运动控制产品

应用说明文档
通用驱动接口使用 Modbus TCP 的 B&R PLC

AN00265

版本 A（中文）

现成的 PLC 功能块，与预编写的 Mint 应用程序联用，轻松通过 Modbus TCP 控制 MicroFlex e190 和 MotiFlex e180 驱动

简介

本应用说明书举例详细说明了包含库功能，使 B&R X20 PLC 能够通过 Modbus TCP 控制与监测 ABB MicroFlex e190 和/或 MotiFlex e180 AC 伺服驱动的 Automation Studio 项目。该库中预编写的数据结构和功能块与基于 Mint 的 GDI 无缝集成，使用户能够编写控制一系列驱动运动的基于 IEC61131 的代码。请注意，订购 MicroFlex e190 和 MotiFlex e180 驱动时，必须同时订购 Mint 内存卡（选件码：+N8020）。

本说明书促进了所有项目的一致性，极大地简化了要求简单点对点运动的 B&R PLC 运动控制应用程序的开发。

本文件假设读者对 B&R PLC、Automation Studio、Mint Workbench 和 Mint GDI 已有基本的了解。建议读者参阅应用说明书 AN00204，详细了解 Mint GDI 的运行与配置。

本应用说明书描述的项目向 B&R X20 PLC（X20CP0410，使用 X20BB52 底座）提供执行以下操作的机制：

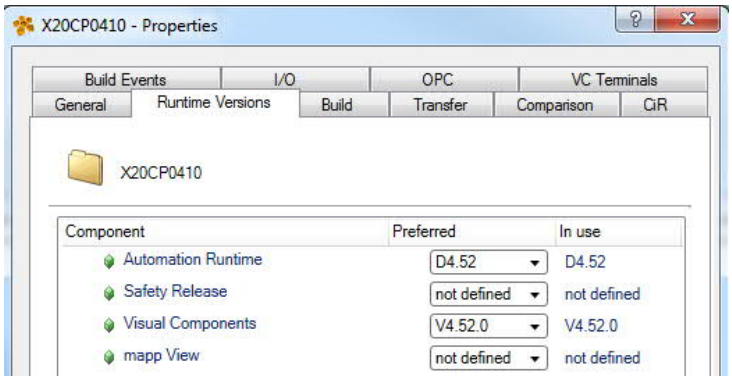
- 发送回位命令
- 发送检测物理轴最终停止位置的命令以及将该位置作为基准位置（需要 5863 及以上版本的驱动固件）
- 发送相对运动命令
- 发送绝对运动命令
- 发送增量相对运动命令（以及选择性地在运动完“快速捕获”位置后的规定距离时停止运动）
- 发送增量绝对运动命令（以及选择性地在运动完“快速捕获”位置后的规定距离时停止运动）
- 设置增量运动的目标偏移（即轴相对于被捕获的快速中断的位置）
- 使轴点动
- 设置轴的位置
- 发送基准速度
- 发送基准转矩
- 启用/禁用轴
- 启用/禁用硬件限制
- 将轴错误复位
- 对轴执行受控停止或急停
- 使轴按照二级编码器输入运动
- 设置所有运动的速度、加速时间、减速时间与急动时间
- 控制模态轴与非模态轴

同时，PLC 能够监测驱动的以下状态信息：

- 启用状态
- 启用就绪状态
- 空转状态
- 就位状态
- 电机制动状态
- 回位状态
- 正向限位状态
- 反向限位状态
- 故障状态
- 停止输入状态
- 快速锁存中断遗漏指示
- 相位搜索状态
- 错误码
- 位置已测量
- 速度已测量
- 随动误差
- 轴运行模式
- RMS 电流

在 PLC 中，这些均通过将输入和输出过程数据映射(PDO)到驱动上的 NETDATA 对象实现。由于已将 32 位数据（UDINT 数据类型）用于接口，因此每个数据均被映射到驱动中的单个 32 位 NETINTEGER 或 NETFLOAT 位置上。
此外，选配的看门狗机制还使驱动能够在通信中断时采取措施（默认采用急停与禁用）。

由于使用 X20CP0410 处理器，因此为打开与使用示例项目，需要 4.5.2.102 或以上版本的 Automation Studio。需要的 Automation Runtime 版本如下所述。



配置通用驱动接口(GDI) Mint 程序

预编写的 GDI Mint 程序仅需稍微自定义就可满足用户的应用需求。详见应用说明书 AN00204。

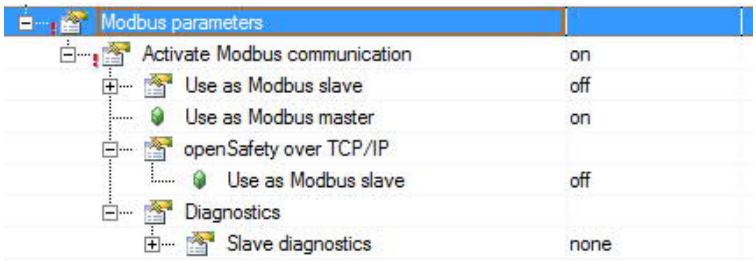
配置基于 Mint 的驱动使用的 Modbus TCP

MicroFlex e190 与 MotiFlex e180 驱动在交付时已“预配置”，能够通过驱动正面的标准以太网端口(E3)与运行 Modbus TCP。用户仅需通过 Mint Workbench 向驱动分配与 PLC 项目中设置的 IP 地址相符的（唯一）IP 地址。在给出的示例项目中，MicroFlex e190 的 IP 地址预期会被设为 192.168.0.1（PLC 项的地址是 192.168.0.109）。
添加附加轴时，必须确保为每个驱动分别设置唯一 IP 地址。谨记，所有驱动均必须与 PLC 位于相同子网中（例如：192.168.0.x）。使用标准以太网开关将所有驱动连至同一网络。

PLC 配置

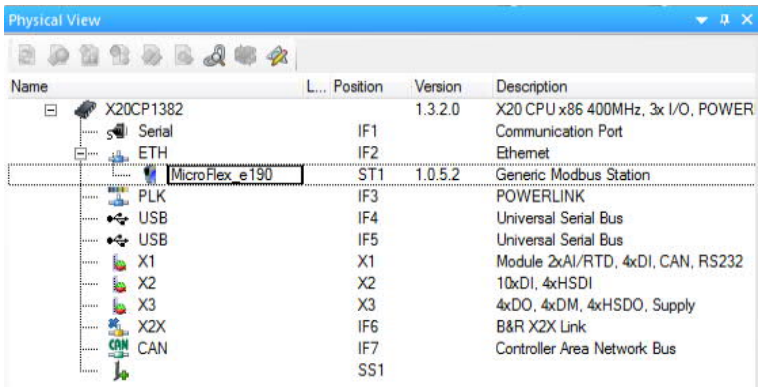
本应用说明书通过示例应用程序解释了为使 X20CP0410 PLC 能够通过 Modbus TCP 与 ABB 运动驱动通信而需完成的配置。如果从头启动新应用程序，则请遵循以下过程：

在 Automation Studio 内的物理视图中，右键单击 PLC，选择“配置”(Configuration)。展开右侧面板中的“Modbus 参数”(Modbus parameters)部分，按下图所示激活 Modbus...

