

可使用 PLC 功能块，
结合预先写好的 Mint 应用程序，
通过 Ethernet/IP 对 MicroFlex e150、MicroFlex e190 和 MotiFlex e180 驱动器
实施简单控制



引言

本应用说明详细介绍如何使用 RSLogix 5000™ 第 19 版（或更高版本）导入和配置 ABB 通用驱动器接口（GDI）用户自定义指令（AOI）。对于在 Studio 5000® 下运行的 Logix Designer 的用户，相同原则可能适用。因此，如果使用更高版本的 Rockwell 软件，请在提到 RSLogix5000 时假定是 Logix Designer。这些 AOI 提供预先编写的数据结构和功能块，可与基于 Mint 的 GDI 完美集成，并允许适当的 Allen Bradley PLC 控制运行 Mint 程序（支持 Ethernet/IP）的 ABB 驱动器（MicroFlex e150/e190 和 MotiFlex e180）。请注意，MicroFlex e190 和 MotiFlex e180 驱动器必须配备 Mint 存储卡（选件代码+N8020）。

这些说明确保所有项目的一致性，并极大地简化了需要简单点对点运动的 Allen Bradley PLC 运动控制应用的开发。

本文假设读者具备有关 Allen Bradley PLC、RSLogix5000 或 Studio 5000、Ethernet/IP 配置、Mint Workbench 和 Mint GDI 的基本知识。建议读者阅读应用说明 AN00204，了解 Mint GDI 操作和配置的详细信息。

AOI 以及本应用说明为配备 Allen Bradley Ethernet/IP 的 PLC 提供若干机制，以便：

- 发送一条寻零命令
- 发出一条命令，要求检测物理轴端点，并将其用作基准位置（如果使用 e150 或 e180 驱动器，则需要 5863 以上的固件版本）
- 发送一条相对运动命令
- 发送一条绝对运动命令
- 发送一条增量相对运动命令（可选择在经过“快速捕获”位置一段设定距离后停止）
- 发送一条增量绝对运动命令（可选择在经过“快速捕获”位置一段设定距离后停止）
- 为增量式运动设置一个偏移目标（即相对于捕获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/停用硬件限值
- 清除轴错误
- 在轴上执行一次受控停止或紧急停止
- 调整轴，使其跟随第二个编码器输入
- 为所有运动设置速度、加速时间、减速时间和加加速时间
- 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
- 电机制动状态
- 寻零状态
- 正向限位状态
- 反向限位状态
- 故障状态
- 停止输入状态
- 错过快速锁存中断的指示
- 寻相状态
- 错误代码
- 实测位置
- 实测速度
- 跟随误差
- 轴运行模式
- RMS 电流

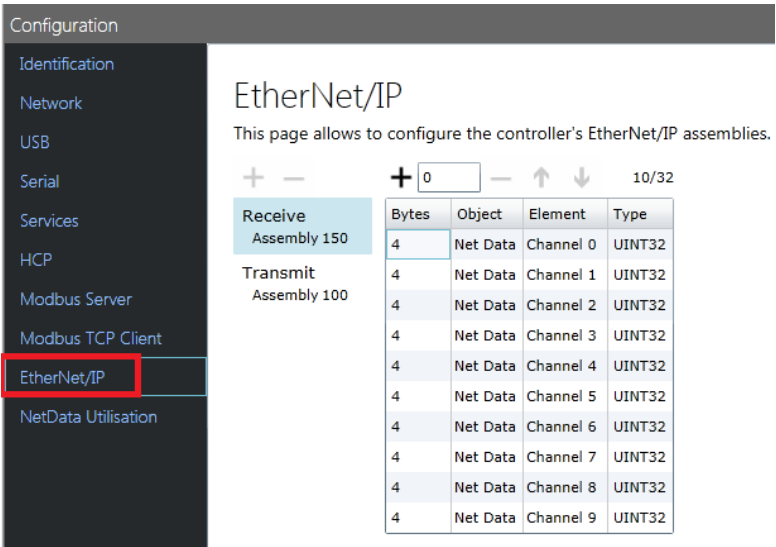
所有这一切都通过作为输入和输出寄存器实现。由于我们使用 32 位数据（DINT 数据类型）作为接口，每个值映射到驱动器上的一个 32 位 NETINTEGER 或 NETFLOAT 位置。
其中还包括可选的看门狗机制，允许驱动器在通讯丢失时采取行动（默认紧急停止）。

配置通用驱动器接口（GDI）Mint 程序

只需要根据用户的应用作少量修改，即可使用 GDI Mint 程序。详细信息请参考应用说明 AN00204。

在基于 Mint 的驱动器上配置 Ethernet/IP

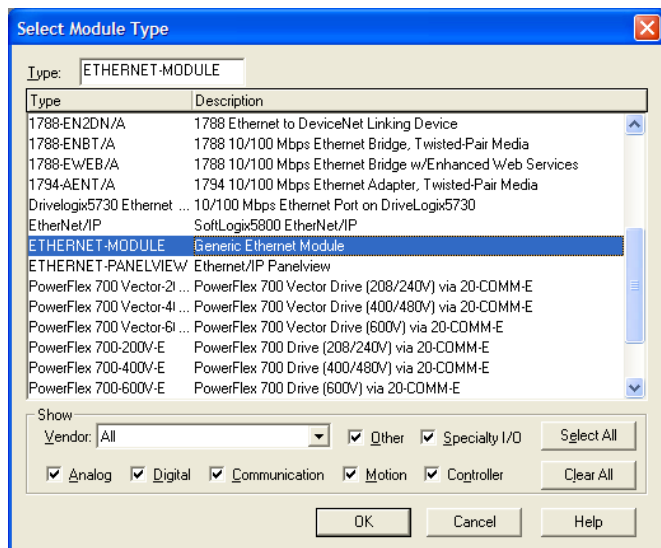
MicroFlex e150 以及启动 Mint 的 MicroFlex e190/MotiFlex e180 驱动器以“预配置”状态交付，可通过 Ethernet/IP 为 GDI 操作。已为 10 个输入字和 10 个输出字配置 Netdata 映射（实际上，GDI 只需要 9 个输入字和 7 个输出字，但各提供 10 个，以方便用户需要时进行扩展）。
如已连接到驱动器，进入 Mint Workbench 中的“配置”屏幕，可看到该配置。从驱动器上传现有配置，并选择“Ethernet/IP”部分...



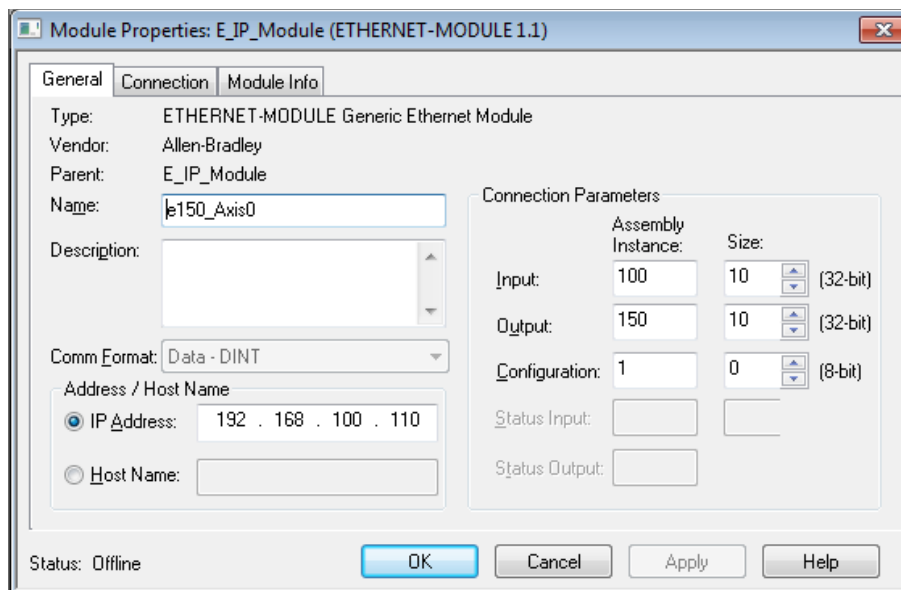
您会发现，程序集 150（驱动器接到的数据）有 10 个映射 Netdata 元素（0 到 9），程序集 100 有 10 个映射 Netdata 元素（100 到 109）（驱动器传输的数据）。对于标准的 GDI 操作，不需要更改任何此类映射。

