

Servo Motion

AC500 V3 PLC 与通用驱动器 Modbus 接口程序应用指导书

Application Note 501

Rev B (CN)



介绍

本应用说明详细介绍了 MicroFlex e190 和 MotiFlex e180 驱动器的“通用驱动器接口”或“GDI”Mint 程序的使用，及其与 AC500 V3 PLC 的配合使用方法。“GDI”是一个在驱动器上运行的简单程序，允许外部 PLC 控制系统访问基于 Mint 的控制功能。这种访问可以通过任何一种可以读写驱动器 Netdata 数组的通讯方式（例如 Modbus TCP、EtherCAT、Ethernet/IP、EPL 或 PROFINET）进行。这使得 PLC（如 ABB AC500 或 eco-PLC）能够监控和控制一个或多个轴上的简单运动功能。

本应用说明重点介绍了 ABB AC500 V3 PLC 通过 Modbus TCP 控制 e190/e180 驱动器的方式，但对于所有可用网络都是相同的原理。有关其他现场总线网络的其他 GDI 应用说明（例如，用于 Ethernet/IP 的 AN00222、用于 EtherCAT 的 AN00234、AN00251 用于 PROFINET 的 GDI 应用说明），请参阅运动控制网站。随附的“第 2 部分”示例 ABB AC500 V3 PLC 程序和库也包括在内，它们利用了 ABB Motion 开发的“PLCopen motion control”样式的功能块。

如果您需要有关旧 ABB 运动产品（如 MicroFlex e150）上的 GDI 应用程序的传统支持信息，请联系您当地的 ABB 支持团队。

兼容性

驱动器 Mint GDI 代码是在运行固件 5902.6 或更高版本的 **e190 和 e180 中实现最大功能而编写的**。最新的驱动器固件可从此处下载：

<https://new.abb.com/drives/low-voltage-ac/servo-products>

PLC 程序和库是使用 **Automation Builder v2.5 或更高版本** 编写的，可以从这里下载：

<https://new.abb.com/plc/automationbuilder>

运行此示例所需的 PLC 硬件是 **ABB AC500 V3 PLC 系列**。它们有两种主要类型：AC500 和 AC500eco。由于所有 CPU 都可以支持 Modbus TCP，因此此范围内的任何 CPU 都可以。所有 AC500 V3 CPU 的型号名称在代码的 PM 部分后都有 4 个数字，例如 **PM5072-T-2ETH**，而上一代 CPU 只有 3 个数字，例如 PM564-ETH。

保修、责任：

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

内容

GDI 中可用的函数（最新版本为 v2.35） 4

通信配置 5

配置 Generic Drive Interface（GDI）Mint 程序..... 6

Mint GDI 详细说明7

示例序列 13

PLC 示例程序 15

创建/定义轴 15

GDI 功能块 16

使用 GDI 结构 29

将 GDI PLC 库导入到项目中 31



GDI 中可用的功能（最新版本为 v2.35）

本应用说明中的示例程序为 ABB PLC 提供了一种机制，可以：

- 发出 Home 回零命令
- 发出命令以检测轴的物理限位器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 及更高版本）
- 发出相对运动
- 发出绝对运动
- 发出增量相对移动（并可选择在“快速捕获”位置之后停止编程距离）
- 发出增量绝对移动（并可选择在“快速捕获”位置之后停止编程距离）
- 为增量移动设置偏移目标（即相对于捕获的快速中断定位轴）
- 点动轴
- 设置轴位置
- 发出速度参考
- 发出扭矩参考
- 启用/禁用轴
- 启用/禁用硬件限制
- 复位轴错误
- 在轴上执行受控停止或紧急停止
- 将轴与输入的第二编码器进行齿轮同步运动
- 为所有运动设置速度、加速时间、减速时间和加加速度时间
- 控制模态轴或非模态轴

同时，PLC 可以监控驱动器的状态信息，包括：

- 已启用状态
- 准备启用状态
- 空闲状态
- 处于 Position 状态
- 电机抱闸制动状态
- 回零状态
- 正向限位状态
- 反转限位状态
- 故障状态
- 停止输入状态
- 丢失快速锁存中断的指示
- 寻相状态
- 错误代码
- 测量的位置
- 测得的速度
- 跟随错误
- 轴操作模式
- RMS 电流

这一切都是通过 PLC 读写 Modbus 寄存器实现的。因为我们在接口中使用了 32 位数据，所以每个值都使用了两个 16 位 Modbus 寄存器，这些寄存器又映射到驱动器中的单个 32 位 NETINTEGER 或 NETFLOAT 寄存器。还包括一个可选的看门狗机制，允许驱动器在通信丢失的情况下采取行动（默认为崩溃停止并去使能）。

通信配置

MotiFlex e180 和 MicroFlex e190 驱动器包括对 Modbus TCP 的本机支持。在 Modbus 寄存器和 Netdata 之间自动进行映射。在 Mint Workbench 中的“配置”部分允许用户配置所有可用通信接口的相关操作参数。“Network”部分允许用户设置驱动器的 IP 地址，例如...

MicroFlex e190 和 MotiFlex e180 驱动器的默认 IP 地址为 192.168.0.1（使用固件版本 59xx 时）。这与 AC500 PLC 完美匹配，因为 AC500 PLC 的 ETH1 端口的默认 IP 地址为 192.168.0.10（即默认情况下它们都位于同一子网上）。

Configuration

Identification

Network

NAT

Services

Modbus TCP Server

Modbus TCP Client

EtherNet/IP

NetData Utilisation

Network

This page allows you to configure the controller's network interface.

Note
Applying new network settings can stop the communication between the wizard and the drive. The wizard must be restarted if it is no longer responsive.

Host port (E3)

DHCP Enabled

Address

Mask

Gateway

Real-time Ethernet port (E1 + E2)

Note
The IP address for a POWERLINK device is derived from the rotary switch position.

Address

Mask

Router Node ID

Default Gateway

Use Host port (E3) gateway

Use Real-time Ethernet port (E1 + E2) gateway

配置页面的“Modbus 服务器”部分允许调整以下与 Modbus TCP 相关的驱动器通信参数：

- 已启用 = TRUE
- 端口号（502 是用于 Modbus TCP 的标准端口）
- 字节顺序（勾选大端顺序以用于 AC500 PLC）
- 字顺序（取消勾选大端顺序以用于 AC500 PLC）

Modbus Server

This page allows you to configure the controller's Modbus TCP

Enabled

Port

Big Endian Byte Order

Big Endian Word Order TCP

配置 Generic Drive Interface (GDI) Mint 程序

预先编写的 GDI Mint 程序只需要少量的定制以适应用户的应用程序。

在主程序的开头（在程序头之后）是一组与应用程序相关的常量.....

应根据需要编辑这些内容以适应应用程序：

不断	功能
_bRemoteEnable	用户可以决定 PLC 是否能够控制驱动器的状态
_bUseWatchdog	用户可以决定驱动器是否启用看门狗机制来检测来自 PLC 的通信丢失（如果设置为 _true 则驱动器将在通信丢失时崩溃、停止和禁用）
_nWatchdogTime	用户可以设置适当的看门狗超时值（以毫秒为单位）。此处设置的值将定义在触发错误之前从 PLC 看到的最大切换时间。
_fScale	用户可以设置一个比例因子，表示电机为一个用户行程单位产生的编码器计数。
_nHomeType	用户可以定义当要求基准时轴执行的回零序列类型（有关可用回零类型及其相关 Mint 常数值完整列表，请参阅 Mint 帮助文件中的 HOME 关键字）
_nHomeInput	允许用户指定将驱动器的哪些本地数字输入（0,1 或 2）用作 Home 传感器输入。如果使用不需要输入的主类型（例如，_hmPOSITIVE_INDEX），则将此值设置为 -1
_nFwdLimit / _nRevLimit	允许用户指定驱动器的哪些本地数字输入（0、1 或 2）将用作方向限位输入。如果应用程序不需要行程限位，则将这些常量设置为 -1
_nLimitMode	允许用户指定驱动器对激活其中一个限位输入的反应模式。默认情况下，驱动器将崩溃、停止和禁用（有关其他选项，请参阅 Mint 帮助文件中的 LIMITMODE 关键字）
_nStopInput	允许用户指定将驱动器的哪些本地数字输入（0、1 或 2）用作 Mint 停止输入（激活 Mint 停止输入会导致 Mint 应用程序中的停止事件被处理）。如果不需要 stop input，则将其设置为 -1
_nStopMode	允许用户指定驱动器对 stop 输入激活的反应模式。默认情况下，驱动器将以当前减速速率减速（有关其他选项，请参阅帮助文件中的 STOPMODE 关键字）
_nInputLevel	允许用户指定输入是高电平有效还是低电平有效。默认值指定为二进制值（输入 0 是最低有效位），将所有可用输入设置为高电平有效。请注意，使用条件编译以确保为相关驱动平台配置正确的输入数量
_nEncoderWrap	如果应用程序使用模态轴（例如，要求轴位置在 0 到 360 度范围内的转盘），则应将此常数设置为轴的一个完整周期中的编码器计数。对于非模态轴，将此常数设置为零
_fFolErrorFatal	设置将导致后续误差停机的跟随误差（需求位置和测量位置之间的差值）的大小。此值通常在微调后设置，此时用户可以识别出运动过程中可能被解释为“正常”跟随误差的内容。
_nMasterChannel	如果应用程序需要适应（即 FOLLOW）主编码器参考，则允许用户设置将哪个编码器通道用作主参考
_nMasterEncMode	允许用户设置主编码器通道的计数方向（根据需要为 0 或 1）
_nFollowMode	允许用户设置所需的 Mint FOLLOWMODE（例如 _fmNO_RAMP、_fmRAMP）—— 请参阅 Mint 帮助文件以获取所有可用的跟随模式
_nMotorDirection	允许用户为正向移动命令设置电动机的旋转方向（根据需要设置为 0 或 1）
_bmoduloaxis	支持非整数齿轮比的模态轴，模态轴限制：(Mutilple bits +single turn bits) /pre scale>=2^32

