运动控制产品

应用说明
通用驱动器接口（GDI）

AN00204

Rev S (CN)

即用型 PLC 功能块，与预先编写的 Mint 应用结合起来，可通过任何可用的网络/现场总线方便的控制 e180 和 e190 驱动器

介绍

本应用说明详细描述了 MicroFlex e190 和 MotiFlex e180 驱动器的“通用驱动器接口(GDI)”Mint 程序。它是一种简单的驱动器控制协议，能通过任何提供对驱动器的 Netdata 数组的访问的通信接口来访问（比如，Modbus TCP、EtherCAT、Ethernet/IP 或 PROFINET）。它允许 PLC（比如 ABB AC500 或 eco-PLC）监控和控制一根或多根轴上的简单运动功能。本文还包括了 ABB PLC 程序的示例（需要 V1.1.2 或更新版本的 Automation Builder），它们采用“PLCopen 运动控制”风格的功能块。

本应用说明着重描述了通过 Modbus TCP 执行的操作，但其原理对所有可用的通讯协议都相同。请参考 ABBmotion 网站了解针对其它现场总线通讯的 GDI 应用说明（比如，针对 Ethernet/IP 的 AN00222，针对 EtherCAT 的 AN00234，针对 PROFINET 的 AN00251）。如果你需要关于较老的 ABB 运动产品，比如 MicroFlex e150 的 GDI 应用的传统支持信息，请联系你当地的 ABB 支持团队。

本应用说明中的示例程序为 ABB PLC 提供功能：

- 发送一条寻零命令
- 发送一条寻终点停车命令（寻零到预设的转矩限值，需要 5868 及以上的固件版本）
- 发送一条相对定位命令
- 发送一条绝对定位命令
- 发送一条增量相对定位命令（可选择在经过“快速锁存”位置一段设定距离后停车）
- 发送一条增量绝对定位命令（可选择在经过“快速锁存”位置一段设定距离后停车）
- 为增量式运动设置一个偏移目标（即相对于记录的快速中断位置来定位轴）
- 点动轴
- 设置轴位置
- 发送一个速度给定值
- 发送一个转矩给定值
- 使能/停用轴
- 使能/停用硬限位
- 复位轴错误

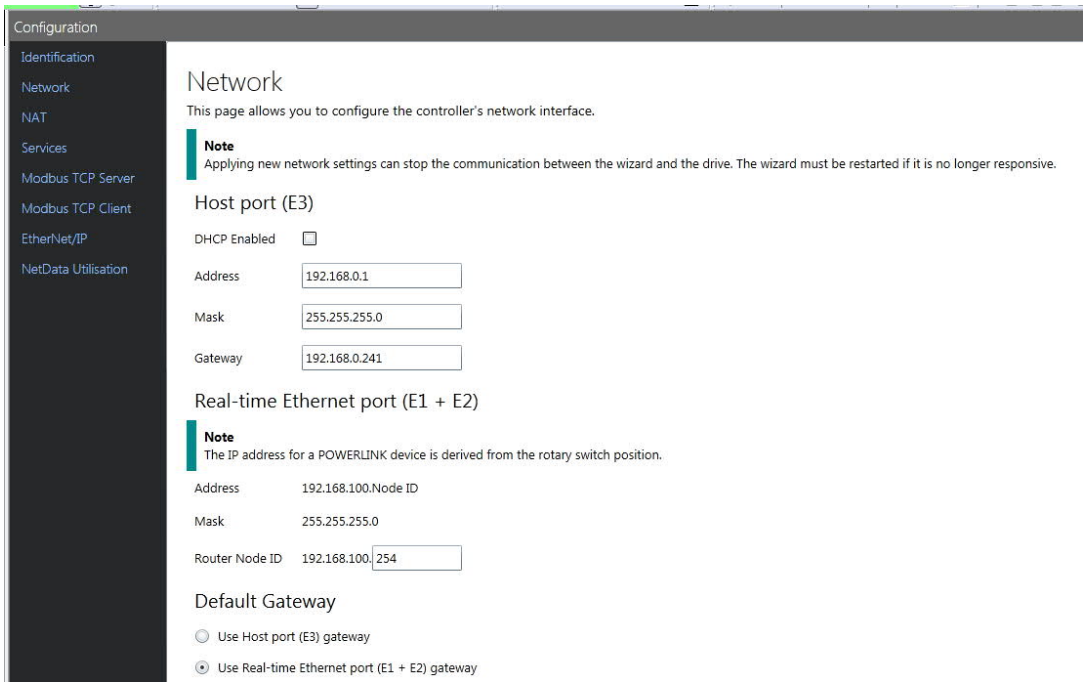
- 在轴上执行受控停车或紧急停车
- 使轴跟随第二个编码器输入
- 为所有运动设置速度、加速时间、减速时间和 S 曲线
- 控制旋转轴或非旋转轴

同时，PLC 还能监视来自驱动器的状态信息，包括：

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

所有这些操作都是通过 Modbus 寄存器来实现的。我们的接口使用 32 位数据，每个值采用两个 16 位的 Modbus 寄存器，而寄存器又被映射到驱动器中的一个 32 位的 NETINTEGER 或 NETFLOAT 位置上。其中还包括了可选的看门狗机制，允许驱动器在通讯丢失时采取动作（默认紧急停车）。

通讯配置

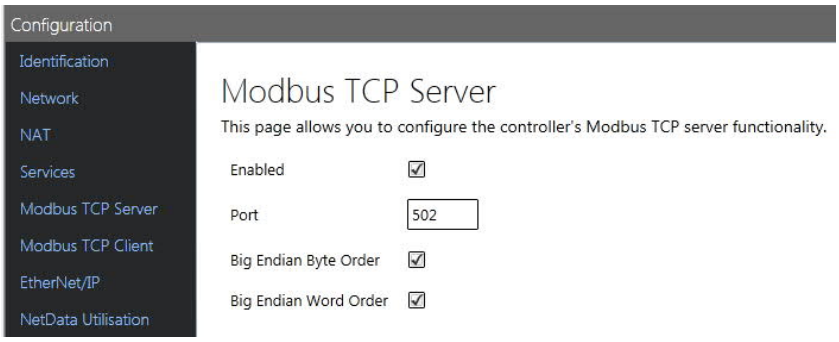


MotiFlex e180 和 MicroFlex e190 驱动器对 Modbus TCP 的原生支持。映射在 Modbus 寄存器和 Netdata 之间自动完成。Mint WorkBench 的“配置”部分允许用户为所有可用的通讯接口配置运行参数。“网络”部分允许用户设置驱动器的 IP 地址，比如...（参见上图）

MicroFlex e190 和 MotiFlex e180 驱动器的默认 IP 地址为 192.168.0.1（在使用推荐的 58xx 版本固件时）。它与 PLC 完美匹配，因为 PLC 的默认 IP 地址为 192.168.0.10（即，它们位于同一子网内）。

配置页面的“Modbus 服务器”部分允许调节与 Modbus TCP 相关的以下驱动器通讯参数：

- 端口编号（比如，502 是 Modbus TCP 使用的标准端口）
- 字节顺序（比如，为配合 AC500 PLC 使用，选择大端顺序）
- 字顺序（比如，为配合 AC500 PLC 使用，选择大端顺序）