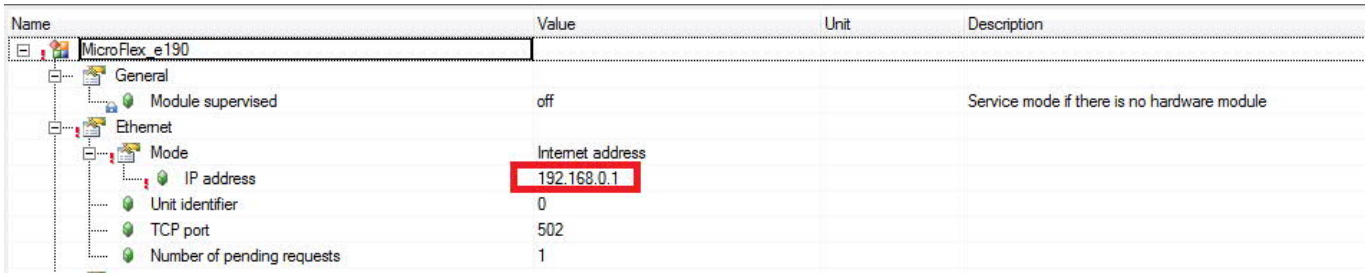


现在，物理视图中的 ETH 图标仍然高亮显示。向下滚动工具框内的设备目录，选择“ModbusTcp_any”设备（将该设备拖到 ETH 图标上）。

如果必要，您可重新设置添加的设备的名称（以便更清楚地识别）...在示例中，我们将设备重命名为“MicroFlex_e190”。



现在，右键单击刚刚添加的设备，选择“配置”(Configuration)。端口编号已设为 502（因为我们选择的是 Modbus TCP 设备），但我们最初需配置将与 PLC 通信的驱动的 IP 地址。



由于我们配置的驱动的 IP 地址是 192.168.0.1，因此我们输入该 IP 地址。

最后，我们需要为 Modbus TCP 添加读/写功能块（功能代码：23），由于在 PLC 与驱动之间传输所有 PDO 数据。该功能块的起始地址以及项目数量（即 Modbus 寄存器的地址与数量）必须适合驱动上的 Mint GDI 程序使用的 Netdata 位置。在我们的标准 GDI 应用程序中，读数据的起始地址为 Netdata(100) / Modbus 寄存器 200（需读取 7 个 Netdata 位置-14 个 Modbus 寄存器）。写数据的起始地址为 Netdata(0) / Modbus 寄存器 0（需写 9 个 Netdata 位置-18 个 Modbus 寄存器）。

更新时间一般设为程序在 PLC 与驱动间传输所有 Modbus 数据使用的周期的一半。在示例中，我们的数据传输程序使用 1 类任务，运行 10ms，因此我们将 Modbus 更新时间设为 5ms。

因此，我们的功能块 1 的最终配置如下。

Block 1			
Function code	FC23: Read/Writ...		
Refresh time	5	ms	
Block send mode	cyclic		
Starting address (read)	200		
Number of items (read)	14		Set to 0 for automatic calculation
Starting address (write)	0		
Number of items (write)	18		Set to 0 for automatic calculation

配置好功能块后，我们可以继续添加与功能块相关的各“通道”的信息。我们必须为每一个 Netdata 位置分别添加一条通道，设置通道的名称（例如：mbStatusWord）、类型(UDINT)（初始，所有数据均以 32 位双整数的形式传输）与方向（读或写）。以下屏幕截图展示了某些通道配置。

Channel 1	
Name	mbStatusWord
Data type	UDINT
Direction	Read
Channel 2	
Name	mbMeasuredPos
Data type	UDINT
Direction	Read
Channel 3	
Name	mbMeasuredVel
Data type	UDINT
Direction	Read
Channel 4	
Name	mbFolError
Data type	UDINT
Direction	Read
Channel 5	
Name	mbAxisMode
Data type	UDINT
Direction	Read
Channel 6	
Name	mbRMSCurrent
Data type	UDINT
Direction	Read

现在，再次右键单击“I/O 映射...”(I/O Mapping...)在打开的面板的右侧，我们需为 ModuleOK 通道选择过程变量（自动添加，指示使用 Modbus TCP 的设备的运行状态）以及选择之前通过通道配置添加的所有 GDI PDO 项目。

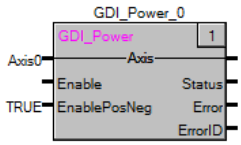
Channel Name	Process Variable	Data Type	Task Class
ModuleOk	::Axis0.NodeOK	BOOL	Automatic
mbStatusWord	::Axis0.PDOIn.pdoSTATUS_WORD	UDINT	Automatic
mbMeasuredPos	::Axis0.PDOIn.pdoMEASURED_POS	UDINT	Automatic
mbMeasuredVel	::Axis0.PDOIn.pdoMEASURED_VEL	UDINT	Automatic
mbFolError	::Axis0.PDOIn.pdoFOL_ERROR	UDINT	Automatic
mbAxisMode	::Axis0.PDOIn.pdoAXIS_MODE	UDINT	Automatic
mbRMSCurrent	::Axis0.PDOIn.pdoRMS_CURRENT	UDINT	Automatic
mbErrorCode	::Axis0.PDOIn.pdoERROR_CODE	UDINT	Automatic
mbCommandWord	::Axis0.PDOOut.pdoCONTROL_WORD	UDINT	Automatic
mbCmdType	::Axis0.PDOOut.pdoCMD_TYPE	UDINT	Automatic
mbValue	::Axis0.PDOOut.pdoVALUE	UDINT	Automatic
mbSpeed	::Axis0.PDOOut.pdoSPEED	UDINT	Automatic
mbAccel	::Axis0.PDOOut.pdoACCEL	UDINT	Automatic
mbDecel	::Axis0.PDOOut.pdoDECEL	UDINT	Automatic
mbAccelJerk	::Axis0.PDOOut.pdoACCELJERK	UDINT	Automatic
mdDecelJerk	::Axis0.PDOOut.pdoDECELJERK	UDINT	Automatic
mbOffset	::Axis0.PDOOut.pdoOFFSET	UDINT	Automatic

B&R GDI 功能块

以下章节详细描述了 B&R GDI 功能块的使用：

GDI_Power

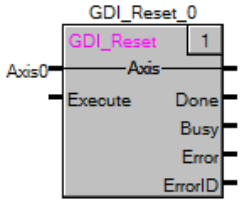
该功能块用于启用/禁用轴。启用输入会启用驱动中的功率级，而非功能块本身。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Enable	BOOL	值为“真”时，表示 PLC 请求将轴启用
EnablePosNeg	BOOL	值为“真”时，表示两个方向上的运动均被允许。值为“假”时，表示禁止运动（或者，如果正在运动，会执行停止命令）
VAR_OUTPUT		
Status	BOOL	指示轴是否已启用（1：已启用；0：未启用）
Error	BOOL	轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_Reset

