode 编码。但是，建议避免使用 Unicode 字符，因为只有本网页能正确显示名称-所有其它屏幕将显示“普通”名称...

驱动器网页上编辑完成后按 Set 按钮来设置驱动器的名称；



站名存储在驱动器的对象字典中（即，非易失性存储）。在设置新值后，驱动器执行一次内部重启
注意： 站名也可以由 TIA Portal 分配，在本文档中不予介绍。

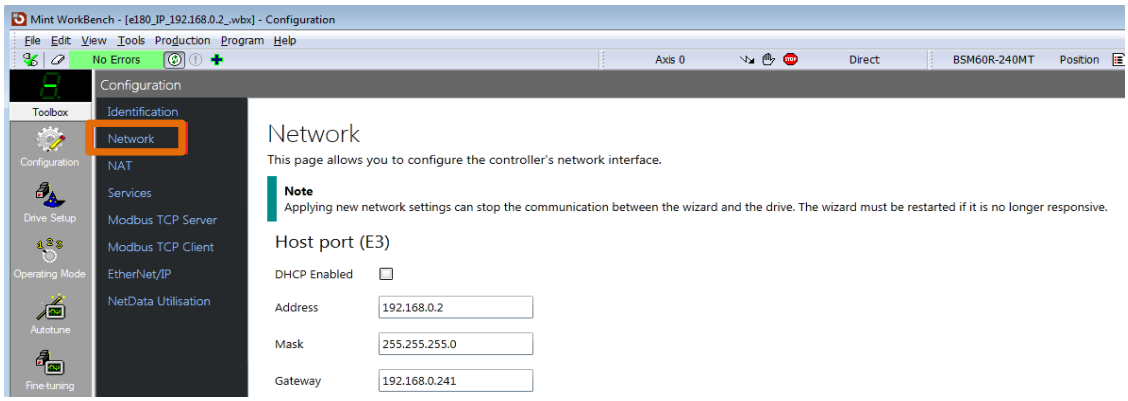
配置驱动器的 IP 地址

如果主设备支持，可在起动过程中分配驱动器的 IP 地址（通过动态主机配置协议-DHCP）。但是，IP 地址的变更可能与通过 Workbench 做出的配置相冲突，因此不建议使用 DHCP。

多数 PROFINET 主设备允许用户保持设备上现有的 IP 配置。这是首选的配置方法，尤其是因为设备的前端口支持多种协议时（比如，PROFINET、Modbus TCP、ICM/Workbench 连接）。本示例中，我们将更改驱动器 IP 地址以避免与 PLC 的 IP 地址冲突。

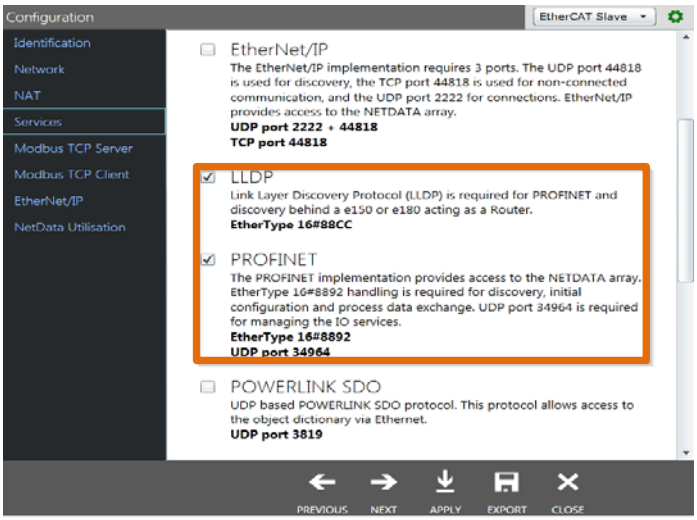
设置驱动器 IP 地址的两种方法：

- 在 Workbench 中，Configuration |Network，为主机端口（E3）输入所需的 IP 地址，网络掩码（通常为 255.255.255.0）和网关地址（只有在 PLC 和驱动器之间有路由器时使用...通常不存在这种情况，因此，网关地址应设置为同一子网内未被使用的 IP 地址）...
- 或使用驱动器的 PROFINET 网页（可通过 Mint Sidebar 应用对话框内的链接访问）



启用 PROFINET 服务

因网络安全原因，可能会禁用 PROFINET。PROFINET 联网前，在 Workbench 中通过 Configuration | Services 配置...



要使 PROFINET 正常运行，LLDP 和 PROFINET 都需要选择。PROFINET 的执行机制没有提供通过 PROFINET 执行闭环轴控制所需的 SYNC 对时信号。作为替代，执行机制开放驱动器的内部 NETDATA 通道（0-999），使 PLC 能够读取/写入 32 位整数或浮点值。比如，它允许通过基于 Mint 的通用驱动器接口控制轴（按照应用说明所含的例子）；另外，用户也可创建自己的 Mint 应用程序，并使用 PROFINET 来交换命令/状态/参数数据。

可在 EtherCAT 或 Ethernet Powerlink（EPL）网络上使用驱动器（使用驱动器顶部的两个实时以太网端口），并同时连接到 PROFINET 主设备。

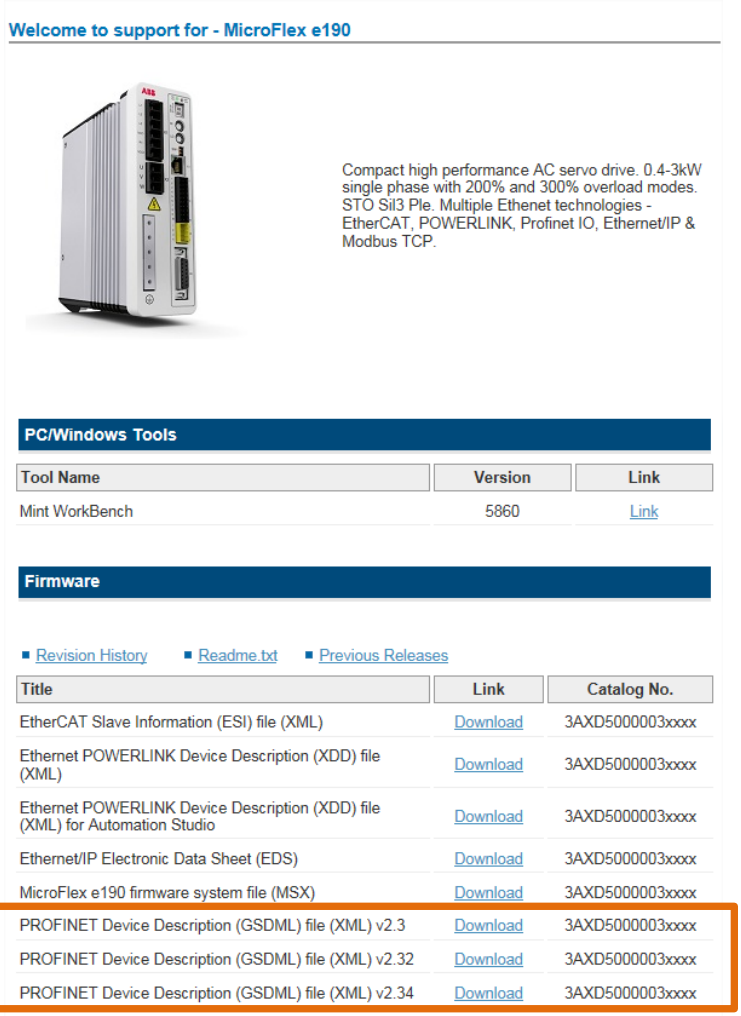
配置通用驱动器接口（GDI）Mint 程序

预先编写的 GDI Mint 程序仅需要少量修改即可适应用户的应用（例如轴比例因子，零点开关和寻零方式的定义）。详细信息，请参阅应用说明 AN00204。

获取 GSDML 文件

PROFINET 主设备需要获取 GSDML 文件导入其“设备库”。可通过两种不同的方法获取 GSDML 文件：

- 从“ABB Motion”网站下载（http://www.abbmotion.com/support 进入“Support by Product”，选择 MotiFlex e180 或 MicroFlex e190）。



Welcome to support for - MicroFlex e190

Compact high performance AC servo drive. 0.4-3kW single phase with 200% and 300% overload modes. STO SIl3 Ple. Multiple Ethenet technologies - EtherCAT, POWERLINK, Profinet IO, Ethernet/IP & Modbus TCP.

PC/Windows Tools

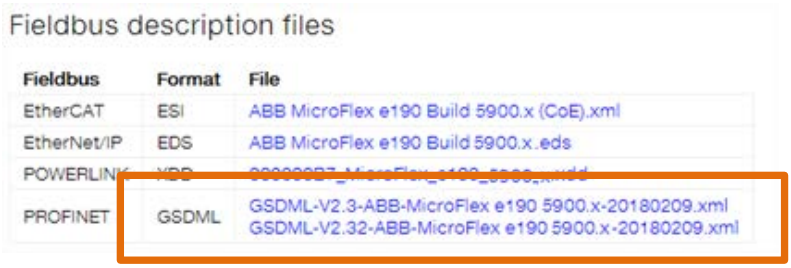
Tool Name	Version	Link
Mint WorkBench	5860	Link

Firmware

Revision History | Readme.txt | Previous Releases

Title	Link	Catalog No.
EtherCAT Slave Information (ESI) file (XML)	Download	3AXD5000003xxxx
Ethernet POWERLINK Device Description (XDD) file (XML)	Download	3AXD5000003xxxx
Ethernet POWERLINK Device Description (XDD) file (XML) for Automation Studio	Download	3AXD5000003xxxx
Ethernet/IP Electronic Data Sheet (EDS)	Download	3AXD5000003xxxx
MicroFlex e190 firmware system file (MSX)	Download	3AXD5000003xxxx
PROFINET Device Description (GSDML) file (XML) v2.3	Download	3AXD5000003xxxx
PROFINET Device Description (GSDML) file (XML) v2.32	Download	3AXD5000003xxxx
PROFINET Device Description (GSDML) file (XML) v2.34	Download	3AXD5000003xxxx

- 从驱动器的主（网）页的索引页下载文件。页面包含了起动中由驱动器固件生成的所有现场总线描述文件：



Fieldbus description files

Fieldbus	Format	File
EtherCAT	ESI	ABB MicroFlex e190 Build 5900.x (CoE).xml
EtherNet/IP	EDS	ABB MicroFlex e190 Build 5900.x.eds
POWERLINK	XDD	[illegible]
PROFINET	GSDML	GSDML-V2.3-ABB-MicroFlex e190 5900.x-20180209.xml GSDML-V2.32-ABB-MicroFlex e190 5900.x-20180209.xml

文件名称包含 GSDML 描述和固件版本。使用指定固件版本发布的所有文件的文件名必须包括发布编号。

文件的使用/导入方式取决于具体的 PROFINET 主站厂商（PLC）。

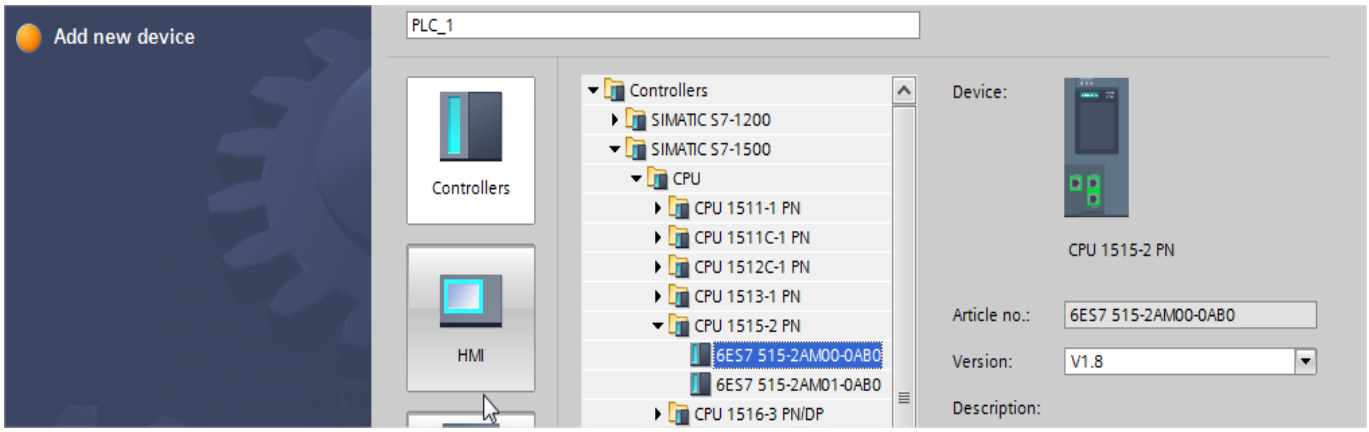
注意： ABB 运动驱动器提供两种不同的 GSDML 版本文件（v2.32 和 v2.3）。用于不同版本的西门子博图软件。下表显示了兼容性。还有一个 v2.34 GSDML 文件可用，仅用于认证测试。

Version (TIA Portal)	Version (GSD file)
TIA Portal V11	GSD file V2.25
TIA Portal V11 SP1	GSD file V2.25
TIA Portal V12	GSD file V2.3
TIA Portal V12 SP1	GSD file V2.31
TIA Portal V13	GSD file V2.31
TIA Portal V13 SP1	GSD file V2.31
TIA Portal V14	GSD file V2.32
TIA Portal V14 SP1	GSD file V2.32
TIA Portal V15	GSD file V2.32

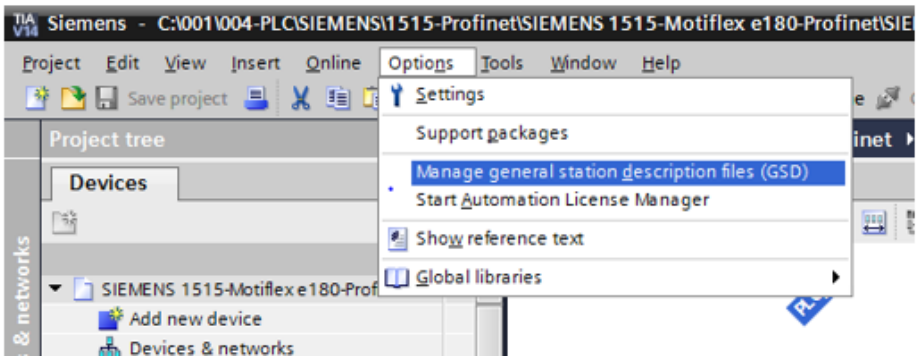
至此，驱动器配置所必需的步骤全部完成。从这里开始，下文都是关于 PROFINET 主站配置的主题（比如 PLC）。

配置并将驱动器的 GDS 文件导入 TIA Portal 项目

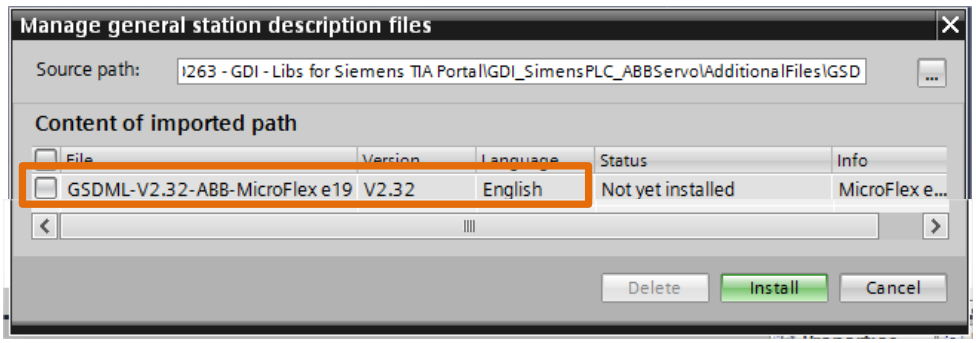
首先，打开 TIA portal 软件并创建一个新项目。接下来我们应该转到“Configure a device”视图。选择‘Add a new device’，并选择匹配的 CPU 类型（在示例中为 CPU 1515-2 PN）。这将是配置中的基础对象。