将驱动器添加到 PLC Ethernet/IP 配置

根据 PLC 型号的不同，这个过程可能会略有不同。在本应用说明中，我们使用带一个 1756-ENET/B Ethernet/IP 模块的 Logix5561™ 处理器（1756-L61）。在 RSLogix5000 中，右键单击 Ethernet/IP 模块，然后选择“New Module（新模块）...”。从可能的以太网设备列表中选择通用以太网模块，如下文所示：



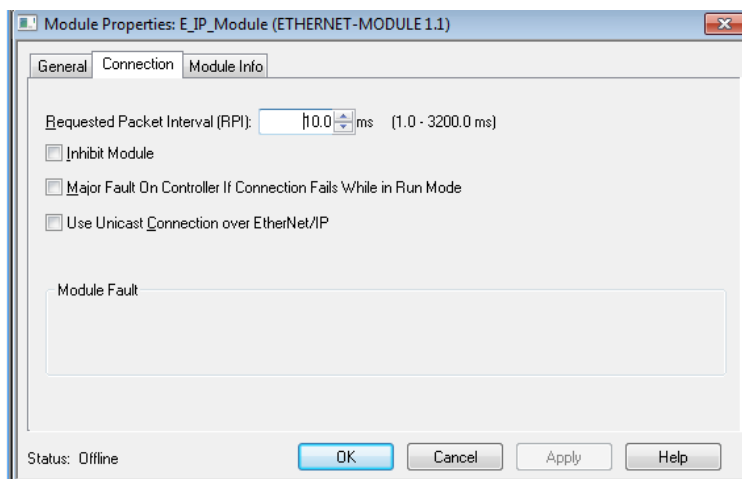
此时，您会被要求输入关于该驱动器的一些基本信息...



给驱动器分配一个可识别的名称。将输入程序集实例设为 100，大小为 10（即与驱动器设置相匹配）。将输出程序集实例设为 150，大小为 10（即与驱动器设置相匹配）。如果您修改驱动器上的映射数据量（例如：添加更多 PDO 值），则这些大小必须与映射数据量完全匹配。为配置程序集实例输入 1，并将其大小设置为 0（即未使用）。

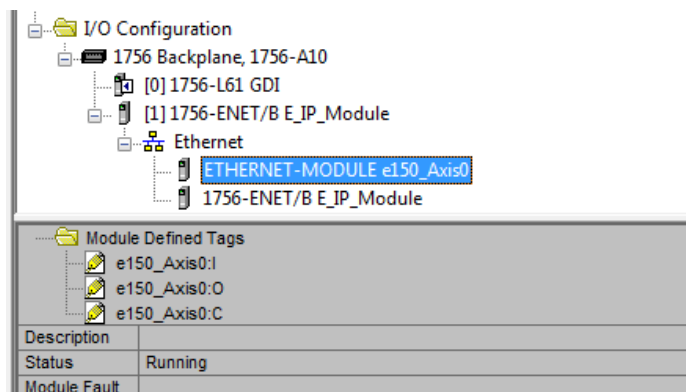
现在，输入驱动器 IP 地址（192.168.0.1 是驱动器的默认 IP 地址，但我们的示例中使用 192.168.100.110-驱动器 IP 地址可以通过 Mint Workbench ‘配置’部分的‘Network’屏幕修改）。任何地址都是可能的，只要它与驱动器设置相匹配，并且前三个八位位组（即 192.168.100）与 PLC 机架中的 Ethernet/IP 模块的设置相匹配（假定用户知道如何在需要时修改 Ethernet/IP 模块地址-如有不确定，请参考 Rockwell 的文件）。我们更改了驱动器 IP 地址，以适应 PLC 的默认子网设置，因为这是最简单的。

如果使用 5853 之前的驱动器固件（如果驱动器运行更高版本的固件，可以根据需要选择单播选项），选择“Connection”选项卡，并确保取消选择“使用 Ethernet/IP 选项上的单播连接”，如下文所示。如有必要，可调整 RPI 间隔时间，但我们发现，10ms 对于大多数应用程序都很适合。



根据需要，重复其它驱动器的该程序（别忘了为每台驱动器分配一个唯一的名称和 IP 地址）。在本应用说明中，我们刚刚增加了一个 MicroFlex e150 驱动器。

此时，您可能需要下载 PLC 项目并转到“Online”，以确定通信是否正常。如果通信正常，您应该在 RSLogix5000 项目树中看到类似这样的内容.....



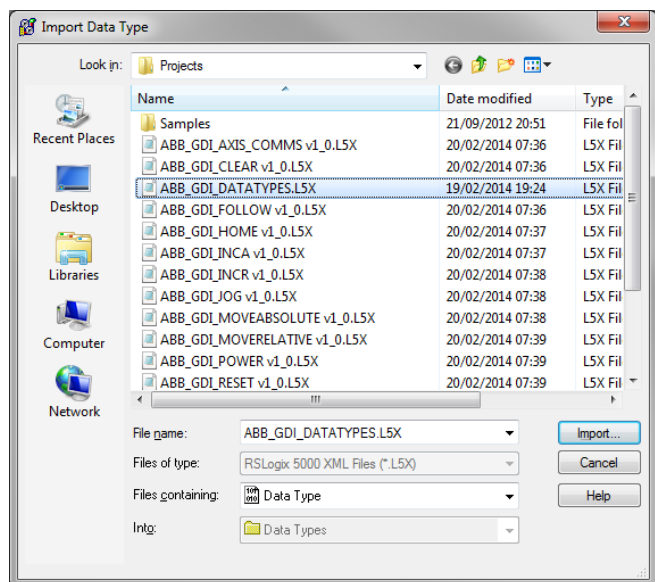
模块状态应指示“正在运行”并且没有故障。如果看不到该状态（例如：显示“I/O Faulted”），请检查您的 Ethernet/IP 模块和通用以太网模块设置。

导入用户定义数据类型和用户自定义指令

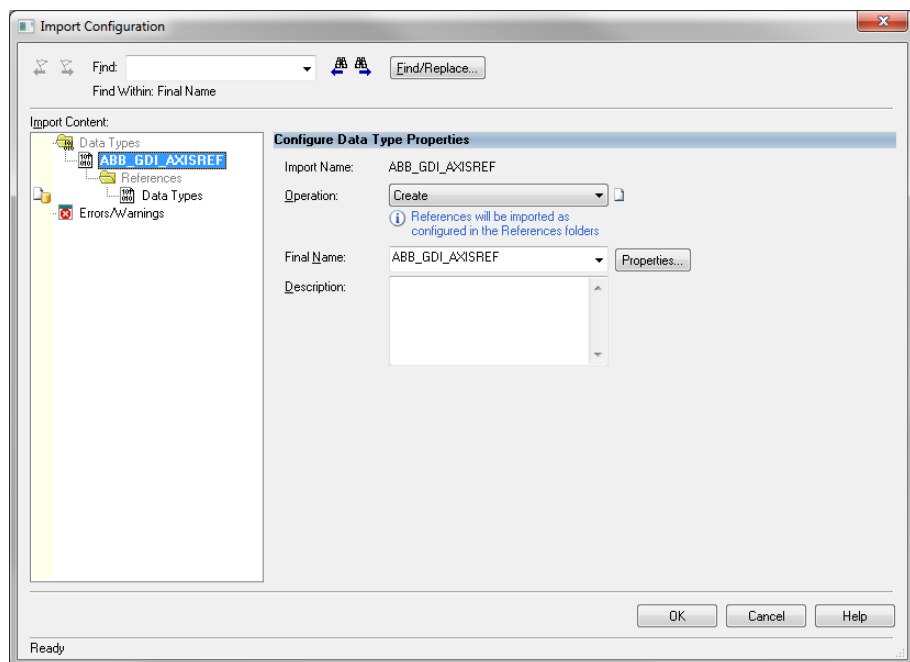
打开 RSLogix 项目中的“Data Types”文件夹，然后右键单击“User-Defined”文件夹...



