MT 系列运动控制产品软件二次开发手册

MT_API.dll 版本 v3.11

更新时间 2015-05-04

声明

由于软件不可避免会有漏洞（Bug）和遗漏，本司也会尽最大努力进行修正和更新，如用户发现有任何问题，请随时联系本司进行修正或者获取最新版本，本司非常感谢用户能帮助我们提高软件质量。

目录

1 SDK 概述

本 SDK 通过提供一个统一的动态链接函数库（Dll）来实现对 MT 系列系列的产品进行控制操作，目前支持的开发语言包括 VC、VB、C#/VB.NET、Delphi，C++ Builder，方便用户进行二次开发，集成到自己的软件系统中。

1.1 SDK 约定

SDK 的函数采用类似 COM 组件的定义方式，方便以后本司提供 COM 组件方式的 SDK 开发包。

SDK 所有的设置参数、执行动作的函数形式为
INT32 MT_Set_xxxx(xxxx)
返回值为 32 位有符号整形型，0 代表函数执行成功，非 0（例如-1）代表执行失败，括号内为设置的一系列参数

SDK 所有的读取参数、读取状态的函数形式为
INT32 MT_Get_xxxx(xxxx)
返回值为 32 位有符号整形数，0 代表函数执行成功，非 0（例如-1）代表执行失败，括号内为以指针或者引用的方式返回的读取值

函数名清楚的表达了本函数的功能

标记为兼容的函数不推荐新软件使用，是为了老用户保持兼容性的函数。

函数所用的单位都以脉冲为单位，对应的速度为 Hz/s,加速度为 Hz/s^2 ,特殊标明的单位除外

另外提供了一组辅助计算函数，帮助用户在使用平移台、旋转台、编码器和光栅尺时进行物理量单位和脉冲单位之间的相互转换。

1.2 运动控制相关的概念

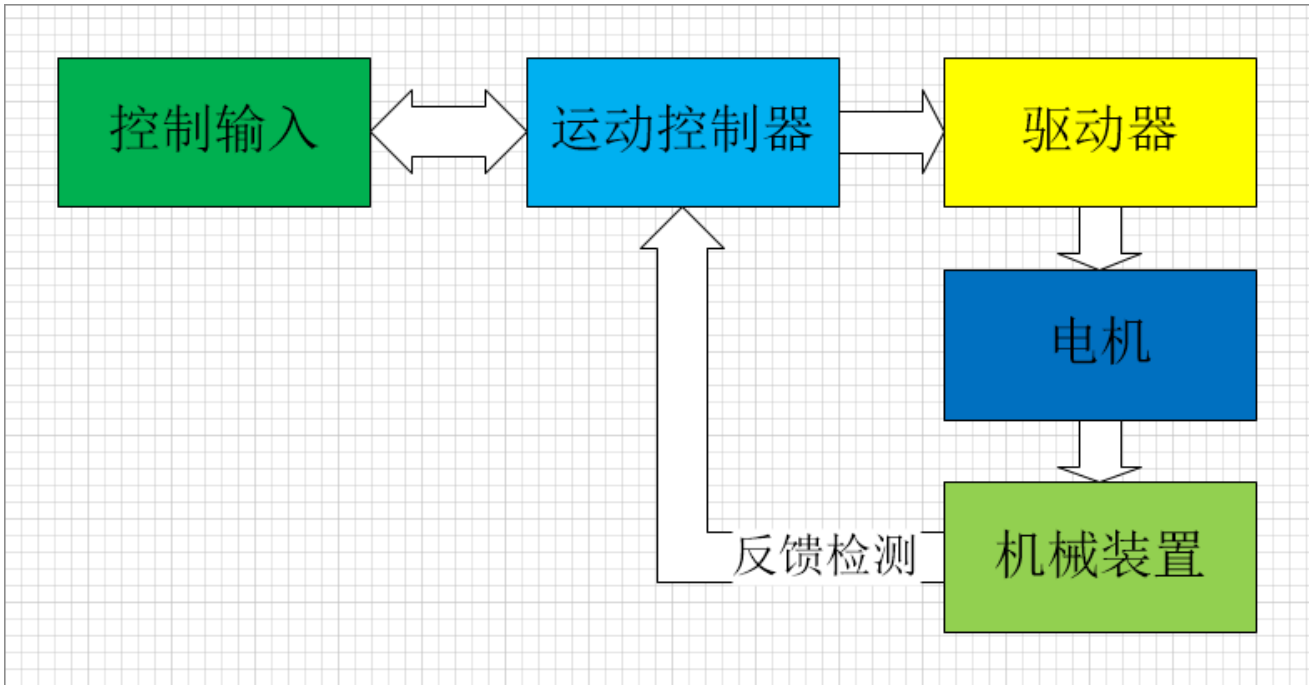
1.2.1 运动的类型

所有的运动都可以简单的抽象为直线运动和圆弧运动，这两种运动组合就可以产生各种运动形式的组合，例如垂直的两个直线运动，如果是等速的，合成的就是一个斜直线运动，如果不等速，就是二维曲线运动。