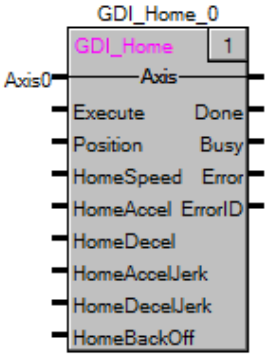
该功能块用于复位当前的所有轴错误。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始复位故障
VAR_OUTPUT		
Done	BOOL	轴不再有错误时，值为“真”。在 Execute 输入撤销之前始终为“真”。Execute 输入在 Done 位变为“真”之前被撤销的，Done 位将仅在单个 PLC 周期内为“真”。无法清除错误时，Done 位不会变为“真”（使用 Busy 输出检测是否在尝试复位故障）
Busy	BOOL	功能块尝试清除轴错误时，值为“真”
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_Home

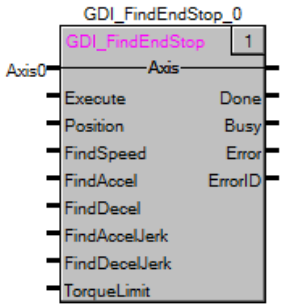
该功能块用于设置轴的基准。详细的基准序列取决于 Mint GDI 程序中设置的零位类型。Position 输入用于在基准序列成功结束时设置轴的位置。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始基准序列
Position	REAL	在基准序列成功结束时设置的绝对位置
HomeSpeed	REAL	以用户单位/s 表示的回位速度
HomeAccel	REAL	以用户单位/s^2 表示的回位加速度
HomeDecel	REAL	以用户单位/s^2 表示的回位减速度
HomeAccelJerk	REAL	以用户单位/s3 表示的回位加速度急动度（梯形运动时，设为 0）
HomeDecelJerk	REAL	以用户单位/s^3 表示的回位减速度急动度（梯形运动时，设为 0）
HomeBackOff	REAL	回位速度与退避速度之比
VAR_OUTPUT		
Done	BOOL	指示轴已成功回到零位。Execute 输入在轴返回零位的过程中被撤销的，当轴完成回位序列时，Done 输出会变为“真”，但仅会坚持一个 PLC 扫描周期。如果 Execute 输入始终为 1，那么 Done 输出会一直为“真”（前提是轴成功回位）
Busy	BOOL	在回位序列执行过程中，值为“真”
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_FindEndStop

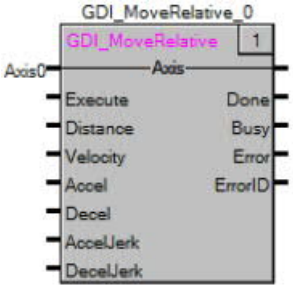
该功能块是在无零位传感器的情况下设置轴基准的替代方法。轴按命令的速度运转，直至达到设置的转矩限值，轴的转速小于设置的空转速度。Position 输入用于在基准序列成功结束时设置轴的位置。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始基准序列
Position	REAL	在基准序列成功结束时设置的绝对位置
FindSpeed	REAL	以用户单位/s 表示的速度（该数值的正负决定寻找方向）
FindAccel	REAL	以用户单位/s^2 表示的加速度
FindDecel	REAL	以用户单位/s^2 表示的减速度
FindAccelJerk	REAL	以用户单位/s^3 表示的加速度急动度（梯形运动时，设为 0）
FindDecelJerk	REAL	以用户单位/s^3 表示的减速度急动度（梯形运动时，设为 0）
TorqueLimit	REAL	序列执行过程中使用的转矩限值（额定驱动电流的%）
VAR_OUTPUT		
Done	BOOL	指示轴已成功找到最终停止位置。Execute 输入在序列执行过程中被撤销的，当轴找到最终停止位置时，Done 输出会变为“真”，但仅会坚持一个 PLC 扫描周期。如果 Execute 输入始终为 1，那么 Done 输出会一直为“真”（前提是序列成功执行）
Busy	BOOL	在寻找序列执行过程中，值为“真”
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_MoveRelative

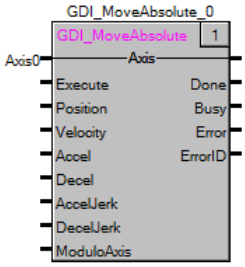
该功能块用于命令轴在受控模式下相对开始位置运动规定的距离。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始运动
Distance	REAL	相对运动距离（以用户单位表示）
Velocity	REAL	以用户单位/s 表示的最大速度（不一定要达到）
Accel	REAL	以用户单位/s^2 表示的加速度
Decel	REAL	以用户单位/s^2 表示的减速度
AccelJerk	REAL	以用户单位/s^3 表示的加速度急动度（梯形运动时，设为 0）
DecelJerk	REAL	以用户单位/s^3 表示的减速度急动度（梯形运动时，设为 0）
VAR_OUTPUT		
Done	BOOL	指示轴已成功到达目标位置。Execute 输入在运动过程中被撤销的，当轴完成相对运动时，Done 输出会变为 1，但仅会坚持一个 PLC 扫描周期。如果 Execute 输入始终为“真”，那么 Done 输出会一直为“真”（前提是轴成功到达目标位置）
Busy	BOOL	在相对运动过程中，值为“真”
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_MoveAbsolute

该功能块用于命令轴在受控模式下运动到规定的绝对位置。该功能可与模态轴联用（在这种情况下，会使用运动到规定位置的最短路径）。

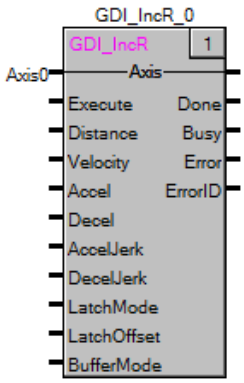


	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始运动
Position	REAL	运动的目标位置（以用户单位表示）
Velocity	REAL	以用户单位/s 表示的最大速度（不一定要达到）
Accel	REAL	以用户单位/s^2 表示的加速度
Decel	REAL	以用户单位/s^2 表示的减速度
AccelJerk	REAL	以用户单位/s^3 表示的加速度急动度（梯形运动时，设为 0）
DecelJerk	REAL	以用户单位/s^3 表示的减速度急动度（梯形运动时，设为 0）
ModuloAxis	BOOL	决定是否使用模态轴（即利用 ENCODERWRAP 定义一个周期内的运动）。使用模态轴时，绝对运动始终使用最短路径（例如：在一条 0-360°的模态轴上，如需完成从 20°到 350°的绝对运动，会正向运动 30°）
VAR_OUTPUT		
Done	BOOL	指示轴已成功到达目标位置。Execute 输入在运动过程中被撤销的，当轴完成绝对运动时，Done 输出会变为“真”，但仅会坚持一个 PLC 扫描周期。如果 Execute 输入始终为“真”，那么 Done 输出会一直为“真”（前提是轴成功到达目标位置）
Busy	BOOL	在绝对运动过程中，值为“真”
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_IncR