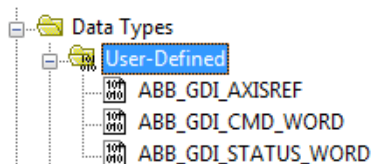
选择“Import Data Type（导入数据类型）...”，然后转到并选择本应用说明中包含的“ABB_GDI_DATATYPES.L5X”文件，如下文本所示...



点击 Import（导入）…按钮。如果项目中没有 GDI 数据类型，您会被要求创建这些数据类型，如下文所示：



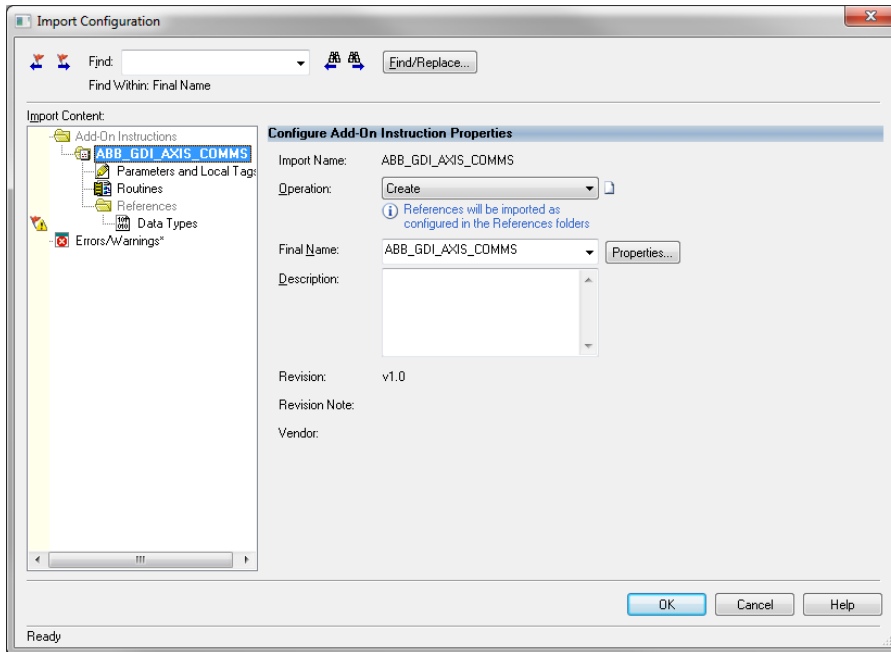
点击 OK。RSLogix 将导入 GDI 操作所需要的所有用户定义数据类型。导入结束时，您的用户定义文件夹应该是这样…



该项目现在包含 GDI 轴的数据类型（ABB_GDI_AXISREF）以及该轴使用的命令和状态字的数据类型。现在右键单击“Add On Instructions”文件夹并选定“Import Add-On Instruction …”，如下文所示…

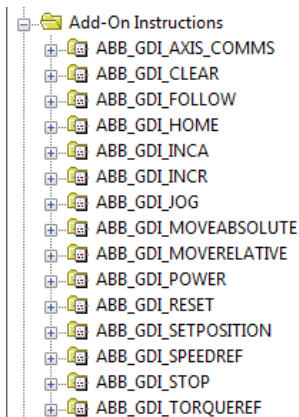


现在转到并选择随本应用说明一起提供的用户自定义指令。ABB_GDI_AXIS_COMMS 指令非常重要，因为它可处理从 PLC 到驱动器的数据传输。对于其余的运动功能，用户可能希望只导入特定应用所需要的功能，但在本示例中，我们选择了所有功能。无法同时选定多个功能，因此必须重复每个功能的导入过程。点击“Import...”以后，点击下一个对话框上的 OK 按钮，创建功能...



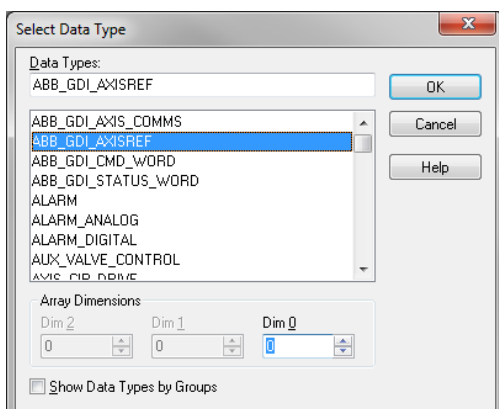
在导入该功能以后，您会看到一条警告。这是可以忽略的，因为函数本身引用了未包含在函数导入中的 ABB_GDI_AXISREF 数据类型。这是因为我们之前/分别导入它-这已经完成。因此，如果将来需要的话，可以单独修改数据类型，无需考虑运动功能。

导入所有运动功能以后，您的 Add On Instructions 文件夹应该看起来像这样（取决于您决定导入哪个运动功能块）...



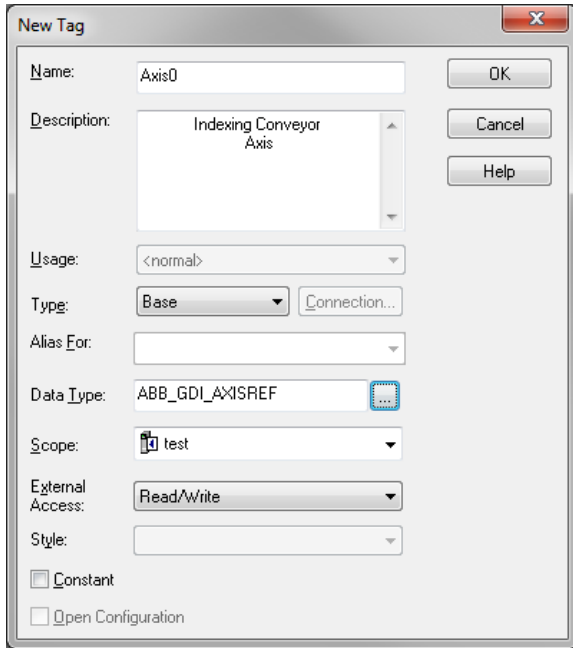
现在，我们可以准备开始使用这些指令在我们 PLC 应用程序内创建一些动作。第一步是在我们的项目中为每个轴创建一个标签。在本应用说明中，我们只涉及一台 MicroFlex e150 驱动器。我们将在 RSLogix5000 内的“Controller Tags”（控制器标签）文件夹中创建该标签（因此它是一个全局变量）。

右键点击 Controller Tags（控制器标签）图标并选择“New Tag（新标签）...”。出现一个对话框，您可为该标签命名，最重要的是为它定义数据类型。我们需要选择我们的 ABB_GDI_AXISREF 作为数据类型，如下文所示...



我们可以为每个轴创建唯一的标签，或者我们可以使用 Dim 0 上/下控件（每个阵列元素都是我们的 ABB_GDI_AXISREF 数据类型）创建一个轴阵列。由于我们只有一个轴，我们只需要创建一个标签。

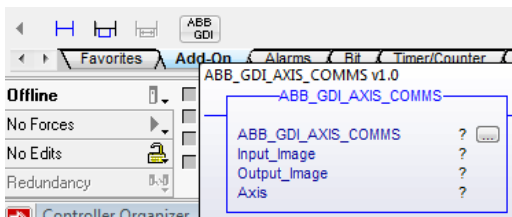
点击 OK，输入一些关于轴标签的其它信息以后，我们结束这个对话框...