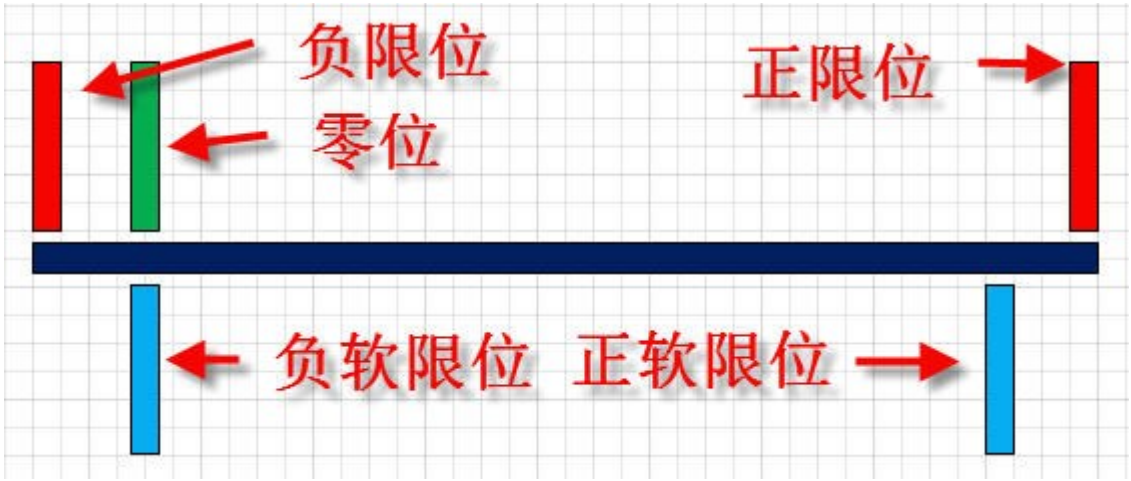
对运动控制卡来说，就是驱动电机旋转，再由机械结构转换成其它的运动形式。控制上的一个控制单位一般叫一步（step），在机械上实际的物理量对应控制上的多少步的比例清楚了，就可以抽象下来只关心电机的运动了。

1.2.2 运动控制系统组成

一般的运动系统由控制输入（电脑、HMI、MCU、开关、摄像头等）+运动控制器（卡）+电机驱动器（功率放大器）+电机+机械装置+反馈检测（限位开关、零位开关、接近开关、编码器、光栅尺）组成。



1.2.3 限位开关、零位开关和软件限位



负限位、正限位和零位一般在硬件上为机械开关、光电开关、接近开关等形式。

负限位和正限位是指物理上的最大位置，通过限位可以通知运动控制卡不超过物理上的限制，可以保护机械装置的安全性，正负不一定都需要有，需要保护的可以加。

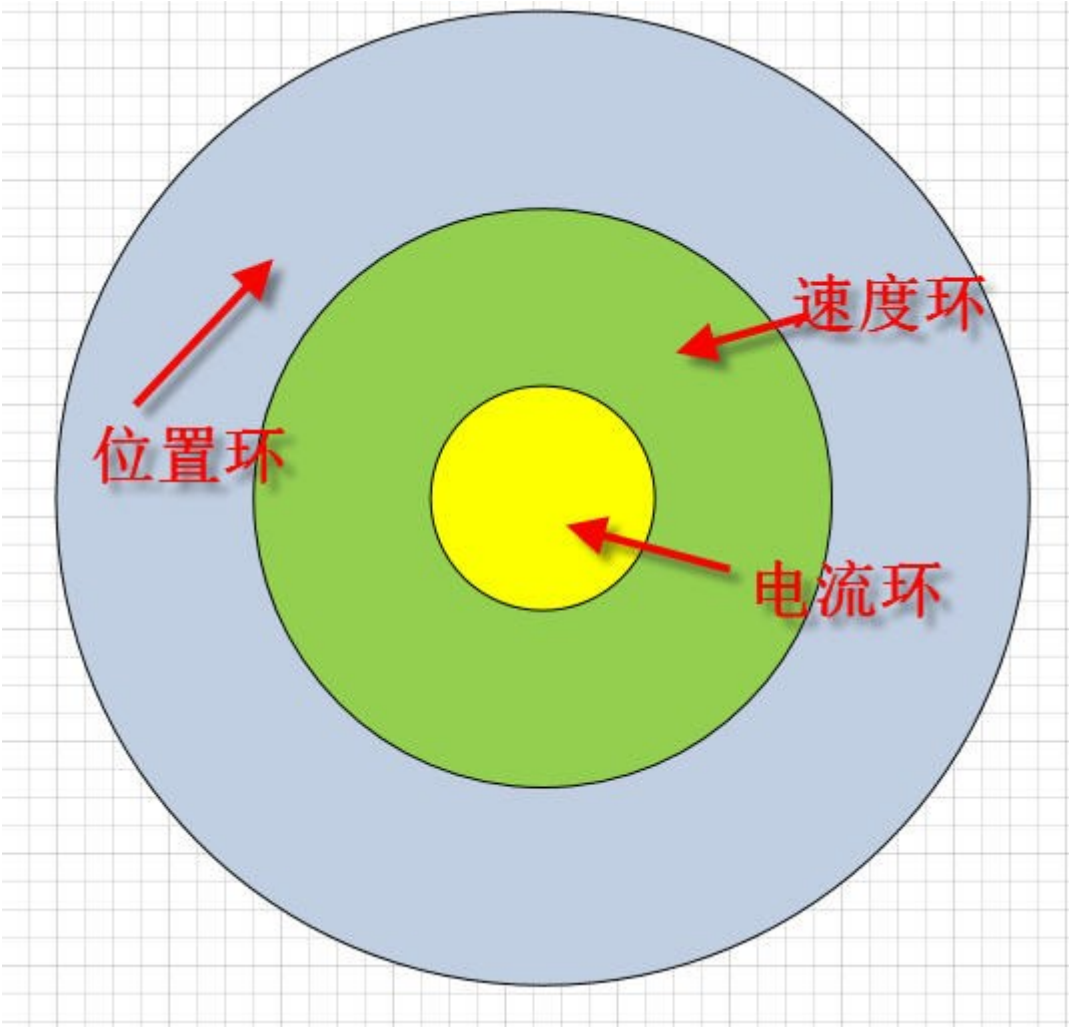
零位为机械上的坐标零点。一般的控制系统断电后不能保持当前的位置信息（绝对式编码器和光栅除外），再次上电后需要用原点来校正一次坐标系，为后续的运动做参考。