该功能块用于命令轴在受控模式下相对于执行该功能块时产生的目标位置运动规定的距离。在运动过程中，可通过以下任何方式变更调用该功能块产生的目标位置：

a. 发送另一 GDI_IncR 或 GDI_IncA 功能命令（前提是 BufferMode 的值为“真”）
b. 将输入参数 Latchmode 的值设为“真”以及设置输入参数 LatchOffset 的值。之后，驱动上的 Mint 代码会自动变更轴的目标位置，使轴在经过设置的快速中断捕获的位置后继续运动一段等于 LatchOffset 的距离才停止。Axis 状态词中的某一位(btLatchMissed)可用于指示快速中断检测失败（这种情况可用于提醒操作员（例如）系统发生了故障）。使用 Latchmode 与 LatchOffset 时，可轻松实施分度输送机应用。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始运动
Distance	REAL	相对运动距离（以用户单位表示）
Velocity	REAL	以用户单位/s 表示的最大速度（不一定要达到）
Accel	REAL	以用户单位/s^2 表示的加速度
Decel	REAL	以用户单位/s^2 表示的减速度
AccelJerk	REAL	以用户单位/s^3 表示的加速度急动度（梯形运动时，设为 0）
DecelJerk	REAL	以用户单位/s^3 表示的减速度急动度（梯形运动时，设为 0）
LatchMode	BOOL	设置轴是否应使用设置的快速锁存中断与是否设置新目标位置（即在被捕获位置后继续运动“LatchOffset”用户单位
LatchOffset	REAL	设置在被捕获的快速中断位置后继续运动的距离（使用用户单位），用于修改 GDI_INCR 的目标位置（当输入参数 LatchMode 的值设为“真”时）
BufferMode	BOOL	决定功能块是否应设置 Done 输出以及是否在运动加载后立即完成。将 BufferMode 设为“真”后，应用程序可在当前运动执行过程中触发进一步增量运动

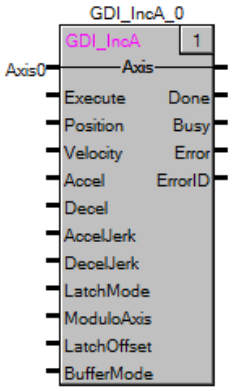
VAR_OUTPUT		
Done	BOOL	“缓冲模式”设为“假”时，这指示轴已成功到达目标位置。Execute 输入在运动过程中被撤销的，当轴完成相对运动时，Done 输出会变为“真”，但仅会坚持一个 PLC 扫描周期。如果 Execute 输入始终为“真”，那么 Done 输出会一直为“真”（前提是轴成功到达目标位置）。BufferMode 设为“真”时，Done 输出在一个 PLC 扫描周期内设为“真”，以指示运动成功加载
Busy	BOOL	在运动过程中，值为“真”
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

需要在运动过程中修改 SPEED/ACCEL/DECEL 时，也可以使用 GDI_IncR。通过 GDI_MoveRelative 加载的运动会按照当时加载的 SPEED/ACCEL/DECEL 执行，一旦开始运动，SPEED/ACCEL/DECEL 将无法更改。使用将输入参数 BufferMode 设为“真”的 GDI_IncR 时，可通过加载另一将输入参数 Distance 设为“零”的 GDI_IncR（含新的 SPEED/ACCEL/DECEL）来更改这些设置参数。

GDI_IncA

该功能块用于命令轴在受控模式下运动到规定的绝对位置。该功能块与 GDI_MoveAbsolute 的区别在于目标位置可在运动过程中通过以下任何方式更改：

a. 发送另一 GDI_IncR 或 GDI_IncA 功能命令（前提是 BufferMode 的值为“真”）
b. 将输入参数 Latchmode 的值设为“真”以及设置输入参数 LatchOffset 的值。之后，驱动上的 Mint 代码会自动变更轴的目标位置，使轴在经过设置的快速中断捕获的位置后继续运动一段等于 LatchOffset 的距离才停止。Axis 状态词中的某一位(btLatchMissed)可用于指示快速中断检测失败（示例程序展示了如何检测连续遗漏了 3 个锁存-这种情况可用于提醒操作员（例如）系统发生了故障）。

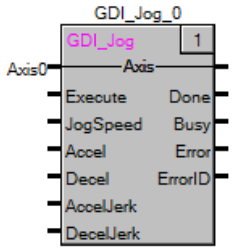


	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始运动
Position	REAL	运动的绝对目标位置（以用户单位表示）
Velocity	REAL	以用户单位/s 表示的最大速度（不一定要达到）
Accel	REAL	以用户单位/s^2 表示的加速度
Decel	REAL	以用户单位/s^2 表示的减速度
AccelJerk	REAL	以用户单位/s^3 表示的加速度急动度（梯形运动时，设为 0）
DecelJerk	REAL	以用户单位/s^3 表示的减速度急动度（梯形运动时，设为 0）
LatchMode	BOOL	设置轴是否应使用设置的快速锁存中断与是否设置新目标位置（即在被捕获位置后继续运动“LatchOffset”用户单位
ModuloAxis	BOOL	决定是否使用模态轴（即利用 ENCODERWRAP 定义一个周期内的运动）。使用模态轴时，绝对运动始终使用最短路径（例如：在一条 0-360°的模态轴上，如需完成从 20°到 350°的绝对运动，会正向运动 30°）
LatchOffset	REAL	设置在被捕获的快速中断位置后继续运动的距离（使用用户单位），用于修改 ABB_GDI_INCA 的目标位置（当输入参数 LatchMode 的值设为“真”时）
BufferMode	BOOL	决定功能块是否应设置 Done 输出以及是否在运动加载后立即完成。将 BufferMode 设为“真”后，应用程序可在当前运动执行过程中触发进一步增量运动
VAR_OUTPUT		
Done	BOOL	“缓冲模式”设为“假”时，这指示轴已成功到达目标位置。Execute 输入在运动过程中被撤销的，当轴完成绝对运动时，Done 输出会变为“真”，但仅会坚持一个 PLC 扫描周期。如果 Execute 输入始终为“真”，那么 Done 输出会一直为“真”（前提是轴成功到达目标位置）。BufferMode 设为“真”时，Done 输出在一个 PLC 扫描周期内设为“真”，以指示运动成功加载
Busy	BOOL	在该功能块执行过程中为“真”
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

需要在运动过程中修改 SPEED/ACCEL/DECEL 时，也可以使用 GDI_IncA。通过 GDI_MoveAbsolute 加载的运动会按照当时加载的 SPEED/ACCEL/DECEL 执行，一旦开始运动，SPEED/ACCEL/DECEL 将无法更改。使用将输入参数 BufferMode 设为“真”的 GDI_IncA 时，可通过先加载 GDI_IncA 运动，然后加载将输入参数 Distance 设为“零”的 GDI_IncR（含新的 SPEED/ACCEL/DECEL）来更改这些设置参数。