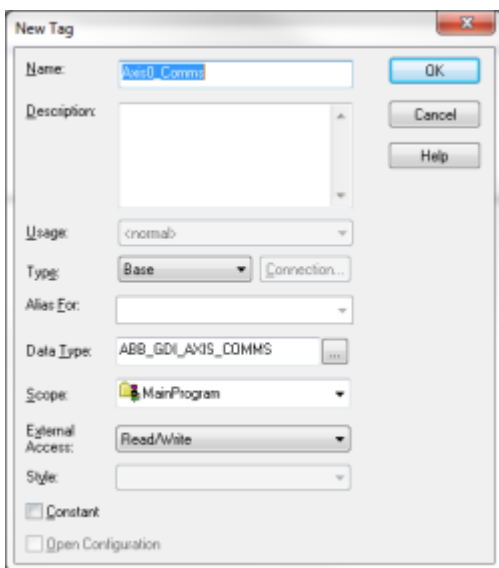
点击 OK，创建这个新标签。我们为这个标签分配的名称是我们从现在开始如何引用轴/驱动器。

现在，我们需要将每个轴的 ABB_GDI_AXIS_COMMS 函数实例包含在我们的应用程序内。

对于本应用说明，我们将只使用 Main Routine 作为我们的程序逻辑。选择第一个梯级，然后从指令工具箱选择与 COMMS 相关的 ABB_GDI 函数（当您鼠标悬停在每个指令按钮上，出现以下功能块图像）。

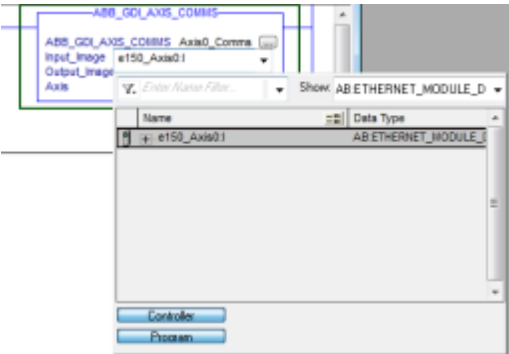


一旦将该功能块添加到梯级中，您需要为该功能块实例创建一个标签（必须为程序中使用的每个功能块实例创建标签）。点击紧邻 ABB_GDI_AXIS_COMMS 的？，输入一个名称（我们称之为 Axis0_Comms，表明它是 0 轴的通信功能）。输入名称以后，右键单击它并选择“New Axis0_Comms”（或您为您的标签指定的名称）。将出现一个对话框，允许您定义这个新标签...



点击 OK，接受默认标签设置。现在，我们需要将通信模块的输入和输出图像链接到 Ethernet/IP 驱动器的输入和输出程序组件。

双击紧邻 Input_Image 的？，然后点击下拉箭头。RSLogix 将显示来自项目的可能输入程序组列表（因为我们只添加了一个驱动器到以太网，我们只能从一个可能项目中选择，如下文所示）...



对于 Output_Image，重复这一过程。最后，我们需要将此功能链接到我们的轴标签，请输入 Axis0（或您之前对 ABB_GDI_AXISREF 标签的称谓）作为轴参数值。如果您严格遵照这些说明，您的第一个程序应该看起来像这样...



现在，我们可以将它下载到 PLC 并测试基本通信是否正常工作。
双击 RSLogix5000 中的 Controller Tags 图标，打开 Monitor Tags 窗口。展开 Axis0 标签，您会看到传出和传入的数据...

Controller Tags - test(controller)								
Scope: test		Show: All Tags						
Name	Value	Force Mask	Style	Data Type	Description	Constant		
Axis0	{...}	{...}		ABB_GDI_AXIS...	Indexing Convey...	<input type="checkbox"/>		
Axis0.CommandWord	{...}	{...}		ABB_GDI_CMD_...	Indexing Convey...			
Axis0.CommandType	0		Decimal	DINT	Indexing Convey...			
Axis0.Value	0.0		Float	REAL	Indexing Convey...			
Axis0.Speed	0.0		Float	REAL	Indexing Convey...			
Axis0.Accel	0.0		Float	REAL	Indexing Convey...			
Axis0.Decel	0.0		Float	REAL	Indexing Convey...			
Axis0.AcceJerk	0.0		Float	REAL	Indexing Convey...			
Axis0.DeceJerk	0.0		Float	REAL	Indexing Convey...			
Axis0.LatchOffset	0.0		Float	REAL	Indexing Convey...			
Axis0.StatusWord	{...}	{...}		ABB_GDI_STAT...	Indexing Convey...			
Axis0.Pos	-466.8025		Float	REAL	Indexing Convey...			
Axis0.Vel	0.0		Float	REAL	Indexing Convey...			
Axis0.FolError	0.0		Float	REAL	Indexing Convey...			
Axis0.AxisMode	0		Decimal	DINT	Indexing Convey...			
Axis0.CurrentMeas	0.024382766		Float	REAL	Indexing Convey...			
Axis0.ErrorCode	10000		Decimal	DINT	Indexing Convey...			
e150_Axis0:C	{...}	{...}		AB:ETHERNET_...		<input type="checkbox"/>		
e150_Axis0:I	{...}	{...}		AB:ETHERNET_...		<input type="checkbox"/>		
e150_Axis0:O	{...}	{...}		AB:ETHERNET_...		<input type="checkbox"/>		

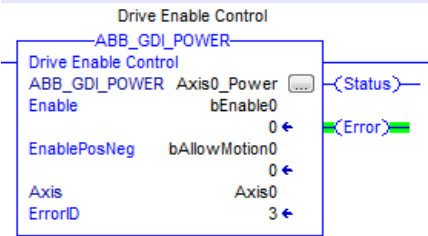
如果 Mint GDI 应用程序正在驱动器上运行，那么驱动器位置、实测速度、实测电流等参数将根据驱动器的实际值进行更新-这可以确认所有部件都正常工作，并且您可以继续在 PLC 上设计您的运动程序。

AOI GDI 功能块

下文详细说明 AOI GDI 功能块的使用：

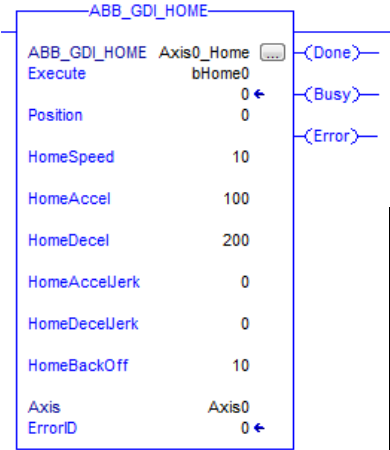
ABB_GDI_POWER

该功能块用于使能/停用轴。使能输入可使能驱动器内的功率级，而不是功能块本身。



	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Enable	BOOL	如果为 True，PLC 将要求启用轴
EnablePosNeg	BOOL	如果为 True，允许两个方向的运动。如果为 false，运动被阻止（或在运动开始以后执行一个停车命令）
VAR_OUTPUT		
Status	BOOL	指示轴是使能（1）还是未使能（0）
Error	BOOL	如果轴出错误，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

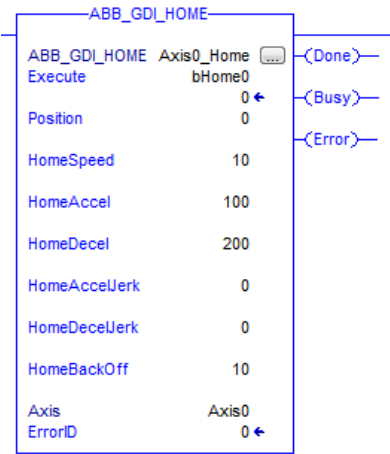
ABB_GDI_HOME



该功能块用作轴的寻零。寻零序列详情取决于 Mint GDI 程序中设置的寻零类型。位置输入用于在一个成功的寻零序列结束时设置轴位置。

	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	在上升沿启动寻零序列
Position	REAL	需要在成功的寻零序列结束时设置的绝对位置
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	寻零加速度曲线光滑度，用户单位/秒 ³ （对于梯形运动，设置为 0）
HomeDecelJerk	REAL	寻零减速度曲线光滑度，用户单位/秒 ³ （对于梯形运动，设置为 0）
HomeBackOff	REAL	寻零速度与回退速度的比率
VAR_OUTPUT		
Done	BOOL	指示轴已经成功寻零。如果在寻零过程中 Execute 输入被移除，并且轴完成了寻零序列，将设置一次 PLC 扫描周期的 Done 输出。如果 Execute 输入仍然是 1，则 Done 输出将保持设置状态（在寻零成功的前提下）。
Busy	BOOL	如果寻零序列正在进行，设为真
Error	BOOL	如果轴出错，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_FIND_END_STOP