软件限位为控制器上通过软件的方式实现的限位，可以在当前坐标线的基础上人工限定一个可以移动的范围，用来做保护。**软件限位一般在零位校正后使用，一般比物理上可以移动的范围小**。

1.2.4 加速和减速

一般的机械装置都有比较大的惯性，为了达到更好的控制效果，都需要在启动的时候有加速过程，在停止的时候有减速过程，一是可以提高控制的精度，二是可以减少惯性对工件的伤害，三是可以保护电控系统的安全。特别是步进系统，可以通过低俗启动提供更大的启动力矩。

1.2.5 电机控制内部逻辑组成



电机控制从内到外分为三个环：电流环、速度环和位置环。不同的环路代表完成不同的功能。电流环一般由驱动器完成，提供电机运动的动力。速度环和位置环为运动控制器实现，实现对目标的速度或者位置控制，完成用户需求。用户只需要根据应用发出指令，有控制器和驱动器协同完成。

1.2.6 步进细分

步进电机一般是每一个整步走一个步距角，一般 2 相 4 线的步进电机的步距角为 1.8°，也就是 200 个脉冲转一圈。驱动器在电气上可以实现对每一个整步进行细分，提供细分后的电流，而不是一步到位到需要的电流，从而可以实现步进电机的细分运动，增加精度和平稳度。

1.2.7 丝杠螺距

电机通过丝杠可以将电机旋转运动转换为丝杠上载物台的直线运动，丝杠的螺距决定了电机旋转一圈转成的直线运动距离。

1.2.8 传动比/减速比

机械上通过减速箱/减速齿轮等方式可以提高旋转的精度，增加力矩。电机旋转多少圈，对应的减速后的机械旋转一圈的数值即为减速比/传动比。

1.2.9 编码器

根据检测原理，编码器可分为光学式、磁式、感应式和电容式。根据其刻度方法及信号输出形式，可分为增量式、绝对式以及混合式三种。MT 系列运动控制产品目前只支持增量式的编码器。增量式编码器可以测量旋转的角度，通过脉冲方式输出。编码器关键的参数为线数，即旋转一圈输出的脉冲数。

1.2.10 光栅尺

光栅尺通过光学的方法可以测量出移动的距离。光栅尺以直线的方式固定时可以测量位移，以圆的方式固定时可以测量角度。光栅尺的关键参数为最小刻度值，即输出一个脉冲对应的位移量。

1.2.11 开环控制

开环控制是指不监测控制对象的状态，不进行反馈调整控制，假设目标按设定的目标进行运动。

1.2.12 闭环控制

闭环控制是指监测控制对象的状态，根据控制对象的反馈进行控制调整，最终让被控对象实现设定的目标。闭环控制需要增加监测的传感器，根据传感器的状态进行反馈控制，一般传感器为编码器、光栅尺等。

1.2.13 相对和绝对

相对是指相对于指令执行的瞬间的状态，比如相对当前的速度再调整的量，相对当前的位置再调整的量；而绝对是指想到达的位置，和当前的速度或者位置等无关。

1.3 运动模式

1.3.1 开环零位模式

在有些系统应用中，系统中需要一个坐标原点，而运动控制器一般把上电时的位置作为当前的坐标原点。利用负限位开关、正限位开关或者零位开关来实现一个坐标原点，后续的定位都是相对这个固定的原点来进行。

零位模式即按用户设定的速度，向正方向或者负方向进行运动，当零位开关有效、限位开关有

效时，停止运动，并置当前的位置为运动控制器的坐标原点。

如果设置零位模式速度为正，则电机正向运动，碰到零位或者正限位后停止，设置当前软件位置为 0；如果设置零位模式速度为负，则电机负向运动，碰到零位或者负限位后停止，设置软件位置为 0。

如果系统需要初始化过程尽可能的短，可以进行一次粗查找零位（有过冲），再进行一次精查找零位动作，如下图所示：

PS：也可以用位置模式或者速度模式进行 0 位查找。位置模式和速度模式在碰到限位后不修改当前的软件计数位置，需要用户自行通过函数设置，而零位模式下碰到零位或者限位后会自动修改当前软件计数位置为 0.