Mint GDI 详细说明

在大多数应用程序中，用户不太可能需要 Mint 应用程序的任何知识（除了编辑上述应用程序相关数据）。但是，为了完整起见，以下各节将详细介绍 Mint GDI。

数据接口

Modbus 寄存器偏移	功能	驱动器内部地址	驱动器数据类型
0/1	控制字	NETINTEGER (0)	32 位整数
2/3	命令类型	NETINTEGER (1)	32 位整数
4/5	数值	NETFLOAT 的 (2)	32 位 IEEE 浮点
6/7	速度	NETFLOAT 浮动 (3)	32 位 IEEE 浮点
8/9	加速率	NETFLOAT (4)	32 位 IEEE 浮点
10/11	减速率	NETFLOAT (5)	32 位 IEEE 浮点
12/13	加加速度	NETFLOAT (6)	32 位 IEEE 浮点
14/15	减加速度	NETFLOAT (7)	32 位 IEEE 浮点
16/17	Latch 偏移	NETFLOAT 浮动 (8)	32 位 IEEE 浮点
18/19	DO 状态强制	NETINTEGER (9)	32 位整数
Modbus 寄存器偏移	功能	驱动器内部地址	驱动器数据类型
200/201	状态词	NETINTEGER (100)	32 位整数
202/203	测量的位置	NETFLOAT 浮动 (101)	32 位 IEEE 浮点
204/205	测得的速度	NETFLOAT 网络 (102)	32 位 IEEE 浮点
206/207	跟随错误	NETFLOAT 浮动 (103)	32 位 IEEE 浮点
208/209	轴模式	NETINTEGER (104)	32 位整数
210/211	RMS 电流	NETFLOAT 浮动 (105)	32 位 IEEE 浮点
212/213	错误代码	NETINTEGER (106)	32 位整数
214/215	DI 状态	NETINTEGER (107)	32 位整数
216/217	DO 状态	NETINTEGER (108)	32 位整数
218/219	实际扭矩 (Nm)	NETFLOAT 的 (109)	32 位 IEEE 浮点

我们将在以下部分中详细研究这些功能中的每一个.....

控制字: Modbus 寄存器偏移量 0/1, NETINTEGER (0)

PLC 使用控制字中的位对驱动器执行特定操作。每当 PLC 写入控制字时，Mint 程序都会自动调用 Event NETDATA0。

位	功能	状态 0	状态 1
0	远程驱动器启用控制	禁用驱动器（如果 Mint 程序常数 <code>_nRemoteEnable</code> 设置为 <code>_true</code> ）	启用驱动器（如果 Mint 程序常数 <code>_nRemoteEnable</code> 设置为 <code>_true</code> ）
1	启用操作*	禁止运动	允许运动
2	锁存控制	禁用位置锁存器	启用位置锁存器
3	正向硬件限位控制	已分配正向硬件限位输入 （由 Mint 常数 <code>_nFwdLimit</code> ）设置	已取消分配正向硬件限位输入
4	反向硬件限制控制	已分配反向硬件限位输入 （由 Mint 常数 <code>_nRevLimit</code> ）设置	已取消分配反向硬件限位输入
5	模态轴	Axis 没有模态位置。 绝对移动使用整个位置范围	轴位置在定义的模态范围内。 绝对移动使用最短路径到达位置
6	故障复位控制	未使用	上升沿到状态 1 会导致驱动器故障复位
7	触发命令	未使用	上升沿到状态 1 使驱动器执行通过 PLC 输出字 1 加载的命令*
8	看门狗	Watchdog 关闭	Watchdog 开启
9	忽略跟随误差	驱动器启用跟随误差检测	驱动器忽略跟随误差检测
10-31	备用	未使用	未使用

* 某些命令不需要执行运动，即使 Enable Operation 位为零也可以发出（例如，Set Position 和 Cancel 命令）

注意：触发器位会激活 Command Type 字中加载的命令。Command 字中的所有其他位连续运行，不需要触发。模态轴命令仅由 MOVEA 和 INCA 运动命令使用。如果使用模态轴，则应编辑 Mint 程序，将 `_nEncoderWrap` 常数设置为等于一个轴周期中的编码器计数。如果在驱动器上启用了 Watchdog 监控，则应足够频繁地切换 Watchdog 位，以防止发生驱动器的 Watchdog 超时（通常将此位切换在 Watchdog 超时值的一半范围内）。

命令类型: Modbus 寄存器偏移量 2/3, NETINTEGER (1)

PLC 使用这些寄存器在驱动器上加载特定命令。Mint 程序示例提供了许多预定义的命令类型。这些可以通过添加到 Mint 程序中的命令枚举列表并在 Mint 的 'ActionTrigger' 任务中包含额外的代码来轻松扩展。

命令编号	功能	Mint 关键字
17	回零	HOME
1	相对移动	MOVER
2	绝对移动	MOVEA
3	以设定速度运行	JOG
4	相对增量移动	INCR
5	绝对增量移动	INCA
6	设置位置	POS
7	跟随	FOLLOW
8	速度参考	VELREF
9	扭矩参考	TORQUEREF
10	取消运动	CANCEL
11	受控停止	STOP
12	查找终点停止	使用扭矩 (%) 值停止
13	查找终点停止 Nm	使用以 Nm 为单位的扭矩值停止, 而不是%
14	运动 Sin 值	计算样条曲线的 Sin 值
15	暂停	为 Active Moves 添加 suspend interrupt 功能
16	绝对位置值读取	执行此操作后, 在新的通电后, 将恢复绝对位置

PLC 应使用适当的命令编号 (以及包含与后面描述的命令相关的参数的其他输出字) 预加载命令类型寄存器, 然后通过生成控制字第 7 位的上升 (0->1) 沿来触发此命令

数值: Modbus 寄存器偏移 4/5, NETFLOAT (2)

PLC 使用这些寄存器来加载与要在驱动器上发出的命令相关的信息。下表详细说明了该数值与每个特定命令的关系:

命令编号	功能	数值的作用
17	回零	回零回退比 ¹
1	相对运动	相对移动距离
2	绝对运动	绝对移动目标
3	以恒定速度运行 (启用位置循环)	未使用
4	相对增量移动	相对增量移动距离 ²
5	绝对增量移动	绝对增量移动目标 ²
6	设置位置	新位置值
7	跟随	设置齿轮比
8	速度参考	未使用
9	扭矩参考	设定扭矩值 (%)
10	取消运动	未使用
11	受控停止	未使用
12	查找终点停止	扭矩限制 (驱动器额定电流的 %) ³
13	查找终点停止 Nm	扭矩限制以 Nm 值而不是标称值 % 为单位

PLC 应使用适当的数据预加载 Value 寄存器 (以及包含与前面和后面描述的命令相关的参数的其他输出寄存器), 然后通过生成控制字第 7 位的上升 (0->1) 沿来触发命令。

¹ 归位回退比率为 Mint HOMEBACKOFF 关键字设置一个值 (有关详细信息, 请参阅 Mint 帮助文件)。此值只是一个标量。所有其他值都被缩放 (即, 使用单位取决于驱动器上的 Mint SCALEFACTOR 设置)。

² 如果在控制字中设置了锁存控制位, 则可以根据存储在 'Latch Offset' 寄存器中的值修改增量移动的目标位置 (这在后面的 Latch Offset 部分中有详细说明)。

³ 驱动器中的扭矩限制实际上与 CURRENTLIMIT 设置串联, 因此通常会设置一个扭矩限制, 从而降低整体电流限制 (例如, 如果 CURRENTLIMIT 设置为驱动器额定电流的 60%, 则在找到末端挡块时, 通常会设置低于 60% 的扭矩限制)。

速度：Modbus 寄存器偏移 6/7，NETFLOAT（3）

PLC 使用这些寄存器在驱动器上加载 slew speed。下表详细说明了 Speed 值与每个特定命令的关系：

命令编号	功能	速度的作用
17	回零	设置回零速度
1	相对运动	设置移动的速度
2	绝对运动	设置移动的速度
3	以恒定速度运行（启用位置循环）	设置要求的点动速度
4	相对增量移动	设置移动的速度
5	绝对增量移动	设置移动的速度
6	设置位置	未使用
7	跟随	未使用
8	速度参考	设置速度参考（以用户单位/秒为单位）
9	扭矩参考	未使用
10	取消运动	未使用
11	受控停止	未使用
12	查找终点停止	设置寻找终点停止的速度
13	查找终点停止 Nm	设置寻找终点停止的速度

Speed 是一个缩放量（即单位与驱动器上的 SCALEFACTOR 设置相关）。PLC 应使用适当的数据预加载 Speed 寄存器（以及包含与前面和后面描述的命令相关的参数的其他输出寄存器），然后通过生成控制字第 7 位的上升（0->1）沿来触发命令。

加速度：Modbus 寄存器偏移 8/9，NETFLOAT（4）

PLC 使用这些寄存器在驱动器上加载加速度。下表详细说明了加速率值与每个特定命令的关系：

命令编号	功能	加速度的作用
17	回零	为回零速度设置加速度
1	相对运动	为相对运动的速度设置加速度
2	绝对运动	为绝对运动的速度设置加速度
3	以恒定速度运行（启用位置循环）	为 jog speed 设置加速度
4	相对增量移动	为相对增量的速度设置加速度
5	绝对增量移动	为绝对增量的速度设置加速度
6	设置位置	未使用
7	跟随	未使用
8	速度参考	根据速度需求设置加速度
9	扭矩参考	未使用
10	取消运动	未使用
11	受控停止	未使用
12	查找终点停止	设置查找终点停止运动速度的加速度
13	查找终点停止 Nm	设置查找终点停止 Nm 运动速度的加速度

加速度的单位是每秒用户单位数 2。请注意，此值必须大于 0.001。超出此范围的值将导致驱动器进入故障状态，并报告错误代码 3（数据超出范围）。

PLC 应使用适当的数据（以及包含与前面和后面描述的命令相关的参数的其他输出寄存器）预加载加速度寄存器，然后通过生成控制字第 7 位的上升（0->1）沿来触发命令