GDI_Jog

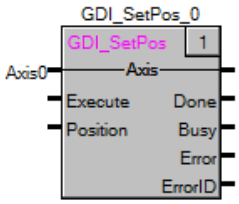
该功能块用于命令轴恒速运动（使用驱动中的位置回路控制器）。只要 Execute 输入的值“真”，就会执行运动。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始运动，只要输入值为“真”，就一直运动。在 Execute 输入变为“假”后，运动速度按设置的 Decel 逐渐降为零。
JogSpeed	REAL	以用户单位/s 表示的轴会达到的速度
Accel	REAL	以用户单位/s^2 表示的加速度
Decel	REAL	以用户单位/s^2 表示的减速度
AccelJerk	REAL	以用户单位/s^3 表示的加速度急动度（梯形运动时，设为 0）
DecelJerk	REAL	以用户单位/s^3 表示的减速度急动度（梯形运动时，设为 0）
VAR_OUTPUT		
Done	BOOL	在发送点动命令后立即变为“真”，在 Execute 输入变为“假”或轴出错之前始终为“真”
Busy	BOOL	在该功能块执行过程中为“真”
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_SetPos

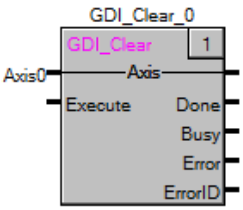
该功能块用于将轴位置（驱动上的编码器和位置值）设为规定值。调用该功能时，轴必须在空转，否则轴会反馈“无法执行-正在运动”的错误消息（错误码：10）。轴使用绝对编码器的，会设置/示教新的绝对位置（v2.17 及以上版本的 GDI Mint 程序）。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时设置新位置
Position	REAL	待设置的轴位置（使用用户单位）
VAR_OUTPUT		
Done	BOOL	只要发送了命令，该输出的值就为“真”（无论命令是否成功被执行-通过 Error 输出判断命令是否成功被执行）在 Execute 输入撤销之前始终为“真”。Execute 输入在 Done 位变为“真”之前被撤销的，Done 位将仅在单个 PLC 周期内为“真”。
Busy	BOOL	在该功能块执行过程中为“真”（一旦设置了“完成”位，该输出值会被清除）
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_Clear

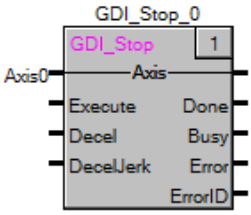
该功能块用于使轴急停以及中断正在进行的运动，轴会保持启用状态（前提是 GDI_Power 在请求启用状态且轴未出错）。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始急停
VAR_OUTPUT		
Done	BOOL	当轴在完成急停后空转或轴在急停命令发送后出错时，该输出值为“真”。在 Execute 输入撤销之前始终为“真”。Execute 输入在 Done 位变为“真”之前被撤销的，Done 位将仅在单个 PLC 周期内为“真”。
Busy	BOOL	在停止过程中为“真”- 一旦设置了“完成”位，该输出值会被清除
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_Stop

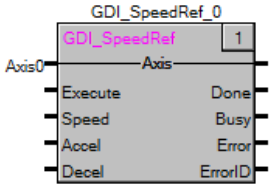
该功能块用于使轴在受控模式下按照设置的减速度停止运转。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始受控停止
Decel	REAL	以用户单位/s^2 表示的减速度
DecelJerk	REAL	以用户单位/s^3 表示的减速度急动度（梯形运动时，设为 0）
VAR_OUTPUT		
Done	BOOL	当轴在完成受控停止后空转或轴在停止命令发送后出错时，该输出值为“真”。在 Execute 输入撤销之前始终为“真”。Execute 输入在 Done 位变为“真”之前被撤销的，Done 位将仅在单个 PLC 周期内为“真”。
Busy	BOOL	在停止过程中为“真”- 一旦设置了“完成”位，该输出值会被清除
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_SpeedRef

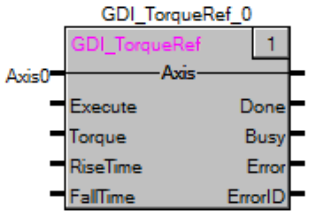
该功能块用于设置轴的基
准速度。在这种运行模式下，驱动不使用位置回路（因此不会记录后续错误或不会对后续错误采取任何措施）。在发送另一控制模式的运动（例如：位置受控的运动）之前，轴始终在速度控制模式（由 Controlmode 状态词中的位指示）保持不变。为从零速运行（速度控制模式）转换为保持位置（位置控制模式），可发送（例如）将相对运动距离设为零（使用用户单位）的 GDI_MoveRelative。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时，轴开始运动，只要输入值为“真”，就一直运动。在 Execute 输入变为“假”后，运动速度按设置的 Decel 逐渐降为零。
Speed	REAL	以用户单位/s 表示的轴会达到的速度可在 Execute 为“真”时修改，以改变轴的运动速度
Accel	REAL	以用户单位/s^2 表示的加速度
Decel	REAL	以用户单位/s^2 表示的减速度
VAR_OUTPUT		
Done	BOOL	在基准速度发送后立即变为“真”（无论命令是否成功执行）。Done 输出在 Execute 变为“假”之前始终为“真”
Busy	BOOL	在该功能块执行过程中为“真”（即 Execute 为“真”时）
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_TorqueRef

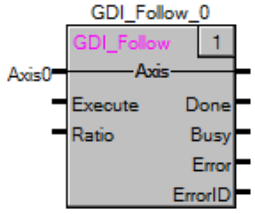
该功能块用于设置轴的基准转矩（电流）。在这种运行模式下，驱动不使用位置回路（因此不会记录后续错误或不会对后续错误采取任何措施）。在发送另一控制模式的运动（例如：位置受控的运动）之前，轴始终在转矩控制模式（由 Controlmode 状态词中的位指示）保持不变。为从零转矩运行（转矩控制模式）转换为保持位置（位置控制模式），可发送（例如）将相对运动距离设为零（使用用户单位）的 GDI_MoveRelative。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时，开始使用基准转矩，只要输入值为“真”，就维持该转矩。在 Execute 输入变为“假”后，转矩按设置的 FallTime 逐渐降为零。
Torque	REAL	轴将使用的基准转矩（DRIVERATEDCURRENT 的%-见 Mint 帮助文件）。可在 Execute 为“真”时修改，以改变产生的转矩
RiseTime	REAL	设置电流从零升到 DRIVEPEAKCURRENT 花费的时间（见 Mint 帮助文件）
FallTime	REAL	设置电流从 DRIVEPEAKCURRENT 降到零花费的时间（见 Mint 帮助文件）
VAR_OUTPUT		
Done	BOOL	在基准转矩发送后立即变为“真”（无论命令是否成功执行）。Done 输出在 Execute 变为“假”之前始终为“真”
Busy	BOOL	在该功能块执行过程中为“真”（即 Execute 为“真”时）
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

GDI_Follow

该功能块用于命令轴按照设置的随动比跟随配置的主编码器参考值运动。



	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构
VAR_INPUT		
Execute	BOOL	在出现上升沿时开始随动 Execute 输入变为“假”之前，轴会始终处于随动模式（如需停止随动，发送另一运动命令或通过 ABB_GDI_CLEAR 清除运动）
Ratio	REAL	轴与主编码器参考速度之间的随动比（齿轮比）（该参数值受轴与主编码器换算的影响 - 见 Mint 帮助文件中的“随动”部分）。为在随动过程中设置新随动比，需发送新 ABB_GDI_FOLLOW 命令。
VAR_OUTPUT		
Done	BOOL	在随动命令发送后立即变为“真”（无论命令是否成功执行）。Done 输出在 Execute 变为“假”之前始终为“真”
Busy	BOOL	在该功能块执行过程中为“真”（即 Execute 为“真”时）
Error	BOOL	在轴出错时，值为“真”
ErrorID	DINT	指示轴报告的 Mint 错误码