1.3.2 闭环零位模式

闭环零位模式可开环零位模式类似，不过判断零位停止的信号由传感器提供，传感器包括光栅尺和编码器。

对于光栅尺，一般是通过查找光栅尺零位信号的方式进行归零，碰到光栅尺零位或者限位后，会置光栅尺的读数为 0。

对于编码器，一般都有传动机构，可以和开环的零位模式一样，通过零位开关和限位开关来实现零位查找，只不过置编码器的读数为 0。

对于编码器和光栅尺，一般不需要进行粗查和精查，即使有过冲，也是正确的光栅尺或者编码器计数。

1.3.3 速度模式

指定轴工作在指定转速模式下，在这个模式下可以随时调整速度。在加速和减速的过程中会按指定的加速度进行加减速

速度为正的情况，电机向正方向旋转；

速度为负的情况下，向负的方向旋转；

有换向的情况，先减速再加速到指定的反向速度。

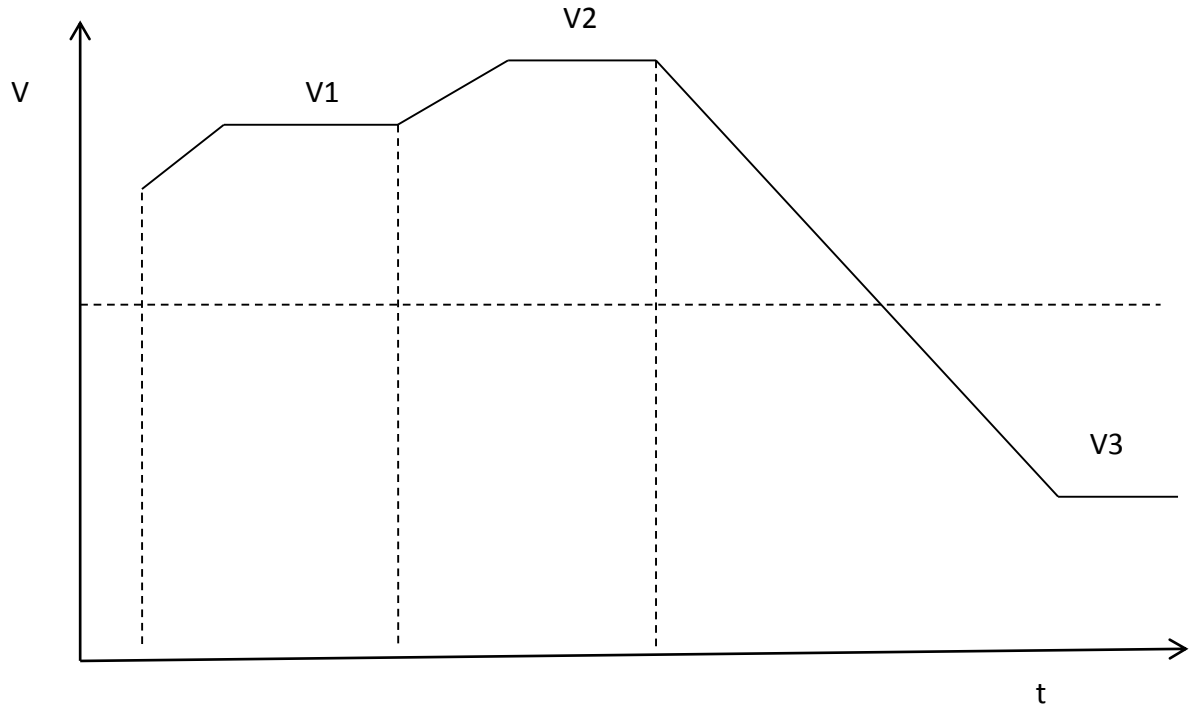
在正向运动时，如果正向限位有效，则电机立即停止动作

在负向运动时，如果负向限位有效，则电机立即停止动作

在速度模式下，位置模式相关的指令无效。

速度模式下最小速度为 +-50Hz/S

以下示意图为三次设置新的速度的工作工程，用户可以随时设置新的运行速度，由控制器自动进行加减速，如果是反向，则自动进行减速再加速的过程。V1，V2，V3 为新的目标速度



1.3.4 开环位置模式

指定轴按指定的步数进行运动，达到指定的位置后停止。在运动过程中会有加减速过程，如果有匀速过程，则按设定的最大速度运动。

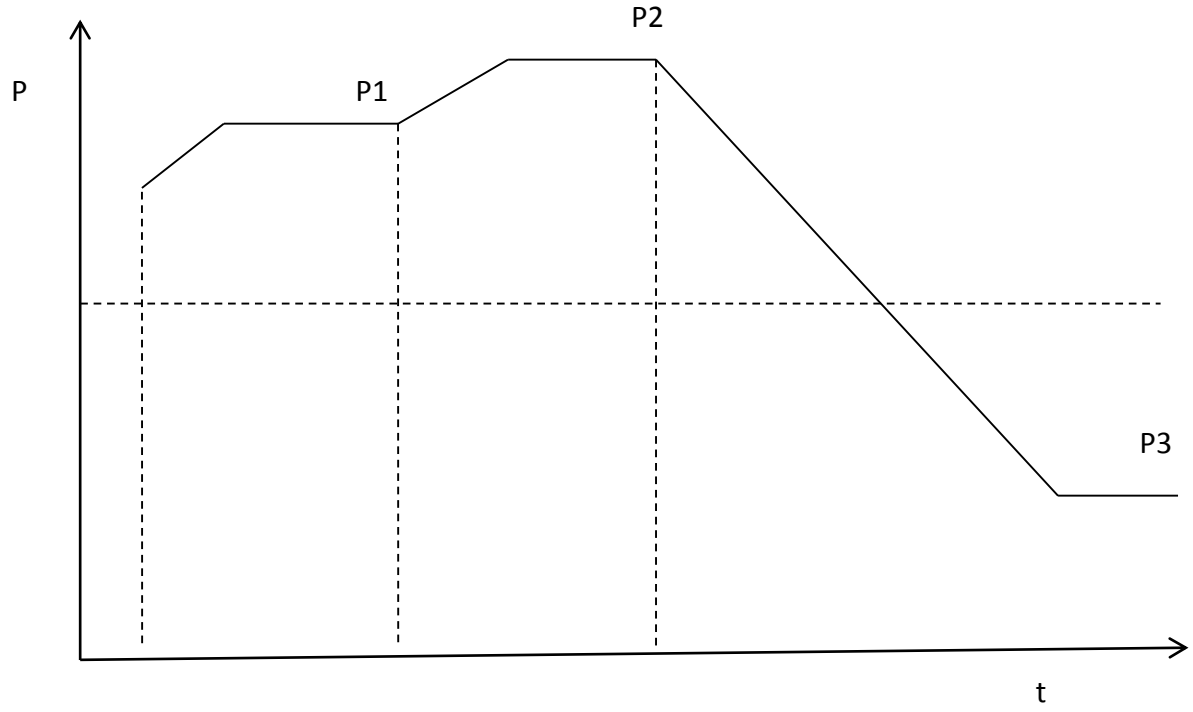
如果指定的位置在当前位置的正向，则电机正向运动；**如果当前正在负向运动，则负向先减速再加速到正向。**