要使用任何 ABB Motion 驱动器，必须首先将 GSDML 文件导入到项目中。要将文件导入 TIA portal 的设备库（存储所有通用站描述（GSD）文件的地方），转到工具栏，选择“选项”菜单并选择“管理通用站描述文件（GSD）”



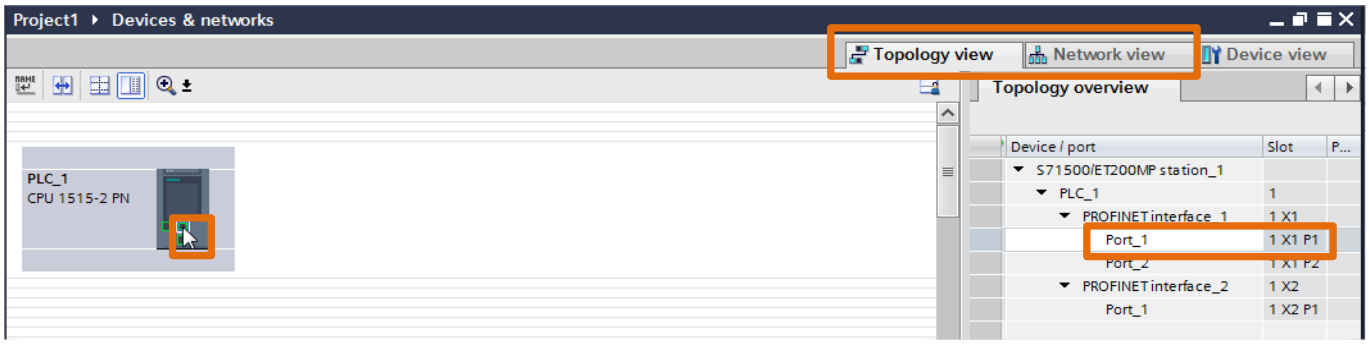
接下来，浏览到已保存 GSDML 文件的 PC 上的位置，选中并单击“安装”。



安装完成后，在“硬件目录”中应出现驱动器的 GSD 文件。

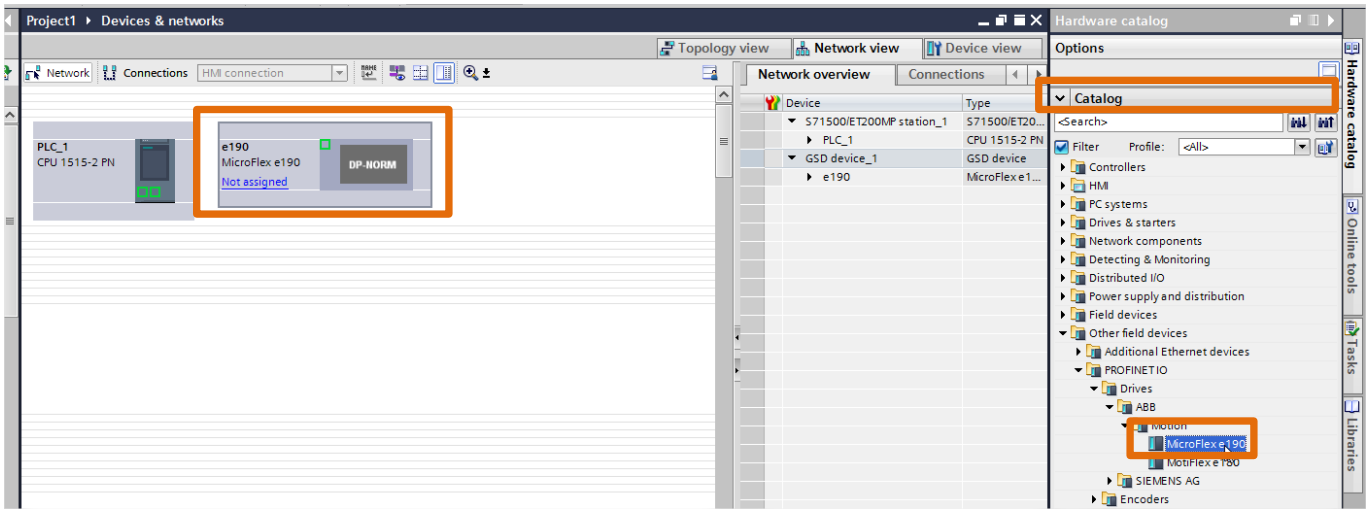
注意： *该文件的名称将采用“Gsdml-v[version]-abb-[drive type] [FW Version] [Date created].xml”的形式*

接下来，可通过顶部的选项卡选择“拓扑视图”或“网络视图”，这将在中央窗格中显示相应视图。选中“网络视图”，可以选择 PROFINET 电缆将连接的端口。在本例中，我们将选择 Port_1。选择端口后，可以在屏幕底部的属性窗口中配置地址。在本例中，将 PLC 地址设置为默认值“192.168.0.1”。

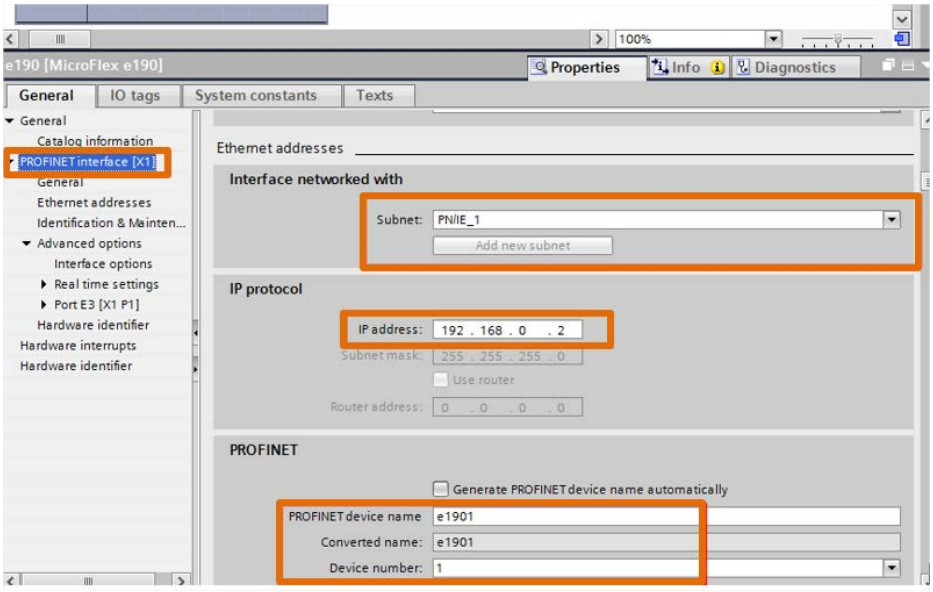


注意： 在继续操作之前，为避免 IP 冲突，请确保已更改驱动器 IP 地址。

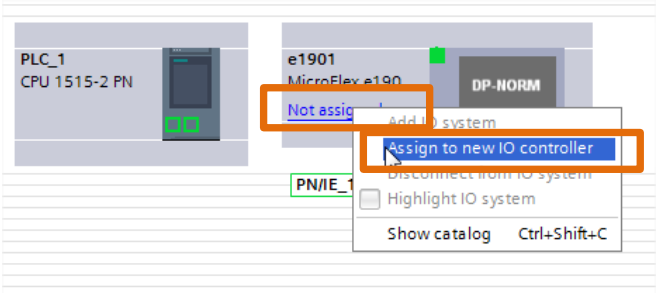
接下来，在主窗口中选择“网络视图”。然后，通过“硬件目录”视图（从窗口右侧选择），您必须从硬件目录中的相关文件中导入设备。对于 ABB 运动控制驱动器，您必须转到“*其他现场设备> PROFINET IO> 驱动器> ABB> 运动> MicroFlex e190*”。然后，驱动器将作为“未分配的设备”出现在主窗口中。



双击新添加的 MicroFlex e190 节点，TIA portal 将自动切换到驱动器的设备视图。在设备视图中可配置设备名称和 IP 地址。选择“常规”菜单，然后选择“PROFINET 接口[X1]”。要在此处输入设置，请按“添加新子网”按钮。在此处输入的设置应与之前在 Mint Workbench 中输入的 IP 地址和设备名称相匹配。还应确保选择正确的子网（在多个网络的情况下），并选择正确的设备编号（如果是多个驱动器）。

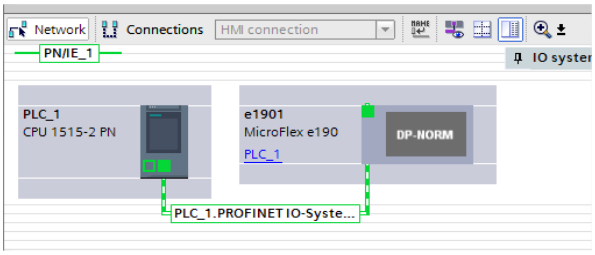


现在已经配置了设备，返回“网络视图”。现在，e190 将显示为“未分配 IO 控制器”的节点。我们必须在 e190 节点符号上选择“未分配”区域。这将为我们提供“分配给新 IO 控制器”的选项。



然后，我们必须选择正确的 IO 控制器，它将是 PLC 的 PROFINET 接口

然后，我们可看到 e190 被指定由 PLC 控制（以下名称为 PLC_1）

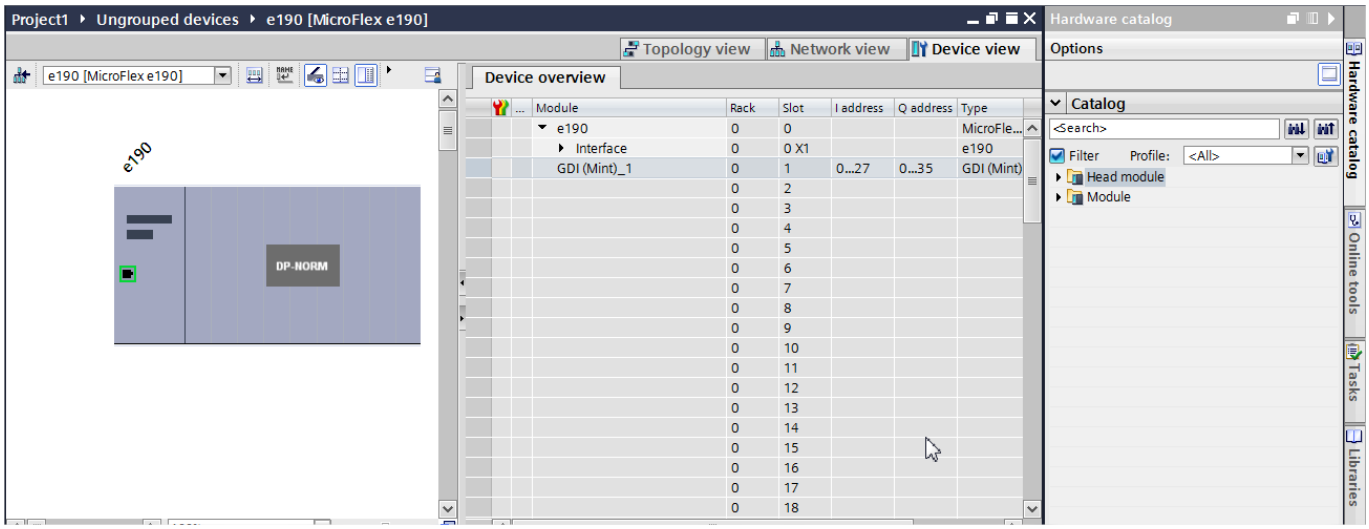


在 PLC 中配置 IO 映射

现在设备配置已经完成，我们现在可以考虑数据映射，以允许它与控制器交换数据。从固件版本 5900 开始，在下载 GSDML 文件时自动创建与 GDI 程序的正确映射（有关更多信息，请参阅标题“使用自动 GDI 映射方法配置 IO 映射”）。如果用户愿意，也需要删除此自动映射并根据自己的要求定制映射（有关详细信息，请参阅“使用手动方法配置 IO 映射”）。

使用自动 GDI 映射方法配置 IO 映射

导入项目后，自动方法不需要进一步的映射操作。如您所见，设备映射中有一个名为“GDI（Mint）_1”的“对象”。该对象包含与 Mint GDI 程序中的标准 GDI 映射通信所需的所有数据；



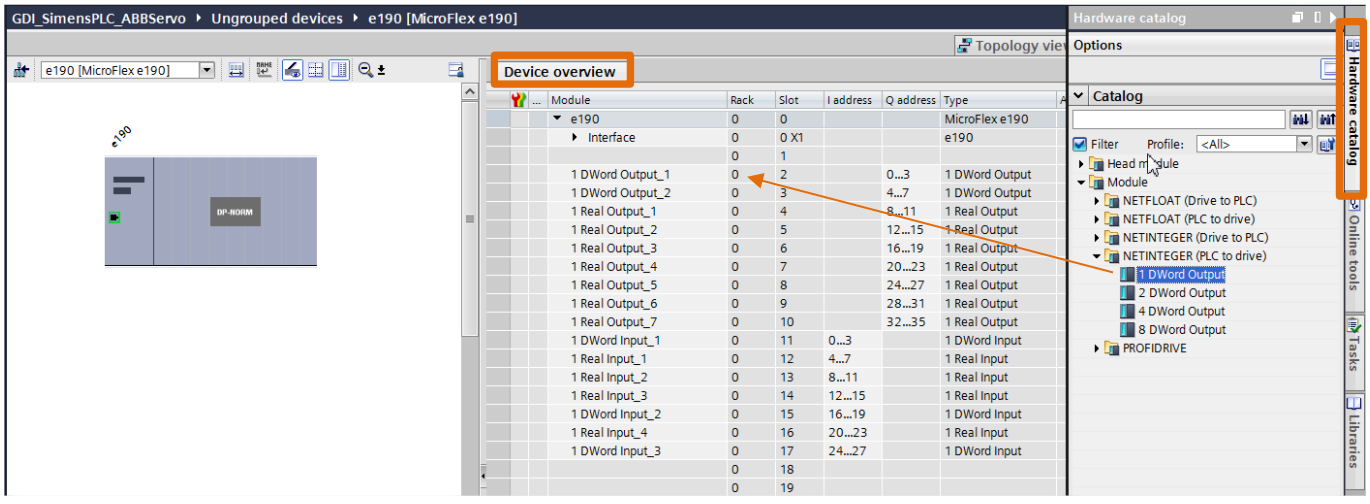
注意：显示的默认 IO 映射区域与所有输入数据地址 0 到 27 和输出数据地址 0 到 35 相关联。