减速度：Modbus 寄存器偏移 10/11，NETFLOAT（5）

PLC 使用这些寄存器在驱动器上加载减速率。下表详细说明了减速率值与每个特定命令的关系：

命令编号	功能	减速度的作用
17	回零	为回零速度设置减速度
1	相对运动	为相对运动的速度设置减速度
2	绝对运动	为绝对运动的速度设置减速度
3	以恒定速度运行（启用位置循环）	为 jog speed 设置减速度
4	相对增量移动	为相对增量的速度设置减速度
5	绝对增量移动	为绝对增量的速度设置减速度

6	设置位置	未使用
7	跟随	未使用
8	速度参考	根据速度需求设置减速度
9	扭矩参考	未使用
10	取消运动	未使用
11	受控停止	未使用
12	查找终点停止	设置查找终点停止运动速度的减速度
13	查找终点停止 Nm	设置查找终点停止 Nm 运动速度的减速度

减速率的单位是用户单位/秒²。请注意，此值必须大于 0.001。超出此范围的值将导致驱动器进入故障状态，并报告错误代码 3（数据超出范围）。

PLC 应使用适当的数据（以及包含与前面和后面描述的命令相关的参数的其他输出寄存器）预加载减速率寄存器，然后通过生成控制字第 7 位的上升（0->1）沿来触发命令

加加速度：Modus 寄存器偏移 12/13，NETFLOAT（6）

PLC 使用这些寄存器在驱动器上加载加加速度（S-ramp）。下表详细说明了加加速度速率值与每个特定命令的关系：

命令编号	功能	加加速度的作用
17	回零	为回零速度设置加加速度
1	相对运动	为相对运动的速度设置加加速度
2	绝对运动	为绝对运动的速度设置加加速度
3	以恒定速度运行（启用位置循环）	为 jog speed 设置加加速度
4	相对增量移动	为相对增量的速度设置加加速度
5	绝对增量移动	为绝对增量的速度设置加加速度
6	设置位置	未使用
7	跟随	未使用
8	速度参考	根据速度需求设置加加速度
9	扭矩参考	未使用
10	取消运动	未使用
11	受控停止	未使用
12	查找终点停止	设置查找终点停止运动速度的加加速度
13	查找终点停止 Nm	设置查找终点停止 Nm 运动速度的加加速度

加加速度的单位是每秒用户单位数³。请注意，此值必须大于 0.001。超出此范围的值将导致驱动器进入故障状态，并报告错误代码 3（数据超出范围）。

注意：将加加速度或减速度加加速度设置为零将强制驱动器使用梯形运动曲线。如果这两个字都包含有效的非零值，则驱动器将使用 S 斜坡运动曲线。PLC 应使用适当的数据（以及其他包含与前面和后面描述的命令相关的参数的输出寄存器）预加载加加速度寄存器，然后通过生成控制字第 7 位的上升（0->1）沿来触发命令

减加速度：Modbus 寄存器偏移 14/15，NETFLOAT（7）

PLC 使用这些寄存器在驱动器上加载减速度（S 斜坡）。下表详细说明了减速度加加速度值与每个特定命令的关系：

命令编号	功能	减加速度的作用
17	回零	为回零速度设置减加速度
1	相对运动	为相对运动的速度设置减加速度
2	绝对运动	为绝对运动的速度设置减加速度
3	以恒定速度运行（启用位置循环）	为 jog speed 设置减加速度
4	相对增量移动	为相对增量的速度设置减加速度
5	绝对增量移动	为绝对增量的速度设置减加速度
6	设置位置	未使用
7	跟随	未使用
8	速度参考	根据速度需求设置减加速度
9	扭矩参考	未使用
10	取消运动	未使用
11	受控停止	未使用
12	查找终点停止	设置查找终点停止运动速度的减加速度

13	查找终点停止 Nm	设置查找终点停止 Nm 运动速度的减加速度
----	-----------	-----------------------

减速度加速度的单位是用户单位/秒³。请注意，此值必须大于 0.001。超出此范围的值将导致驱动器进入故障状态，并报告错误代码 3（数据超出范围）。

将加速加加速度或减速度加加速度设置为零将强制驱动器使用梯形运动曲线。如果这两个字都包含有效的非零值，则驱动器将使用 S 斜坡运动曲线。

PLC 应使用适当的数据预加载减速度加加速度寄存器（以及包含与前面和后面描述的命令相关的参数的其他输出寄存器），然后通过生成控制字第 7 位的上升（0→1）沿来触发命令。

锁存器偏移：Modbus 寄存器偏移 16/17，NETFLOAT（8）

PLC 使用这些寄存器加载与捕获的快速中断位置的偏移距离，该位置随后用作驱动器的新目标位置（此功能通常用于分度输送机，其中轴必须始终停在经过参考传感器的固定距离处）。下表详细说明了 Latch Offset 值与每个特定命令的关系：

命令编号	功能	Latch Offset 作用
17	回零	未使用
1	相对运动	未使用
2	绝对运动	未使用
3	以恒定速度运行（启用位置循环）	未使用
4	相对增量移动	设置与锁存位置的偏移
5	绝对增量移动	设置与锁存位置的偏移
6	设置位置	未使用
7	跟随	未使用
8	速度参考	未使用
9	扭矩参考	未使用
10	取消运动	未使用
11	受控停止	未使用
12	查找终点停止	未使用
13	查找终点停止 Nm	未使用

如果 PLC 已设置控制字的第 2 位（以启用“快速”位置锁存），并且驱动器在相对增量移动或绝对增量移动期间收到锁存事件，则驱动器将使用 Latch Offset 字中包含的数据为移动设置新的目标位置。移动目标将成为捕获的（快速）位置加上定义的偏移距离。

Latch Offset 是一个缩放的数量（即，单位与驱动器上的 SCALEFACTOR 设置相关）。PLC 应该用适当的数据预加载 Latch Offset 寄存器（以及包含与前面描述的命令相关的参数的其他输出寄存器），然后通过生成控制字第 7 位的上升（0→1）沿来触发增量移动命令之一。

DO 状态强制：Modbus 寄存器偏移 18/19，NETFLOAT（9）

PLC 使用这些寄存器来强制打开或关闭数字输出。

命令编号	功能	Latch Offset 作用
0	强制数字输出 0	未使用
1	强制数字输出 1	未使用
2	强制数字输出 2	未使用
3	强制数字输出 3（e180 + e190 仅适用于 SIO-01）	未使用
4	强制数字输出 4（e180+ e190 仅适用于 SIO-01）	未使用
5	强制数字输出 5（e190 仅适用于 SIO-01）	未使用
6	强制数字输出 6（e190 仅适用于 SIO-01）	未使用

注意：只有在 Mint 程序中没有分配的函数时，这才有效 - 例如 motorbrakeoutput（0）

状态字: Modbus 寄存器偏移量 200/201, NETINTEGER (100)

PLC 使用状态字中的位来确定驱动器上的特定条件。PLC 可以使用这些位来确定先前发出的命令何时完成。Mint 程序会自动填充此单词中的位。

位	功能	状态 0	状态 1
0	启用状态	驱动器已禁用	驱动器已启用
1	空闲状态	驱动器未空闲 ¹	驱动器处于空闲状态 ¹
2	位置	驱动器不在最后一个目标位置的值 IDLEPOS2	驱动器在最后一个目标位置的值 IDLEPOS2 则驱动器在 Value 范围内
3	电机制动状态 ³	电机制动器被释放	电机制动器被应用
4	回零状态	驱动器未成功完成 Home Sequence	驱动器已成功完成 Home Sequence
5	正向限位状态	正向限位输入处于非激活状态	正向限位输入处于激活状态
6	反转限位状态	反向限位输入处于非激活状态	反向限位输入处于激活状态
7	故障状态	无故障	驱动器上出现故障（有关更多信息，请参阅错误代码字）
8	停止输入状态	停止输入处于非激活状态	停止输入处于激活状态
9	准备启用状态	驱动器尚未准备好启用	驱动器已准备好启用
10-11	控制模式	这两个位组合报告驱动器的激活控制模式（电流/速度/位置控制）——请参阅 Mint 帮助文件中的 CONTROLMODE	
12	触发器状态	控制字 bit 7 已清除	控制字 bit 7 已设置
13	启用操作状态	控制字 bit 1 已清除	控制字 bit 1 已设置
14	丢失的锁存状态	已检测到锁存（如果使用）	锁存（如果使用）已丢失
15	故障复位状态	控制字 bit 6 已清除	控制字 bit 6 已设置
16	寻相状态	寻相未完成	寻相已完成
17	暂停状态	移动未暂停	移动暂停
18	Watchdog Okay	通信不正常	通信正常
19	已达到目标速度	未达到目标速度	达到目标速度
20-31	备用	未使用	未使用

¹ 有关构成空闲条件的详细信息，请参阅有关 IDLE 的 Mint 帮助文件主题（另请参阅 IDLEMODE、IDLEVEL、IDLEPOS 和 IDLETIME）

² 请参阅 IDLEPOS 上的 Mint 帮助文件主题

³ 仅当使用 Mint Workbench 设置与此功能相关的各种 MOTORBRAKExxxxxxx 参数时，电机制动控制才会激活。请参阅所有以 MOTORBRAKE 开头的 Mint 帮助文件主题

测量位置: Modbus 寄存器偏移 202/203, NETFLOAT (101)

PLC 使用这些寄存器来读取驱动器的当前位置。该值是一个缩放量（即它使用的单位取决于驱动器上的 Mint SCALEFACTOR 设置）。为了允许 GDI 控制模态轴和非模态轴，这个 register 实际上反映了 ENCODER (0) 的缩放值。这允许根据 ENCODERWRAP (0) 的设置来获得轴的位置。有关更多详细信息，请参阅有关 ENCODER 和 ENCODERWRAP 的 Mint 帮助文件主题。

测量速度: Modbus 寄存器偏移 204/205, NETFLOAT (102)

PLC 使用这些寄存器来读取驱动器的当前速度。该值是一个缩放量（即它使用的单位取决于驱动器上的 Mint SCALEFACTOR 设置）。有关更多详细信息，请参阅有关 VEL 的 Mint 帮助文件主题。

跟随误差: Modbus 寄存器偏移 206/207, NETFLOAT (103)