配置通用驱动器接口（GDI）Mint 程序

只需要少量修改，即可使用 GDI Mint 程序。

在主程序开始部分（程序头之后）是一组与应用相关的常量 ...

```
' Application specific (edit as required)...
Const _bRemoteEnable As Integer    = _true   'can PLC control enable?
Const _bUseWatchdog As Integer     = _true
Const _nWatchdogTime As Integer    = 2000    'ms
Const _fScale As Float             = 131072    'counts per user unit
Const _nHomeType As Integer        = _hmNEGATIVE_SWITCH 'modify to suit home type
Const _nHomeInput As Integer       = 1       '
Const _nFwdLimit As Integer        = -1      'digital input
Const _nRevLimit As Integer        = -1      'number
Const _nLimitMode As Integer       = _emCRASH_STOP_DISABLE
Const _nStopInput As Integer       = -1      '
Const _nStopMode As Integer        = _smDECEL
#If _platform = _nMotiFlexE180 Then
  Const _nInputLevel As Integer    = 2#11111111    '0=Active low, 1=Active high
#Else
  Const _nInputLevel As Integer    = 2#1111
#End If
Const _nEncoderWrap As Integer     = 0       'Set to counts in 1 cycle for modulo axis
Const _fFolErrorFatal As Float     = 1       'User units - edit to suit application
Const _nMasterChannel As Integer   = 2       '1 = Fast inputs, 2 = Secondary encoder input
Const _nMasterEncMode As Integer   = 0       'Set to 0 or 1 as required for encoder direction
Const _nFollowMode As Integer      = _fmNO_RAMP  'Edit to use required Mint follow mode _fmXXXX
Const _nMotorDirection As Integer = 0       'Set to 0 or 1 as required for correct motor direction
```

常量	功能
_bRemoteEnable	用户可决定 PLC 是否对驱动器的使能状态有控制能力
_bUseWatchdog	用户可决定驱动器是否使用看门狗机制来检测 PLC 通讯的丢失（如果设置为_true，则驱动器会在通讯丢失时紧急停车和停用）
_nWatchdogTime	用户可设置合理的超时值（以 ms 为单位）。PLC 需要在本时间段内（默认情况下，示例 PLC 程序将每 500ms 切换这些寄存器中的一位-该操作可定制，在后文有说明）反复向 Modbus 寄存器写入 0/1

	（Netinteger0）
_fScale	用户可设置比例因数，代表一个用户行程单位中由包含多少个电机编码器脉冲。默认值为 131072，代表配备 Smartabs 反馈的 ESM 伺服电机的用户单位“转”
_nHomeType	用户可定义轴执行的寻零序列的类型（参考 Mint 帮助文件中的 HOME 关键字，以了解完整的可用寻零类型列表及其相关的 Mint 常量值）
_nHomeInput	允许用户指定把驱动器的哪一个本地数字输入（0、1 或 2）用作寻零传感器输入。如果使用的寻零类型不需要输入（比如，_hmPOSITIVE_INDEX），则把这个值设置为-1
_nFwdLimit / _nRevLimit	允许用户指定把驱动器的哪些本地数字输入（0、1 或 2）用作方向限位输入。如果应用不需要行程限位，则把这些常量设置为-1
_nLimitMode	允许用户指定驱动器对限位输入被激活时的响应方式。默认情况下，驱动器将紧急停车并停用（参考 Minit 帮助文件中的 LIMITMODE 关键词，以了解其它选项及其相关的 Mint 常量值）
_nStopInput	允许用户指定把驱动器的哪些本地数字输入（0、1 或 2）用作 Mint 停车输入（Mint 停车输入激活可以触发 Mint 应用程序的停车事件）。如果没有对停车输入的要求，则把该常量设置为-1
_nStopMode	允许用户指定驱动器对停车输入激活的响应方式。默认情况下，驱动器将按照当前定义的减速度减速到静止（参考 Mint 帮助文件中的 STOPMODE 关键词，以了解其它选项及其相关的 Mint 常量值）
_nInputLevel	允许用户指定数字量输入是高电平有效还是低电平有效。默认值为二进制值（输入 0 是最低有效位），把所有可用的输入设置为高电平有效。注意根据不同型号的驱动器，使用不同数量的数字量输入设置值。
_nEncoderWrap	如果应用使用旋转轴（比如，转盘要求在 0 到 360 度的范围内），则本常量应该被设置为一个完整的轴循环内的编码器脉冲数。对非旋转轴，把本常量设置零。
_fFolErrorFatal	设置跟随误差限值（给定和测量位置之间的差值）。这个值一般在精细参数整定之后，用户能够明确运动中正常的跟随误差值。然后把该设定值设置为略高于这个“正常”跟随误差值即可
_nMasterChannel	如果应用要求与主编码器给定值有关（如 FOLLOW），则它允许用户把某一个编码器通道设置为主轴给定值。
_nMasterEncMode	允许用户设置主编码器通道的计数方向（按照要求设置为 0 或 1）
_nFollowMode	允许用户设置所需的 Mint FOLLOWMODE（比如，_fmNO_RAMP，_fmRAMP）－参考 Mint 帮助文件了解完整的可用跟随模式设定。
_nMotorDirection	允许用户配置在正向运动命令下电机的旋转方向（按照要求设置为 0 或 1）。

Mint GDI 详细说明

在大多数应用中，用户不需要了解有关 Mint 应用的所有知识（除编辑上述与应用相关的数据外）。但是，为了完整性，以下章节对 Mint GDI 进行了完整的说明。

数据接口

Modbus 寄存器	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	控制类型	NETINTEGER (1)	32 位整数
4/5	值	NETFLOAT(2)	32 位 IEEE 浮点数
6/7	速度	NETFLOAT(3)	32 位 IEEE 浮点数
8/9	加速度	NETFLOAT (4)	32 位 IEEE 浮点数

10/11	减速度	NETFLOAT (5)	32 位 IEEE 浮点数
12/13	加加速度	NETFLOAT (6)	32 位 IEEE 浮点数
14/15	减减速度	NETFLOAT (7)	32 位 IEEE 浮点数
16/17	锁存偏移	NETFLOAT(8)	32 位 IEEE 浮点数
Modbus 寄存器	功能	驱动器内部地址	驱动器数据类型
200/201	状态字	NETINTEGER (100)	32 位整数
202/203	测量位置	NETFLOAT(101)	32 位 IEEE 浮点数
204/205	测量速度	NETFLOAT(102)	32 位 IEEE 浮点数
206/207	跟随误差	NETFLOAT(103)	32 位 IEEE 浮点数
208/209	轴模式	NETINTEGER (104)	32 位整数
210/211	均方根电流	NETFLOAT(105)	32 位 IEEE 浮点数
212/213	错误代码	NETINTEGER (106)	32 位整数

在以下章节中，我们将仔细介绍其中的每项功能 ...

控制字：Modbus 寄存器 0/1，NETINTEGER(0)

PLC 使用控制字中的位来执行驱动器上的指定操作。PLC 在任何时候向控制字写入新值时，Mint 程序自动调用事件 NETDATA0。

位	功能	状态 0	状态 1
0	远程驱动器使能控制	停用驱动器（如果 Mint 程序常量 _nRemoteEnable 被设置为_true）	使能驱动器（如果 Mint 程序常量 _nRemoteEnable 被设置为_true）
1	使能操作*	禁止运动	允许运动
2	锁存控制	停用位置锁存	使能位置锁存
3	正向硬件限位控制	允许正向硬件限位功能（取决于 Mint 常量 _nFwdLimit 的设置）	取消正向硬件限位功能
4	反向硬件限位控制	允许反向硬件限位功能（取决于 Mint 常量 _nRevLimit 的设置）	取消反向硬件限位功能
5	旋转轴	轴没有旋转位置。绝对定位使用整个轴位置范围	轴位置在定义的旋转轴位置范围内。绝对定位使用最短路径
6	故障复位控制	未使用	上升沿进入状态 1，驱动器故障被复位
7	触发命令	未使用	上升沿进入状态 1，驱动器执行通过 PLC 给定的命令*
8	看门狗	看门狗关	看门狗开
9	忽略跟随误差	跟随误差检测被使能	驱动器忽略跟随误差
10-31	备用	未使用	未使用