现在可以跳到标题“将配置下载到 PLC”。

手动配置 IO 映射

使用此方法，我们必须手动添加应用程序所需的地址映射。可以调整所需映射的地址，以提高网络的通讯速度或增加可连接的驱动器的数量。在本例中，我们假设需要映射所有 GDI 地址。

欲手动将所需的 IO 映射到驱动器，要选择 e190 节点并转到“设备概览”，然后删除项目“GDI（Mint）_1”（这是默认映射）。然后，用户可以从右侧的“硬件目录”窗格添加项目。这里将展示了可以在“模块”文件夹中选择的所有数据类型：“PLC 到驱动器”和“驱动器到 PLC”两个方向上的“NETINTEGER”和“NETFLOAT”。然后，我们可以按正确的顺序将它们拖到设备概览中。



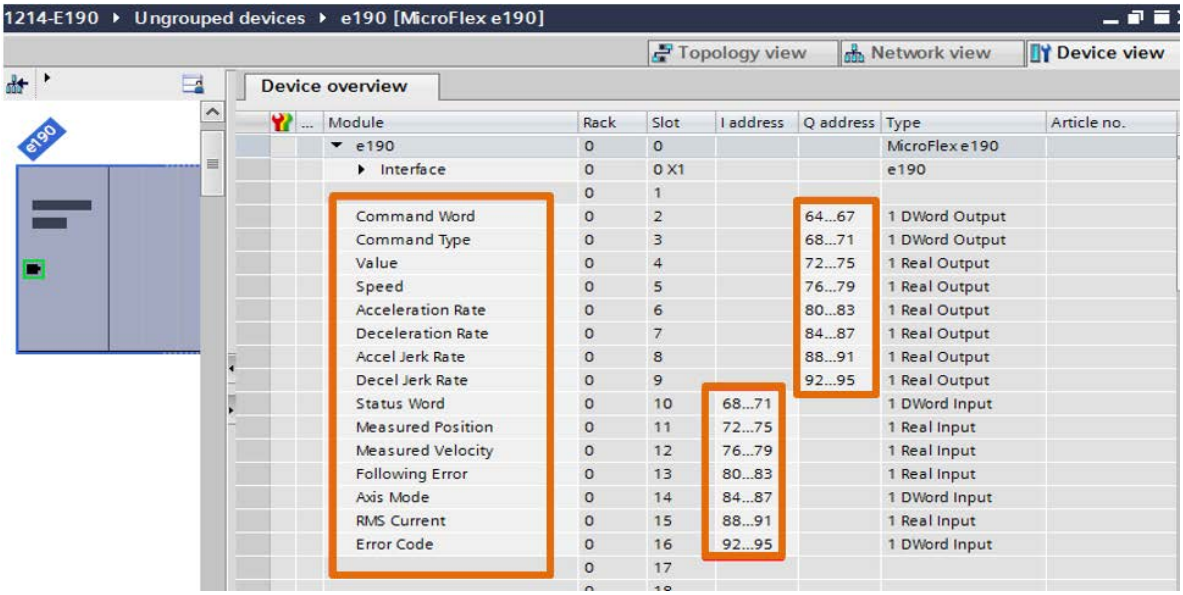
GDI 所需的所有标准地址顺序如下表所示；

描述	Mint Netdata 通道	Mint NetData 类型	PLC 地址 DoubleWord	PLC 数据类型	要选择的 IO 选项
Command word	0	NetInteger 0	%QD[xx]	十进制	PLC 到驱动器（1 个 DWord）
Command type	1	NetInteger 1	%QD[xx+4]	十进制	PLC 到驱动器（1 个 DWord）
Value	2	NetFloat 2	%QD[xx+8]	浮点	PLC 到驱动器（1 个 Real）
Speed	3	NetFloat 3	%QD[xx+12]	浮点	PLC 到驱动器（1 个 Real）
Accel	4	NetFloat 4	%QD[xx+16]	浮点	PLC 到驱动器（1 个 Real）
Decel	5	NetFloat 5	%QD[xx+20]	浮点	PLC 到驱动器（1 个 Real）
Acceljerk	6	NetFloat 6	%QD[xx+24]	浮点	PLC 到驱动器（1 个 Real）
Deceljerk	7	NetFloat 7	%QD[xx+28]	浮点	PLC 到驱动器（1 个 Real）
偏移	8	NetFloat 8	%QD[xx+32]	浮点	PLC 到驱动器（1 个 Real）
状态字	100	NetInteger 100	%QI[xx]	十进制	驱动器到 PLC（1 个 DWord）
测量位置	101	NetFloat 101	%QI[xx+4]	浮点	驱动器到 PLC（1 个 Real）
测量速度	102	NetFloat 102	%QI[xx+8]	浮点	驱动器到 PLC（1 个 Real）
跟随误差	103	NetFloat 103	%QI[xx+12]	浮点	驱动器到 PLC（1 个 Real）
Axis mode	104	NetInteger 104	%QI[xx+16]	十进制	驱动器到 PLC（1 个 DWord）
均方根电流	105	NetFloat 105	%QI[xx+20]	浮点	驱动器到 PLC（1 个 Real）
错误代码	106	NetInteger 106	%QI[xx+24]	十进制	驱动器到 PLC（1 个 DWord）

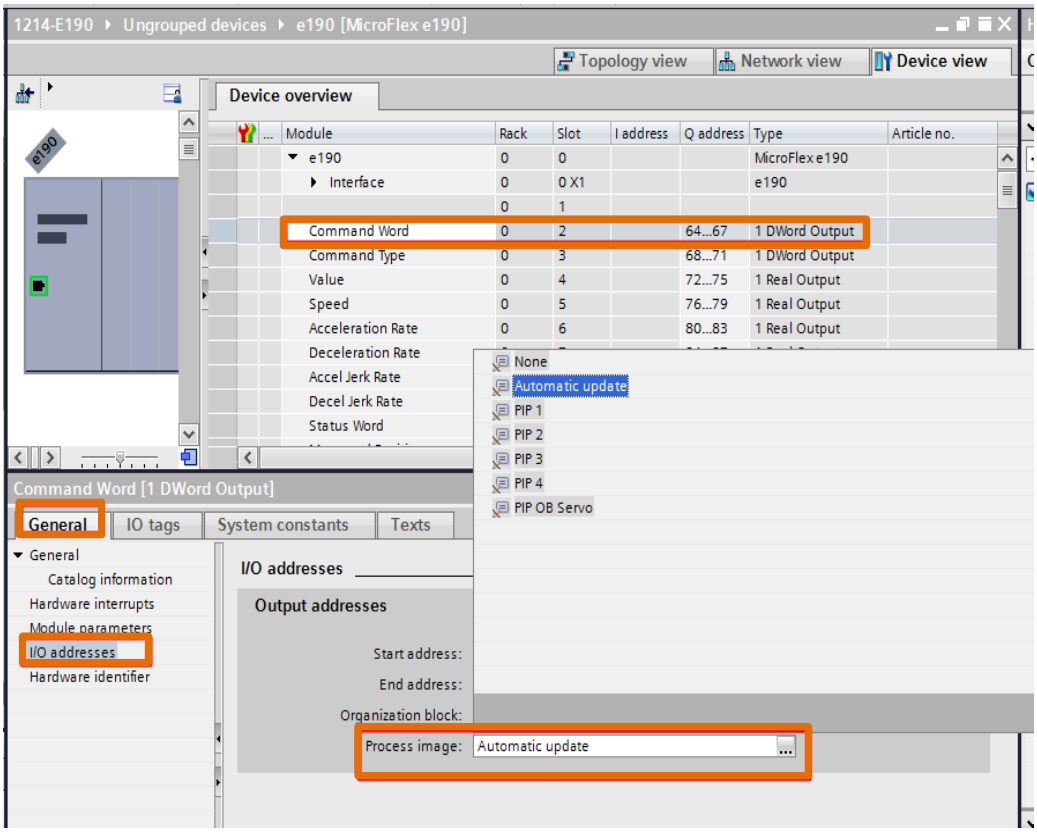
注意： 上述地址值“xx”取决于数据分配和 CPU 类型。“xx”是系统给出的第一个地址的占位符，可以以“xx”起点计算出其他地址。

注意： 我们默认只能使用%ID0-%ID128 和%QD0-%QD128 地址区域。如果这个地址范围不够，可以使用不同的区域，在 TIA portal 上有帮助信息。

我们可以根据表中的信息为所有 IO 添加描述，以便于理解。这可以在“模块”列中完成，该列显示在“设备概览”页面中“驱动器 IO 映射”的右侧。

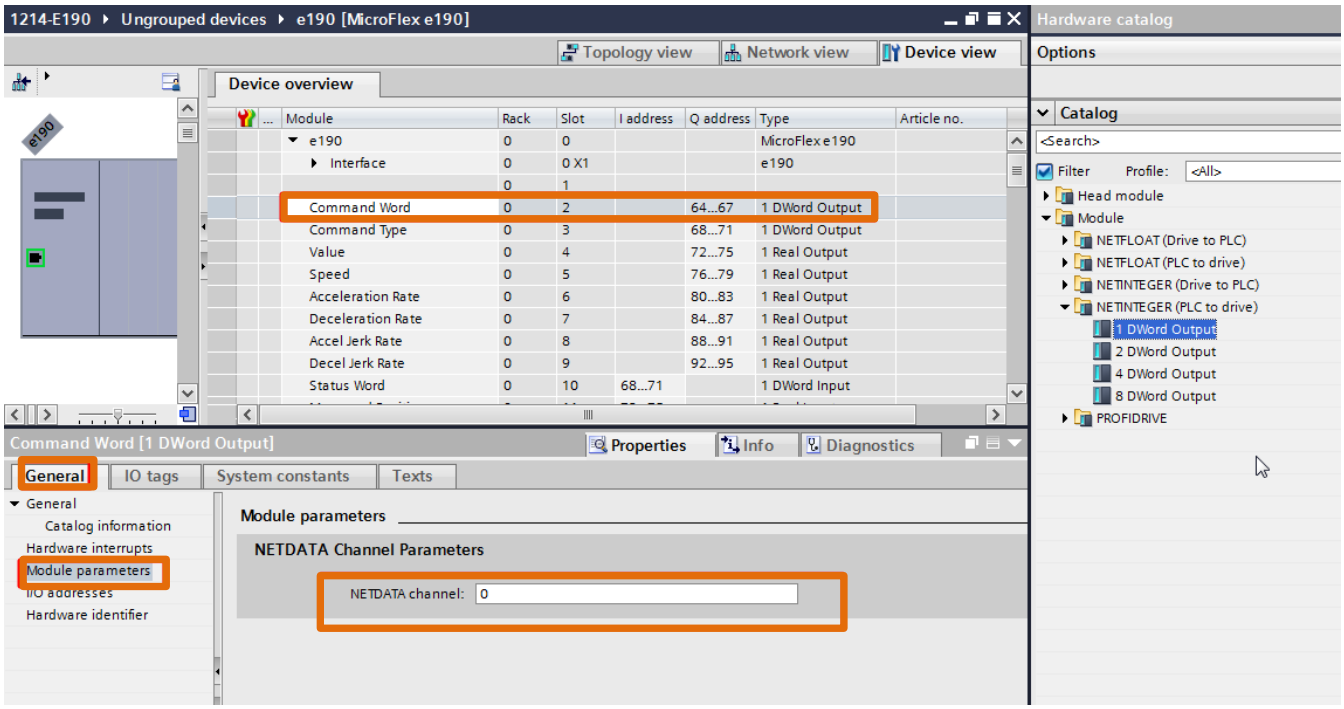


当添加了所有 IO 对象（PZD）后，我们必须确保它们配置正确。对于所有 IO，我们必须检查所有 IO 的处理映射（数据交换）被设置为“自动更新”。为此，请选择每个 PZD 的 IO 范围，然后转到“常规”>“IO 地址”>，并确保选中“自动更新”。