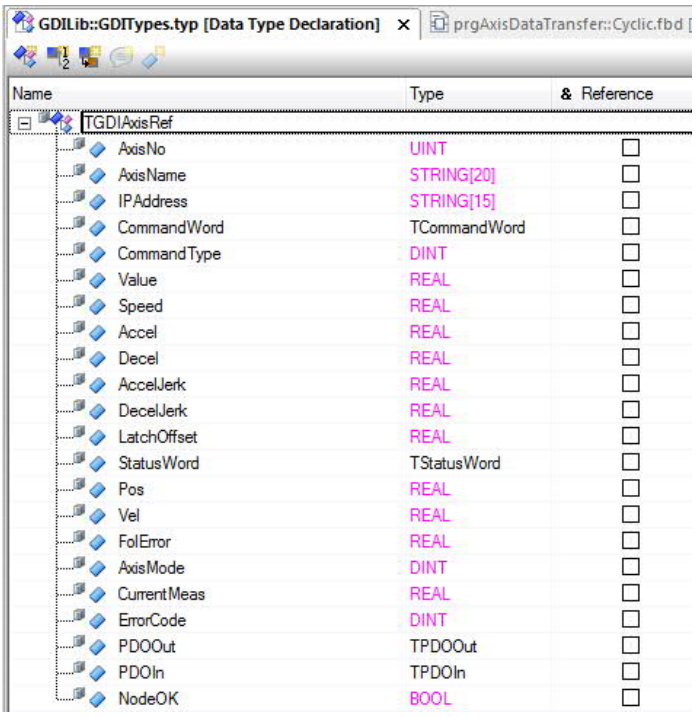
GDI_DataInterface

该功能块用于在 PLC 与 ABB 运动驱动之间传输命令/状态数据。应用程序中必须有每个轴的相关功能块实例。

	类型	描述
VAR_IN_OUT		
Axis	TGDIAxisRef	指轴的结构

使用轴结构

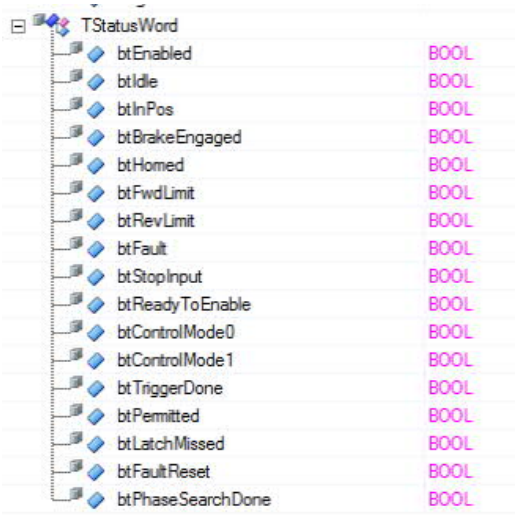
GDI 的大部分功能均作为库功能块提供，但在某些情况下，应用逻辑可能需访问轴结构数据。TGDIAxisRef 数据类型宣告如下：



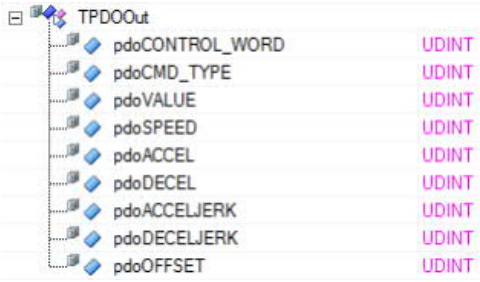
这种数据结构细分为四种数据结构（TCommandWord、TStatusWord、TPDOOut 与 TPDOIn）。这些数据结构宣告如下：

命令字

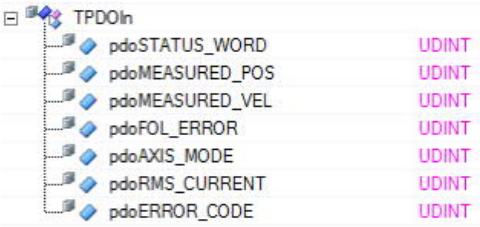
状态字



PDO 数据输出（至驱动）

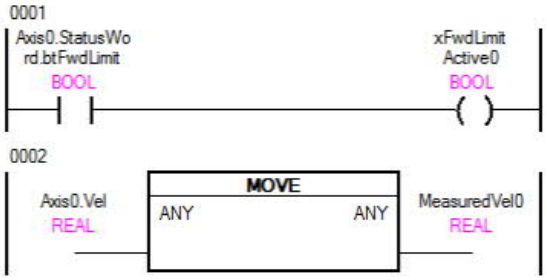


PDO 数据输入（来自驱动）



PLC 代码可以通过这些结构访问任何数据，但过程数据结构仅用于使 PDO 映射变量包含在主结构中，以免在向项目添加附加轴时需要为每次映射创建唯一变量名称，因此，通常不会从通用应用逻辑访问过程数据结构。

例如：
两个 rung 直接访问轴结构数据，一个读取驱动上的正向限位输入的状态，另一个保存测得的轴速度。



由于能够直接访问轴结构数据，PLC 应用代码拥有极大的灵活性（例如：对于分度输送机应用，PKC 应用程序可以访问锁存遗漏状态位，利用该数据驱动计数器在连续遗漏的锁存（快速中断）达到一定数量时使轴停止运动）。

通信看门狗

Mint GDI 默认使用看门狗机制。在收到 PLC 发送的第一条消息后，Mint 程序会检查通信是否未断开。如果通信断开，轴会停止运动，在错误被清除之前，无需继续运动。可以在驱动端禁用看门狗（详见 AN00204），但为了完整性，看门狗机制包含在 GDI_DataInterface 功能块中。