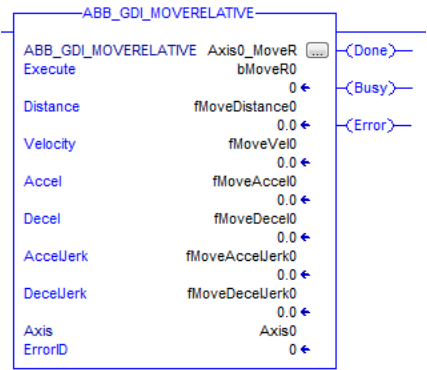


没有内部传感器时，该功能块用作一个轴的寻零。轴将以指定速度运行，并带有预设转矩极限，直到达到该转矩极限并且轴速度小于预设空闲速度。位置输入用于在一个成功的寻零序列结束时设置轴位置。

		型号	说明
VAR_IN_OUT			
Axis	ABB_GDI_AXISREF	参考轴结构	
VAR_INPUT			
Execute	BOOL	在上升沿启动寻零序列	
Position	REAL	需要在一个成功的寻零序列结束时设置的绝对位置	
FindSpeed	REAL	速度（用户单位/秒）（该值的符号决定查找方向）	
FindAccel	REAL	加速度，用户单位/秒 ²	
FindDecel	REAL	减速度，用户单位/秒 ²	
FindAccelJerk	REAL	加速度曲线光滑度（用户单位/秒 ³ ）（对于梯形运动，设置为 0）	
FindDecelJerk	REAL	减速度曲线光滑度（用户单位/秒 ³ ）（对于梯形运动，设置为 0）	
TorqueLimit	REAL	序列执行过程中的转矩极限（驱动器额定电流的%）	
VAR_OUTPUT			
Done	BOOL	显示轴是否已经成功找到终点停止位置。如果 Execute 输入在此过程中被移除，并且轴找到终点，将设置一个 PLC 扫描周期的 Done 输出。如果 Execute 输入仍然是 1，则 Done 输出将保持设置状态（如果过程顺利）。	
Busy	BOOL	如果查找序列正在进行，设为 True	
Error	BOOL	如果轴出错，设为 True	
ErrorID	DINT	指示轴报告的 Mint 错误代码	

ABB_GDI_MOVERELATIVE

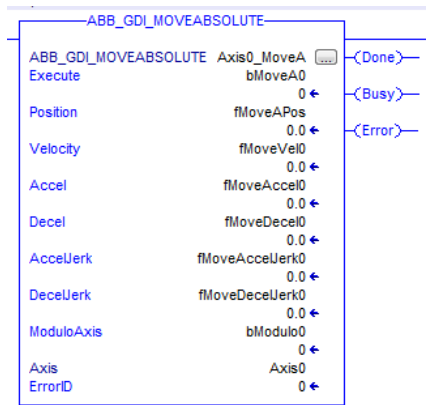
该功能块控制相对于起始位置的指定距离的受控运动。



	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	上升沿启动运动
Distance	REAL	运动的相对距离（用户单位）
Velocity	REAL	最高速度（无需达到）（用户单位/秒）
Accel	REAL	加速度，用户单位/秒 ²
Decel	REAL	减速度，用户单位/秒 ²
AccelJerk	REAL	加加速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）
DecelJerk	REAL	减减速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）
VAR_OUTPUT		
Done	BOOL	显示轴是否已经成功找到目标位置。如果 Execute 输入在运动过程中被移除，并且相对移动完成，将设置一个 PLC 扫描周期的 Done 输出。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	在相对运动正在进行时，设置为 True
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_MOVEABSOLUTE

该功能块发出一条命令，控制至指定绝对位置的受控运动。该功能可结合旋转轴使用（在这种情况下，将采用距指定位置最近的路线）。

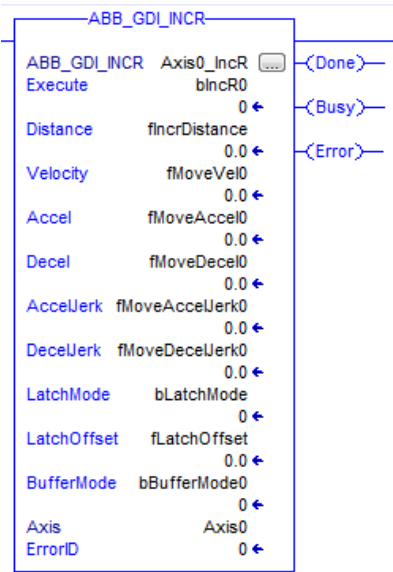


		型号	说明
VAR_IN_OUT			
Axis	ABB_GDI_AXISREF	参考轴结构	
VAR_INPUT			
Execute	BOOL	上升沿启动运动	
Position	REAL	运动的目标位置（用户单位）	
Velocity	REAL	最高速度（无需达到）（用户单位/秒）	
Accel	REAL	加速速率，用户单位/秒 ²	
Decel	REAL	减速速率，用户单位/秒 ²	
AccelJerk	REAL	加加速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）	
DecelJerk	REAL	减减速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）	
ModuloAxis	BOOL	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运动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 度的绝对运动将引发一次 30 度的向前运动）	
VAR_OUTPUT			
Done	BOOL	显示轴是否已经成功找到目标位置。如果在运动过程中 Execute 输入被移除，并且绝对定位完成，将设置一个 PLC 扫描周期的 Done 输出。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	在绝对运动进行时设置为 True	
Error	BOOL	如果轴出错，设为 True	
ErrorID	DINT	指示轴报告的 Mint 错误代码	

ABB_GDI_INCR

该功能块用于发出命令，以控制相对于目标位置的指定距离的受控运动。当运动仍在进行时，可通过以下任何一种方法修改本功能块的目标位置：

- a. 发送另一个 ABB_GDI_INCR 或 ABB_GDI_INCA 函数（如果输入参数 BufferMode 为 True）
- b. 把输入参数 Latchmode 设置为 True，并为输入参数 LatchOffset 指定一个值。然后，驱动器上的 Mint 代码将自动修改轴目标位置，确保在通过快速中断所捕获的轴位置时停止 LatchOffset。可使用轴状态字中的一个位（btLatchMissed）来指示未能检测到快速中断（然后，可利用这种情况告诉操作员系统有故障）。使用 Latchmode 和 LatchOffset，可简化分度输送机应用的执行。



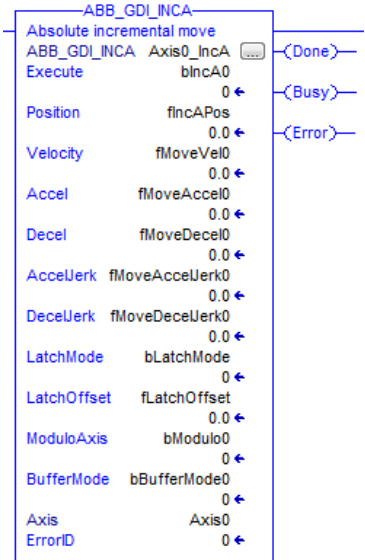
	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	上升沿启动运动
Distance	REAL	相对运动距离（用户单位）
Velocity	REAL	最高速度（无需达到）（用户单位/秒）
Accel	REAL	加速率，用户单位/秒 ²
Decel	REAL	减速率，用户单位/秒 ²
AccelJerk	REAL	加加速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）
DecelJerk	REAL	减减速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）
LatchMode	BOOL	确定轴是否应该利用配置的快速锁存中断，并在与捕获的位置距离为“LatchOffset”个用户单位的地方设置新目标位置
LatchOffset	REAL	确定通过用以修改 GDI_INCR 目标的捕获快速位置（用户单位）的距离（当输入参数 LatchMode 设为 True）
BufferMode	BOOL	确定功能块是否应该设置 Done 输出，并在运动载入时尽快完成。设置 BufferMode True，允许应用在目前的运动正在进行时触发更多的增量运动。
VAR_OUTPUT		
Done	BOOL	当 BufferMode 设为 False 时，表明轴已成功到达目标位置。如果运动过程中 Execute 输入被移除，并且相对运动完成，将设置一个 PLC 扫描周期的 Done 输出。如果 Execute 输入仍然为 True，则 Done 输出也将保持设置状态（在成功到达目标位置的前提下）。当 BufferMode 设为 True，将设置一个 PLC 扫描周期的 Done 输出，表明运动已经成功载入。
Busy	BOOL	当功能块正在进行，设置为 True
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