PLC 使用这些寄存器来读取驱动器的当前位置误差。该值是一个缩放量（即它使用的单位取决于驱动器上的 Mint SCALEFACTOR 设置）。有关更多详细信息，请参阅有关 FOLERROR 的 Mint 帮助文件主题。

轴模式: Modbus 寄存器偏移 208/209, NETINTEGER (104)

PLC 使用这些寄存器来读取驱动器上的当前操作模式。有关更多详细信息，请参阅有关 AXISMODE 的 Mint 帮助文件主题。

测得的 RMS 电流: Modbus 寄存器偏移 210/211, NETFLOAT (105)

PLC 使用这些寄存器来读取驱动器产生的测得的 RMS 电流（以安培为单位）。有关更多详细信息，请参阅 CURRENTMEAS 上的 Mint 帮助文件主题。

错误代码: Modbus 寄存器偏移量 212/213, NETINTEGER (106)

PLC 使用这些寄存器从驱动器读取当前故障/错误代码。零值表示没有故障处于激活状态（此外，Status 字的 bit 7 表示是否存在故障）。有关更多详细信息，请参阅有关 ERRCODE 的 Mint 帮助文件主题。

DI 状态: Modbus 寄存器偏移 214/215, NETINTEGER (107)

PLC 使用这些寄存器来读取驱动器数字输入的状态。这将查看 digital inputs 的当前状态。如果 input 处于活动状态，则将设置 bit。对于 edge triggered inputs，如果 edge 已被锁定，则 bit 将被设置。

DO 状态: Modbus 寄存器偏移量 216/217, NETINTEGER (108)

PLC 使用这些寄存器来读取驱动器数字输出的状态。

实际扭矩 (Nm) : Modbus 寄存器偏移 218/219, NETINTEGER (109)

PLC 使用这些寄存器来读取驱动器计算的电机的实际扭矩。该值以 Nm 表示。

示例序列

本应用说明中包含的示例程序中的 IEC 61131 PLC 功能块可自动执行下述序列，使 GDI 对所有应用都非常易于使用。这些序列仅用于用户希望设计/实现自己的序列控制功能时的信息。

启用驱动器

在下载 the Mint 程序中，_bRemoteEnable Mint 常数设置为 _True（因此 PLC 最终控制启用状态）。确保驱动器上都存在本地联锁装置（例如，停止输入、驱动器启用输入、交流电源）。

如果驱动器已准备好启用，则状态字将通过“Ready to Enable”位指示这一点

如果是这样，则可以在控制字中设置 Enable bit

驱动器应启用，并且应设置 Status 字的 bit 0 以指示这一点

执行 Home

确保驱动器已启用（见上文）

将 Home back off ratio 的值加载到 Value 字中

将回零速度的值加载到 Speed 字中

将回零加速度的值加载到 Accel 字中

将回零减速率的值加载到 Decel 字中

如果需要，提供加速和减速加速度的负载值

在 Command type 字中写入 17 以此将命令类型设置为 Homing

写入 Command 字的 1 到 bit 1 以允许执行运动

通过写入 Command 字的 1 到 bit 7 位来触发移动

应该进行编程的移动。使用状态位（Idle、InPos、Homed 和 Fault）来检查这一点。

将控制字的 0 到 bit 7 写入，为下一个命令做好准备。

执行相对移动

确保驱动器已启用（见上文）

将移动距离的值加载到 Value 字中

加载移动到 Speed 字的 slew speed 值

将加速时间的值加载到 Accel 字中

将减速时间的值加载到 Decel 字中

如果需要，加载加速和减速加速度的值

将 1 写入 Command type 字以选择相对移动
写入 Command 字的 1 到 bit 1 以允许执行运动
通过写入 Command 字的 1 到 bit 7 位来触发移动
应该进行编程的移动。使用状态位 (Idle, InPos 和 Fault) 来检查这一点。
将控制字的 0 到 bit 7 写入, 为下一个命令做好准备。

注意: 如果需要进一步移动, 则无需为所有或任何不需要更改的参数重新设置值 - 驱动器将保留当前值。

执行点动

确保驱动器已启用 (见上文), 并确保将控制字的 bit 1 设置为允许移动
将 Jog Speed 的值加载到 Speed 字中
根据需要设置加速/减速和加加速度 (请参阅前面的示例)
将 3 写入 Command type 字以选择 Jog
将 Command 字的 bit 7 位设置为 1 以开始轴点动

可以通过写入值 0、清除 Enable Operation 位、发出新命令、激活驱动器上的本地停止输入 (如果已分配) 或禁用驱动器来停止点动

执行跟随运动

确保驱动器已启用 (见上文)
确保将控制字的 bit 1 位设置为允许移动
将跟随比率的值加载到 Value word
确保轴为 IDLE (状态字的 bit 1, 已设置 NETINTEGER (100)) 或已跟随 (Axismode / NETINTEGER (104) 返回值 128)
将 7 写入 Command type 字以选择 Following
将 Command 字的 bit 7 位设置为 1 以开始主参考之后的轴 (如果轴已经跟随, 则发出新的跟随比率)

可以通过写入值 0 并触发另一个 follow、清除 Enable Operation 位、发出新命令、激活驱动器上的本地 stop input (如果已分配) 或禁用驱动器来停止 Follow。

执行速度参考运动

确保驱动器已启用 (见上文)
确保将控制字的 bit 1 位设置为允许移动
将速度参考的值 (以用户单位/秒为单位) 加载到 Speed 字中
加载 accel/decel/acceljerk/deceljerk 的值, 如前面的“执行点动”中所述
将 8 写入 Command type 字以选择速度参考模式
将控制字的 bit 7 设置为 1, 以启动轴以编程的速度参考运行

可以通过将速度参考写入零、清除 Enable Operation 位、发出新命令、激活驱动器上的本地停止输入 (如果已分配) 或禁用驱动器来停止速度参考。请注意, 在发送需要不同操作模式的新命令之前, 驱动器将保持速度参考模式。如果用户要求驱动器返回位置控制模式, 保持其当前位置并具有最大扭矩, 则 PLC 应发出 0 的相对移动 (Command type 为 1, Value 为 0)。

执行扭矩参考运动

确保驱动器已启用 (见上文)
确保将控制字的 bit 1 位设置为允许移动
将扭矩参考的值 (以百分比表示) 加载到 Value 中
将 9 写入 Command type 字以选择扭矩参考模式
将控制字的 bit 7 设置为 1, 以启动以编程的扭矩参考运行的轴
可以通过清除 Enable Operation 位、发出新命令、激活驱动器上的本地停止输入 (如果已分配) 或禁用驱动器来停止扭矩参考。请注意, 在发送需要不同操作模式的新命令之前, 驱动器将保持扭矩参考模式。如果用户要求驱动器返回位置控制模式, 以最大扭矩保持其当前位置, PLC 应发出 0 的相对移动 (Command type 为 1, Value 为 0)。

查找限位块（扭矩限制回零）

确保驱动器已启用（见上文）

确保将控制字的 bit 1 设置为允许移动

将扭矩限制值（占驱动器额定电流的百分比）加载到 Value 中

将用于查找终点停止点（以用户单位/秒为单位）的速度参考值加载到速度字中（该值的符号确定行进方向）

加载 accel/decel/acceljerk/deceljerk 值，如前面的“执行点动”中所述

将 12 写入 Command type 字以选择查找结束停止模式

将 Command 字的 bit 7 设置为 1，以启动以编程的速度参考运行的轴

可以通过清除 Enable Operation 位、发出新命令、激活驱动器上的本地停止输入（如果已分配）或禁用驱动器来停止查找结束停止。请注意，在发送需要不同操作模式的新命令之前，驱动器将保持位置控制模式（在确定的终端停止处保持位置）。

PLC 示例程序

随附的 zip 文件附带一个用于控制驱动器的示例程序，其中包含应用说明（为通过 Modbus TCP 进行控制而编写），其中包括三个主要元素：

- 一. Mint 通用驱动接口（GDI）中包含的每种运动命令类型的功能块。功能块在操作上与用于运动控制的 PLCopen 功能块非常相似
- 二. 一个数据接口功能块。此代码负责将应用程序级数据（例如，从功能块使用）路由到 Modbus 通信层
- 三. 读/写 Modbus 数据的功能块

为了说明 PLC 代码和 Mint GDI 的使用，每个 PLC 示例程序都包含两个可视化。

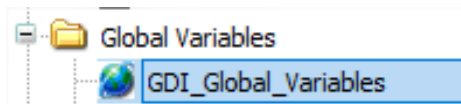
- 一. 用于操作每个可用 GDI 功能块并监控驱动器状态信息的可视化。此可视化项由几个较小的预先编写的可视化项（也包括在内）组成。这些可以非常轻松地重新用于其他轴，只需将占位符分配给与每个新轴关联的相应功能块即可
- 二. 用于监控 Modbus 数据传输运行和 Modbus 通信状态的可视化

创建/定义轴

编写示例 PLC 程序以允许通过 Modbus TCP 控制一个驱动器（通过 GDI）。但是，添加其他轴非常简单。

对于 Modbus TCP，添加附加轴的过程如下：

- 一. 在 CoDeSys 应用程序的“资源”选项卡中，双击“GDI_Global_Variables”图标...



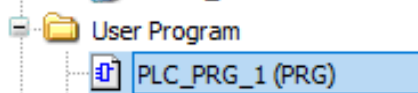
- 二. 对于每个需要的额外轴，请为 TGDIAxisRef 类型的轴结构添加一个额外的声明。下面的示例显示了两个额外的轴...
(* 根据需要添加轴数据类型 *)

```
t 轴 1;TGDIAxisRef;
t 轴 2;TGDIAxisRef;
tAxis3 : TGDIAxisRef;
```

- 三. 根据需要编辑 WatchdogTime 的声明。这是驱动器预期“watchdog bit”打开和关闭的时机。
(* 看门狗时间.....需要是驱动器看门狗时间的一半 (或更少) *)

```
看门狗时间: 时间 := T#500MS;
```

- 四. 现在双击 PLC_PRG_1 POU 图标或创建新的 POU...