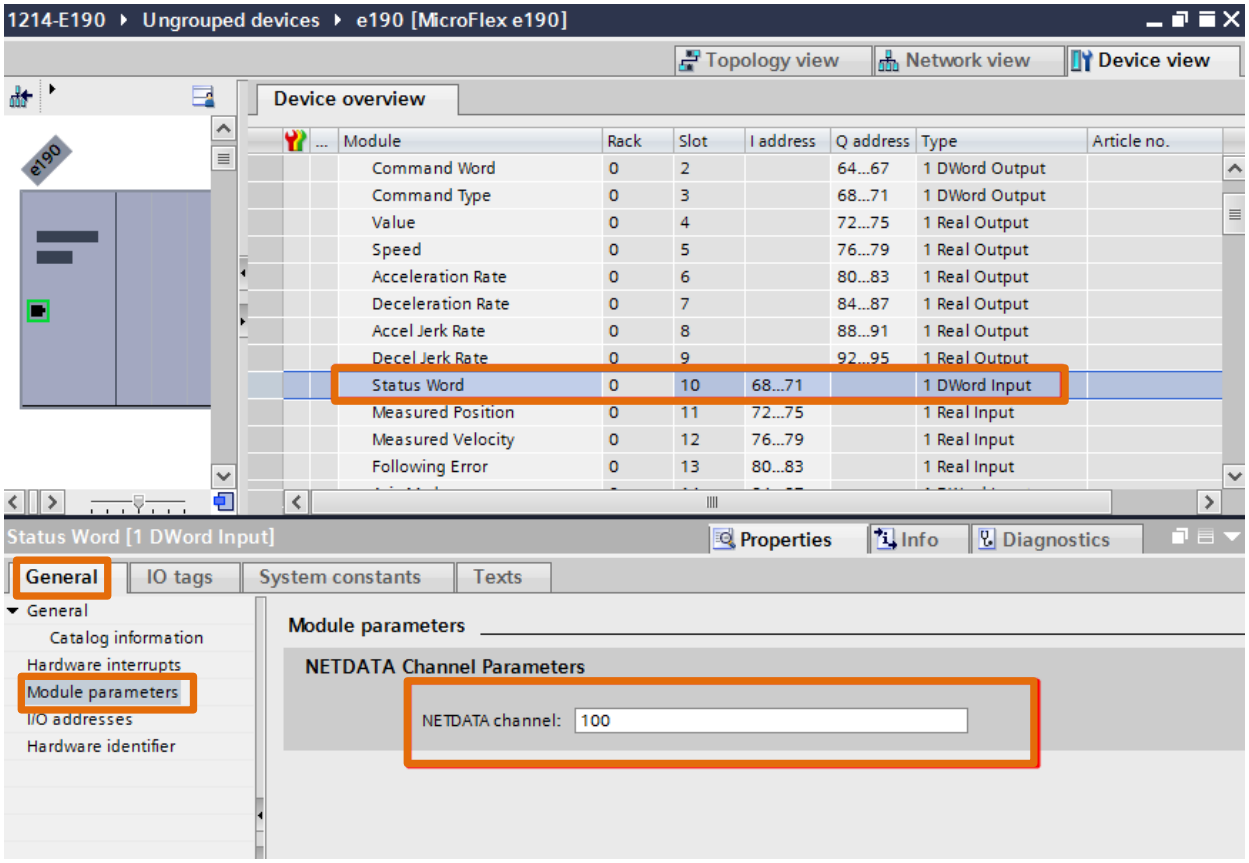


接下来，须确保 PLC 的 ID（输入）和 QD（输出）区域映射到正确的 NETDATA 通道。

对于所有输出数据，逐个选中并转到“常规”>“模块参数”，确保选择正确的 Netdata 通道（输出数据对应为 0-8）。

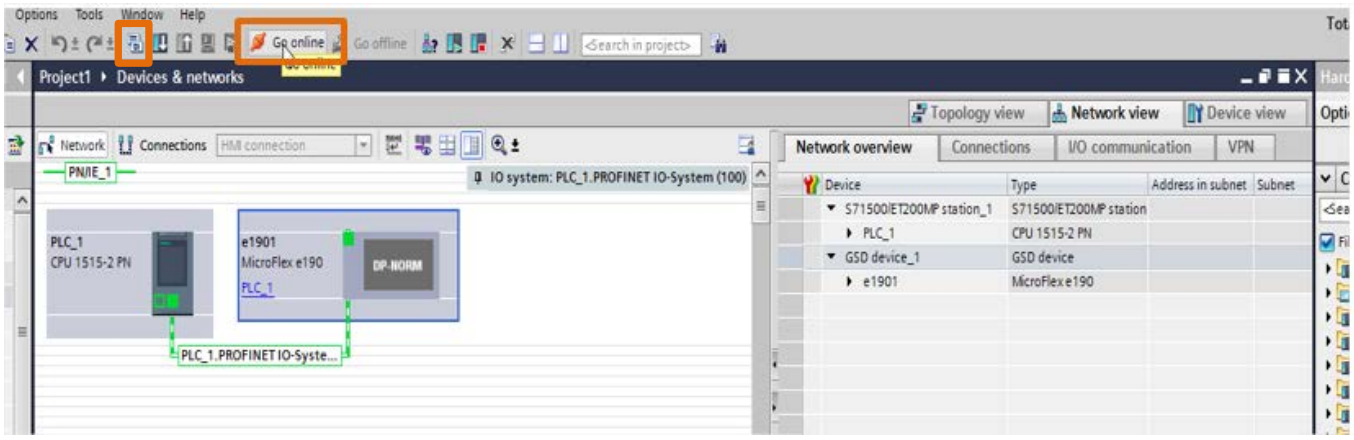


对于所有输入数据，逐个选中并转到“常规”>“模块参数”，确保选择正确的 Netdata 通道（所有输入数据对应 100-106）。

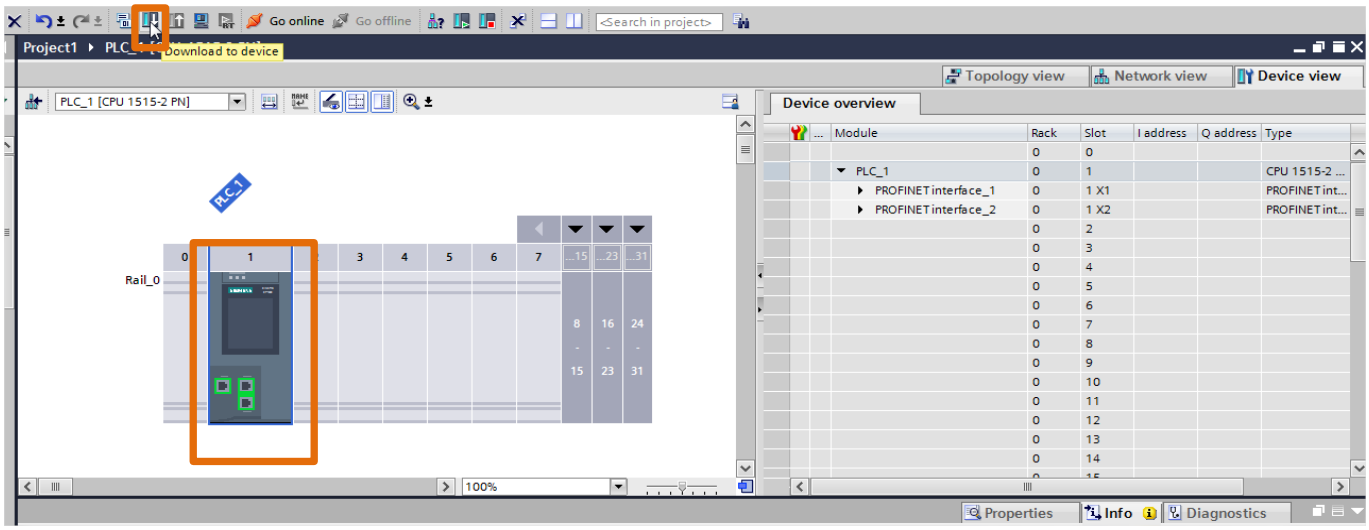


将配置下载至 PLC

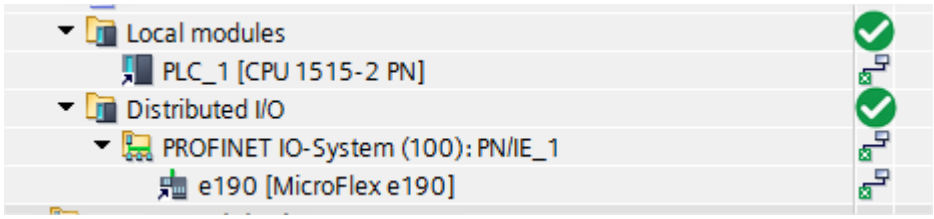
在配置完 IO 数据后，须将此配置下载到 PLC，以便检查网络运行情况。首先：用“编译”按钮编译所有硬件更改，然后“联机”到相关的 PLC。首次选择“联机”TIA portal 时，博图将弹出一个对话框，用于选择需要连接的 PLC。我们必须首先确保选择正确的网络适配器，然后我们可以扫描 PLC。TIA portal 将使用 mac 扫描功能连接 PLC。因此，请确保禁用任何防病毒软件以允许扫描。



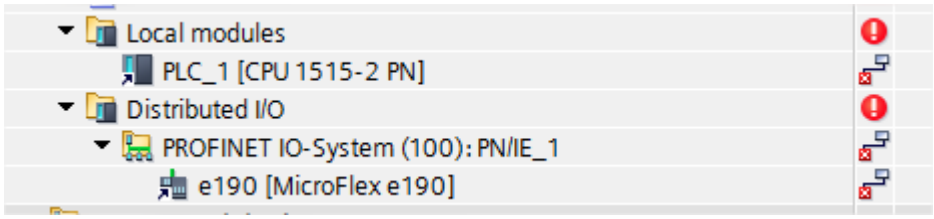
联机后，可以通过选择“下载到设备”将配置下载到 PLC，选中连接的 PLC，然后选择“加载”。



下载并联机后，可以使用左侧窗口中的“设备”视图来确定系统的状态。此视图将显示任何配置错误。PLC 和 MicroFlex e190 驱动器都有绿色勾号图标，这意味着它们已正确连接。



如果出现问题，未正确连接或配置的设备旁边会出现红叉。如果发生这种情况，请检查是否存在使用错误版本的 GSDML 文件，所选设备的地址或名称或驱动器或 PLC 的固件版本等问题。

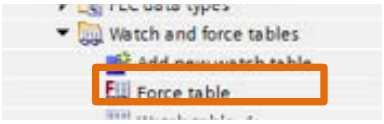


如果您遇到任何网络问题，可以使用“设备”窗口中的在线访问组来获取有关系统状态的更多信息。

下载到 PLC，在 PLC 中进行数据监视

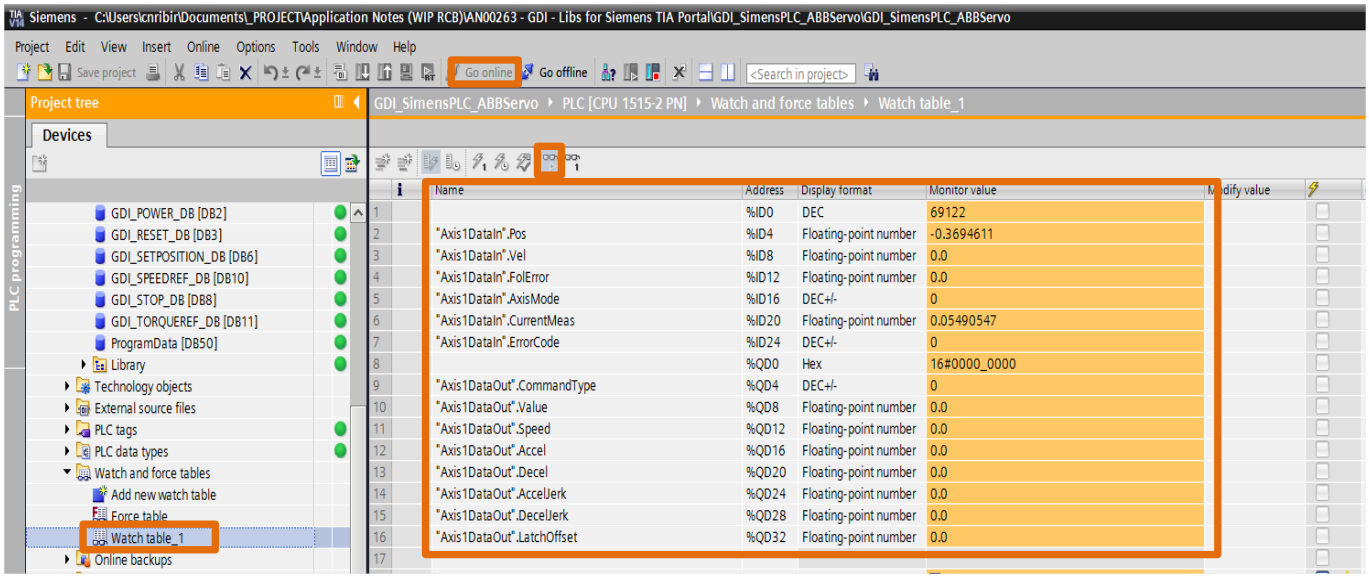
如需检查通信状态，可在 PLC 和 Mint Workbench 中添加变量监测表。变量被映射后，我们就可以在程序中自由使用它们，从长远来看，把它们添加监视表中有利于调试。程序中存在的任何变量都可以映射到监视列表中。这样，我们就可以将我们刚为 e190 添加的任何变量映射到新的监视列表中。为此，我们必须先离线，以便设置数据监控“监视表”便于调试。

要创建监视表，首先转到设备视图，选择“[PLC 名称]>监视和强制表>添加新监视表”。它将自动获得一个名称，例如“Watch_Table_1”。



然后，可以通过向监视列表中添加元素，然后在地址列中选择地址，为驱动节点添加当前映射的数据。

然后，我们可以联机并监视位于“[PLC 名称]>监视和强制表> Watch_Table_1”中的监视表中的数据。完成后，您可以使用“在线查看”按钮来监控变量值。

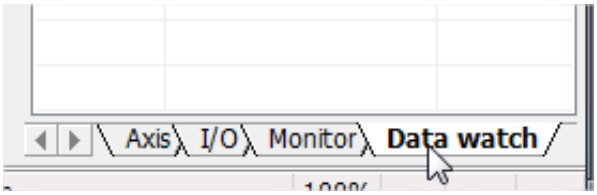


注意： “监视值”列将显示变量的当前值（应选择正确的数据类型）。一致的数据值表示 PROFINET IO 通信正常。

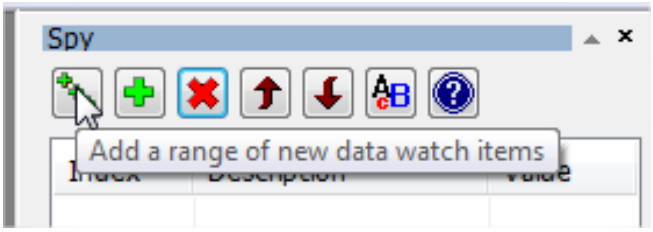
在驱动器中进行数据监视

现在已经可以在 PLC 中监控数据。还可以在驱动器中设置数据监控，以检查数据的一致性。为此，必须转到 Mint Workbench 软件，转到工具箱的“Edit & Debug”部分；

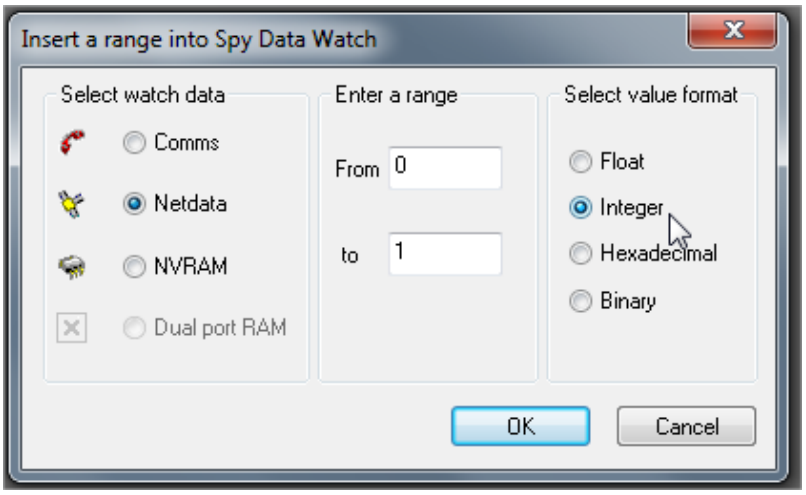
接下来转到“监视”窗口，并选择“数据监视”选项卡。



然后，我们可以使用前面显示的表格在此处添加数据，以便为每个网络数据位置选择正确的数据类型。为此，请在监视窗口中选择“+++”图标。

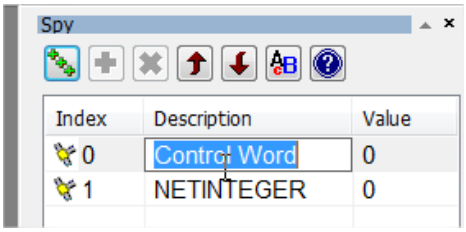


然后会出现一个对话框，可以在其中选择将添加到 “数据监视列表”中的 Netdata。我们一次添加多个 Netdata，然后为每个 Netdata 选择正确的地址（Mint NetData 通道）和格式（数据类型）；



添加完所有地址后， Netdata 0~8，和 NetData 100~106 就被映射。再次检查所有 Netdata 的数据格式（整数/浮点数）是否正确。

为了更容易跟踪变量，最好根据 GDI 中根据 Netdata 的功能重命名变量。您可以通过单击其描述，键入名称，直接更名。



注意： 在文档 AN00204 Mint GDI 操作和配置中更详细地概述了该过程。

ABB 运动 GDI 库简介

为了方便用户使用 Mint GDI，ABB 创建了一个功能块库，它提供了一种从 PLC 向驱动器发送运动命令的简便方法。本文档包含该库。可以把它轻松导入到您的程序中，为您提供易于使用的命令，以实现简单的单轴运动。

库指令概述

本应用说明的库为具有 PROFINET IO 接口的西门子 PLC 提供如下功能：

- 发送一条寻零命令
- 发出命令以检测物理轴终点挡板，并将其用作基准位置（如果使用 e190 或 e180 驱动器，则需要固件版本 5868）
- 发送一条相对定位命令
- 发送一条绝对定位命令
- 发送一条增量相对定位命令（可选择在经过“快速锁存”位置一段设定距离后停车）
- 发送一条增量绝对定位命令（可选择在经过“快速锁存”位置一段设定距离后停车）
- 为增量式运动设置一个偏移目标（即相对于记录的快速中断位置来定位轴）
- 点动轴
- 设置轴位置
- 发送一个速度给定值
- 发送一个转矩给定值
- 使能/停用轴
- 使能/停用硬件限位
- 复位轴错误
- 在轴上执行一次受控停车或紧急停车
- 使轴与第二个编码器输入相关联
- 为所有运动设置速度、加速时间、减速时间和加加速度时间
- 控制旋转轴或非旋转轴