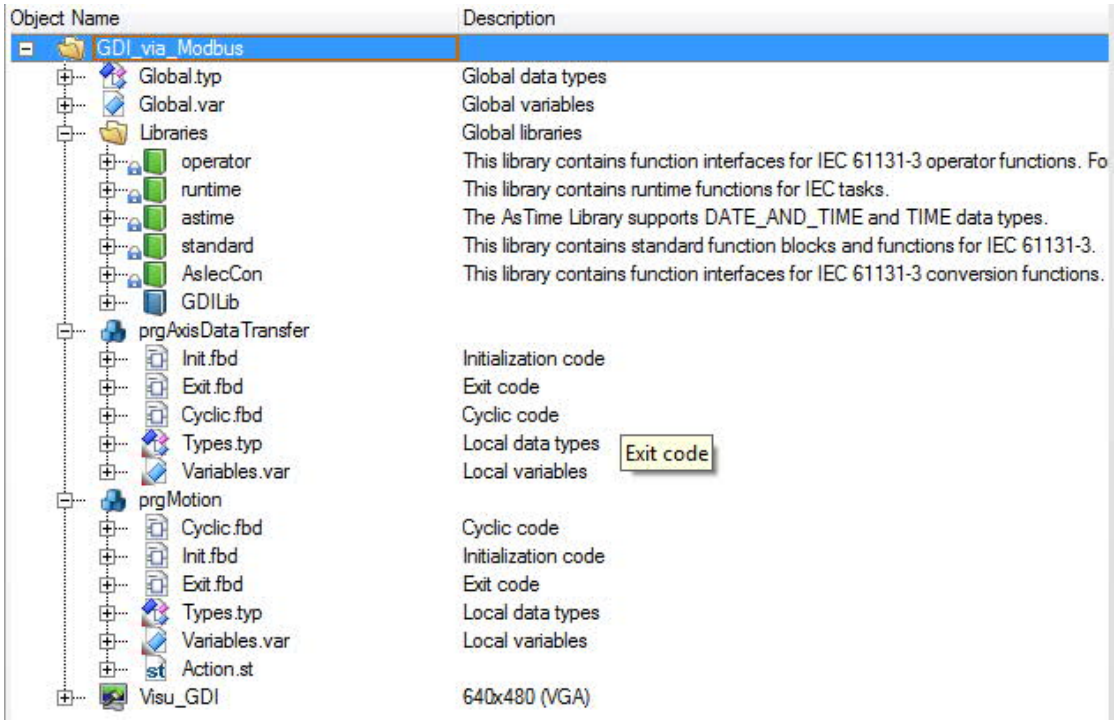
```
WatchdogBlink(Enable:=TRUE, TimeLow := _WatchdogTime / 2, TimeHigh := _WatchdogTime / 2);
WatchdogRTrig(CLK:=WatchdogBlink.Out);
IF WatchdogRTrig.Q = TRUE THEN
    iWatchdog := (iWatchdog + 1) MOD 2;
END_IF;
```

示例应用

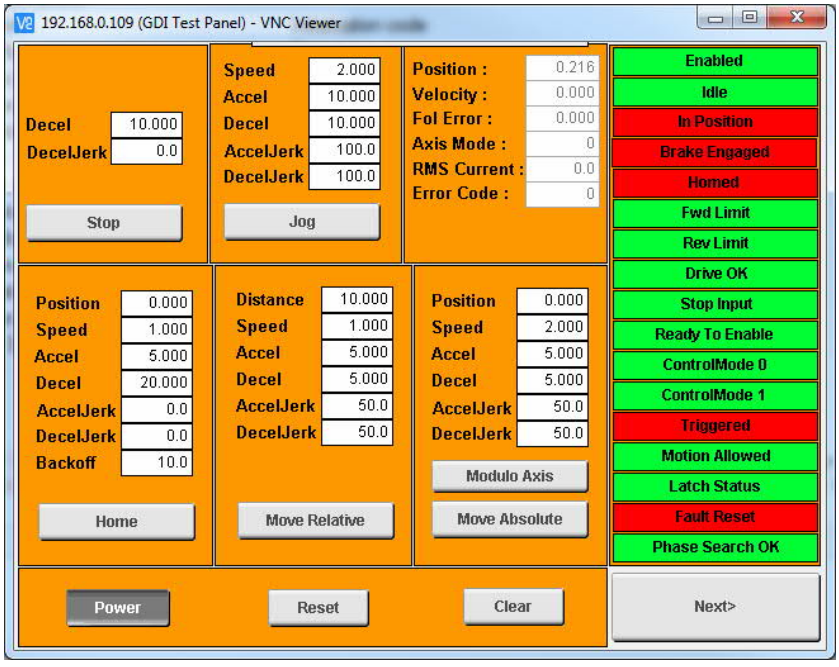
在本应用说明书提供的 Automation Studio 项目示例中，可以控制单个 MicroFlex e190 驱动（从 X20CP0410 PLC 控制）。主要有两种程序文件（均写在功能块图/FBD 中）。

prgAxisDataTransfer – 该程序的 cyclic.fbd 元素调用 GDI_DataInterface 的一个实例来传输/接收所有 PDO 数据
prgMotion – 该程序的 cyclic.fbd 元素包含每个 GDI 运动功能块的实例，为用于轴 0 进行了预配置（轴 0 在 Global.var 中被定义为 TGDIAxisRef 类型）。该程序的 Init.fbd 元素用于预加载各运动功能块的某些默认值

这两种程序均配置成作为 1 类循环任务运行（10ms 周期）。可按照应用要求和/或使用的处理器的规格调整周期。



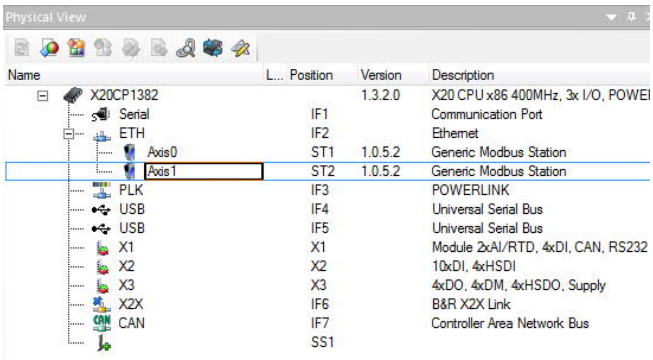
在示例应用中，PLC 的 IP 地址为 192.168.0.109。配置了 VNC 服务器，使 VNC 查看器（例如：https://www.realvnc.com）能够利用项目包含的可视化功能。可视化功能使用户能够测试 GDI 库支持的每个运动功能。



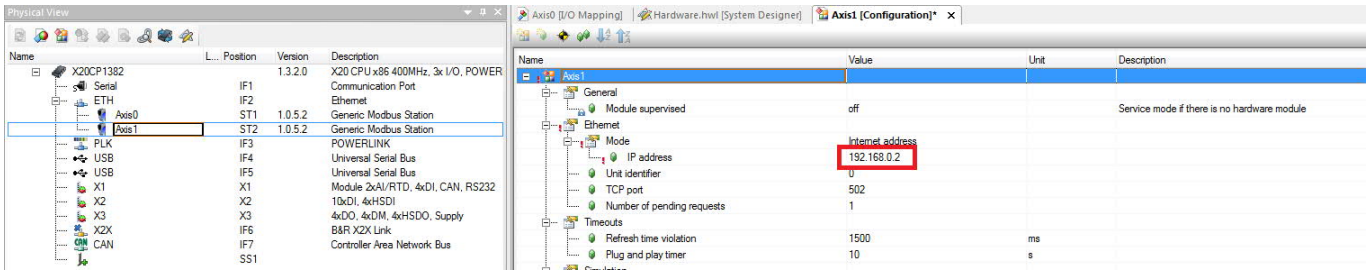
添加附加轴

应按照以下步骤扩展示例应用程序与在以太网(Modbus TCP)中添加附加轴：

将新驱动复制黏贴到设备树中，从而将驱动添加到物理视图(Physical View)（附加驱动应显示在硬件（系统设计程序(System Designer)）屏幕中，如下所示）。