注意，触发位激活在控制类型字中写入的运动类型。控制字中的所有其它位持续有效，不需要触发。

Modulo axis 命令位只针对 MOVEA 和 INCA 运动命令使用。如果使用旋转轴，应该编辑 Mint 程序来设置_nEncoderWrap 常量，使其等于一个轴循环周期内编码器总脉冲数。

如果在驱动器上使能了看门狗监控，则应该以足够的频率切换 Watchdog 位，以防止驱动器的看门狗超时故障（一般是在半个看门狗超时值内切换该位）。

* 某些命令不需要执行运动，即使在 Enable Operation 位为零时也可以发送（比如，Set Position 和 Cancel 命令）。

控制类型：Modbus 寄存器 2/3，NETINTEGER(1)

PLC 使用这些寄存器在驱动器上给定控制类型命令。Mint 示例程序提供了多种预先定义的控制类型。可通过在 Mint 程序中添加命令枚举列表和在 Mint“命令触发器”任务中添加额外的代码来轻松的扩展控制类型。

命令编号	功能	Mint 关键字
0	寻零	HOME
1	相对定位	MOVER
2	绝对定位	MOVEA
3	以恒速运行	JOG

4	相对增量定位运动	INCR
5	绝对增量定位运动	INCA
6	设置位置	POS
7	跟随	FOLLOW
8	速度给定值	VELREF
9	转矩给定值	TORQUEREF
10	取消运动	CANCEL
11	受控停车	STOP
12	查找终点停车	不适用（使用定义的命令序列）

PLC 应该在控制类型寄存器中预先写入适当的命令编号（以及其它后文描述的相关设定值），然后通过控制字第 7 位的上升（0->1）沿来触发本命令。

值：Modbus 寄存器 4/5，NETFLOAT(2)

PLC 使用这些寄存器来给定驱动器上要执行的控制类型命令相关的信息。下表详细说明了该值是如何与具体的控制类型命令相关联的。

命令编号	功能	值的用途
0	寻零	寻零回退速度比率 [1]
1	相对定位	相对定位距离
2	绝对定位	绝对定位目标
3	以恒速运行（使能位置环）	未使用
4	相对增量定位运动	相对增量定位运动距离 [2]
5	绝对增量定位运动	绝对增量定位运动目标 [2]
6	设置位置	新的位置值
7	跟随	设置齿轮比
8	速度给定值	未使用
9	转矩给定值	设置转矩值（%）
10	取消运动	未使用
11	受控停车	未使用
12	查找终点停车	转矩限值（驱动器额定电流的百分比） [3]

PLC 应该在值寄存器中预先写入适当的数据（以及其它前文和后文所描述的命令相关参数的寄存器），然后通过控制字第 7 位的上升（0->1）沿来触发本命令。

1 寻零回退速度比率是为 Mint HOMEBACKOFF 关键字设置的一个值（详见 Mint 帮助文件）。这个字只是设定一个换算比例。所有其它值均按比例缩放（即按照驱动器上的 Mint SCALEFACTOR 设置使用单位）。

2 如果在控制字内设置了锁存控制位，可按照“锁存偏移”寄存器中存储的值修改增量定位运动的目标位置（在下方的锁存偏移一节中有详细说明）。

3 驱动器中的转矩限值与 CURRENTLIMIT 设置是等效的。因此，通常是设置转矩限值来限制电流（比如，如果把 CURRENTLIMIT 设置为驱动器额定电流的 60%，在查找终点停车时通常设置一个低于 60%的转矩限值）

速度：Modbus 寄存器 6/7，NETFLOAT(3)

PLC 使用这些寄存器在驱动器上写入运行速度。下表详细说明了速度值是如何与各具体的命令相关联的。

命令编号	功能	速度的用途
0	寻零	设置寻零速度
1	相对定位	设置运动的旋转速度
2	绝对定位	设置运动的旋转速度
3	以恒速运行	点动速度要求
4	相对增量定位运动	设置运动的旋转速度
5	绝对增量定位运动	设置运动的旋转速度
6	设置位置	未使用
7	跟随	未使用
8	速度给定	设置速度给定值（用户单位/秒）
9	转矩给定值	未使用
10	取消运动	未使用
11	受控停车	未使用
12	查找终点停车	设置用于查找终点停车的旋转速度

速度是一个换算量（即，单位与驱动器上的 SCALEFACTOR 设置有关）。PLC 应该在速度寄存器中预先写入适当的数据（和其它包含前文和后文描述的命令相关参数的寄存器），然后通过控制字第 7 位的上升（0->1）沿来触发本命令。

加速度：Modbus 寄存器 8/9，NETFLOAT(4)

PLC 使用这些寄存器在驱动器上写入加速度。下表详细说明了加速度值是如何与各具体的命令相关联的：

命令编号	功能	加速度的用途
0	寻零	设置到达寻零速度的加速度
1	相对定位	设置到达旋转速度的加速度
2	绝对定位	设置到达旋转速度的加速度
3	以恒速运行	设置到达点动速度的加速度
4	相对增量定位运动	设置到达旋转速度的加速度
5	绝对增量定位运动	设置到达旋转速度的加速度
6	设置位置	未使用
7	跟随	未使用
8	速度给定值	设置达到速度要求的加速度
9	转矩给定值	未使用
10	取消运动	未使用
11	受控停车	未使用
12	查找终点停车	设置到达旋转速度的加速度

加速度的单位是用户单位每秒 2。注意，此值必须大于 0.001。值超出该范围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 应该在加速度寄存器中预先写入适当的数据（和其它包含前文和后文描述的命令相关参数的寄存器），然后通过控制字第 7 位的上升（0->1）沿来触发本命令。

减速度：Modbus 寄存器 10/11，NETFLOAT(5)

PLC 使用这些寄存器在驱动器上写入减速度。下表详细说明了减速度值是如何与各具体的命令相关联的：

命令编号	功能	减速度的用途
0	寻零	设置始于寻零速度的减速度
1	相对定位	设置始于旋转速度的减速度
2	绝对定位	设置始于旋转速度的减速度
3	以恒速运行	设置始于点动速度的减速度
4	相对增量定位运动	设置始于旋转速度的减速度
5	绝对增量定位运动	设置始于旋转速度的减速度
6	设置位置	未使用
7	跟随	未使用
8	速度给定	设置始于速度的减速度
9	转矩给定值	未使用
10	取消运动	未使用
11	受控停车	设置始于当前速度的减速度
12	查找终点停车	设置始于旋转速度的减速度

减速度的单位是用户单位每秒 2。注意，这个值必须大于 0.001。值超出该范围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 应该在减速度寄存器中预先写入适当的数据（和其它包含前文和后文描述的命令相关参数的寄存器），然后通过控制字第 7 位的上升（0->1）沿来触发本命令。

加加速度: Modbus 寄存器 12/13，NETFLOAT(6)

PLC 使用这些寄存器在驱动器上写入加加速度（S 曲线）。下表详细说明了加加速度的值是如何与各具体的命令相关联的：

命令编号	功能	加加速度的用途
0	寻零	设置达到寻零加速度的加加速度
1	相对定位	设置达到加速度的加加速度
2	绝对定位	设置达到加速度的加加速度
3	以恒速运行	设置达到加速度的加加速度
4	相对增量定位运动	设置达到加速度的加加速度
5	绝对增量定位运动	设置达到加速度的加加速度
6	设置位置	未使用
7	跟随	未使用
8	速度给定	设置达到加速度的加加速度
9	转矩给定值	未使用
10	取消运动	未使用
11	受控停车	未使用
12	查找终点停车	设置达到加速度的加加速度

加加速度的单位是用户单位每秒 3。注意，这个值必须大于 0.001。值超出该范围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）。