如果应用要求修改正在进行的相对运动的 SPEED/ACCEL/DECEL，ABB_GDI_INCR 也很有用。使用 ABB_GDI_MOVERELATIVE 加载运动的 SPEED/ACCEL/DECEL 时，当运动开始后，这些无法修改。使用 ABB_GDI_INCR（输入参数 BufferMode 设为 True），可通过载入另一个 ABB_GDI_INCR（带新 SPEED/ACCEL/DECEL，输入参数 Distance 设为零），修改曲线参数。

ABB_GDI_INCA

该功能块用以控制至指定绝对位置的一次受控运动。该功能与 ABB_GDI_MOVEABSOLUTE 的不同之处在于，它可在运动过程中采用下列任何一种方式改变目标位置：

- 发送另一个 ABB_GDI_INCR 或 ABB_GDI_INCA 函数（如果输入参数 BufferMode 为 True）
- 把输入参数 Latchmode 设为 True，并为输入参数 LatchOffset 指定一个值。然后，驱动器上的 Mint 代码将自动修改轴目标位置，确保在通过快速中断所捕获的轴位置 LatchOffset 距离后停止。可使用轴状态字中的一个位（bLatchMissed）来指示未能检测到快速中断（示例程序说明如何能够检测到连续三次锁存失败 - 例如：可利用这种情况提醒操作人员存在系统故障）。

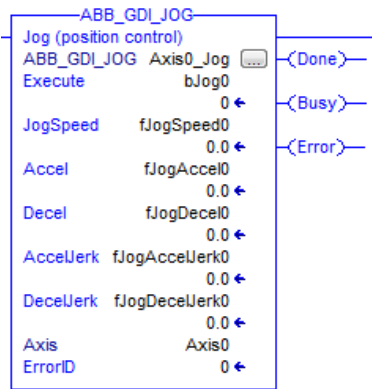


	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	上升沿启动运动
Position	REAL	运动的绝对位置目标（用户单位）
Velocity	REAL	最高速度（无需达到）（用户单位/秒）
Accel	REAL	加速度，用户单位/秒 ²
Decel	REAL	减速度，用户单位/秒 ²
AccelJerk	REAL	加加速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）
DecelJerk	REAL	减减速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）
LatchMode	BOOL	确定轴是否应该利用配置的快速锁存中断，并在与捕获的位置距离为“LatchOffset”个用户单位的地方设置新目标位置
ModuloAxis	BOOL	确定轴是否为旋转轴（即使用 ENCODERWRAP 来定义一个周期内的行程）。使用旋转轴时的绝对运动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 度的绝对运动将引发一次 30 度的向前运动）
LatchOffset	REAL	确定通过用以修改 ABB_GDI_INCA 目标的捕获快速位置（用户单位）的距离（当输入参数 LatchMode 设为 True）
BufferMode	BOOL	确定功能块是否应该设置 Done 输出，并在运动载入后立即完成。设置 BufferMode 为 True，允许应用在目前的运动正在进行时触发更多的增量运动
VAR_OUTPUT		
Done	BOOL	在 BufferMode 设为 False，表明轴已经成功到达目标位置。如果在运动过程中 Execute 输入被移除，并且相对运动完成，将设置一个 PLC 扫描周期的 Done 输出。如果 Execute 输入仍然为 True，则 Done 输出也将保持设置状态（在成功到达目标位置的前提下）。在 BufferMode 设为 True，将设置一个 PLC 扫描周期的 Done 输出，表明运动已经成功载入。
Busy	BOOL	当功能块正在进行，设置为 True
Error	BOOL	如果出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

如果应用要求修改正在进行的绝对定位的 SPEED/ACCEL/DECEL，也可使用 ABB_GDI_INCA。使用 ABB_GDI_MOVEABSOLUTE 载入的运动，运动开始以后无法修改 SPEED/ACCEL/DECEL。使用 ABB_GDI_INCA（输入参数 BufferMode 设为 True）修改曲线参数，可首先载入一个 ABB_GDI_INCA 运动、然后载入另一个 ABB_GDI_INCR（带新 SPEED/ACCEL/DECEL，输入参数 Position 设为零）实现。

ABB_GDI_JOG

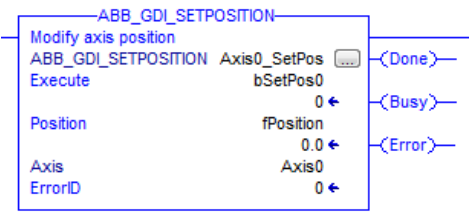
该功能块用于控制轴上的一次恒速运动（使用驱动器内的位置回路控制器）。只要 Execute 输入保持 True，则会执行运动。



	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	在上升沿开始运动，只要输入保持 True，则保持运动。当 Execute 变为 False，运动以配置的减速度逐渐下降到零速。
JogSpeed	REAL	轴将达到的速度值，用户单位/秒
Accel	REAL	加速度，用户单位/秒 ²
Decel	REAL	减速度，用户单位/秒 ²
AccelJerk	REAL	加加速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）
DecelJerk	REAL	减减速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）
VAR_OUTPUT		
Done	BOOL	在成功发送点动命令后立即设置为 True 并保持该设置，直到 Execute 变为 False 或出现轴错误
Busy	BOOL	当功能块正在进行，设置为 True
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_SETPOSITION

该功能块用于将轴位置（驱动器上的编码器和位置值）设置为整定值。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，则会设置/发送一个新绝对位置（GDI Mint 程序 V2.17 及更新版本）。



	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
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）
VAR_OUTPUT		
Done	BOOL	在发送命令后立即设置为 True（无论其是否成功-使用错误输出来确定命令是否成功）。保持 True，直到 Execute 输入被移除。如果 Execute 输入在设置 Done 位以前被移除，则设置一个 PLC 周期的 Done 位。
Busy	BOOL	当功能块正在运行，设置为 True（设置 Done 位以后立即清除）
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_STOP

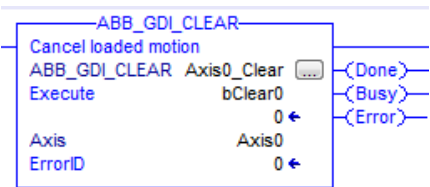
本功能块按照预设的减速度在轴上执行有控制的停止。



	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	在上升沿开始有控制的停止
Decel	REAL	减速度，用户单位/秒 ²
DecelJerk	REAL	减减速（用户单位/秒 ³ ）（对于梯形运动，设置为 0）
VAR_OUTPUT		
Done	BOOL	在完成受控停止以后，当轴空闲时，或者如果在发送停止命令时出错，则设置为 True。保持 True，直到 Execute 输入被移除。如果 Execute 输入在设置 Done 位以前被移除，则设置一个 PLC 周期的 Done 位。
Busy	BOOL	当停止正在进行，设置为 True-设置 Done 位后立即清除
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_CLEAR

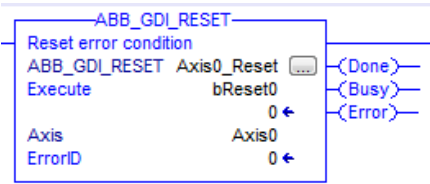
本功能块用于紧急停止轴，并中断任何正在进行的运动。轴将保持使能（如果 ABB_GDI_POWER 要求使能状态，并且轴没有出错）。