要允许 PLC 与附加驱动器交换控制和状态数据，请调用 GDI_DATAINTERFACE_TCP 功能块的新实例（将相关的 Axis 结构和 IP 地址作为参数传递给此块）。

- 五. 根据需要将应用程序代码添加到 PLC 程序中，以控制附加轴（例如，包括对所需 GDI 功能块的调用的新实例）

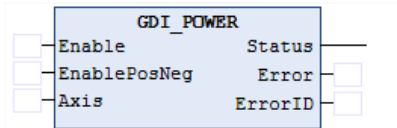
PLC 可以控制的轴数是 PLC 可用内存的函数。示例程序是为 PM5072-T-2ETH eco-PLC 编写的，该 PLC 为应用程序代码提供 1024kB 内存。

作为参考，双轴示例程序使用大约 126kB（12%）的 PM5072 可用程序代码和数据存储。每个额外的轴会额外使用 13kb 的存储空间和（1.1%）的总存储空间。

GDI 功能块

GDI_POWER

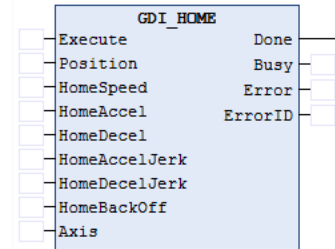
此功能块用于启用/禁用轴。使能输入启用驱动器中的功率阶段，而不是功能块本身。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
Enable	Bool	当 True 时，PLC 将请求启用轴
启用 PosNeg	Bool	为 True 时允许两个方向的运动。如果为 False 时，阻止 motion（如果运动已经在进行中，则执行停止）
VAR_OUTPUT		
状态	Bool	指示轴是否已启用（True）或未启用（False）
错误	Bool	如果轴出错，则设置 True
错误 ID	DINT	指示轴报告的 Mint 错误代码

GDI_HOME

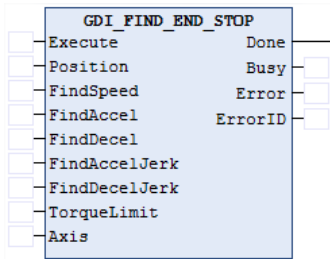
此功能块用于将轴位置基准化到专用的 Home 开关/传感器位置。基准序列的详细信息取决于在 Mint GDI 程序中设置的 Home 类型。Position（位置）输入用于设置成功基准序列结束时的轴位置。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升边上开始回零序列
位置	Real	在成功的回零序列结束时设置的绝对位置
HomeSpeed	Real	回零速度（以用户单位/秒为单位）
HomeAccel	Real	以用户单位/sec ² 为单位的回零加速速率
HomeDecel	Real	回零减速率（以用户单位/sec ² 为单位）
HomeAccelJerk	Real	以用户单位/sec ³ 为单位的回零加速加速度（对于梯形运动，设置为 0）
HomeDecelJerk	Real	以用户单位/sec ³ 为单位的回零减速加速度（对于梯形运动，设置为 0）
HomeBackOff	Real	原点速度与回退速度的比率
VAR_OUTPUT		
Done	Bool	指示轴已成功回零。如果在回零期间删除了 Execute 输入，并且轴完成了 home 序列，则 Done 输出一个 PLC 扫描周期。如果成功且 Execute input（执行）输入保持为 True，则 Done（完成）输出将保持设置状态。
Busy	Bool	在回零序列正在进行时设置 True
错误	Bool	如果轴出错，则设置 True
错误 ID	DINT	指示轴报告的 Mint 错误代码

GDI_FIND_END_STOP

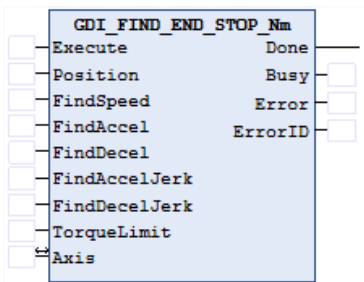
在没有专用的原点开关/传感器的情况下，此功能块可以用作 GDI_HOME 的替代方案，以将轴基准到行程终点的物理限制位置。Position（位置）输入用于设置成功基准序列结束时的轴位置。此处，用于确定终端止动位置的扭矩限制以驱动器标称电流的 % 为单位。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升边上开始回零序列
位置	Real	在成功的回零序列结束时设置的绝对位置
查找速度	Real	以用户单位/秒为单位的速度（+ve 值导致 +ve 方向，-ve 值导致 -ve 方向）
FindAccel	Real	以用户单位/sec ² 为单位的回零加速速率
FindDecel	Real	回零减速率（以用户单位/sec ² 为单位）
FindAccelJerk	Real	以用户单位/sec ³ 为单位的回零加速加速度（对于梯形运动，设置为 0）
FindDecelJerk	Real	以用户单位/sec ³ 为单位的回零减速加速度（对于梯形运动，设置为 0）
扭矩限制	Real	扭矩（电流）限制以驱动器额定电流的百分比表示
VAR_OUTPUT		
Done	Bool	指示轴已成功找到终点。如果在回零期间删除了 Execute 输入，并且轴完成了查找结束停止序列，则 Done 输出一个 PLC 扫描周期。如果 Execute 输入保持 True，则 Done 输出也将保持设置状态（前提是查找结束停止序列成功）
Busy	Bool	在 find end stop sequence 正在进行时输出为 True
错误	Bool	如果轴出错，则输出为 True
错误 ID	DINT	指示轴报告的 Mint 错误代码

GDI_FIND_END_STOP_Nm

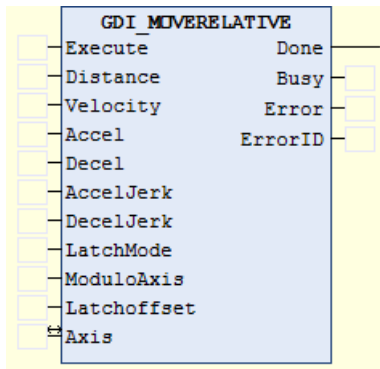
在没有专用的原点开关/传感器的情况下，此功能块可以用作 GDI_HOME 的替代方案，以将轴基准到行程的终点物理限制位置。Position（位置）输入用于设置成功基准序列结束时的轴位置。这是 **GDI_FIND_END_STOP** 的替代方案，此处用于确定终端挡块位置的扭矩限制以牛顿米（Nm）为单位，在电机驱动轴处计算。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升边上开始回零序列
位置	Real	在成功的回零序列结束时设置的绝对位置
查找速度	Real	以用户单位/秒为单位的速度（+ve 值导致 +ve 方向，-ve 值导致 -ve 方向）
FindAccel	Real	以用户单位/sec ² 为单位的回零加速速率
FindDecel	Real	回零减速率（以用户单位/sec ² 为单位）
FindAccelJerk	Real	以用户单位/sec ³ 为单位的回零加速加速度（对于梯形运动，设置为 0）
FindDecelJerk	Real	以用户单位/sec ³ 为单位的回零减速加速度（对于梯形运动，设置为 0）
扭矩限制	Real	电机驱动轴上的扭矩极限以 Nm 表示
VAR_OUTPUT		
Done	Bool	指示轴已成功找到终点。如果在回零期间删除了 Execute 输入，并且轴完成了查找结束停止序列，则 Done 输出一个 PLC 扫描周期。如果 Execute 输入保持 True，则 Done 输出也将保持设置状态（前提是查找结束停止序列成功）
Busy	Bool	在 find end stop sequence 正在进行时输出为 True
错误	Bool	如果轴出错，则输出为 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_MOVERELATIVE

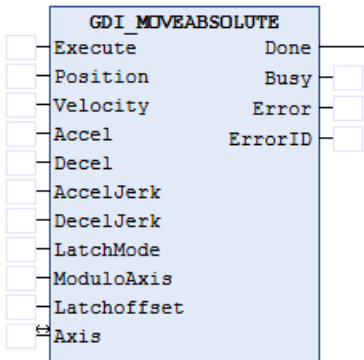
此功能块用于命令相对于执行时设定位置指定的距离的受控运动。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升边上开始运动
距离	Real	移动的相对距离（以用户单位为单位）
速度	Real	最大速度（不一定达到）的值（以用户单位/秒为单位）
加速	Real	加速度速率（以用户单位/秒 ² 为单位）
减速度	Real	减速度速率（以用户单位/sec ² 为单位）
加速度	Real	以用户单位/sec ³ 为单位的加速加加速度（对于梯形运动，设置为 0）
DecelJerk	Real	减速加加速度（以用户单位/sec ³ 为单位）（对于梯形运动，设置为 0）
LatchMode 锁存模式	Bool	未使用。请忽略
ModuloAxis 模轴	Bool	定义轴是否为模态轴（即使用 ENCODERWRAP 定义一个周期内的行程）。使用模态轴时的绝对移动始终通过最短路径实现（例如，在 0-360 度模态轴上从 350 度绝对移动到 20 度将导致向前移动 30 度）
Latchoffset 锁存偏移值	Real	未使用。请忽略
VAR_OUTPUT		
Done	Bool	指示轴已成功到达目标位置。如果在运动过程中删除了 Execute 输入，并且相对移动完成，则 Done 输出一个 PLC 扫描周期。如果 Execute 输入保持 True，则 Done 输出也将保持设置状态（前提是成功达到目标位置）
Busy	Bool	在相对移动正在进行时输出为 True
错误	Bool	如果轴出错，则输出为 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_MOVEABSOLUTE