把加加速度或减减速度设置为零将使驱动器使用梯形运动轨迹。如果这些字均包含有效的非零值，驱动器将使用 S 曲线运动轨迹。

PLC 应该在加加速度寄存器中预先写入适当的数据（和其它包含前文和后文描述的命令相关参数的寄存器），然后通过控制字第 7 位的上升（0->1）沿来触发本命令。

减减速度: Modbus 寄存器 14/15，NETFLOAT(7)

PLC 使用这些寄存器在驱动器上写入减减速度（S 曲线）。下表详细说明了减减速度的值是如何与各具体的命令相关联的：

命令编号	功能	减减速度的用途
0	寻零	设置达到寻零减速度的减减速度
1	相对定位	设置达到减速度的减减速度
2	绝对定位	设置达到减速度的减减速度
3	以恒速运行	设置达到减速度的减减速度
4	相对增量定位运动	设置达到减速度的减减速度
5	绝对增量定位运动	设置达到减速度的减减速度
6	设置位置	未使用
7	跟随	未使用
8	速度给定	设置达到减速度的减减速度
9	转矩给定值	未使用
10	取消运动	未使用
11	受控停车	设置达到减速度的减减速度
12	查找终点停车	设置达到减速度的减减速度

减减速度的单位是用户单位每秒 3。注意，这个值必须大于 0.001。值超出该范围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）。

把加加速度或减减速度设置为零将使驱动器使用梯形运动轨迹。如果这些字均包含有效的非零值，驱动器将使用 S 曲线型运动轨迹。

PLC 应该在减减速度寄存器中预先写入适当的数据（和其它包含前文和后文描述的命令相关参数的寄存器），然后通过控制字第 7 位的上升（0->1）沿来触发本命令。

锁存偏移：Modbus 寄存器 16/17，NETFLOAT(8)

PLC 使用这些寄存器写入一个相对于记录的快速中断位置的偏移距离。该偏移距离随后被用作驱动器的新目标位置（本功能一般在转位输送机上使用。在转位输送机中，轴必须始终在经过基准传感器一段固定的距离后停止）。下表详细说明了锁存偏移值是如何与各个具体命令相关联的：

命令编号	功能	锁存偏移的用途
0	寻零	未使用

1	相对定位	未使用
2	绝对定位	未使用
3	以恒速运行	未使用
4	相对增量定位运动	设置距离锁存位置的偏移值
5	绝对增量定位运动	设置距离锁存位置的偏移值
6	设置位置	未使用
7	跟随	未使用
8	速度给定	未使用
9	转矩给定值	未使用
10	取消运动	未使用
11	受控停车	未使用
12	查找终点停车	未使用

如果 PLC 已经设置控制字的第 2 位（以使能“快速”位置锁存），并且驱动器在相对增量定位运动或绝对增量定位运动中收到锁存事件，则驱动器将使用锁存偏移字中包含的数据来为运动设置一个新的目标位置。运动目标变为记录的（快速）位置加上定义的偏移距离。

锁存偏移是一个换算量（即，单位与驱动器上的 SCALEFACTOR 设置有关）。PLC 应该在锁存偏移寄存器中预先写入适当的数据（和其它包含前文描述的命令相关参数的寄存器），然后通过控制字第 7 位的上升（0->1）沿来触发其中一个增量定位运动命令。

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

PLC 使用状态字中的位来确定驱动器的具体状态。PLC 可使用这些位来确定之前发送的命令完成的时间。Mint 程序自动更新该字中的位。

位	功能	状态 0	状态 1
0	使能状态	驱动器被停用	驱动器被使能
1	空闲状态	驱动器未空闲*	驱动器空闲*
2	就位	驱动器不在最后的目标位置的 IDLEPOS**值范围内	驱动器在最后的目标位置的 IDLEPOS**值范围内
3	电机抱闸状态	电机抱闸被释放	电机抱闸被使用
4	寻零状态	驱动器没有完成一次成功的寻零序列	驱动器已经完成一次成功的寻零序列
5	正向限位状态	正向限位输入未激活	正向限位输入激活
6	反向限位状态	反向限位输入未激活	反向限位输入激活
7	故障状态	无故障	驱动器上存在故障（参考错误代码字了解更多信息）
8	停车输入状态	停车输入未激活	停车输入已激活
9	使能就绪状态	驱动器未做好使能准备	驱动器已做好使能准备
10-11	控制模式	这两位一起报告驱动器的当前控制模式 （电流/速度/位置控制）-参考 Mint 帮助文件中的	

		CONTROLMODE	
12	触发状态	控制字第 7 位已清除	控制字第 7 位已设置
13	启动运行状态	控制字第 1 位已清除	控制字第 1 位已设置
14	忽略锁存状态	已经检测到锁存（如果已使用）	已经忽略锁存（如果已使用）
15	故障复位状态	控制字第 6 位已清除	控制字第 6 位已设置
16	寻相状态	寻相未完成	寻相已完成
17-31	备用		

* 参考 Mint 帮助文件中的 IDLE 主题，以了解空闲状态的详情（同时参考 IDLEMODE、IDLEVEL、IDLEPOS 和 IDLETIME）

** 参考 Mint 帮助文件中的 IDLEPOS 主题

*** 只有在使用 Mint WorkBench 来设置与本功能相关的各个 MOTORBRAKExxxxxxx 参数时，才会激活电机抱闸控制。参考所有以 MOTORBRAKE 开头的 Mint 帮助文件主题

测量位置：Modbus 寄存器 202/203，NETFLOAT(101)

PLC 使用这些寄存器读取驱动器的当前位置。这个值是一个换算量（即按照驱动器上的 Mint SCALEFACTOR 设置使用单位）。要使 GDI 控制旋转轴和非旋转轴，这个寄存器实际上反映了 ENCODER(0)的换算值。它允许按照 ENCODERWRAP(0)的设置折叠轴的视位置。要了解详细情况，参考 Mint 帮助文件中的 ENCODER 和 ENCODERWRAP 主题。

测量速度：Modbus 寄存器 204/205，NETFLOAT(102)

PLC 使用这些寄存器读取驱动器的当前速度。这个值是一个换算量（即按照驱动器上的 Mint SCALEFACTOR 设置使用单位）。要了解详细情况，参考 Mint 帮助文件中的 VEL 主题。

跟随误差 Modbus 寄存器 206/207，NETFLOAT(103)

PLC 使用这些寄存器读取驱动器的当前位置误差。这个值是一个换算量（即按照驱动器上的 Mint SCALEFACTOR 设置使用单位）。要了解详细情况，参考 Mint 帮助文件中的 FOLERROR 主题。

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

PLC 使用这些寄存器读取驱动器的当前操作模式。要了解详细情况，参考 Mint 帮助文件中的 AXISMODE 主题。

测量均方根电流：Modbus 寄存器 210/21，NETFLOAT(105)

PLC 使用这些寄存器读取驱动器当前产生的测量均方根电流（以安培为单位）。 要了解详细情况，参考 Mint 帮助文件中的 CURRENTMEAS 主题。

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

PLC 使用这些寄存器读取驱动器的当前故障/错误代码。零值表示当前没有故障（同时，由状态字的第 7 位来指示是否存在故障）。要了解详细情况，参考 Mint 帮助文件中的 ERRCODE 主题。

序列示例：

本应用说明附带的示例程序中有 IEC 61131 PLC 功能块，它利用对所有应用来说都非常简单的 GDI 来自动执行下述序列。这些序列仅用作参考。用户可以设计/实现他们自己的序列控制功能。