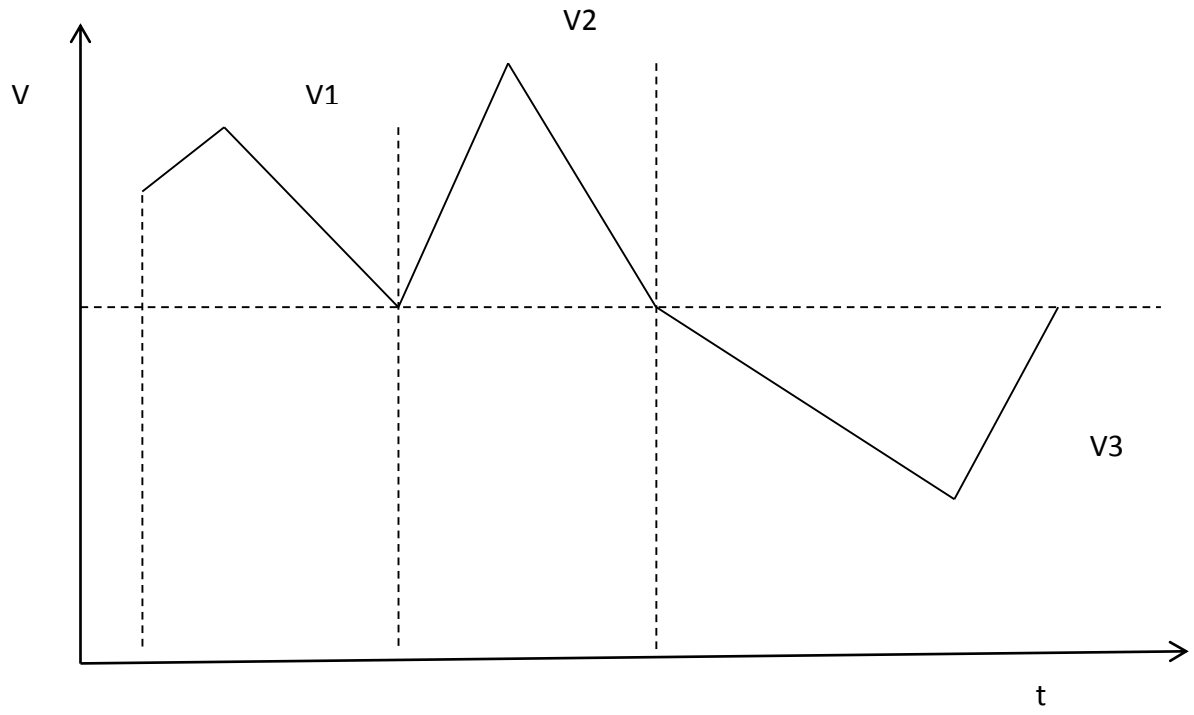
如果指定的位置在当前位置的负向，则电机负向运动；**如果当前正在正向运动，则正向先减速再加速到负向。**

如果指定的位置和当前的位置一样，则电机不动

在正向运动时，如果正向限位有效，则电机立即停止动作

在负向运动时，如果负向限位有效，则电机立即停止动作
下图为定位运动时速度大小和方向和位置目标的示意图：位置不变时，表示速度为 0，停止运动了，P1、P2 和 P3 为设置的目标位置





1.3.5 闭环位置模式

闭环位置模式和开环位置模式类似，只不过位置的判断是以传感器计数为准，不是以脉冲数为准。

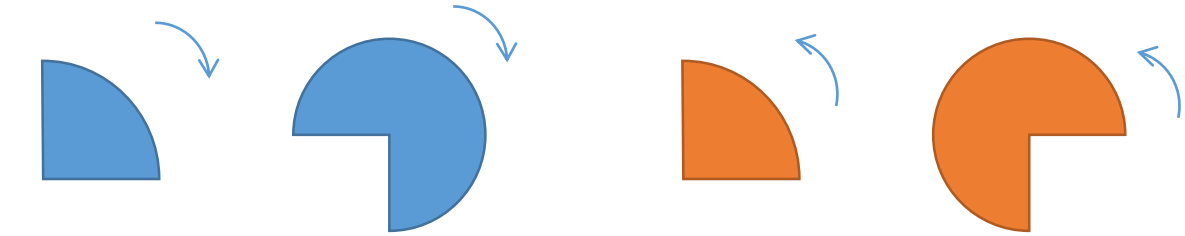
1.3.6 联动插补模式

联动插补是指多个轴同时启动，同时停止的运动，适合需要多个轴有严格时间要求的场合。操作模式和位置模式类似，只是需要多多个指定的轴同时进行指定目标位置，运动控制器自动判别运动方向；设置的速度为最快的轴的运动速度，最快的轴由运动控制器自动判别；任意一个轴碰到限位后，会立即中止本次插补运动；参与插补的轴可以任意组合，至少 2 个，最多支持运动控制器的所有的轴同时插补。