此功能块用于命令受控运动到指定的绝对位置。此功能可以与 Modulo 轴一起使用（在这种情况下，将采用到指定位置的最短路线）。



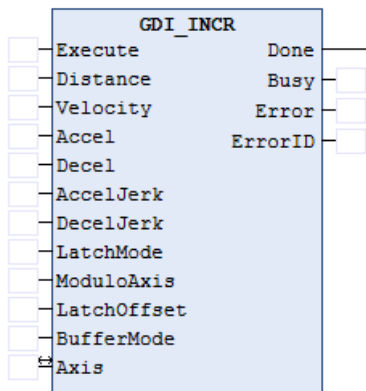
	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升边上开始运动
位置	Real	移动的目标位置（以用户单位为单位）
速度	Real	最大速度（不一定达到）的值（以用户单位/秒为单位）
加速	Real	加速度速率（以用户单位/秒 ² 为单位）
减速度	Real	减速度速率（以用户单位/sec ² 为单位）
加速度	Real	以用户单位/sec ³ 为单位的加速加加速度（对于梯形运动，设置为 0）
DecelJerk	Real	减速加加速度（以用户单位/sec ³ 为单位）（对于梯形运动，设置为 0）
ModuloAxis 模轴	Bool	定义轴是否为模态轴（即使用 ENCODERWRAP 定义一个周期内的行程）。使用模态轴时的绝对移动始终通过最短路径实现（例如，在 0-360 度模态轴上从 350 度绝对移动到 20 度将导致向前移动 30 度）
LatchMode 锁存模式	Bool	未使用，请忽略
LatchOffset（锁存器偏移）	Real	未使用，请忽略
VAR_OUTPUT		
Done	Bool	指示轴已成功到达目标位置。如果在运动过程中删除了 Execute 输入并且绝对移动完成，则 Done 输出为一个 PLC 扫描周期。如果 Execute 输入保持 True，则 Done 输出也将保持设置状态（前提是成功达到目标位置）
Busy	Bool	在绝对移动正在进行时输出为 True
错误	Bool	如果轴出错，则输出为 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_INCR

此功能块用于在执行时命令相对于目标位置指定距离的受控运动。当运动仍在进行时，可以通过以下任一方法修改调用此功能块产生的目标位置：

- 通过发出另一个 GDI_INCR 或 GDI_INCA 函数（提供输入参数 BufferMode 为 True）
- 通过将输入参数 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，以便停止超过定义的快速中断捕获的轴位置的 'LatchOffset distance'。轴状态字（btLatchMissed）中的位可用于指示无法检测到此快速中断（示例程序显示了如何检测连续缺失的 3 个锁存器 - 例如，此条件可用于提醒操作员系统故障）。

如果应用程序需要修改正在进行的相对移动的 SPEED/ACCEL/DECEL GDI_INCR 也很有用。使用 GDI_MOVERELATIVE 加载的移动将使用当时加载的 SPEED/ACCEL/DECEL 进行分析，一旦移动开始，这些就无法更改。通过使用 GDI_INCR 并将输入参数 BufferMode 设置为 True，然后通过加载另一个输入参数 Distance 设置为零的 GDI_INCR（使用新的 SPEED/ACCEL/DECEL）来修改配置文件参数。



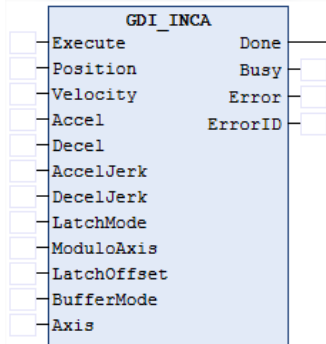
	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升边上开始运动
距离	Real	移动的相对距离（以用户单位为单位）
速度	Real	最大速度（不一定达到）的值（以用户单位/秒为单位）
加速	Real	加速度速率（以用户单位/秒 ² 为单位）
减速度	Real	减速度速率（以用户单位/sec ² 为单位）
加速度	Real	以用户单位/sec ³ 为单位的加速加速度（对于梯形运动，设置为 0）
DecelJerk	Real	减速加速度（以用户单位/sec ³ 为单位）（对于梯形运动，设置为 0）
LatchMode 锁存模式	Bool	设置轴是否应使用配置的快速锁存中断，并在捕获位置之后设置新的目标位置 'LatchOffset' 用户单位
ModuloAxis 模轴	Bool	定义轴是否为模态轴（即使用 ENCODERWRAP 定义一个周期内的行程）。使用模态轴时的绝对移动始终通过最短路径实现（例如，在 0-360 度模态轴上从 350 度绝对移动到 20 度将导致向前移动 30 度）
LatchOffset（锁存器偏移）	Real	定义 GDI_INCR 目标应修改的经过捕获的快速位置（以用户单位为单位）的距离（当输入参数 LatchMode 设置为 True 时）
BuffMode	Bool	定义功能块是否应设置 Done 输出并在加载移动后立即完成。设置 BufferMode True 允许应用程序在现有移动正在进行时触发进一步的增量移动
VAR_OUTPUT		
Done	Bool	当 BufferMode 设置为 False 时，表示轴已成功到达目标位置。如果在运动过程中删除了 Execute 输入，并且相对移动完成，则 Done 输出一个 PLC 扫描周期。如果 Execute input 保持 True，则 Done 输出也将保持设置状态（假设成功达到目标位置）。当 BufferMode 设置为 True 时，将为一次 PLC 扫描设置 Done 输出，以指示移动加载成功
Busy	Bool	在功能块正在进行时输出为 True
错误	Bool	如果轴出错，则输出为 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_INCA

此功能块用于命令受控运动到指定的绝对位置。此功能与 GDI_MOVEABSOLUTE 的不同之处在于，在运动过程中，可以通过以下任一方法修改目标位置：

- 通过发出另一个 GDI_INCR 或 GDI_INCA 函数（提供输入参数 BufferMode 为 True）
- 通过将输入参数 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，以便停止 LatchOffset 距离超过定义的快速中断捕获的轴位置。轴状态字（btLatchMissed）中的位可用于指示无法检测到此快速中断（示例程序显示了如何检测连续缺失的 3 个锁存器 - 例如，此条件可用于提醒操作人员系统故障）。

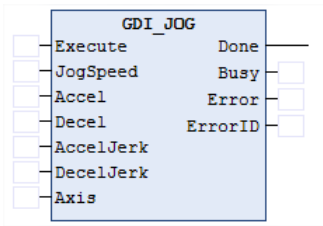
与 GDI_INCR GDI_INCA 一样，如果应用程序需要修改已经在进行的绝对移动的 SPEED/ACCEL/DECEL 也很有用。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升边上开始运动
位置	Real	移动的绝对位置目标（以用户为单位）
速度	Real	最大速度（不一定达到）的值（以用户单位/秒为单位）
加速	Real	加速度速率（以用户单位/秒 ² 为单位）
减速度	Real	减速度速率（以用户单位/sec ² 为单位）
加速度	Real	以用户单位/sec ³ 为单位的加速加加速度（对于梯形运动，设置为 0）
DecelJerk	Real	减速加加速度（以用户单位/sec ³ 为单位）（对于梯形运动，设置为 0）
LatchMode 锁存模式	Bool	设置轴是否应使用配置的快速锁存中断，并在捕获位置之后设置新的目标位置 'LatchOffset' 用户单位
ModuloAxis 模轴	Bool	定义轴是否为模态轴（即使用 ENCODERWRAP 定义一个周期内的行程）。使用模态轴时的绝对移动始终通过最短路径实现（例如，在 0-360 度模态轴上从 350 度绝对移动到 20 度将导致向前移动 30 度）
LatchOffset（锁存器偏移）	Real	定义 GDI_INCA 目标应修改的经过捕获的快速位置（以用户单位为单位）的距离（当输入参数 LatchMode 设置为 True 时）
BuffMode	Bool	定义功能块是否应设置 Done 输出并在加载移动后立即完成。设置 BufferMode True 允许应用程序在现有移动正在进行时触发进一步的增量移动
VAR_OUTPUT		
Done	Bool	当 BufferMode 设置为 False 时，表示轴已成功到达目标位置。如果在运动过程中删除了 Execute 输入，并且相对移动完成，则 Done 输出一个 PLC 扫描周期。如果 Execute input 保持 True，则 Done 输出也将保持设置状态（假设成功达到目标位置）。当 BufferMode 设置为 True 时，则 Done 输出一个 PLC 扫描周期，以指示移动加载成功
Busy	Bool	在功能块正在进行时输出为 True
错误	Bool	如果轴出错，则输出为 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_JOG

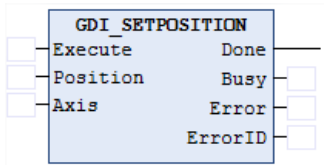
此功能块用于命令轴上的恒定速度移动（使用驱动器中的位置回路控制器）。只要 Execute 输入保持 True，就会执行 Motion。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升沿开始运动，只要输入保持 True，就保持运动。当 Execute 变为 False 时，运动以配置的 Decel 速率斜坡变为零速度
JogSpeed	Real	轴将达到的速度值（以用户单位/秒为单位）
加速度	Real	加速度速率（以用户单位/秒 ² 为单位）
减速度	Real	减速度速率（以用户单位/sec ² 为单位）
加加速度	Real	以用户单位/sec ³ 为单位的加速加加速度（对于梯形运动，设置为 0）
减加速度	Real	减速加加速度（以用户单位/sec ³ 为单位）（对于梯形运动，设置为 0）
VAR_OUTPUT		
Done	Bool	成功发出 Jog 命令后达到设定速度设置为 True，并保持设置状态，直到 Execute 变为 False 或发生轴错误
Busy	Bool	在功能块正在进行时设置 True
错误	Bool	如果轴出错，则设置 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_SETPOSITION

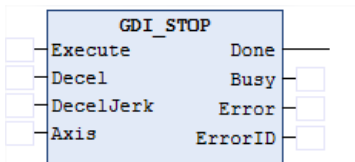
此功能块用于将轴位置（驱动器上的编码器和位置值）设置为编程值。调用此函数时，轴必须处于空闲状态，否则轴将返回“action not possible - motion in progress”错误（错误代码 10）。如果轴使用绝对编码器，这将设置/示教新的绝对位置（GDI Mint 程序 v2.17 及更高版本）。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升沿上设置新位置
位置	Real	要设置的轴位置值（以用户为单位）
VAR_OUTPUT		
Done	Bool	在发出命令后立即设置 True（无论命令是否成功 – 使用 Error 输出来确定命令是否成功）。在 Execute（执行）输入被删除之前保持 True。如果在设置 Done 位之前删除了 Execute 输入，则 Done 位将输出一个 PLC 循环周期。
Busy	Bool	在功能块正在进行时设置 True（设置 Done 位后清除）
错误	Bool	如果轴出错，则设置 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_STOP