同时，PLC 还能监视来自驱动器的状态信息，包括：

- 使能状态
- 准备进入使能状态
- 空闲状态
- 到达目标位置状态
- 电机抱闸状态
- 是否完成寻零
- 正向限位状态
- 反向限位状态
- 故障状态
- 停车输入状态
- 错过快速锁存中断的指示
- 寻相状态
- 错误代码
- 测量位置
- 测量速度
- 跟随误差
- 轴运行模式
- 均方根电流

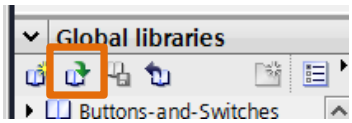
这一切都是通过 PLC 所见的输入和输出寄存器实现的。

看门狗

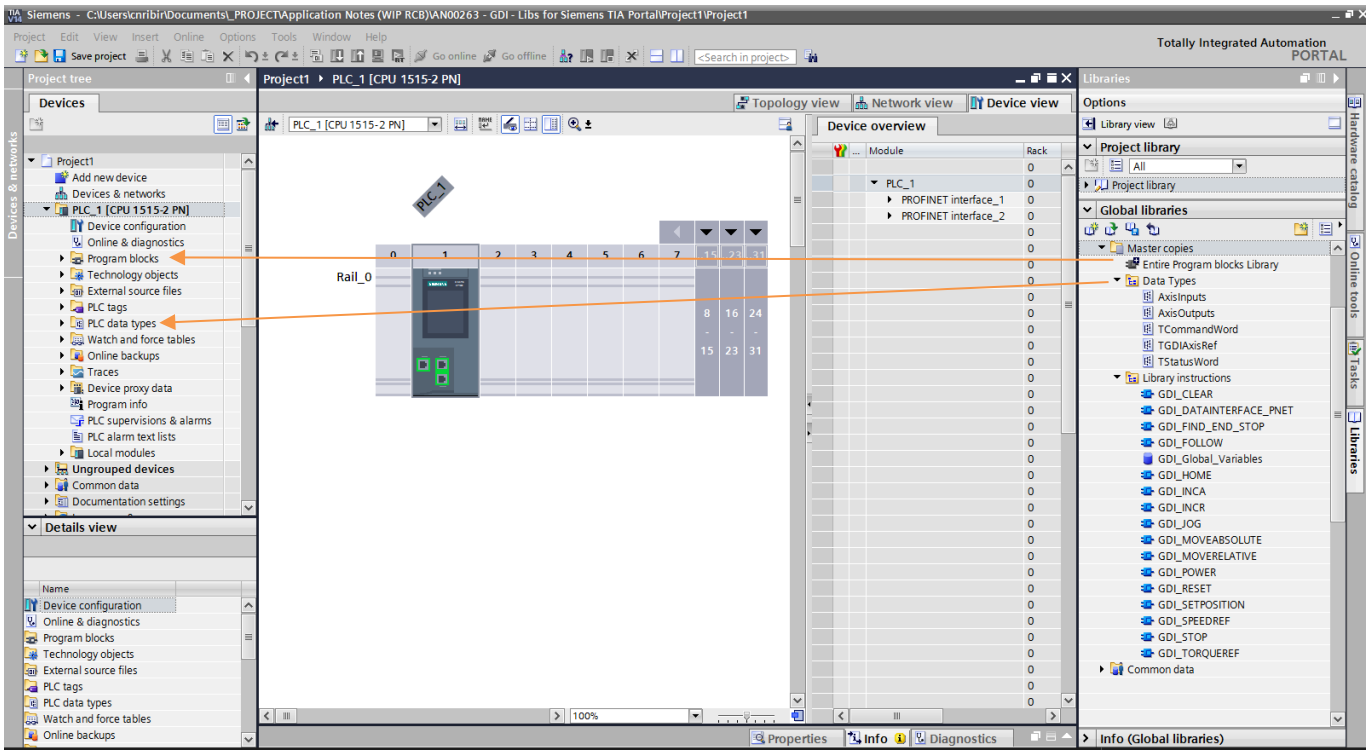
其中还包括了可选的看门狗机制，允许驱动器在通讯丢失时采取动作（默认紧急停车或停用）。

导入用户定义的数据类型和库指令

确保在左侧的“设备”窗口中展开 PLC，以便可以访问“程序块”和“PLC 数据类型”文件夹。接下来我们必须选择“库”视图，（从窗口右侧选择），然后点击图标“打开全局库”。



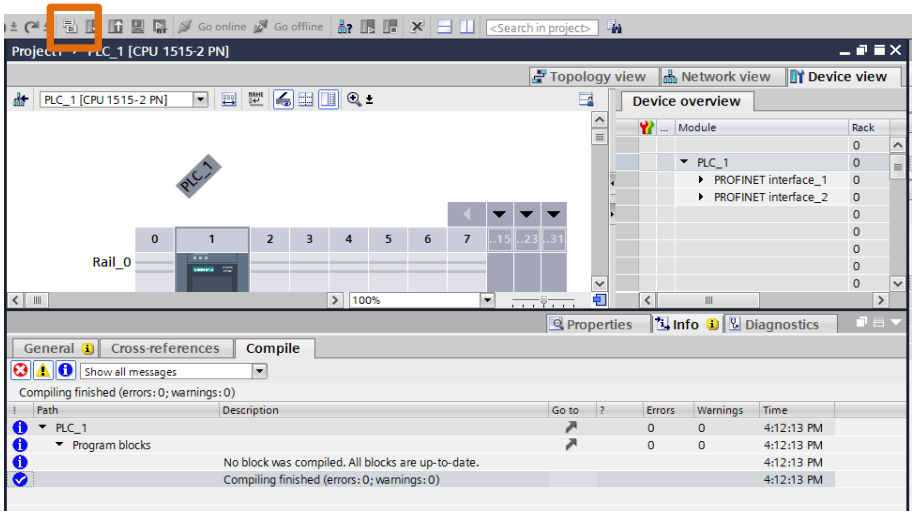
接下来浏览到您保存库的位置，并找到文件“*ABB Motion GDI library.al14*”并打开它。然后，这将库里的所有数据批量导入到项目中。然后，需要做的就是将您需要的元素拖到项目文件夹中；

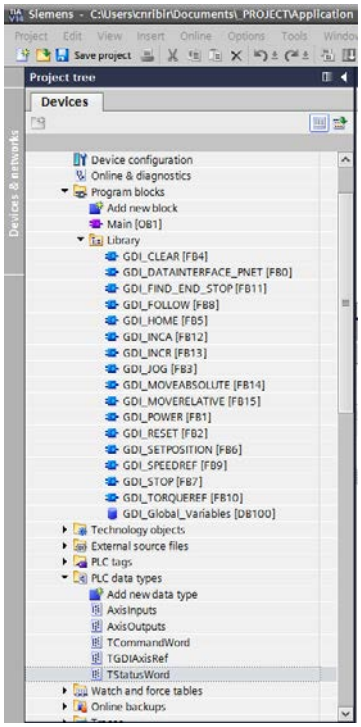


如果将“整个程序块库”导入“程序块”文件夹，您将自动拥有所有 GDI 块。如果您只想使用某些库块，则可以从“库指令”目录中一次导入一个库块。

在所有情况下，都需要导入所有五种数据类型。

接下来必须编译代码；





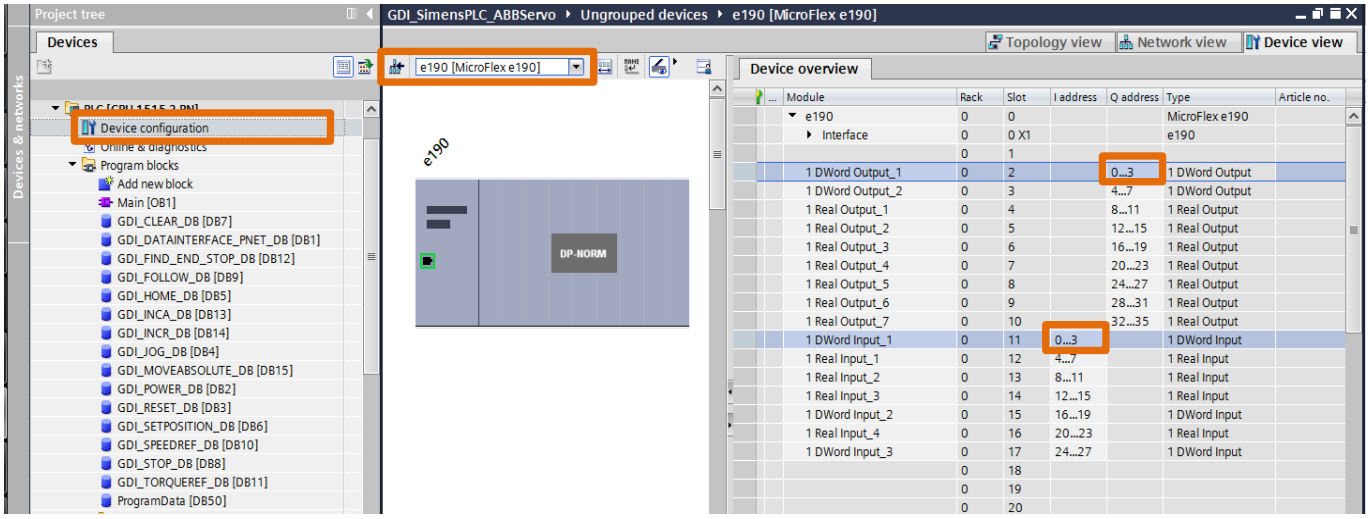
编译代码后，TIA Portal 将可以访问导入的所有数据类型和块。该项目现在包含 GDI 轴的数据类型（TGDIAxisRef），以及此轴使用的命令和状态字，以及输入和输出结构的数据类型。

可以看包含在库中的是一个名为“*GDI_Global_Variables*”的全局变量列表的示例。这样，就给出了一个模板，可以依照该模板向系统添加更多轴。它包含一个名为“Axis1”的变量，它的数据类型为“TGDIAxisRef”，用于在 GDI 控件和接口块之间传递“Axis1”的数据。对于其他轴，您可以为每个附加轴将更多数据类型为“TGDIAxisRef”的变量添加到此数据块结构中。

您还会看到有一个名为“WatchdogTime”的变量，类型为“time”。该变量定义从 PLC 发送到驱动器的看门狗脉冲的周期时间。无论您是否使用它，PLC 中的看门狗都将处于激活状态。

将数据从驱动器映射到 PLC

在我们开始使用该程序之前，我们必须在驱动器 IO 和程序预期使用的数据类型之间创建数据链接。为此，我们必须检查 TIA portal 分配给 e190 驱动器的地址，然后创建一个列出此信息的标签列表。要检查分配给驱动器的地址，必须通过选择“设备配置”检查，选择“设备视图”并从下拉菜单中选择 e190。如下所示，第一个输出地址是 QW0，第一个输入地址是 IW0。



接下来，我们必须创建一个 PLC 标签列表，其中包含地址声明，并使用输入数据类型“*AxisInputs*”和输出“*AxisOutputs*”。这些特殊数据类型的结构允许只使用指向驱动器输入或输出的起始地址的指针一次性读入驱动器中的所有变量。我们必须输入输入“起始”地址%I0.0，选择数据类型为“AxisInputs”并为其命名（在下面的示例中为“Axis1DataIn”）。同样，对于输出，选择“AxisOutputs”作为数据类型，输入%Q0.0 作为“开始”地址并为其命名（在下面的示例中为“Axis1DataOut”）。

AxisIOVariables

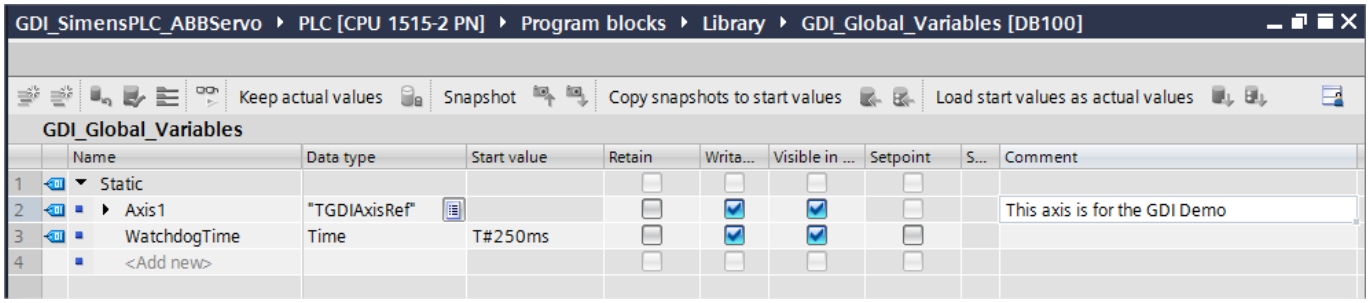
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Supervision
1	Axis1DataIn	"AxisInputs"	%I0.0	☐	☑	☑	☑	
2	Axis1DataOut	"AxisOutputs"	%Q0.0	☐	☑	☑	☑	

注意： 在库中，AxisIOVariables 文件夹中有一个名为“AxisIOVariables”的示例。

编写程序

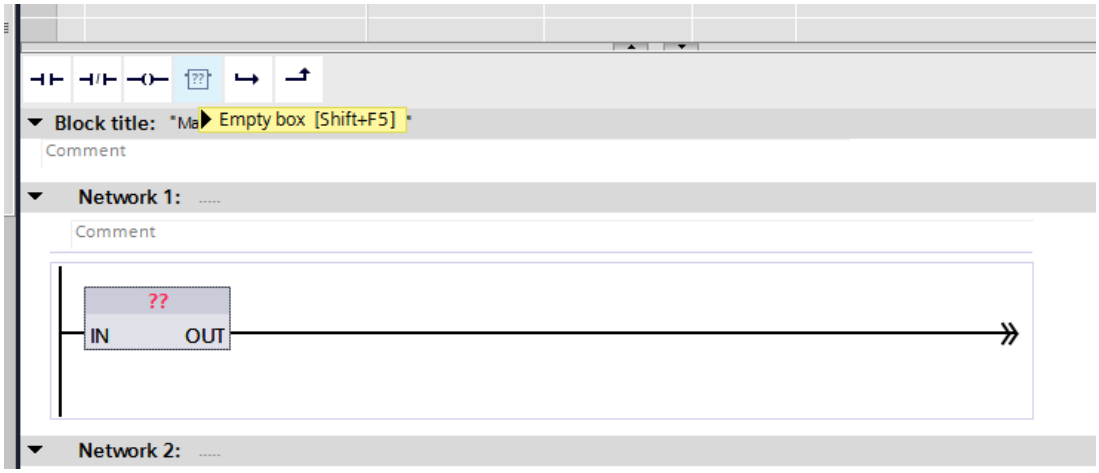
我们现在准备开始使用这些指令在我们的 PLC 应用程序中创建一些运动。对于每个轴，我们必须添加一个轴参考，以将功能块中的控制数据与 PZD 数据关联起来。PZD 数据将通过 PROFINET 传递给驱动器。