使能驱动器

在下载的 Mint 程序中，把 Mint 常量_bRemoteEnable 设置为_True（由 PLC 来控制使能状态）。

确保驱动器上有所有的本地联锁（比如，停车输入、驱动器使能输入、交流电源）。如果驱动器已经做好使能准备，状态字将通过“使能就绪”位显示出来。

设置控制字中的使能位

要显示这种状态，驱动器应该使能，同时应该设置状态字的位 0。

发送一条寻零命令

确保驱动器已经使能（见上文）

把寻零回退速度比率的值写入值字内

把寻零速度的值写入速度字内

把寻零加速度的值写入加速字内

把寻零减速度的值写入减速字内。

如果需要，写入加加速和减减速度的值

在控制类型字中写入 0，以把控制类型设置为寻零

把 1 写入控制字的第 1 位，使运动允许

把 1 写入控制字的第 7 位，以触发运动

程序的运动应该允许。使用状态位（空闲、就位、寻零和故障）来检查执行情况。

在控制字的第 7 位写入 0，为下一条命令做好准备。

发送一条相对定位命令

确保驱动器已经使能（见上文）

把运动距离的值写入值字内

把运动的旋转速度的值写入速度字内

把加速度的值写入加速字内

把减速度的值写入减速字内

如果需要，写入加加速和减减速度的值

在控制类型字中写入 1，以选择相对定位

把 1 写入控制字的第 1 位，使运动允许

把 1 写入控制字的第 7 位，以触发运动

程序的运动应该允许。使用状态位（空闲、就位和故障）来检查执行情况

在控制字的第 7 位写入 0，为下一条命令做好准备

注意，如果需要执行更多运动，不需要为所有或任何不需要修改的运动参数重新发送值-驱动器将保留旋转速度、加速度等的当前值。因此，举例来说，如果 PLC 只需要发送距离不同的新运动命令但使用相同的速度设定，它只需要设置一个新值并在触发命令位上生成一个上升沿。

发送一条点动命令

确保驱动器已经使能（见上文）

确保控制字的第 1 位已经设置，使运动允许

把点动速度的值写入速度字内

按照要求设置加速度/减速度和 S 曲线速率（见上面的例子）。

在控制类型字中写入 3，选择点动。

把控制字的第 7 位设为 1，以启动轴的点动。

可通过写入为零的值、清除使能操作位、发送一条新的命令、激活驱动器上的本地停车输入（如果已经指定）或停用驱动器来停止点动。

发送一条跟随命令

确保驱动器已经使能（见上文）

确保控制字的第 1 位已经设置，使运动允许

把跟随比率的值写入值字内

确保轴是空闲（设置状态字的第 1 位，NETINTEGER(100)）或已经是跟随（轴模式/ NETINTEGER(104)返回的值为 128）

在控制类型字中写入 7，选择跟随

把控制字的第 7 位设置为 1，使轴开始跟随主给定值（或在轴已经在跟随时发送一个新的跟随比率）

可通过写入为零的值并触发另一次跟随、清除使能操作位、发送一条新的命令、激活驱动器上的本地停车输入（如果已经指定）或停用驱动器来停止跟随

发送一个速度给定值

确保驱动器已经使能（见上文）。

确保控制字的第 1 位已经设置，使运动允许。

把速度给定值（用户单位/秒）写入速度字中。

按照上文“发送一条点动命令”中的说明，写入加速/减速/加加速/减减速的值。

在控制类型字中写入 8，以选择速度给定值模式。

把控制字的第 7 位设置为 1，以按照已经编程的速度给定值启动轴的运行。

可通过写入为零的速度给定值、清除使能操作位、发送一条新的命令、激活驱动器上的本地停车输入（如果已经指定）或停用驱动器来停止点动。注意，在发送一条需要不同操作模式的新命令前，驱动器将保持速度给定值模式。如果用户需要驱动器返回位置控制模式，并以可用的全部转矩保持其当前位置，PLC 应该发送一条距离为零相对定位命令（控制类型为 1，值写入 0）。

发送一个转矩给定值

确保驱动器已经使能（见上文）

确保控制字的第 1 位已经设置，使运动允许

把转矩给定值（百分比形式）写入值字中

在控制类型字中写入 9，以选择转矩给定值模式

把控制字的第 7 位设置为 1，以按照已经编程的转矩给定值启动轴的运行

可通过清除使能操作位、发送一条新的命令、激活驱动器上的本地停车输入（如果已经指定）或停用驱动器来停止转矩给定值。注意，在发送一条需要不同操作模式的新命令前，驱动器将保持转矩给定值模式。如果用户需要驱动器返回位置控制模式，并以可用的全部转矩保持其当前位置，PLC 应该发送一条距离为零相对定位命令（控制类型为 1，值写入 0）。

查找终点停车（寻零到转矩限值）

确保驱动器已经使能（见上文）

确保控制字的第 1 位已经设置，使运动允许

把转矩限值（驱动器额定电流的百分比）写入值字中

把用于查找终点停车（用户单位/秒）的速度给定值写入速度字中（这个值的符号决定了行程方向）

按照上文“发送一条点动命令”中的说明，写入加速/减速/加加速/减减速的值

在控制类型字中写入 12，以选择终点停车模式。

把控制字的第 7 位设置为 1，以按照已经编程的速度给定值启动轴的运行

可通过清除使能操作位、发送一条新的命令、激活驱动器上的本地停车输入（如果已经指定）或停用驱动器来停止查找终点停车。注意，在发送一条需要不同操作模式的新命令前，驱动器将保持位置控制模式（把位置保持在确定的终点停车位置）。

PLC 示例程序

本应用说明中包括了控制单台驱动器的一个示例程序（针对通过 Modbus TCP 控制编写）。每个示例都由三个主要元素组成：

1. Mint 通用驱动器接口（GDI）中包括的每个运动控制类型的功能块。功能块在操作上与用于运动控制的 PLCopen 功能块非常相似
2. 数据接口功能块。本代码负责把应用层数据（比如，从功能块使用）按一定路线发送到 Modbus 通信层
3. 读/写 Modbus 数据的功能块

为说明 PLC 代码和 Mint GDI 的使用方法，在每个 PLC 示例程序中包含了两种可视化。

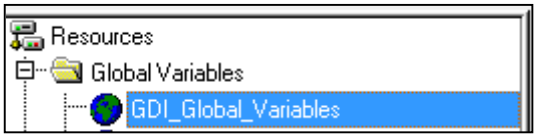
1. 操作每个可用的 GDI 功能块和监控驱动器状态信息的可视化。这种可视化由大量更小的预先编写的可视化组成（同样已经包括）。很容易把这些可视化用于其它的轴，只需要把占位符指定给与新轴对应的功能块
2. 监控 Modbus 数据传输的操作和监控 Modbus 通讯状态的可视化

创建/定义轴

编写的 PLC 示例程序用于通过 Modbus TCP 控制一台驱动器（通过 GDI）。但是，很容易为它添加更多的轴。

对 Modbus TCP，添加额外轴的步骤如下：

1. 在 CoDeSys 应用的“资源”选项卡下双击“GDI_Global_Variables”图标...



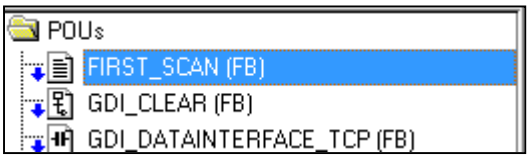
2. 为每条额外的轴另外添加一条 TGDIAxisRef 型轴结构的声明。下文的示例显示了两条额外的轴...

```
（*按照要求添加轴数据类型*）
tAxis0：TGDIAxisRef；
tAxis1：TGDIAxisRef；
tAxis2：TGDIAxisRef；
```

3. 编辑读写数组的声明，与正在使用的轴的数量相匹配。写入数据使用 9 个数据元素，读取数据使用 7 个数据元素，每个轴数据的起始索引值为 10 的倍数。下文的示例显示了编辑它们以匹配三条轴的方式...

```
（* 轴写入数据 @ 0-8, 10-18, 20-28：轴读取数据 @ 0-6, 10-16, 20-26 *）
WriteData:ARRAY [0..28] OF DINT;
ReadData :ARRAY [0..26] OF DINT;
```

4. 双击 CoDeSys 应用的“ POUs ”选项卡下的“ FIRST_SCAN ”POU ...