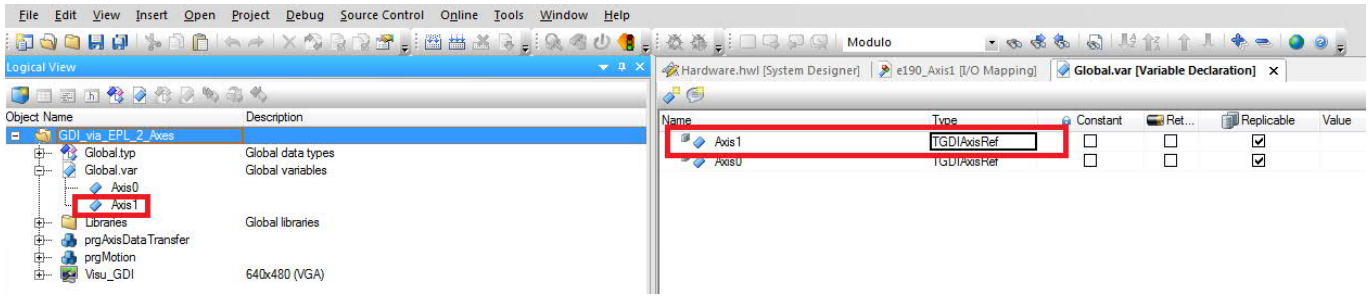


我们决定将驱动重命名为轴 0 与轴 1（因为其皆为 MicroFlex e190 驱动）。右键单击新驱动，选择“配置”(Configuration)，设置适合新驱动的 IP 地址。

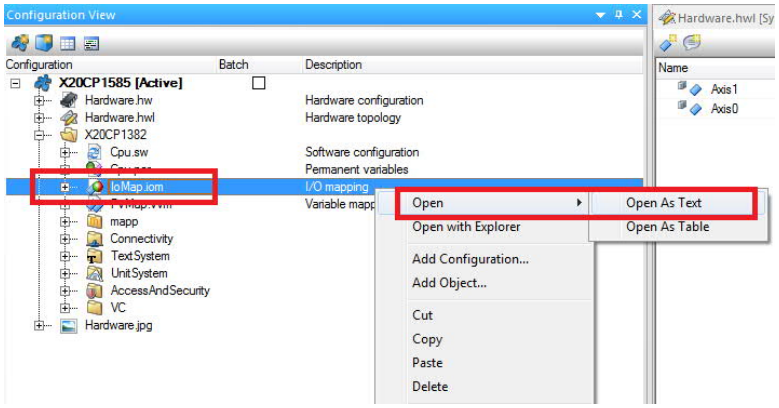


请注意，复制黏贴驱动时（而非将新设备从设备目录(Device Catalog)拖入），软件会自动复制 GDI 接口运行所需的所有 PDO 映射。

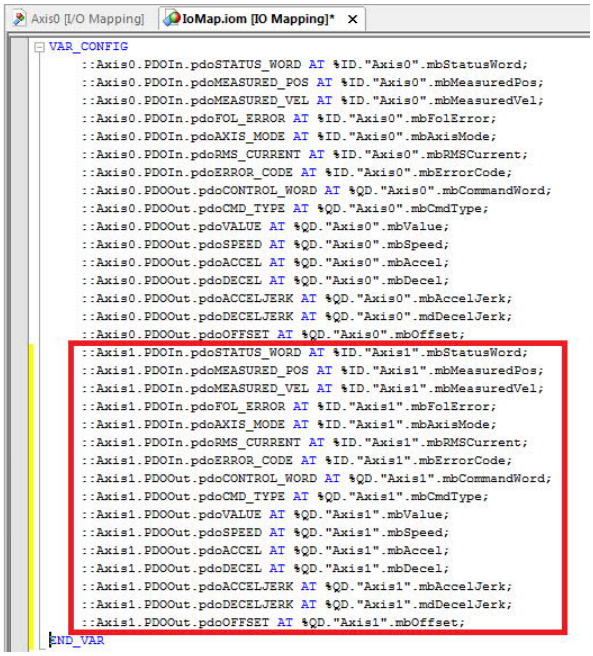
现在，选择 Automation Studio 中的“逻辑视图”(Logical View)，为新轴添加新全局变量，该变量应作为“TGDIAxisRef”类型变量添加，如下所示。



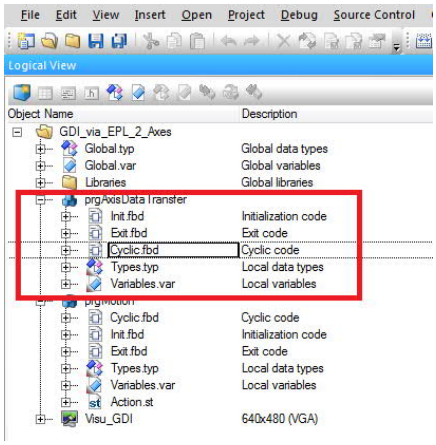
选择 Automation Studio 中的“配置视图”(Configuration View)，展开 PLC 文件夹（示例中的文件夹为 X20CP0410），右键单击 IoMap.iom，然后选择“打开(Open)>以文本形式打开(Open As Text)”。



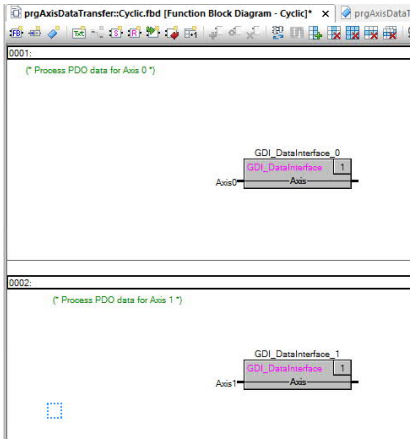
复制第一个轴（例如：轴 0）的所有现有设置，将这些设置黏贴到文件末尾的编辑器中，然后编辑这些设置，将提及第一个轴及其相关设备名称之处改为提及新轴，如下所示。

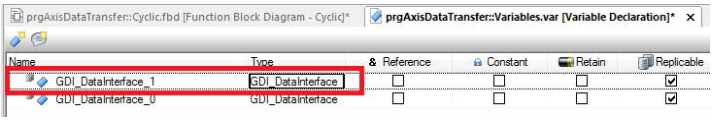


现在，我们需更新用于通过以太网将 PDO 数据传输至 ABB 运动驱动/传输 ABB 运动驱动发送的数据的程序文件。返回“逻辑视图”(Logical View)，双击 prgAxisDataTransfer 程序的“Cyclic.fbd”选项。



在 Cyclic.fbd 程序中添加新网络，插入 GDI_DataInterface 类型的新功能块（在库的 GDILib 区域内）。您需要宣告一个该类型的新变量，将新变量分配给新功能块实例。





至此，新轴添加步骤全部完成，您已准备好添加新应用代码来使添加的第二个轴运动，这在所有运动功能块中提及新轴的名称（例如：轴 1）即可实现。

联系我们

如需更多信息，请联系
您所在地的 ABB 代表或访问以下任一网站：

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