在下面的例子中，我们创建了一个“*GDI_Global_Variables*”数据块（DB100）来映射这个轴以及我们将添加到程序中的任何其他轴的所有数据。对于第一个驱动器，创建了一个名为“Axis1”、数据类型为“TGDIAxisRef”的条目（之前导入的数据类型）。在关于轴应用程序的注释中添加一些信息也是一个好主意；

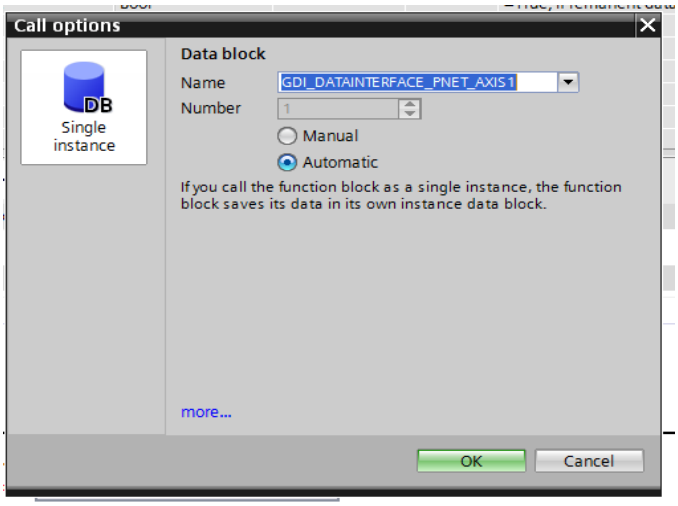


注意： “Axis 1”是我们通过 *“GDI_Global_Variables.Axis1 ”*在程序中引用轴/驱动器的方式。

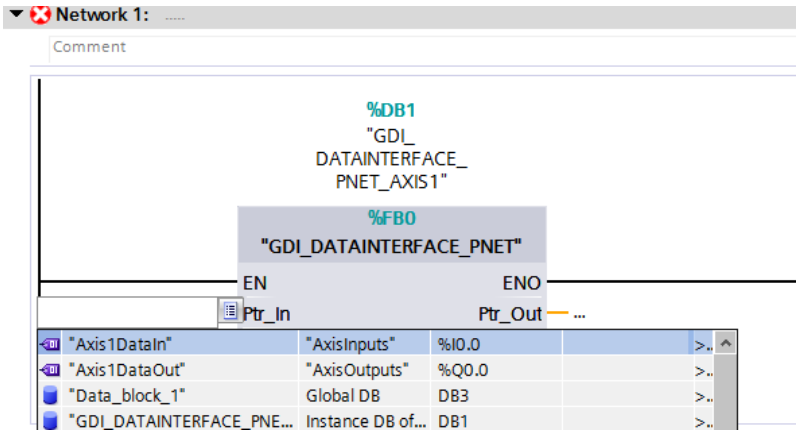
接下来需要在应用程序中为每个轴包含一个 GDI_DATAINTERFACE_PNET 函数的实例。在本文档中，我们将使用主程序（OB1）作为我们的程序逻辑。我们在这个例子中使用了梯形图（LDR）。选择第一个行，然后使用指令工具箱添加一个新的“空框”。



单击框中的“??”，然后键入功能块名称。在本例中，键入 GDI_DATAINTERFACE_PNET 然后按回车键，或从列表中选择。在完成此操作后，将出现“调用选项”对话框，以帮助您选择此实例的数据块地址。如果需要，可以在此处更改名称，也可以手动选择数据块地址。最简单的方法是将地址分配保留为自动，并更改块的名称以引用轴，例如“*GDI_DATAINTERFACE_PNET_AXIS1*”。完成后 - 选择“确定”；

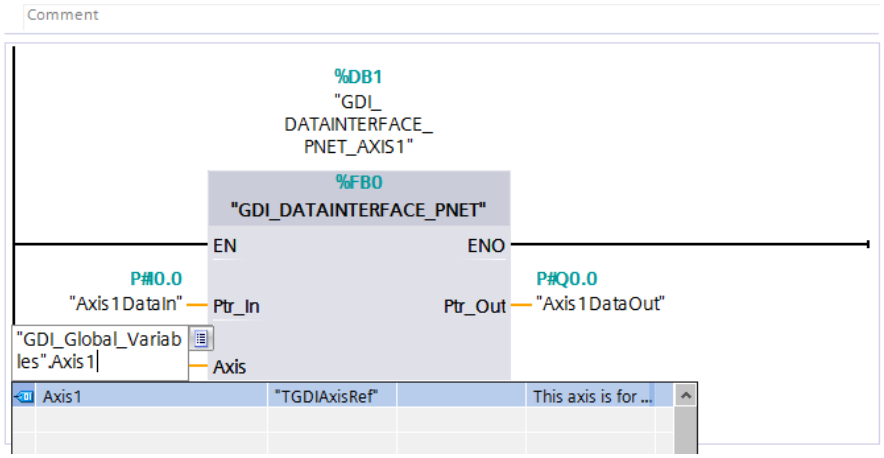


我们现在需要将通信块的输入和输出端子链接到 PROFINET IO 驱动器的输入和输出组件。双击“Ptr_In”旁边的“...”，然后单击“页面”图标并选择驱动器输入数据：“Axis1DataIn” - 然后会自动为此指定一个指针（对于输入，它将是 P＃I0.0）。



类似地，对于输出，您须将“Axis1DataOut”添加到“Ptr_Out”输出（同样，这将由指针分配，在本例中为 P＃Q0.0）。

接下来，给通信块指定参考轴。为此，从之前创建的“GDI_Global_Variables”列表中选择 Axis1。在本例中，轴是 Axis1（尽管也可以把其他轴添加到列表中）。



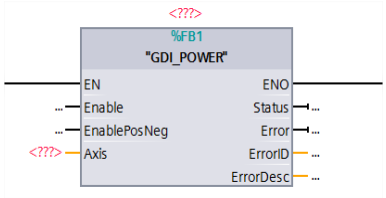
我们现在可以将程序下载到 PLC，并测试基本通信是否正常运行。

GDI 功能块

以下部分详细说明了 GDI 功能块的用法：

GDI_POWER

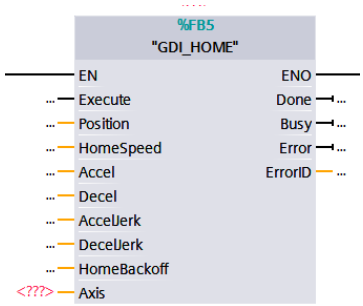
本功能块用于使能/停用轴。使能输入会使能驱动器的功率模块，而不是功能块本身。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Enable	BOOL	在它为真时，PLC 将要求使能轴
EnablePosNeg	BOOL	在它为真时，允许两个方向上的运动。如果为假，运动被阻止（或在运动已经在进行时执行一个停车命令）
输出变量		
Status	BOOL	指示轴是（1）否（0）被使能
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见**错误处理**部分
ErrorDesc	字符串	以纯文本指示 Mint 错误代码描述

GDI_HOME

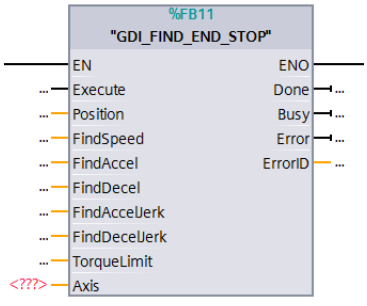
本功能块用于为轴提供基准。基准的详情取决于 Mint GDI 程序中设置的寻零类型。输入的“位置”用于在成功回零后，设置的轴的当前值。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	在上升沿启动基准序列
Position	REAL	需要在成功的基准序列末尾处设置的绝对位置
HomeSpeed	REAL	寻零速度，用户单位/秒
HomeAccel	REAL	寻零加速速率，用户单位/秒 ^{2}
HomeDecel	REAL	寻零减速速率，用户单位/秒 ^{2}
HomeAccelJerk	REAL	寻零加加速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
HomeDecelJerk	REAL	寻零减减速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
HomeBackOff	REAL	寻零速度与回退速度的比率
输出变量		
Done	BOOL	指示轴已经成功寻零。如果在寻零中 Execute 输入被移除，并且轴完成了寻零序列，将 Done 输出为一个 PLC 扫描周期。如果 Execute 输入仍然为真，则 Done 输出将仍然保持为真（在寻零成功的前提下）。
Busy	BOOL	在寻零序列正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_FIND_END_STOP

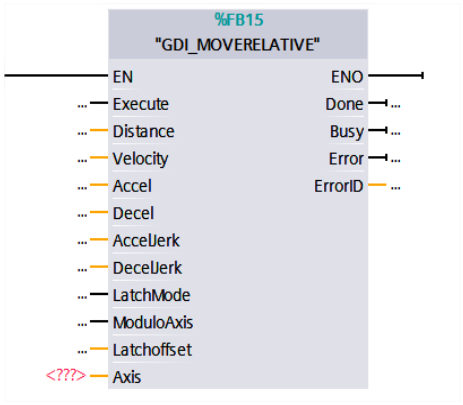
该功能块用作在没有零点传感器的情况下对轴进行回零。轴将以指令速度运行，并具有编程的转矩限值，当转矩达到转矩限值，并且轴的速度小于编程的怠速时。把位置输入作为当前值，回零结束。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	在上升沿启动基准序列
Position	REAL	需要在成功的基准序列末尾处设置的绝对位置
FindSpeed	REAL	以用户单位/秒为单位的速度（此值的符号决定了搜索方向）
FindAccel	REAL	加速速率，用户单位/秒 ^{2}
FindDecel	REAL	减速速率，用户单位/秒 ^{2}
FindAccelJerk	REAL	加加速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
FindDecelJerk	REAL	减减速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
TorqueLimit	REAL	序列中应用的转矩限值（驱动器额定电流的百分比）
输出变量		
Done	BOOL	显示轴是否已经成功找到终点停车位置。如果在序列中删除了 Execute 输入并且轴达到端部止动器，则将 Done 输出设置为一个 PLC 扫描周期。如果 Execute 输入仍然为 1，则 Done 输出将仍然保持为真（在序列成功的前提下）。
Busy	BOOL	在查找序列正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_MOVERELATIVE

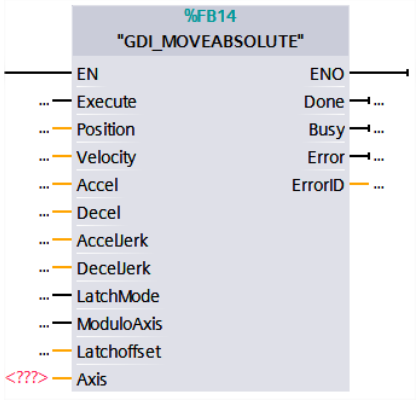
本功能块用于发出命令，以启动一段针对当前值的相对运动。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	起动上升沿上的运动
Distance	REAL	运动的相对距离（用户单位）
Velocity	REAL	最大速度（无需达到），用户单位/秒
Accel	REAL	加速速率，用户单位/秒 ^{2}
Decel	REAL	减速速率，用户单位/秒 ^{2}
AccelJerk	REAL	加加速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
输出变量		
Done	BOOL	显示轴是否已经成功找到目标位置。如果在运动中 Execute 输入被移除，并且相对定位完成，则 Done 输出为 1，即一个 PLC 扫描周期。如果 Execute 输入仍然为真，则 Done 输出将仍然保持为真（在成功达到目标位置的前提下）。
Busy	BOOL	在相对定位正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_MOVEABSOLUTE

本功能块用于发出一条命令，以执行一次到达指定绝对位置的运动。本功能可配合模态轴使用（在这种情况下，将运行最短路线）。

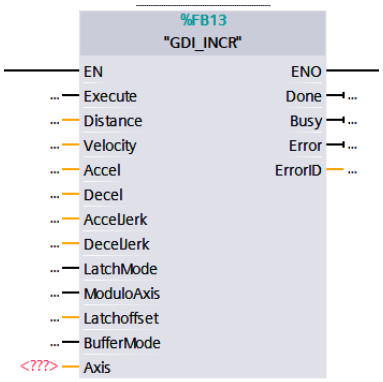


	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	起动上升沿上的运动
Position	REAL	运动的目标位置（用户单位）
Velocity	REAL	最大速度（无需达到），用户单位/秒
Accel	REAL	加速速率，用户单位/秒 ^{2}
Decel	REAL	减速速率，用户单位/秒 ^{2}
AccelJerk	REAL	加加速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
ModuloAxis	BOOL	定义轴是否为旋转轴（即使用 ENCODERWRAP 来定义一个周期内的行程）。使用旋转轴时的绝对定位始终通过最短的路径实现（比如，在 0-360 度的旋转轴上，从 350 度到 20 度的绝对定位将引发一次 30 度的向前运动）
输出变量		
Done	BOOL	显示轴是否已经成功找到目标位置。如果在运动中 Execute 输入被移除，并且绝对定位完成，则 Done 输出为一个 PLC 扫描周期。如果 Execute 输入仍然为真，则 Done 输出将仍然保持为真（在成功达到目标位置的前提下）。
Busy	BOOL	在绝对定位正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_INCR

本功能块用于发出命令，以执行做出相对于目标位置的一段指定距离的受控运动。在运动过程中，可通过以下任一种方法修改运行的目标位置：

a. 发送另一个 GDI_INCR 或 GDI_INCA 功能（在输入参数 BufferMode 为真的前提下）
b. 把输入参数 Latchmode 设置为真，并为输入参数 LatchOffset 指定值。之后，驱动器上的 Mint 代码将自动修改轴目标位置，以在与定义的快速中断记录的轴位置的距离为 LatchOffset 时停止。可使用轴状态字中的一位（btLatchMissed）来指示未能检测到快速中断（比如，这种条件之后可用于提醒操作人员存在系统故障）。使用 Latchmode 和 LatchOffset 能简化分度输送带应用的执行。



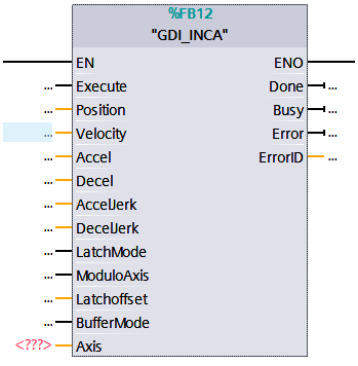
	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	起动上升沿上的运动
Distance	REAL	运动的相对距离（用户单位）
Velocity	REAL	最大速度（无需达到），用户单位/秒
Accel	REAL	加速速率，用户单位/秒 ^{2}
Decel	REAL	减速速率，用户单位/秒 ^{2}
AccelJerk	REAL	加加速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
LatchMode	BOOL	设置轴是否应该利用配置的快速锁存中断，并在与记录的位置距离为“LatchOffset”用户单位的地方设置新目标位置
LatchOffset	REAL	定义应该修改 GDI_INCR 的目标时（在输入参数 Latchmode 设置为真时）经过记录的快速位置的距离（用户单位）
BufferMode	BOOL	定义功能块是否应该设置完成输出，并在运动载入时尽快完成。设置缓冲模式为真时，允许应用在现有运动正在进行时触发更多的增量运动。
输出变量		
Done	BOOL	在缓冲模式设置为假时，表示轴已经成功到达目标位置。如果在运动中 Execute 输入被移除，并且相对定位完成，则 Done 输出为一个 PLC 扫描周期。如果 Execute 输入仍然为真，则 Done 输出将仍然保持为真（在成功达到目标位置的前提下）。在缓冲模式设置为真时，则 Done 输出为一个 PLC 扫描周期，表明运动已经成功载入。
Busy	BOOL	在功能块正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