5. 对需要的每条额外的轴，添加上文第 2 步中声明的轴结构元素的额外初始化代码。下文的示例显示了可用于三条控制轴的代码。注意，轴编号必须是连续的（从零开始），并且 IP 地址必须与驱动器的实际 IP 地址（见其配置参数的设置 ）相匹配 ...

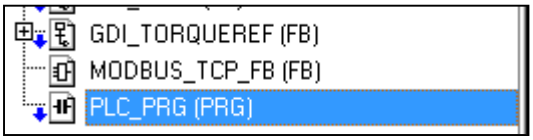
```
tAxis0.AxisName := 'Axis 0';
tAxis0.AxisNo   := 0;
tAxis0.IPAddress := '192.168.0.1';

tAxis1.AxisName := 'Axis 1';
tAxis1.AxisNo   := 1;
tAxis1.IPAddress := '192.168.0.6';

tAxis2.AxisName := 'Axis 2';
tAxis2.AxisNo   := 2;
tAxis2.IPAddress := '192.168.0.14';
```

在示例代码中，使用子网 192.168.0 上的地址来匹配 e180/e190 驱动器的默认值。

6. 现在双击 PLC_PRG POU 图标 ...



为了使 PLC 与额外的驱动器交换控制和状态数据，调用了 GDI_DATAINTERFACE_TCP 功能块的新实例（把相关轴结构作为参数传递给本模块）。

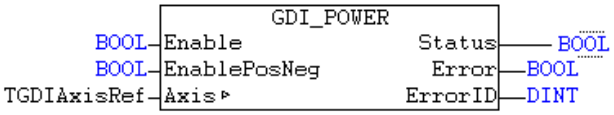
7. 按照要求把应用代码添加到 PLC 程序，以控制额外的轴（比如，包括调用所需要的 GDI 功能块的新实例）。
PLC 可控制的轴的数量与 PLC 可用存储器的容量相关。示例程序是针对 PM554-ETH eco-PLC 编写，它配备了 128kB 的应用代码存储器和 10kB 的变量存储器。
作为参考，单条轴的示例程序使用了约 100kB（78.13%）的 PM554 的可用程序存储器，以及 8.23kB（80.4%）的可用变量存储器。每条额外的轴进一步使用 2.33kB 的变量存储器。
对多轴应用，应该考虑 AC500 PLC（程序和变量存储器的存储容量更大）（例如 PM573-ETH）。

GDI 功能块

以下部分详细说明了 PLC GDI 功能块的用法：

GDI_POWER

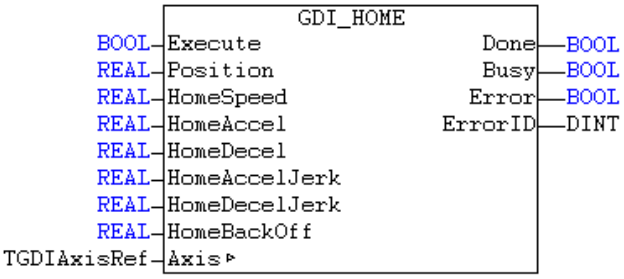
本功能块用于使能/停用轴。使能输入会使能驱动器中的功率单元，而不是功能块本身。



	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Enable	BOOL	在它为真时，PLC 将要求使能轴
EnablePosNeg	BOOL	在它为真时，允许两个方向上的运动。如果为假，运动被阻止（或运动已经在进行时执行一个停车命令）
输出变量		
Status	BOOL	指示轴是否被使能
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

GDI_HOME

本功能块用于把轴零位关联到专用的寻零开关/传感器。寻零的过程取决于 Mint GDI 程序中设置的寻零类型。位置输入用于设置寻零结束后的轴位置。

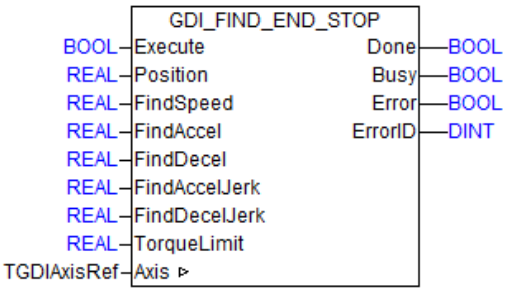


	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	在上升沿启动寻零过程
Position	REAL	在寻零过程成功后设置轴的绝对位置
HomeSpeed	REAL	寻零速度，用户单位/秒
HomeAccel	REAL	寻零加速度，用户单位/秒 ^{2}
HomeDecel	REAL	寻零减速度，用户单位/秒 ^{2}
HomeAccelJerk	REAL	寻零加加速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
HomeDecelJerk	REAL	寻零减减速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
HomeBackOff	REAL	寻零速度与回退速度的比率

输出变量		
Done	BOOL	指示轴已经成功寻零。如果在寻零中 Execute 输入被移除，并且轴完成了寻零过程，则 Done 输出为一个 PLC 扫描周期。如果 Execute 输入仍然为真，则 Done 输出将仍然保持为真（在寻零成功的前提下）。
Busy	BOOL	在寻零序过程正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

GDI_FIND_END_STOP

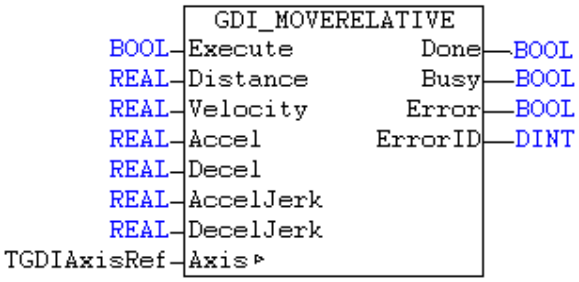
本功能块作为 GDI_HOME 的备选措施，用于在没有专用寻零开关/传感器的情况下把轴的零位关联到行程末端的物理限值上。位置输入用于在寻零成功后设置轴位置。



	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	在上升沿启动寻零过程
Position	REAL	在寻零过程成功后设置轴的绝对位置
FindSpeed	REAL	速度，用户单位/秒（+ve 值向正方向运行，-ve 值向负方向运行）
FindAccel	REAL	寻零加速度，用户单位/秒 ^{2}
FindDecel	REAL	寻零减速度，用户单位/秒 ^{2}
FindAccelJerk	REAL	寻零加加速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
FindDecelJerk	REAL	寻零减减速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
Torque Limit	REAL	转矩（电流）限值被表示为驱动器额定电流的百分比
输出变量		
Done	BOOL	显示轴是否已经成功找到终点停车位置。如果在寻零中 Execute 输入被移除，并且轴完成了查找终点停车过程，则 Done 输出为一个 PLC 扫描周期。如果 Execute 输入持续为真，则 Done 输出将仍然保持为真（在查找终点停车序列成功的前提下）。
Busy	BOOL	在查找终点停车过程正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