	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	在上升沿启动紧急停止
VAR_OUTPUT		
Done	BOOL	在完成紧急停止以后，当轴空闲时，或者如果在发送紧急停止命令时出错，则设置为 True。保持 True，直到 Execute 输入被移除。如果 Execute 输入在设置 Done 位以前被移除，则设置一个 PLC 周期的 Done 位。
Busy	BOOL	当停止正在进行，设置为 True - 在设置 Done 位以后立即清除
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_RESET

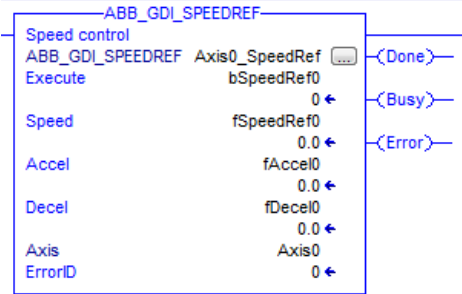
本功能块用于清除任何轴错误。



	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	在上升沿启动故障清除
VAR_OUTPUT		
Done	BOOL	在轴错误已清除，设置为 True。保持 True，直到 Execute 输入被移除。如果 Execute 输入在设置 Done 位以前被移除，则设置一个 PLC 周期的 Done 位。如果无法清除错误，将不会设置 Done 位（利用 Busy 输出检测何时开始故障清除）
Busy	BOOL	当功能块正在尝试清除任何轴错误，设置 True
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_SPEEDREF

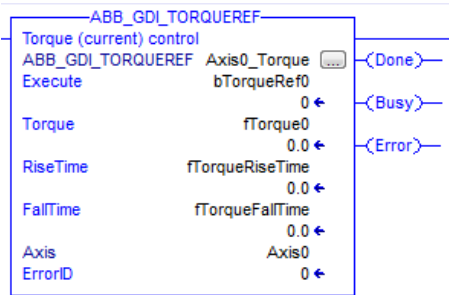
本功能块用于控制轴上的速度/参考速度。在该操作模式下，驱动器上不使用位置回路（因此不会记录跟随误差或者就此采取行动）。轴将保持在速度控制模式（由 Controlmode 的 Statusword 位所指示），直到发送另一种控制模式的运动（比如位置控制运动）。要从零速运行（速度控制模式）切换到保持位置（位置控制模式），可发送一个 ABB_GDI_MOVERELATIVE，相对运动距离为零个用户单位。



	型号	说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	在上升沿启动轴，只要输入保持 True，则保持运动。当 Execute 变为 False，运动以配置的减速度逐渐下降到零速。
Velocity	REAL	轴将达到的速度值，用户单位/秒。可在 Execute 为 True 时修改，以改变轴的速度
Accel	REAL	加速度，用户单位/秒 ²
Decel	REAL	减速度，用户单位/秒 ²
VAR_OUTPUT		
Done	BOOL	在发送参考速度以后，立即设置为 True（无论其是否成功）。Done 输出保持设置状态，直到 Execute 变为 False
Busy	BOOL	当功能块正在进行（即在 Execute 为 True 时），设置为 True
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_TORQUEREF

本功能块用于控制轴上的一个转矩（电流）参考值。在该操作模式下，驱动器上不使用位置回路（因此不会记录跟随误差或就此采取行动）。轴将保持在转矩控制模式（由 Controlmode 的 Statusword 位所指示），直到发送另一种控制模式的运动（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），可发送一个 ABB_GDI_MOVERELATIVE，相对运动距离为零个用户单位。



型号		说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	在上升沿启动参考转矩，只要输入保持 True，则保持转矩。当 Execute 变为 False，转矩以配置的 FallTime 逐渐速度下降到零。
Torque	REAL	轴将使用的转矩参考值（DRIVERATEDCURRENT 的百分比-详见 Mint 帮助文件）。当 Execute 为 True，可进行修改，以改变所产生的转矩
RiseTime	REAL	设置电流从零上升到 DRIVEPEAKCURRENT 所需的时间（ms）（详见 Mint 帮助文件）
FallTime	REAL	设置电流从 DRIVEPEAKCURRENT 下降到零所需的时间（ms）（详见 Mint 帮助文件）
VAR_OUTPUT		
Done	BOOL	发送参考转矩以后，立即设置为 True（无论其是否成功）。Done 输出保持设置状态，直到 Execute 变为 False。
Busy	BOOL	当功能块正在进行（即当 Execute 为 True 时），设置为 True
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_FOLLOW

该功能块用于操纵轴，使之开始以预设跟随比跟随主编码器给定（e100 驱动器仅支持位置跟随，e150 驱动器有其它跟随模式）。



型号		说明
VAR_IN_OUT		
Axis	ABB_GDI_AXISREF	参考轴结构
VAR_INPUT		
Execute	BOOL	在上升沿启动跟随，当 Execute 输入变为 False，轴将保持跟随模式（要停止跟随，则发送另一条运动命令或使用 GDI_CLEAR 清除运动）
Ratio	REAL	轴与主编码器给定之间跟随（关联）比（值将受到轴缩放比例和主编码器缩放比例的影响-详见 Mint 帮助文件中有关 FOLLOW 的主题）。要在跟随过程中设置一个新比率，需要发送一条新 ABB_GDI_FOLLOW 命令。
VAR_OUTPUT		
Done	BOOL	跟随命令处理后，立即设置为 True（无论其是否成功）。Done 输出保持设置状态，直到 Execute 变为 False。
Busy	BOOL	当功能块正在进行（即当 Execute 为 True 时），设置为 True
Error	BOOL	如果轴出错，设为 True
ErrorID	DINT	指示轴报告的 Mint 错误代码

ABB_GDI_AXIS_COMMS

该功能块用于 PLC 程序的应用层与通信层之间的命令/状态数据传送。对于应用中的每条轴，必须有相关功能块的实例。



	型号	说明
VAR_IN_OUT		
Input_Image	AB:ETHERNET_MODULE_DINT_40Bytes:I:0	参考轴的输入程序集实例
Output_Image	AB:ETHERNET_MODULE_DINT_40Bytes:O:0	参考轴的输出程序集实例
Axis	ABB_GDI_AXISREF	参考轴结构

使用轴结构

GDI 的多数功能都封装在用户自定义指令提供的各种 GDI 功能块中。但在某些情况下，访问轴结构数据可能会很有用。ABB_GDI_AXISREF 数据类型结构如下：

	Name	Data Type	Style	Description	External Access
	Command/word	ABB_GDI_CMD_WORD			Read/Write
	CommandType	DINT	Decimal		Read/Write
	Value	REAL	Float		Read/Write
	Speed	REAL	Float		Read/Write
	Accel	REAL	Float		Read/Write
	Decel	REAL	Float		Read/Write
	AccelJerk	REAL	Float		Read/Write
	DecelJerk	REAL	Float		Read/Write
	LatchOffset	REAL	Float		Read/Write
	Status/word	ABB_GDI_STATUS_WORD			Read/Write
	Pos	REAL	Float		Read/Write
	Vel	REAL	Float		Read/Write
	FolError	REAL	Float		Read/Write
	AxisMode	DINT	Decimal		Read/Write
	CurrentMeas	REAL	Float		Read/Write
	ErrorCode	DINT	Decimal		Read/Write

该数据结构包含另外两种数据结构（ABB_GDI_CMD_WORD 和 ABB_GDI_STATUS_WORD）。它们的声明如下：

命令字

	Name	Data Type	Style	Description	External Access
	btEnable	BOOL	Decimal	Enable	Read/Write
	btMotionAllowed	BOOL	Decimal	Allow Motion	Read/Write
	btPosLatchEnable	BOOL	Decimal	Enable Position Latch	Read/Write
	btDisFwdLimit	BOOL	Decimal	Disable Fwd Limit	Read/Write
	btDisRevLimit	BOOL	Decimal	Disable Reverse Limit	Read/Write
	btModulo	BOOL	Decimal	Modulo Axis	Read/Write
	btFaultReset	BOOL	Decimal	Reset Fault	Read/Write
	btTriggerCmd	BOOL	Decimal	Trigger Motion Cmd	Read/Write
	btWatchDog	BOOL	Decimal	Watchdog	Read/Write
	btIgnoreFE	BOOL	Decimal	FolErrorMode	Read/Write