如果应用需要修改正在进行的相对定位的 SPEED/ACCEL/DECEL，也可使用 GDI_INCR。使用 GDI_MOVERELATIVE 载入的运动，在运动开始后无法修改 SPEED/ACCEL/DECEL 配置。通过使用 GDI_INCR 并把输入参数 BufferMode 设置为真，可通过载入另一个 GDI_INCR（新的 SPEED/ACCEL/DECEL）并把输入参数 Distance 设置为零来修改配置文件的参数。

GDI_INCA

本功能块用于发出一条命令，以执行一次到达指定绝对位置的运动。本功能与 GDI_MOVEABSOLUTE 的不同之处在于，可在运动进行时通过以下任一方法修改目标位置：

a. 发送另一个 GDI_INCR 或 GDI_INCA 功能（在输入参数 BufferMode 为真的前提下）
b. 把输入参数 Latchmode 设置为真，并为输入参数 LatchOffset 指定值。之后，驱动器上的 Mint 代码将自动修改轴目标位置，以在与定义的快速中断记录的轴位置的距离为 LatchOffset 时停止。可使用轴状态字中的一位（btLatchMissed）来指示未能检测到快速中断（示例程序示范了检测到连续三次错过锁存的方式-比如，这种条件之后可用于提醒操作人员存在系统故障）。

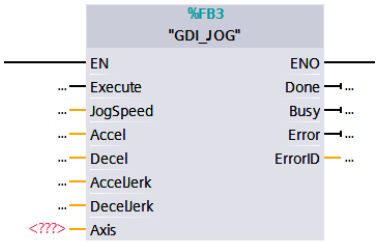


	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	起动上升沿上的运动
Position	REAL	运动的绝对位置目标（用户单位）
Velocity	REAL	最大速度（无需达到），用户单位/秒
Accel	REAL	加速速率，用户单位/秒 ^{2}
Decel	REAL	减速速率，用户单位/秒 ^{2}
AccelJerk	REAL	加加速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
LatchMode	BOOL	设置轴是否应该利用配置的快速锁存中断，并在与记录的位置距离为“LatchOffset”用户单位的地方设置新目标位置
ModuloAxis	BOOL	定义轴是否为旋转轴（即使用 ENCODERWRAP 来定义一个周期内的行程）。使用旋转轴时的绝对定位始终通过最短的路径实现（比如，在 0-360 度的旋转轴上，从 350 度到 20 度的绝对定位将引发一次 30 度的向前运动）
LatchOffset	REAL	定义应该修改 GDI_INCR 的目标时（在输入参数 Latchmode 设置为真时）经过记录的快速位置的距离（用户单位）
BufferMode	BOOL	定义功能块是否应该设置完成输出，并在运动载入时尽快完成。设置缓冲模式为真时，允许应用在现有运动正在进行时触发更多的增量运动。
输出变量		
Done	BOOL	在缓冲模式设置为假时，表示轴已经成功到达目标位置。如果在运动中 Execute 输入被移除，并且相对定位完成，则 Done 输出为一个 PLC 扫描周期。如果 Execute 输入仍然为真，则 Done 输出将仍然保持为真（在成功达到目标位置的前提下）。在缓冲模式设置为真时，则 Done 输出为一个 PLC 扫描周期，表明运动已经成功载入。
Busy	BOOL	在功能块正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

如果运行中需要修改 SPEED/ACCEL/DECEL，也可使用 GDI_INCR。使用 GDI_MOVERELATIVE 载入的运动，在运动开始后无法修改 SPEED/ACCEL/DECEL 配置。通过使用 GDI_INCR 并把输入参数 BufferMode 设置为真，可通过载入另一个 GDI_INCR（新的 SPEED/ACCEL/DECEL）并把输入参数 Distance 设置为零来修改配置文件的参数。

GDI_JOG

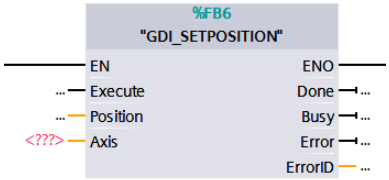
本功能块用于发出一条命令，以在轴上执行一次恒速运动（使用驱动器中的位置环控制器）。如果 Execute 输入保持为真且 JogSpeed 不是“0”，则执行运动。如果 JoGrice 变成“0”，则需要考虑一个新的执行命令，因为轴将把它看作是运动的结束。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	在上升沿时开始运动，并在输入为真是保持运动。在 Execute 变为假时，运动以斜坡形式按照配置的减速速率下降到零速。
JogSpeed	REAL	轴将达到的速度值，用户单位/秒
Accel	REAL	加速速率，用户单位/秒 ^{2}
Decel	REAL	减速速率，用户单位/秒 ^{2}
AccelJerk	REAL	加加速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
输出变量		
Done	BOOL	在成功发送点动命令后立即设置为真并保持该设置，直到 Execute 变为假或出现轴错误
Busy	BOOL	在功能块正在进行时设置为真如果执行功能块但 JogSpeed=0，则将打开
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_SETPOSITION

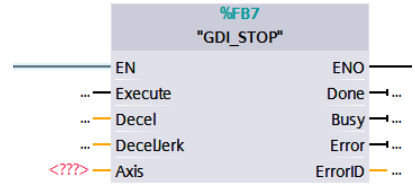
本功能块用于将轴位置（驱动器上的编码器和位置值）设置为程序内的值。在调用本功能时，轴必须为空闲状态；否则，轴将返回一个“操作无法执行-运动进行中”的错误（错误代码 10）。如果轴使用绝对值编码器，这时会设置/发送新的绝对位置（GDI Mint 程序 V2.17 及更新版本）。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	在上升沿设置新位置
Position	REAL	要设置的轴位置的值（用户单位）
输出变量		
Done	BOOL	在发送命令后立即设置为真（无论其是否成功 - 使用错误输出来确定命令是否成功）。保持为真，直到 Execute 输入被移除。如果 Execute 输入在设置完成位前被移除，则 Done 输出为一个 PLC 扫描周期。
Busy	BOOL	在功能块正在运行时设置为真（在设置 Done 后立即清除）
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_STOP

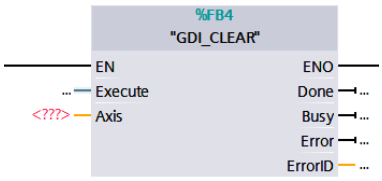
本功能块用于按照预设的减速速率在轴上执行受控停车。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	上升沿起动受控停车
Decel	REAL	减速速率，用户单位/秒 ^{2}
DecelJerk	REAL	减减速速率，用户单位/秒 ^{3}（对梯形运动，设置为 0）
输出变量		
Done	BOOL	在完成受控停车后轴静止时，或在发送停车命令后发生错误时，设置为真。保持为真，直到 Execute 输入被移除。如果 Execute 输入在设置完成位前被移除，则 Done 输出为一个 PLC 扫描周期。
Busy	BOOL	在停车正在进行时设置为真-在设置 Done 后立即清除
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_CLEAR

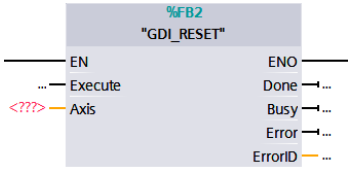
本功能块用于轴的紧急停车，并中断任何正在进行的运动。轴将保持使能（在 GDI_POWER 正在要求使能状态，并且轴没有出错的前提下）。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	上升沿起动紧急停车
输出变量		
Done	BOOL	在完成紧急停车后轴静止时，或在发送紧急停车命令后发生错误时，设置为真。保持为真，直到 Execute 输入被移除。如果 Execute 输入在设置完成位前被移除，则 Done 输出为一个 PLC 扫描周期。
Busy	BOOL	在停车正在进行时设置为真-在设置 Done 后立即清除
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_RESET

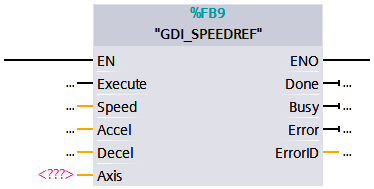
本功能块用于复位存在的任何轴错误。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	上升沿起动故障复位
输出变量		
Done	BOOL	在轴不再有错误时，设置为真。保持为真，直到 Execute 输入被移除。如果 Execute 输入在设置完成位前被移除，则 Done 输出为一个 PLC 扫描周期。如果无法清除错误，不能设置 Done（使用 Busy 输出来检测尝试故障复位的时间）
Busy	BOOL	在功能块尝试清除任何轴错误时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_SPEEDREF

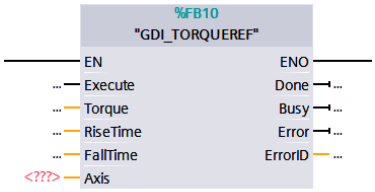
本功能块用于对轴进行速度控制。在本操作模式下，不使用驱动器的位置环（因此不会记录跟随误差）。轴将保持速度控制模式（见控制模式的状态字位的指示），直到发送另一种控制模式类型的运动（比如，位置控制运动）。要从零速运行（在速度控制模式下）切换到保持位置（在位置控制模式下），可发送比如定位距离为零的 GDI_MOVERELATIVE 指令。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	在上升沿启动轴，并在输入为真时保持运动。在 Execute 变为假时，运动以斜坡形式按照配置的减速速率下降到零速。
Speed	REAL	轴将达到的速度值，用户单位/秒。可在 Execute 为真时修改，以改变轴的速率
Accel	REAL	加速速率，用户单位/秒 ^{2}
Decel	REAL	减速速率，用户单位/秒 ^{2}
输出变量		
Done	BOOL	在发送速度给定值后立即设置为真（无论其是否成功）。Done 输出保持设置，直到 Execute 变为假。
Busy	BOOL	在功能块正在执行时设置为真（即在 Execute 为真时）
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_TORQUEREF

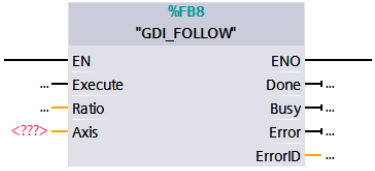
本功能块用于控制轴的扭矩输出。在本操作模式下，不使用驱动器的位置环（因此不会记录跟随误差或做出响应）。轴将保持转矩控制模式（见控制模式的状态字位的指示），直到发送另一种控制模式类型的运动（比如，位置控制运动）。要从零转矩运行（在转矩控制模式下）切换到保持位置（在位置控制模式下），可发送比如定位距离为零的 GDI_MOVERELATIVE 指令。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	在上升沿启动转矩给定值，并在输入为真时保持转矩。在 Execute 变为假时，转矩以斜坡形式按照配置的下降时间比率下降到零。
Torque	REAL	轴将使用的转矩给定值（形式为 DRIVERATEDCURRENT 的百分比-见 Mint 帮助文件）。在 Execute 为真时可进行修改，以改变产生的转矩
RiseTime	REAL	设置电流从零上升到 DRIVEPEAKCURRENT 所需的时间（单位为 ms）（见 Mint 帮助文件）
FallTime	REAL	设置电流从 DRIVEPEAKCURRENT 下降到零所需的时间（单位为 ms）（见 Mint 帮助文件）
输出变量		
Done	BOOL	在发送转矩给定值后立即设置为真（无论其是否成功）。Done 输出保持设置，直到 Execute 变为假。
Busy	BOOL	在功能块正在执行时设置为真（即在 Execute 为真时）
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_FOLLOW

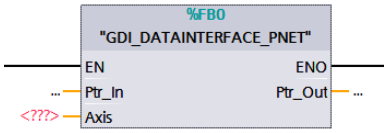
本功能块用于命令轴按照预设的跟随比率，跟随配置的主编码器值。



	类型	说明
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考
输入变量		
Execute	BOOL	在出现上升沿时启动跟随。在 Execute 输入变为假时，轴将保持跟随模式（为停止跟随，发送另一条运动命令或使用 GDI_CLEAR 清除运动）
Ratio	REAL	轴和主编码器给定值之间跟随（关联）比率的值（值将受到轴换算和主编码器换算的影响-见 Mint 帮助文件中有关 FOLLOW 的主题）。要在跟随时设置一个新的比率，有必要发送一条新的 GDI_FOLLOW 命令。
输出变量		
Done	BOOL	在发送跟随命令后立即设置为真（无论其是否成功）。Done 输出保持设置，直到 Execute 变为假。
Busy	BOOL	在功能块正在执行时设置为真（即在 Execute 为真时）
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码 – 参见错误处理部分

GDI_DATAINTERFACE_PNET

本功能块用于在 PLC 程序的应用层和通信层之间传送命令/状态数据。对应用中的每个轴，必须有一个对应的此功能实例。



	类型	说明
输入变量		
Ptr_In	“AxisInputs” - 将使用指针引用 40 字节的输入数据	对轴的输入结构的参考
输出变量		
Ptr_Out	“AxisOutputs” - 将使用指针引用 40 字节的输出数据	对轴的输出结构的参考
输入输出变量		
Axis	TGDI_AxisRef	对轴结构的参考

使用轴结构

GDI 的大多数功能由提供的各种 GDI 功能块封装。但是，在某些情况下，对轴结构数据的访问对应用程序逻辑可能会很有用。TGDI_AxisRef 数据类型的声明如下所示：

TGDIAxisRef

	Name	Data type	Default value
1	AxisNo	UInt	0
2	CommandWord	"TCommandWord"	
3	CommandType	DInt	0
4	Value	Real	0.0
5	Speed	Real	0.0
6	Accel	Real	0.0
7	Decel	Real	0.0
8	AcceJerk	Real	0.0
9	DeceJerk	Real	0.0
10	LatchOffset	Real	0.0
11	StatusWord	"TStatusWord"	
12	Pos	Real	0.0
13	Vel	Real	0.0
14	FolError	Real	0.0
15	AxisMode	DInt	0
16	CurrentMeas	Real	0.0
17	ErrorCode	DInt	0

本数据结构另外包括了两种数据结构（TCommandWord 和 TStatusWord）。它们的声明如下所示：

TCommandWord

	Name	Data type	Default value
1	Spare_14	Bool	false
2	Spare_15	Bool	false
3	Spare_16	Bool	false
4	Spare_17	Bool	false
5	Spare_18	Bool	false
6	Spare_19	Bool	false
7	Spare_20	Bool	false
8	Spare_21	Bool	false
9	Spare_6	Bool	false
10	Spare_7	Bool	false
11	Spare_8	Bool	false
12	Spare_9	Bool	false
13	Spare_10	Bool	false
14	Spare_11	Bool	false
15	Spare_12	Bool	false
16	Spare_13	Bool	false
17	btWatchdog	Bool	false
18	btIgnoreFE	Bool	false
19	Spare	Bool	false
20	Spare_1	Bool	false
21	Spare_2	Bool	false
22	Spare_3	Bool	false
23	Spare_4	Bool	false
24	Spare_5	Bool	false
25	btEnable	Bool	false
26	btMotionAllowed	Bool	false
27	btPosLatchEnable	Bool	false
28	btDisFwdLimit	Bool	false
29	btDisRevLimit	Bool	false
30	btModulo	Bool	false
31	btFaultReset	Bool	false
32	btTriggerCmd	Bool	false

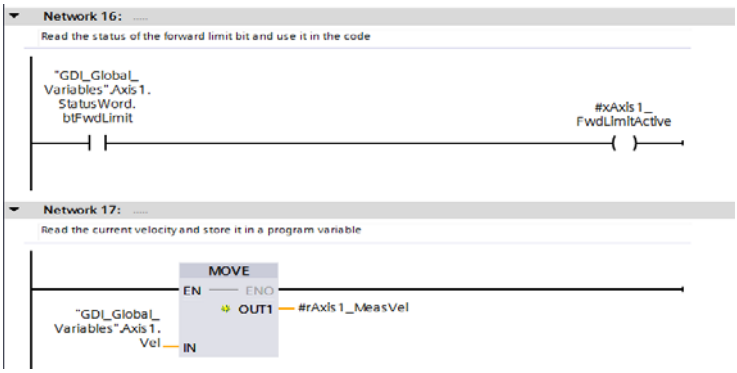
TStatusWord

	Name	Data type	Default value
1	Spare_14	Bool	false
2	Spare_15	Bool	false
3	Spare_16	Bool	false
4	Spare_17	Bool	false
5	Spare_18	Bool	false
6	Spare_19	Bool	false
7	Spare_20	Bool	false
8	Spare_21	Bool	false
9	btPhaseSearchDone	Bool	false
10	Spare_6	Bool	false
11	Spare_7	Bool	false
12	Spare_8	Bool	false
13	Spare_9	Bool	false
14	Spare_10	Bool	false
15	Spare_11	Bool	false
16	Spare_12	Bool	false
17	btStopInput	Bool	false
18	btReadyToEnable	Bool	false
19	btControlMode0	Bool	false
20	btControlMode1	Bool	false
21	btTriggerDone	Bool	false
22	btPermitted	Bool	false
23	btLatchMissed	Bool	false
24	btFaultReset	Bool	false
25	btEnabled	Bool	false
26	btIdle	Bool	false
27	btInPos	Bool	false
28	btBrakeEngaged	Bool	false
29	btHomed	Bool	false
30	btFwdLimit	Bool	false
31	btRevLimit	Bool	false
32	btFault	Bool	false

结构可能看起来很复杂（与 ABB 伺服产品寻址相比）- 这与西门子 PLC 的字节和字高低位交换有关。因此，PLC 代码可以通过这些结构访问任何这些数据。

示例：

两行梯图直接访问轴数据结构，一个读取驱动器上的正向限位输入的状态，另一个保存测量的轴速度...