GDI_MOVERELATIVE

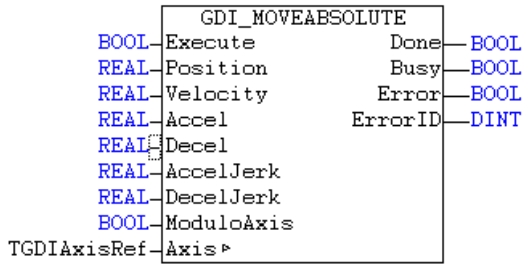
本功能块用于发出命令，用于执行相对当前位置的一段指定距离的定位运动。



	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	在上升沿启动运动
Distance	REAL	运动的相对距离（用户单位）
Velocity	REAL	最大速度的值（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	加加速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
输出变量		
Done	BOOL	显示轴是否已经成功到达目标位置。如果在运动中 Execute 输入被移除，并且相对定位完成，则 Done 输出为一个 PLC 扫描周期。如果 Execute 输入持续为真，则 Done 输出将仍然保持为真（在成功达到目标位置的前提下）。
Busy	BOOL	在相对定位正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

GDI_MOVEABSOLUTE

本功能块用于发出一条命令，用于执行一次到达指定绝对位置的定位运动。本功能可配合旋转轴使用（在这种情况下，将采用最短距离运行路线）。



	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	上升沿启动运动
Position	REAL	运动的目标位置（用户单位）

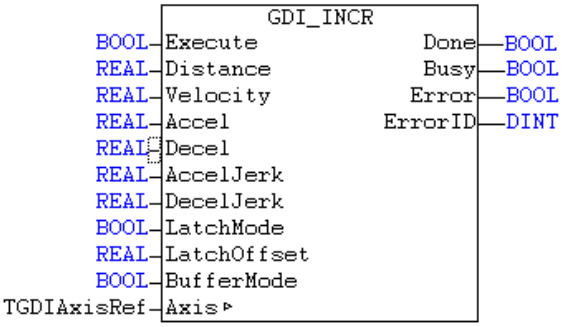
Velocity	REAL	最大速度的值（无需达到），用户单位/秒
Accel	REAL	加速度，用户单位/秒 ^{2}
Decel	REAL	减速度，用户单位/秒 ^{2}
AccelJerk	REAL	加加速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
ModuloAxis	BOOL	定义轴是否为旋转轴（即使用 ENCODERWRAP 来定义一个周期旋转的范围）。使用旋转轴时的绝对定位始终通过最短的路径实现（比如，在 0-360 度的旋转轴上，从 350 度到 20 度的绝对定位将引发一次 30 度的向前运动）
输出变量		
Done	BOOL	显示轴是否已经成功到达目标位置。如果在运动中 Execute 输入被移除，并且绝对定位完成，则 Done 输出为一个 PLC 扫描周期。如果 Execute 输入持续为真，则 Done 输出将仍然保持为真（在成功达到目标位置的前提下）。
Busy	BOOL	在绝对定位正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

GDI_INCR

本功能块用于发出命令，用于执行相对于当前位置的一段指定距离的定位运动。在运动仍在进行时，可通过以下任一种方法修改本功能块的目标位置。

a. 发送另一个 GDI_INCR 或 GDI_INCA 功能（在输入参数 BufferMode 为真的前提下）

b. 把输入参数 Latchmode 设置为真，并为输入参数 LatchOffset 指定值。之后，驱动器上的 Mint 代码将自动修改轴目标位置，以在与定义的快速中断记录的轴位置的距离为 LatchOffset 时停止。可使用轴状态字中的位（btLatchMissed）来指示未能检测到快速中断（示例程序示范了检测到连续三次错过锁存的方式-比如，这种条件之后可用于提醒操作人员存在系统故障）。使用 Latchmode 和 LatchOffset 能简化转位输送机应用的执行。



	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	上升沿启动运动
Distance	REAL	运动的相对距离（用户单位）
Velocity	REAL	最大速度的值（无需达到），用户单位/秒

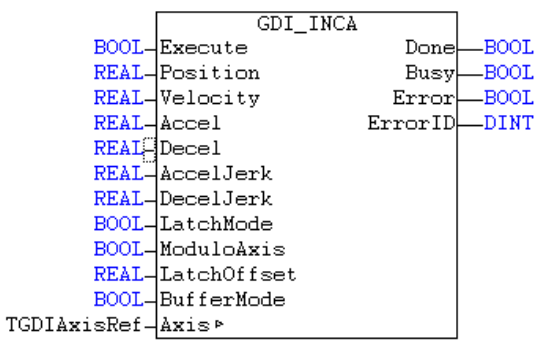
Accel	REAL	加速度，用户单位/秒 ^{2}
Decel	REAL	减速度，用户单位/秒 ^{2}
AccelJerk	REAL	加加速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
Latchmode	BOOL	设置轴是否应该利用配置的快速锁存中断，并在与记录的位置距离为“ LatchOffset ”个用户单位的地方设置新目标位置
LatchOffset	REAL	定义应该修改 GDI_INCR 的目标时（在输入参数 Latchmode 设置为真时）经过记录的快速位置的距离（用户单位）
BufferMode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

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

GDI_INCA

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

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

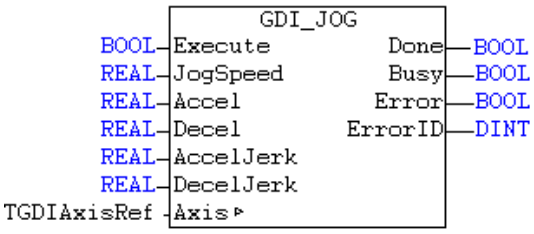


	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	上升沿启动运动
Position	REAL	运动的绝对位置目标（用户单位）
Velocity	REAL	最大速度的值（无需达到），用户单位/秒
Accel	REAL	加速度，用户单位/秒 ^{2}
Decel	REAL	减速度，用户单位/秒 ^{2}
AccelJerk	REAL	加加速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速度，用户单位/秒 ^{3}（对梯形运动，设置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

如果应用需要修改正在进行的绝对定位的 SPEED/ACCEL/DECEL，也可使用 GDI_INCA。使用 GDI_MOVERELATIVE 写入的运动，在运动开始后无法修改 SPEED/ACCEL/DECEL 配置。通过使用 GDI_INCA 并把输入参数缓冲模式设置为真，可通过首先写入 GDI_INCA（及新的 SPEED/ACCEL/DECEL）并把输入参数 Position 设置为零来修改配置文件的参数。

GDI_JOG

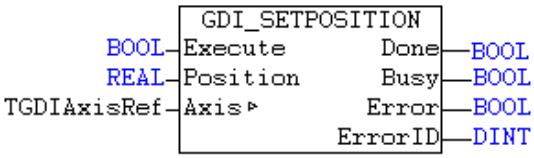
本功能块用于发出一条命令，用于在轴上执行恒速运动（使用驱动器中的位置环控制器）。只要执行输入保持为真，则会一直执行运动。



	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	在上升沿时开始运动，并在输入为真是保持运动。在执行变为假时，运动以斜坡形式按照配置的减速度下降到零速。
JogSpeed	REAL	轴设定的速度值，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	加加速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
DecelJerk	REAL	减减速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
输出变量		
Done	BOOL	在成功发送点动命令后立即设置为真并保持该设置，直到 Execute 变为假或出现轴错误
Busy	BOOL	在功能块正在进行时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

GDI_SETPOSITION

本功能块用于将轴位置（驱动器上的编码器的值和实际位置值）设置为程序设定值。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，这时会设置/发送新的绝对位置（GDI Mint 程序 V2.17 及更新版本）。



	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	上升沿设置新位置
Position	REAL	要设置的轴位置的值（用户单位）
输出变量		
Done	BOOL	在发送命令后立即设置为真（无论其是否成功 - 使用错误输出来确定命令是否成功）。保持为真，直到 Execute 输入被移除。如果 Execute 输入在设置完成位前被移除，则 Done 输出为一个 PLC 扫描周期。

Busy	BOOL	在功能块正在运行时设置为真（在设置 Done 后立即清除）
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