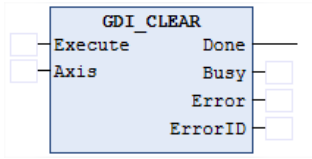
此功能块用于以编程的减速速率在轴上执行受控停止。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升沿开始受控停止
Decel	Real	减速度速率（以用户单位/sec ² 为单位）
DecelJerk	Real	减速加加速度（以用户单位/sec ³ 为单位）（对于梯形运动，设置为 0）
VAR_OUTPUT		
Done	Bool	当轴在完成受控停止后变为空闲状态或在发出停止命令时发生错误时，请设置为 True。在 Execute（执行）输入被删除之前保持 True。如果在设置 Done 位之前删除了 Execute 输入，则 Done 位将输出一个 PLC 循环周期。
Busy	Bool	在停止正在进行时设置 True – 设置 Done 位后清除
错误	Bool	如果轴出错，则设置 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_CLEAR

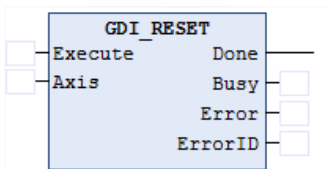
此功能块用于使轴崩溃、停止和中断任何正在进行的运动。轴将保持启用状态（前提是 GDI_POWER 请求启用状态并且轴没有错误）。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升边上启动碰撞停止
VAR_OUTPUT		
Done	Bool	当轴在完成碰撞停止后变为空闲状态或在发出碰撞停止命令时发生错误时，请设置 True。在 Execute（执行）输入被删除之前保持 True。如果在设置 Done 位之前删除了 Execute 输入，则 Done 位将输出一个 PLC 循环周期。
Busy	Bool	在停止正在进行时设置 True – 设置 Done 位后清除
错误	Bool	如果轴出错，则设置 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_RESET

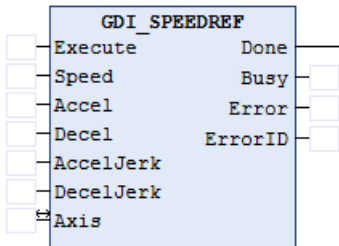
此功能块用于重置存在的任何轴错误。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升沿启动故障复位
VAR_OUTPUT		
Done	Bool	当轴不再存在错误时，设置 True。在 Execute（执行）输入被删除之前保持 True。如果在设置 Done 位之前删除了 Execute 输入，则 Done 位将输出一个 PLC 循环周期。如果无法清除错误，则不会设置 Done 位（使用 Busy 输出检测何时尝试错误重置）
Busy	Bool	在功能块尝试清除任何轴错误时设置 True
错误	Bool	如果轴出错，则设置 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_SPEEDREF

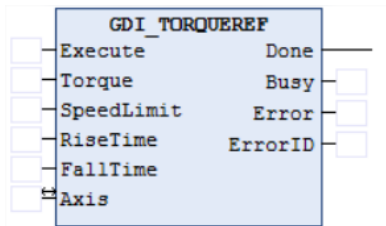
此功能块用于命令轴上的速度/速度参考。在这种操作模式下，驱动器上不使用位置环（因此不会记录或处理以下错误）。轴将保持速度控制模式，直到发出另一种控制模式类型的运动。要从零速操作（在速度控制模式下）切换到保持位置（在位置控制模式下），可以发出一个 GDI_MOVERELATIVE，例如，相对移动距离为零用户单位。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	从上升沿开始轴，只要输入保持 True，就保持运动。当 Execute 变为 False 时，运动以配置的 Decel 速率斜坡变为零速度
速度	Real	轴将达到的速度值（以用户单位/秒为单位）。可以在 Execute 为 True 时进行修改，以更改轴速度
加速	Real	加速度速率（以用户单位/秒 ² 为单位）
减速度	Real	减速度速率（以用户单位/sec ² 为单位）
加速度	Real	以用户单位/sec ³ 为单位的加速加加速度（对于梯形运动，设置为 0）
DecelJerk	Real	减速加加速度（以用户单位/sec ³ 为单位）（对于梯形运动，设置为 0）
VAR_OUTPUT		
Done	Bool	在发出速度参考后达到设定速度后设置 True。Done 输出将保持设置状态，直到 Execute 变为 False
Busy	Bool	在功能块正在进行时设置 True（即当 Execute 为 True 时）
错误	Bool	如果轴出错，则设置 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_TORQUEREF

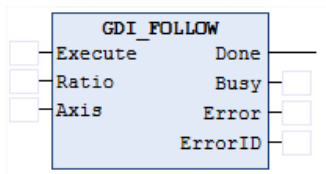
此功能块用于命令轴上的扭矩（电流）参考。在这种操作模式下，驱动器上不使用位置环（因此不会记录或处理以下错误）。轴将保持扭矩控制模式，直到发出另一种控制模式类型的运动。要从零扭矩操作（扭矩控制模式）切换到保持位置（在位置控制模式下），可以发出 GDI_MOVERELATIVE，例如，相对移动距离为零用户单位。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在上升沿启动扭矩参考，只要输入保持 True，就保持扭矩。当 Execute 变为 False 时，扭矩以配置的 FallTime 速率斜坡变为零
力矩	Real	轴将使用的扭矩参考值（以 DRIVERATEDCURRENT 的百分比 - 参见 Mint 帮助文件）。可以在 Execute 为 True 时进行修改，以更改产生的扭矩
速度限制	Real	给定当前扭矩参考值，轴可以运行的最大速度（单位/秒）的值。
上升时间	Real	设置电流从零上升到 DRIVEPEAKCURRENT 所需的时间（以毫秒为单位）（请参阅 Mint 帮助文件）
下降时间	Real	设置电流从 DRIVEPEAKCURRENT 降至零所花费的时间（以毫秒为单位）（请参阅 Mint 帮助文件）
VAR_OUTPUT		
Done	Bool	一旦发出扭矩参考，达到设定的扭矩限制时设置为 True（无论它是否成功）。Done 输出将保持设置状态，直到 Execute 变为 False
Busy	Bool	在功能块正在进行时设置 True（即当 Execute 为 True 时）
错误	Bool	如果轴出错，则设置 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_FOLLOW

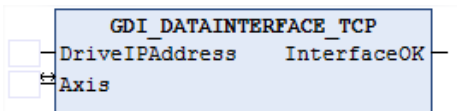
此功能块用于命令轴以编程的跟随比开始跟随配置的主编码器参考。



	类型	描述
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_INPUT		
执行	Bool	在 rising edge 上启动 follow 条件。当 Execute 输入变为 False 时，轴将保持跟随模式（要停止跟随，请发出另一个运动命令或使用 GDI_CLEAR 清除运动）
率	Real	轴和主编码器参考之间的跟随（齿轮）比值（该值将受轴的缩放和主编码器的缩放影响 - 请参阅 FOLLOW 的 Mint 帮助文件主题）。要在跟随的同时设置新的比率，必须发出新的 GDI_FOLLOW 命令
VAR_OUTPUT		
Done	Bool	在发出 follow 后立即设置 True（无论是否成功）。Done 输出将保持设置状态，直到 Execute 变为 False
Busy	Bool	在功能块正在进行时设置 True（即当 Execute 为 True 时）
错误	Bool	如果轴出错，则设置 True
错误 ID	Dint	指示轴报告的 Mint 错误代码

GDI_DATAINTERFACE_TCP

这些功能块用于在 PLC 程序的应用层和通信层之间传输命令/状态数据。应用程序中的每个轴都必须存在相关功能块的实例。

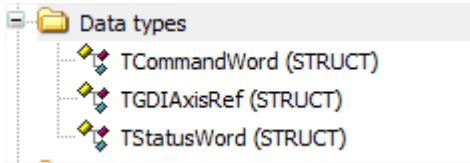


	类型	描述
VAR_IN		
驱动器 IP 地址	字符串 (80)	连接到此轴的驱动器的 IP 地址
VAR_IN_OUT		
轴	TGDIAxisRef	对轴结构的引用
VAR_OUTPUT		
接口确定	Bool	如果 Modbus 通信运行无误，则为 True。

使用 GDI 结构

为了便于使用，GDI 库包括 Structures，用于对轴引用中定义的数据进行排序，并理解 mint 程序中预定义的状态和控制字数据结构（在 mint 中称为 Bitfields）。

在这里，您可以看到存储在 'Data Types' 文件夹中的 3 个结构



TGDIAxisRef 数据类型减速如下所示：

```

1  TYPE TGDIAxisRef :
2  STRUCT
3      AxisNo: UINT;
4      CommandWord: TCommandWord;
5      CommandType: DINT;
6      Value: REAL;
7      Speed: REAL;
8      Accel: REAL;
9      Decel: REAL;
10     AccelJerk: REAL;
11     DecelJerk: REAL;
12     LatchOffset: REAL;
13     DOForce: DINT; //New
14
15     StatusWord: TStatusWord;
16     Pos: REAL;
17     Vel: REAL;
18     FolError: REAL;
19     AxisMode: DINT;
20     CurrentMeas: REAL;
21     ErrorCode: DINT;
22     DIStatus : DINT; //New
23     DOStatus : DINT; //New
24     TorqueNM : REAL; //New
25
26 END_STRUCT
27 END_TYPE

```

此数据结构又包含另外两个数据结构（*TCommandWord*和 *TStatusWord*）。这些的声明如下所示：

```

TYPE TCommandWord :
STRUCT
(*bit00*) btEnable : BOOL;
(*bit01*) btMotionAllowed : BOOL;
(*bit02*) btPosLatchEnable : BOOL;
(*bit03*) btDisFwdLimit : BOOL;
(*bit04*) btDisRevLimit : BOOL;
(*bit05*) btModulo : BOOL;
(*bit06*) btFaultReset : BOOL;
(*bit07*) btTriggerCmd : BOOL;
(*bit08*) btWatchdog: BOOL;
(*bit09*) btIgnoreFE: BOOL;