注意：联动插补模式和位置模式相比，在运动的过程中不能重新设置新的系列目标值，只能停止或者等待本次插补运动完毕后进行下一个插补。

1.3.7 圆弧插补模式

圆弧插补模式是指任意两个轴进行一个符合圆弧的插补，可以任意指定运动控制器的 2 个轴。圆弧插补模式需要指定目标坐标点和插补半径；任意一轴碰到限位后会中止本次插补；同样的目标位置和半径，圆弧插补分为顺时针劣弧、顺时针优弧、逆时针劣弧和逆时针优弧 4 种情况。



顺时针劣弧　　顺时针优弧　　逆时针劣弧　　逆时针优弧

注意：圆弧插补模式，在运动的过程中不能重新设置新的系列目标值，只能停止或者等待本次插补运动完毕后进行下一个插补。

1.3.8　流指令模式

流指令模式是指指令可以连续发送，由运动控制器来执行的指令的工作模式，适合于需要响应较快的连续动作。流指令模式相当于其它运动控制器（卡）的 FIFO 指令模式，但又比一般的 FIFO 指令模式强大，一般的其它运动控制器（卡）的 FIFO 指令模式只能支持连续定位点等功能（例如连续的直线插补），而 MT 系列控制器的流指令模式除了支持连续的定位指令（直线插补、圆弧插补、单轴定点等），还支持通用 IO 输出和延时（1ms 实时精度）。

流指令模式下，会在运动控制器开辟一个指令缓冲区，Stream 打头的指令会进入这个缓冲区，由运动控制器来进行连续执行。应用程序只需要查询缓冲区的空间大小，如果有空闲空间，就可以连续发送指令，大大减少了用户应用程序的负担，特别是在需要响应比较快的情况下，或者需要 1ms 精度延时的情况下（**应用程序在 windows 上做不到这么高的精度**）。

一般的应用程序不建议使用这种模式。

每种流指令占用的指令空间不一样长，判断空间剩余建议需要 20 个单位以上。

1.3.9　PLC 模式

PLC 模式是指类似 PLC 的工作模式，在这种模式下，用户可以下载一段程序到运动控制器中，由运动控制器根据自定义的状态自动进行设定的动作，而且在这种状态下还可以继续接收通信口的指令。

独立脱机控制：如果要离开电脑或者液晶触摸屏，可以直接使用 PLC 模式，这种模式下，用户可以通过外接按钮或者开关来启动、停止、暂停程序，可以执行固定的动作。

配合应用程序：可以将固定的动作编辑到控制器内部，通过全局变量进行通信和控制，可以实现动作时固定的，但是参数时可以调的。
PLC 模式需要 MTEditor 软件、MTRun 软件和 MTPLC 软件配合。

1.3.10 软件限位模式

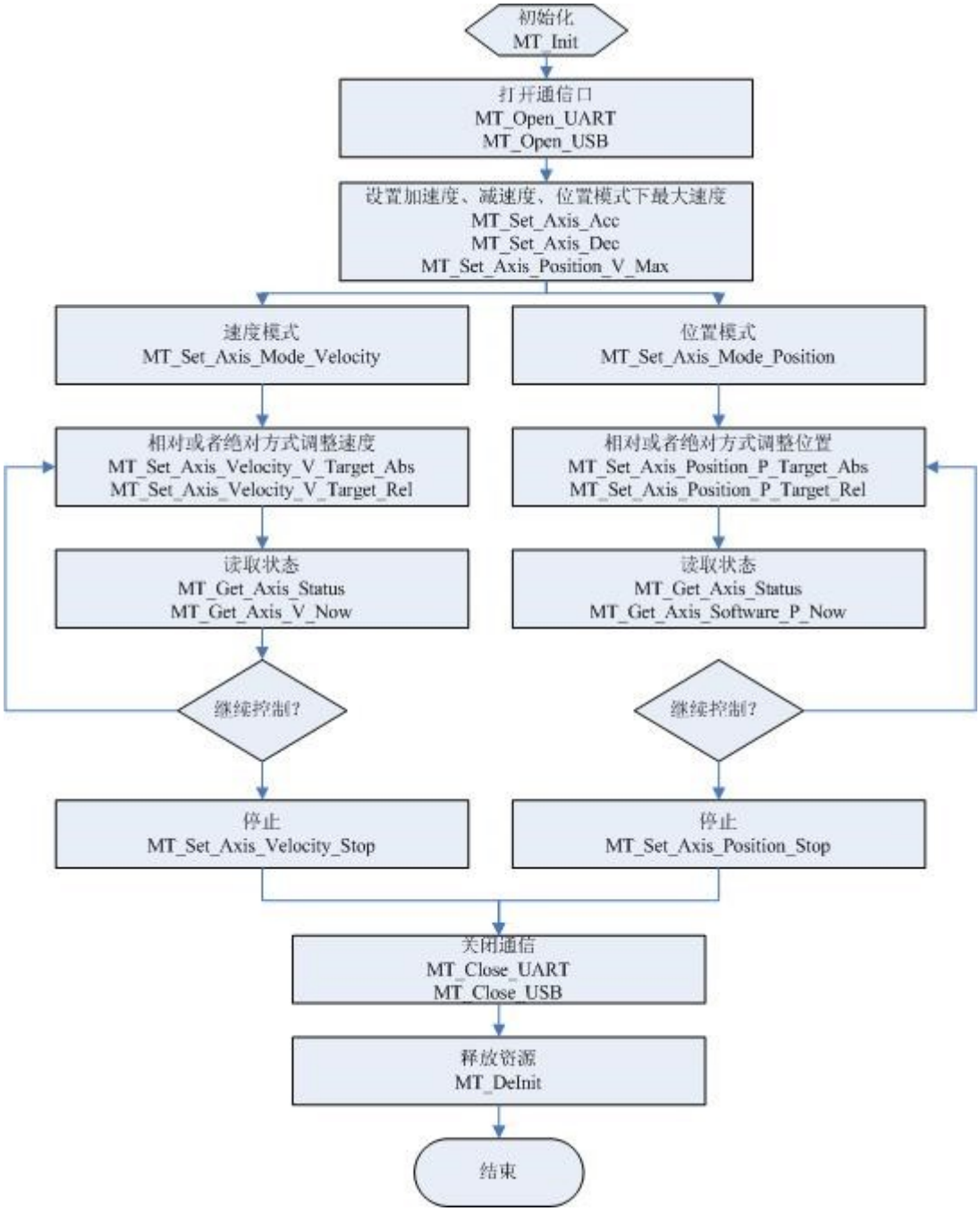
软件限位模式是指每个轴可以设定一个软件限位功能，可以在软件上限定这个轴可以移动的范围，从而实现类似硬件限位的功能。