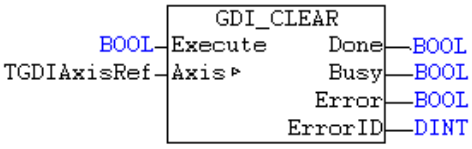
GDI_STOP

本功能块用于按照预设的减速度在轴上执行受控停车。

	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	上升沿启动受控停车
Decel	REAL	减速度，用户单位/秒 ^{2}
DecelJerk	REAL	减减速度，用户单位/秒 ^{3}（对梯形运动，设置为 0）
输出变量		
Done	BOOL	在完成受控停车后轴静止时，或在发送停车命令后发生错误时，设置为真。保持为真，直到 Execute 输入被移除。如果 Execute 输入在设置完成位前被移除，则 Done 输出为一个 PLC 扫描周期。
Busy	BOOL	在停车正在进行时设置为真-在设置 Done 后立即清除
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

GDI_CLEAR

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



	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	上升沿启动紧急停车
输出变量		
Done	BOOL	在完成紧急停车后轴静止时，或在发送紧急停车命令后发生错误时，设置为真。保持为真，直到 Execute 输入被移除。如果 Execute 输入在设置完成位前被移除，则 Done 输出为一个 PLC 扫描周期。
Busy	BOOL	在停车正在进行时设置为真-在设置 Done 后立即清除

Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

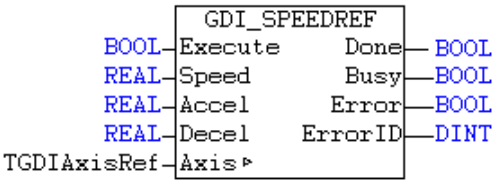
GDI_RESET

本功能块用于复位存在的任何轴错误。

	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	上升沿启动故障复位
输出变量		
Done	BOOL	在轴不再有错误时，设置为真。保持为真，直到 Execute 输入被移除。如果 Execute 输入在设置完成位前被移除，则 Done 输出为一个 PLC 扫描周期。如果无法清除错误，不能设置 Done（使用 Busy 输出来检测尝试故障复位的时间）
Busy	BOOL	在功能块尝试清除任何轴错误时设置为真
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

GDI_SPEEDREF

本功能块用于发送关于轴上速率/速度给定值的命令。在本操作模式下，不在驱动器上使用位置环（因此不会记录跟随误差或做出响应）。轴将保持速度控制模式（见控制模式的状态字位的指示），直到发送另一种控制模式类型的运动（比如，位置控制运动）。要从零速运行（在速度控制模式下）切换到保持位置（在位置控制模式下），可发送比如 GDI_MOVERELATIVE，相对定位距离为零个用户单位。

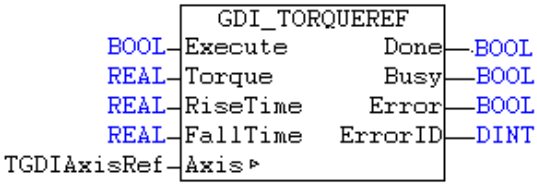


	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输入变量		
Execute	BOOL	在上升沿启动轴，并在输入为真时保持运动。在 Execute 变为假时，运动以斜坡形式按照配置的减速度下降到零速。
Speed	REAL	轴设定的速度值，用户单位/秒。可在 Execute 为真时修改，以改变轴的速率
Accel	REAL	加速度，用户单位/秒 ^{2}
Decel	REAL	减速度，用户单位/秒 ^{2}
输出变量		
Done	BOOL	在发送速度给定值后立即设置为真（无论其是否成功）。Done 输出保持设置，直到 Execute

		变为假。
Busy	BOOL	在功能块正在执行时设置为真（即在 Execute 为真时）
Error	BOOL	如果轴有错误，设为真
ErrorID	DINT	指示轴报告的 Mint 错误代码

GDI_TORQUEREF

本功能块用于发送一条命令，为轴提供一个转矩（电流）给定值。在本操作模式下，不在驱动器上使用位置环（因此不会记录跟随误差或做出响应）。轴将保持转矩控制模式（见控制模式的状态字位的指示），直到发送另一种控制模式类型的运动（比如，位置控制运动）。要从零转矩运行（在转矩控制模式下）切换到保持位置（在位置控制模式下），可发送比如 GDI_MOVERELATIVE，相对定位距离为零个用户单位。



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

GDI_FOLLOW

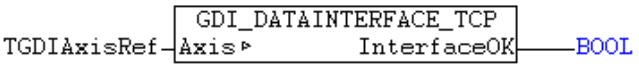
本功能块用于控制轴跟随配置好的主编码器给定值，并以预设的跟随速率启动。

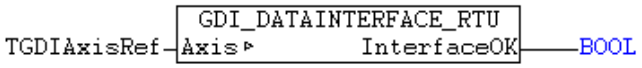
	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）

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

GDI_DATAINTERFACE_TCP / GDI_DATAINTERFACE_RTU

这些功能块用于在 PLC 程序的应用层和通信层之间传送命令/状态数据。对应用中的每根轴，必须配置相关功能块的实例。





	类型	描述
输入输出变量		
Axis	TGDIAxisRef	对轴结构的引用（在使用 Modbus TCP 时，使用轴结构数组的元素）
输出变量		
InterfaceOK	BOOL	如果 Modbus 通信没有运行错误，则为真

使用轴结构

GDI 的多数功能都封装在各种 GDI 功能块中。同时提供的还有 PLC 程序示例。但是，在某些情况下，应用逻辑程序可以直接访问轴结构数据（比如，在示例程序中，逻辑程序通过访问空闲状态和锁存漏失状态来确定是否错过了连续三个锁存）。

TGDIAxisRef 数据类型的声明如下所示：

```
TYPE TGDIAxisRef :
STRUCT
  AxisNo: UINT;
  AxisName: STRING(20);
  NodeAddress: BYTE;
  CommandWord: TCommandWord;
  CommandType: DINT;
  Value: REAL;
  Speed: REAL;
  Accel: REAL;
  Decel: REAL;
  AccelJerk: REAL;
  DecelJerk: REAL;
  LatchOffset: REAL;
  StatusWord: TStatusWord;
  Pos: REAL;
  Vel: REAL;
  FolError: REAL;
  AxisMode: DINT;
  CurrentMeas: REAL;
  ErrorCode: DINT;
END_STRUCT
END_TYPE
```

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

```
TYPE TCommandWord :
STRUCT
  btEnable : BOOL;
  btMotionAllowed : BOOL;
  btPosLatchEnable : BOOL;
  btDisFwdLimit : BOOL;
  btDisRevLimit : BOOL;
  btModulo : BOOL;
  btFaultReset : BOOL;
  btTriggerCmd : BOOL;
  btWatchdog : BOOL;
END_STRUCT
END_TYPE
```

```
TYPE TStatusWord :
STRUCT
  btEnabled: BOOL;
  btIdle: BOOL;
  btInPos: BOOL;
  btBrakeEngaged: BOOL;
  btHomed: BOOL;
  btFwdLimit: BOOL;
  btRevLimit: BOOL;
  btFault: BOOL;
  btStopInput: BOOL;
  btReadyToEnable: BOOL;
  btControlMode0: BOOL;
  btControlMode1: BOOL;
  btTriggerDone: BOOL;
  btPermitted: BOOL;
  btLatchMissed: BOOL;
  btFaultReset: BOOL;
  btPhaseSearchDone : BOOL;
END_STRUCT
END_TYPE
```

因此，PLC 代码可通过这些结构访问其中任意数据。

示例：
读取正向限位输入的状态 ...

tAxis0.StatusWord.btFwdLimit

联系我们

要获得更多信息，请联系你
当地的 ABB 代表，或以下一种方式：

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