```

```

TYPE TStatusWord :
STRUCT
(*bit00*) btEnabled: BOOL;
(*bit01*) btIdle: BOOL;
(*bit02*) btInPos: BOOL;
(*bit03*) btBrakeEngaged: BOOL;
(*bit04*) btHomed: BOOL;
(*bit05*) btFwdLimit: BOOL;
(*bit06*) btRevLimit: BOOL;
(*bit07*) btFault: BOOL;
(*bit08*) btStopInput: BOOL;
(*bit09*) btReadyToEnable: BOOL;
(*bit10*) btControlMode0: BOOL;
(*bit11*) btControlModel: BOOL;
(*bit12*) btTriggerDone: BOOL;
(*bit13*) btPermitted: BOOL;
(*bit14*) btLatchMissed: BOOL;
(*bit15*) btFaultReset: BOOL;
(*bit16*) btPhaseSearchDone : BOOL;
(*bit17*) bSTOState:BOOL;
(*bit18*) bSuspendState:BOOL;
(*bit19*) bTargetVelReached:BOOL;
(*bit20*) bWatchdogOkay: BOOL;

```

GDI 的大部分功能都由示例 PLC 程序提供的各种 GDI 函数封装。但是，在某些情况下，application logic 可能会发现对 axis

structure 数据的访问很有用（例如，如示例程序所示，其中 logic 访问 idle status 和 latch missing 状态，以确定是否连续错过了 3 个 latches）。

因此，PLC 代码可以通过这些结构访问任何这些数据。

示例 1:

读取 Forward Limit Input 的状态.....

```
tAxis1.StatusWord.btFwdLimit —— Axis1ForwardLimitStatus
```

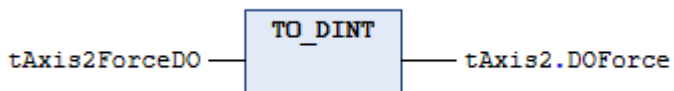
示例 2:

读取数字输入位

```
tAxis2.DIStatus.0 —— tAxis2DI0Status
```

示例 3:

强制数字输出位



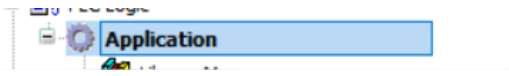
示例 4:

将实际扭矩读取到程序中

```
tAxis2.TorqueNM —— tAxis2TorqueInNm
```

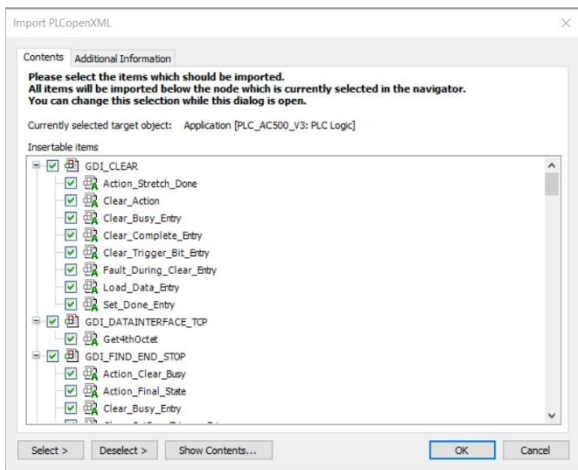
将 GDI PLC 库导入到项目中

此项目还包括一个库文件，称为“AC500 V3 (TCP) GDI Library Rev B.xml”。它包含所有相关的变量、功能块和可视化，以便在另一个 PLC 程序中使用 GDI。要导入到项目中，请先在导航树中选择 application。



注意：如果您错过了此步骤，则当您到达该步骤时，导入窗口中将看不到任何内容。

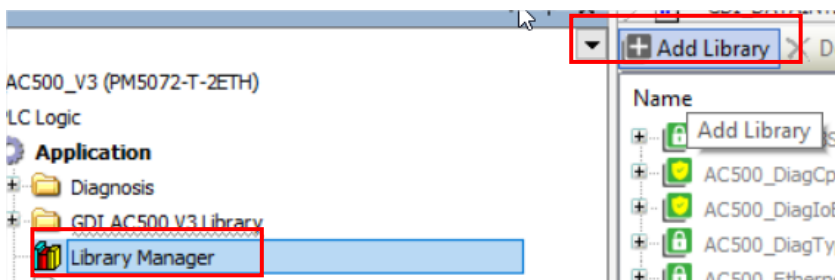
然后 **Project > Import > PLCopen XML...** 从菜单中。然后导航到文件位置，将其解压缩到您的 PC 上。然后，将打开一个导航屏幕，显示 xml 的内容，并允许您仅选择某些部分。



注意：建议您 不要在此处取消选择 任何内容，如果不需要，可以在导入后删除它们。

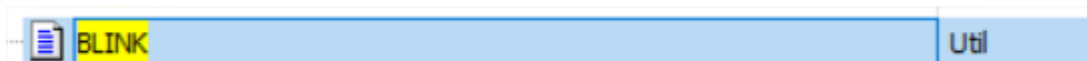
预期错误：缺少一些 PLC 库

导入 xml 文件后，将出现一些错误，这是因为 GDI 块使用了一些标准功能块元素，这些元素在默认库中不可用。要添加这些缺失的库，首先转到库管理器，然后选择“添加库”。

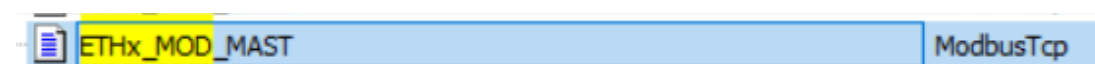


然后，您可以在 'fulltext search' 栏中一个一个地输入两个缺失的块标题，然后选择并单击 OK 添加。区块名称如下：

- 'BLINK' (Util lib)



- ETHx_MOD_MAST' (ModbusTCP 库)



添加这些库后，users 项目现在应该可以构建，没有错误，并且可以在程序中使用这些库。

它包含什么

该库当然将包含上述所有上述库块，此外，它还包含一个与之配套的可视化库。要使用这些实例，您需要将它们逐个添加到用户可视化中，然后将它们与您希望它们控制或显示的 GDI 库元素实例相关联。

有关如何执行此操作的说明，请参阅随附的示例程序。请参阅以下屏幕列表

The image displays 18 distinct GDI library blocks arranged in a grid. Each block is a rounded rectangle with a title and various input fields and buttons. The blocks are:

- Drive Status Info:** Shows status indicators (Enabled, Halt, Ready) and various drive parameters like Measured Position, Velocity, and Error Code.
- GDI_Home:** Includes fields for HomeSpeed, HomeAccel, HomeDecel, HomeAccelJerk, HomeDecelJerk, Home BackOff, and Home Position.
- GDI_TorqueRef:** Features Torque, SpeedLimit, Torque Rise Time, and Torque Fall Time.
- GDI_SpeedRef:** Contains Speed, Accel, and Decel.
- GDI_MoveAbsolute:** Includes Move Target, Speed, Accel, Decel, AccelJerk, DecelJerk, and LatchOffset.
- GDI_MoveRelative:** Includes Move Distance, Speed, Accel, Decel, AccelJerk, DecelJerk, and LatchOffset.
- GDI_Find_End_Stop:** Features FindSpeed, FindAccel, FindDecel, FindAccelJerk, FindDecelJerk, Torque Limit, and Datum Position.
- GDI_IncR:** Includes Move Distance, Speed, Accel, Decel, AccelJerk, DecelJerk, and LatchOffset.
- GDI_IncA:** Includes Move Target, Speed, Accel, Decel, AccelJerk, DecelJerk, LatchOffset, Buffer, and Modulo Axis.
- GDI_Jog:** Features Jog Speed, Accel, Decel, AccelJerk, and DecelJerk.
- GDI_Follow:** Includes Follow Ratio.
- GDI_SetPosition:** Features Position.
- GDI_Power:** Includes Enable and Allow Pos/Neg buttons.
- GDI_Reset:** Includes a Reset Fault button.
- GDI_Clear:** Includes a Cancel Motion button.
- GDI_Stop:** Includes Decel, DecelJerk, and Stop Motion buttons.
- GDI_Find_End_Stop_Nm:** Features FindSpeed, FindAccel, FindDecel, FindAccelJerk, FindDecelJerk, Torque Limit in Nm, and Datum Position.

ABB Motion AC500 PLC & Mint GDI 库中的修改历史

AC500 PLC 库 v1.7 和 Mint Library v2.35 的更改 (2024 年 12 月)

No.	描述	Mint Library	AC500 可编程控制器 Library
1	添加模态轴模式以支持模态轴应用 (_bmoduloaxis=_True 到集成模态轴)	2.35 版	
2	当 abs 编码器溢出时, 重启后动作位置错误。	已在 v2.35 中修复	
3	将 Home 命令类型从 0 更改为 17 以避免错误触发	已在 v2.35 中修复	
4	更改 bInPosition 的条件。 <2* 空闲值以适应较差的调优结果	已在 v2.35 中修复	
5	修复 issue 618283: GDI_JOG: Error ocurred, 但 errorID 为 0	已在 v2.35 中修复	
6	修复问题 619738: 状态字 (NetInteger 100) 位 8 (Stopactive) 问题	已在 v2.35 中修复	
7	解决 STOP 执行时, 带 Jerk 的 JOG 第一次触发时无法正常运行问题	已在 v2.35 中修复	
8	解决 STOP 执行时, 带 Jerk 的 SpeedRef 在第一次触发时无法正常运行问题	已在 v2.35 中修复	
9	将 bDiagnosticsOn 更改为 bCaptureOn, 如果保持原始设置, 它总是报告错误。	已在 v2.35 中修复	
10	避免纵轴触碰限位开关后禁用驱动器问题, 禁用驱动器会导致驱动器轴下降。	已在 v2.35 中修复	
11	执行定位功能 (MoveA/ MoveR/ IncR/ IncA) 时, 执行后偶现无动作。		已修复 1.7 版
12	当控制模式从位置控制更改为扭矩控制时, SpeedLimit 更新延迟导致电机快速飞动。		已修复 1.7 版
13	执行定位功能 (IncR/ IncA) 时, 偶尔不会有 Done 输出。		已修复 1.7 版

库 V1.0 中的第一个版本。

AC500 PLC 库 - 由 ABB 中国开发

项目 - 由 ABB 中国开发

联系我们

有关更多信息，请联系您的
当地 ABB 代表或以下人员之一：

new.abb.com/drives/low-voltage-ac/servo-products
new.abb.com/drives
new.abb.com/drivespartners
new.abb.com/PLC

© Copyright 2022 ABB. All rights reserved.
Specifications subject to change without
notice.