因为软件限位的功能是指在软件脉冲位置，软件脉冲位置在上电的时候默认为 0，所以一般作为硬件限位或者零位的补充，经过硬件限位的初始化后，再通过软件限位功能来实现保护。

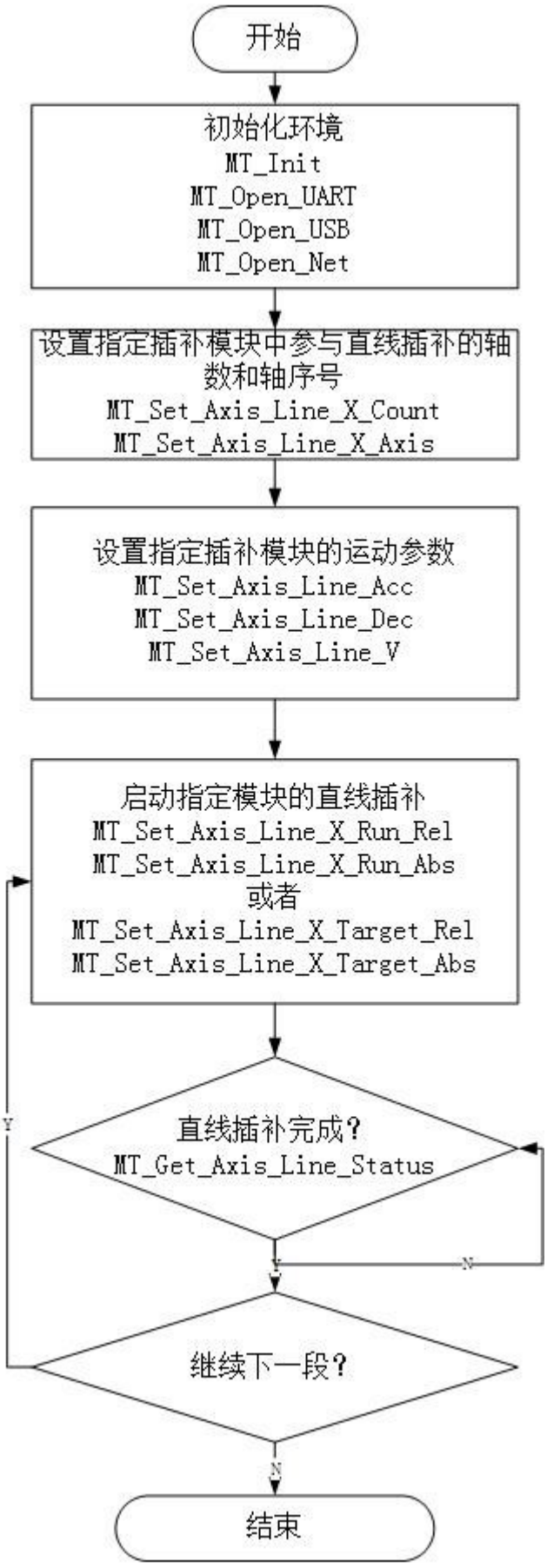
一般情况下推荐使用硬件限位和零位。

1.4 利用 SDK 编程的一般流程图

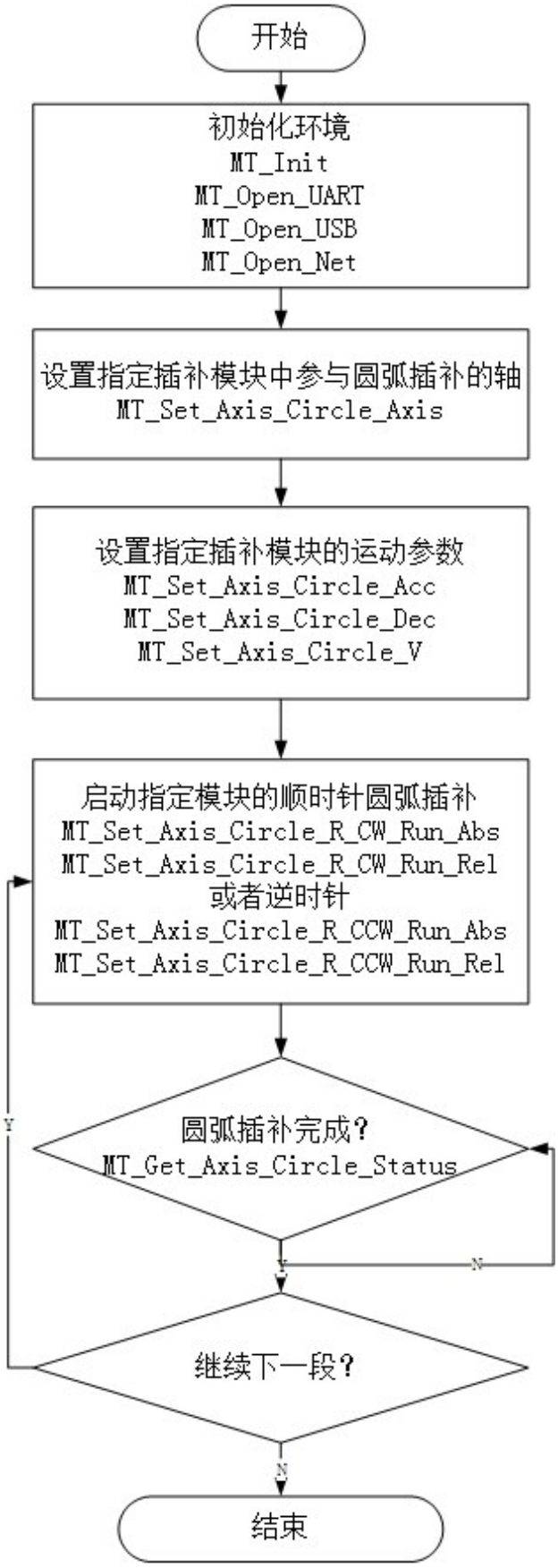
1.4.1 速度模式和位置模式