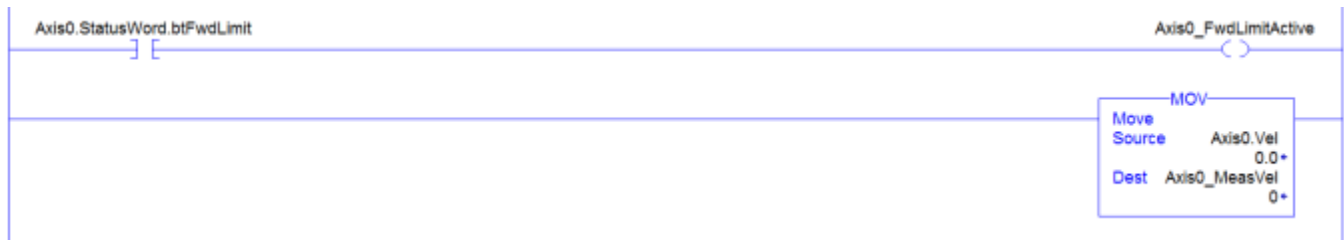
状态字

	Name	Data Type	Style	Description	External Access
	btEnabled	BOOL	Decimal		Read/Write
	btIdle	BOOL	Decimal		Read/Write
	btInPos	BOOL	Decimal		Read/Write
	btBrakeEngaged	BOOL	Decimal		Read/Write
	btHomed	BOOL	Decimal		Read/Write
	btFwdLimit	BOOL	Decimal		Read/Write
	btRevLimit	BOOL	Decimal		Read/Write
	btFault	BOOL	Decimal		Read/Write
	btStopInput	BOOL	Decimal		Read/Write
	btReadyToEnable	BOOL	Decimal		Read/Write
	btControlMode0	BOOL	Decimal		Read/Write
	btControlMode1	BOOL	Decimal		Read/Write
	btTriggerDone	BOOL	Decimal		Read/Write
	btPermitted	BOOL	Decimal		Read/Write
	btLatchMissed	BOOL	Decimal		Read/Write
	btFaultReset	BOOL	Decimal		Read/Write
	btPhaseSearchDone	BOOL	Decimal		Read/Write

因此，可通过这些结构访问任意数据。

例如：

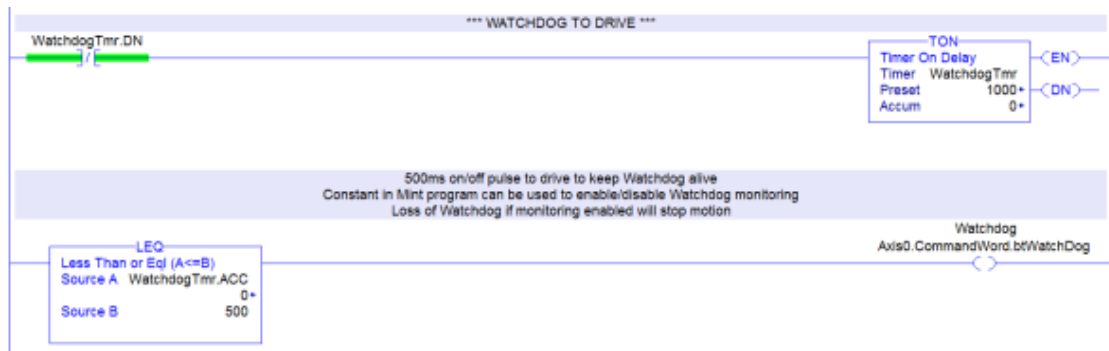
两条梯形图语句直接访问轴数据结构，一个读取驱动器上的 Forward Limit Input 状态，另一个存储实测轴速度...



由于可直接存取这些数据，PLC 应用程序代码拥有了很大的灵活性（例如，对于分度输送机应用，PLC 应用可以访问锁存丢失状态位【btLatchMissed】；当一定数量的锁存器（快速中断）接连丢失，可使用一个计数器使运动停止。）

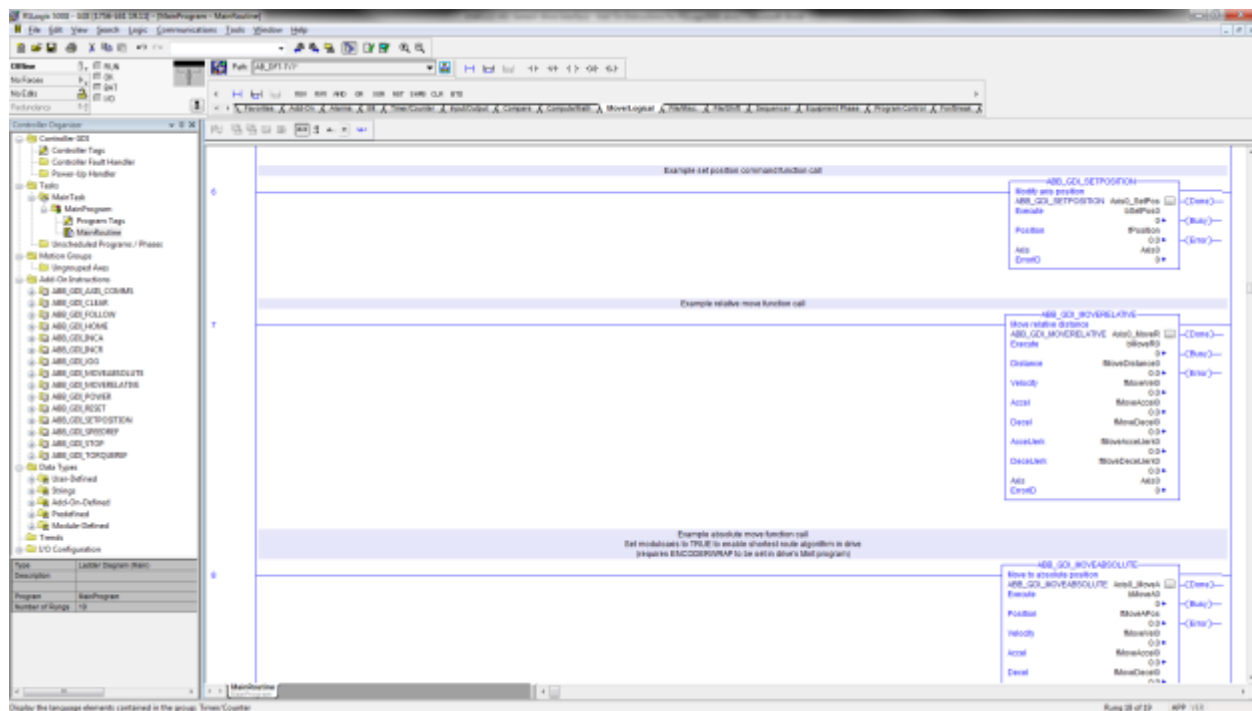
通信看门狗

默认情况下, Mint GDI 被配置为使用看门狗机制。从收到来自 PLC 的第一条消息起, Mint 程序检查通信是否仍然有效。如果发生丢失, 轴将停止转动, 直到错误被清除, 才允许进一步的运动。可禁用看门狗 (详见 AN00204), 但为了保证完整性, PLC 看门狗实例如下文所示 (它包含在本应用说明包含的 PLC 应用实例中)。



实例应用

除本应用说明中包含的 AOI 文件以外, 下载还包含一个 PLC 应用程序实例。PLC 应用程序实例包含看门狗实例、轴结构直接访问实例以及每个 GDI 功能块的实例 (RSLogix5000 数据监视可用于将数据插入相关输入参数中来执行这些功能)。



联系我们

要了解更多信息, 请您联系当地的 ABB 代表或访问下列网站:

www.abb.com/motion
www.abb.com/drives
www.abb.com/drivespartners
www.abb.com/PLC

EtherNet/IP 是 Open DeviceNet Vendor Association 的商标。

RSLogix5000、Logix Designer 和 Studio 5000 是罗克韦尔软件公司的商标。

© ABB 公司, 2014 年, 版权所有。保留所有权利。
 技术规格如有变更, 恕不另行通知。