能够直接访问这些数据可以给 PLC 编程带来很大的灵活性（例如，对于分度传送机，PLC 可以访问锁存丢失状态位（btLatchMissed）并对其计数。当丢失次数达到一定值时，可停止轴运动。 并且，如果一定数量的锁存（快速中断）连续丢失，则可以使用它来驱动计数器。

通信看门狗

看门狗机制使控制字（由 GDI_DATAINTERFACE_PNET 功能块生成）中的一个位反复打开 250 毫秒再关闭 250 毫秒，以显示驱动器仍在与主控制 PLC 通信。驱动器必须以小于 mint 变量“_nWatchdogTime”的速率检测此功能块的状态更改，该变量默认设置为 1000ms。如果检测到通信中断，则驱动器将自动紧急停止。

用户可以通过将 mint 代码中的第 133 行编辑为新值来更改看门狗时间的值，但不得小于 PLC 库中使用的看门狗切换时间的 2 倍。默认设置为 1000ms；

‘Const _nWatchdogTime As Integer = 1000 'ms’

用户可以通过设置 mint 变量（在 mint 代码的第 132 行）来启用或禁用 GDI 的此功能；

‘_bUseWatchdog = _true’ OR ‘_bUseWatchdog = _false’。

错误处理

如果任何功能块使驱动器出错，则该功能块将显示“Done”和“Error”输出= TRUE。 “ErrorID”输出还将给出与驱动器错误代码相对应的当前错误。

一些常见的错误码如下：

错误代码	说明	详细信息	解决方法
40	轴未启用	此错误是由于在禁用轴时试图启动定位而引起的。在程序中，使用 DRIVEENABLE 来启用轴。	在不启用驱动器的情况下，要求执行一个 GDI 功能块。找出其未启用的原因并继续执行
10000	运动中止	此错误是因使用 ABORT 关键字或中断 Mint 程序引起的。参见 ABORT 和 ABORTMODE。	可能是因 GDI 对诸如看门狗脉冲丢失或非法转换请求（例如，在以速度模式定位时发出位置定位命令）等意外情况作出反应而引起的。
10002	反向硬限位命中	反向限位输入被配置为真	找出是否已设置限位或输入是否打开
10005	超出致命跟随误差	驱动器已超过用户设置的以下致命错误的值。默认情况下，将其设置为 1 个电机转速（第 148 行-Const _fFolErrorFatal As Float = 1）	它有许多可能的原因； 用户设置的致命跟随误差可能太低-尝试增加这个值，或者可能需要改善驱动器调节-在工作台中使用微调，或者轴配置文件无法实现，或者轴有机械阻塞或不良惯性比。
10014	过电流脱扣	轴将紧急停止和停用。	轴需要大于可接受的电流来执行所选择的配置文件。
10033	STO 激活	一个或两个安全转矩取消输入未通电。只有启用驱动器时，才会发生此错误。	检查黄色 STO 端子的接线/检查连接设备的状态

有关错误代码的完整列表，请打开 Workbench 并按“帮助”按钮，然后

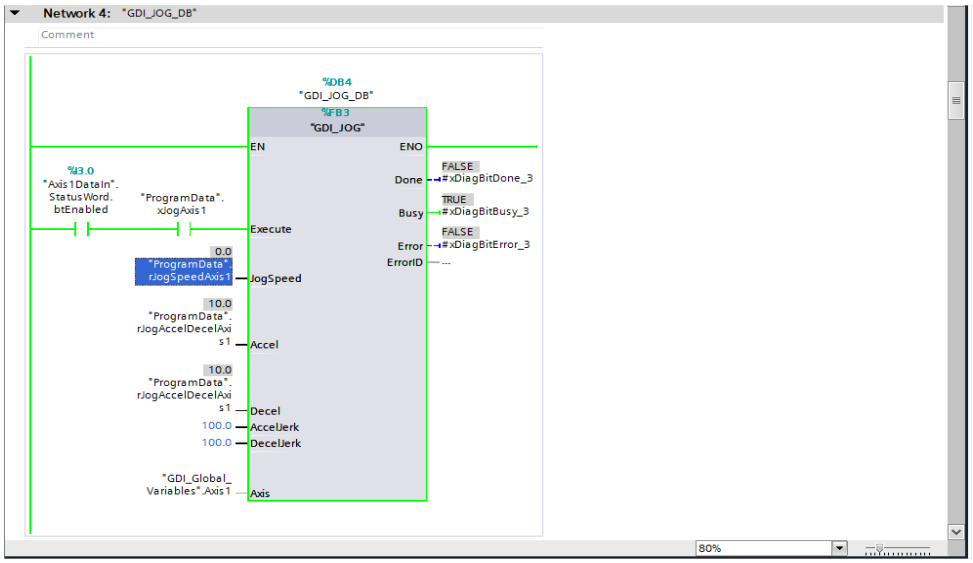
选择“Contents”选项卡>选择“Mint Basic Programming”>“Error Handling”>“Error categories”。

这将为您提供所有错误代码范围的列表。您可以在这里选择“Error categories”并找到错误描述。

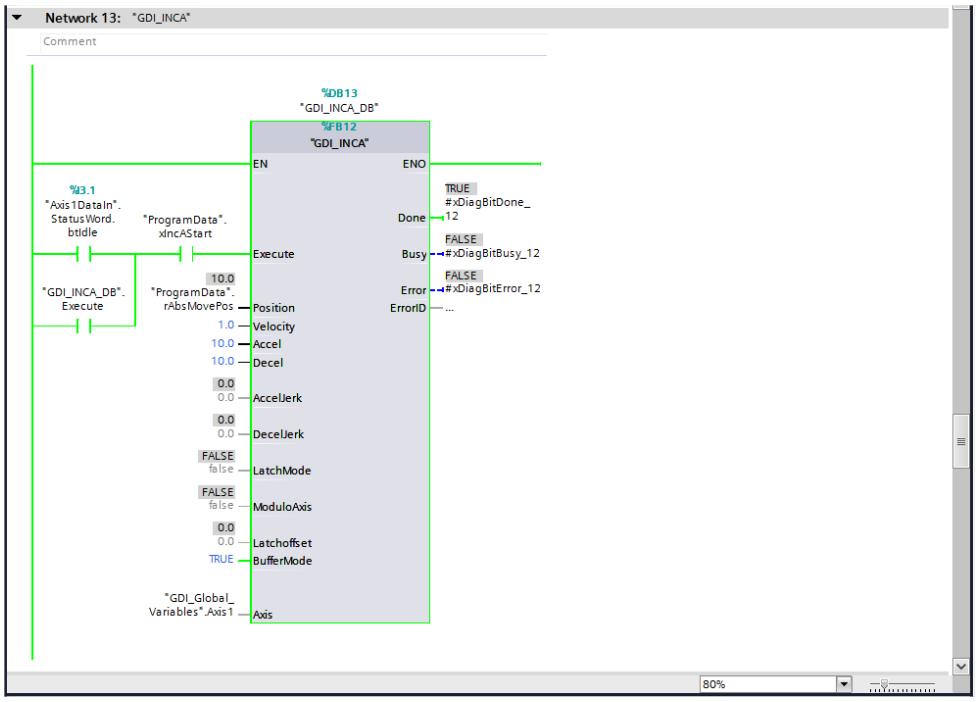
编写 PLC 代码

GDI 库实际上是驱动器内发生的事件序列或联锁的控制接口-如果命令序列错误，则驱动器将无法正确处理命令，并且您将开始收到驱动器生成的错误。始终应该通过驱动器中的 GDI 来处理这个序列。但为了确保这一操作，在 PLC 内部进行一些基本的序列控制总是一个好主意。

在下面的示例中，我们可以看到只有当驱动器状态字显示其已启用并且处理功能块的命令都为 TRUE 时，才允许执行输入。这能确保功能块在驱动器做好准备之前不进行处理。



另一种有用的排序方法是使用驱动器的“空闲”状态作为执行运动功能块的先决条件。只有在轴处于静止状态，而不是正在执行动作并准备好执行新的命令时，“btIdle”位的状态才为 TRUE。如我们所见，我已经使用这个命令和“xIncAStart”命令来控制 Execute 命令。功能块 Execute 输入状态用于维护命令（因为 btidle 在开始定位时将变 FALSE），所以我们会在它处理完成时获得状态输出。



您可以使用任何块模块的“Done”输出而不是“Error”输出来确定它们已经完成且没有错误，因此可以在需要时完成下一条命令。

这些网络都取自示例应用。

示例应用

除了本应用说明附带的库文件外，下载内容还包括一个示例 PLC 应用。示例 PLC 应用的“GDI_DATAINTERFACE_PNET”功能块梯形图中有嵌入式看门狗，用于说明轴结构和每个 GDI 功能块的使用（SIMATIC S7 数据监视器可用于查看相关输入参数，以执行这些功能）。

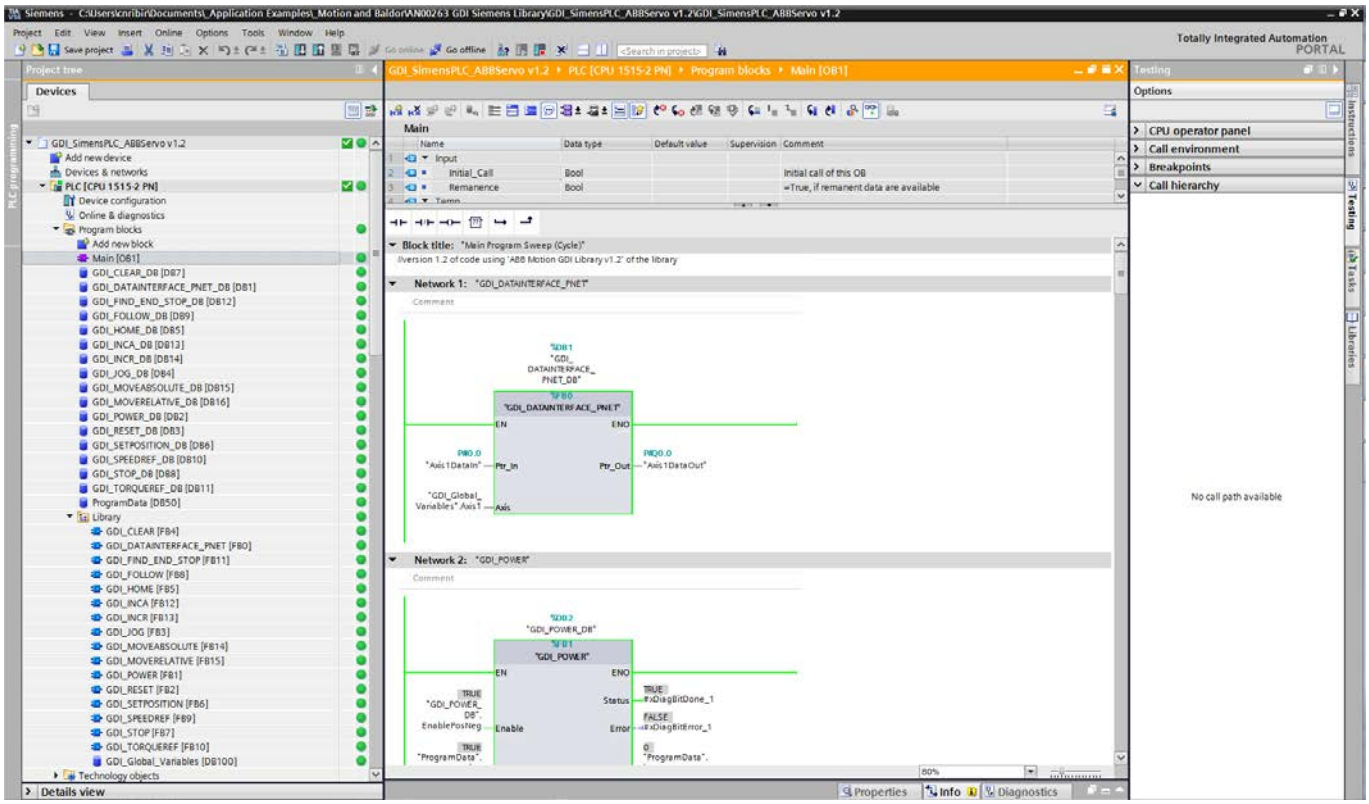


ABB 运动 GDI 库第 1.2 版中新的修改

与第 1.1 版的库相比：

- 添加了 MC_Power “ErrorDesc”输出以帮助调试-它将以字符串形式描述错误。
- 把 ErrorID 的数据类型从 Dword 改为 Dint
- 添加了控制功能，因此现在只有当 JogSpeed 不是 0 时，jog 功能块才会开始处理。但是，在这之后返回到 0 将禁用它。
- 添加了在 Jog 已经正在运行时，重新发送 Accel/Decel Jerk 和 Decel 的功能
- GDIVelRef 是操纵杆类应用的正确选择，但现在使用 GDI_Jog 比以前更好。

联系我们

要了解更多信息，请联系您的当地的 ABB 代表，
或以下一种方式：

new.abb.com/drives/low-voltage-ac/motion
new.abb.com/drives
new.abb.com/channel-partners
new.abb.com/plc