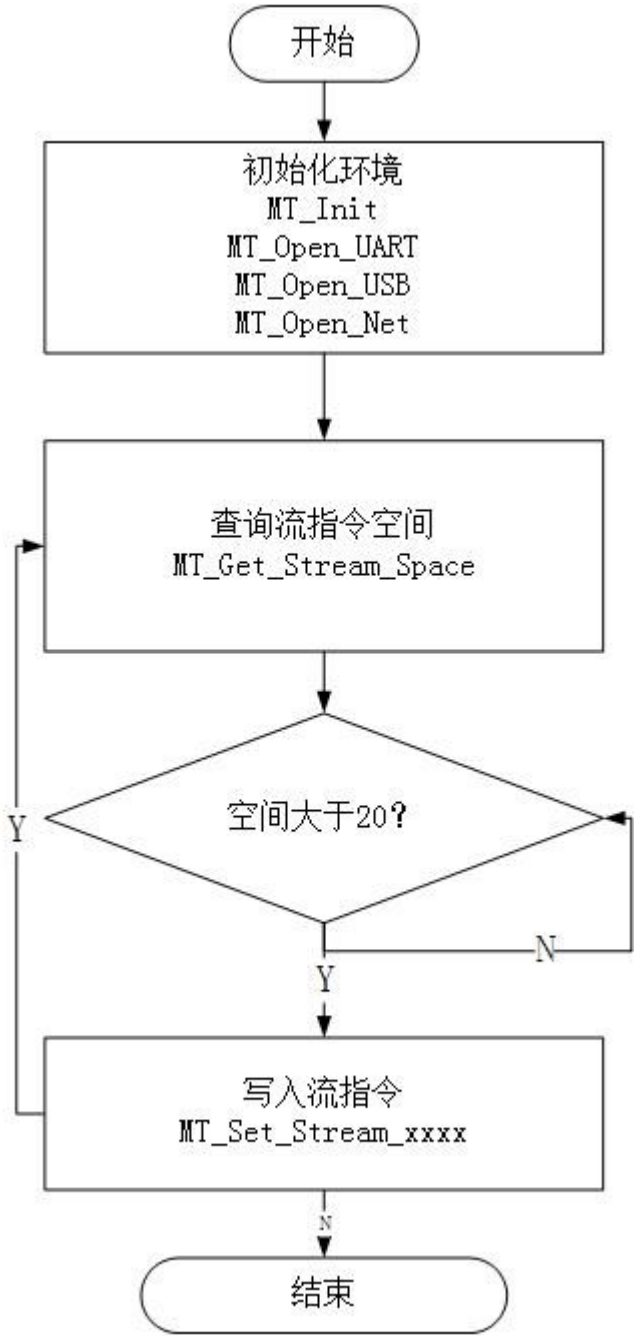
1.4.2 多轴联动插补模式(直线插补)


开始
初始化环境
MT_Init
MT_Open_UART
MT_Open_USB
MT_Open_Net
设置指定插补模块中参与直线插补的轴数和轴序号
MT_Set_Axis_Line_X_Count
MT_Set_Axis_Line_X_Axis
设置指定插补模块的运动参数
MT_Set_Axis_Line_Acc
MT_Set_Axis_Line_Dec
MT_Set_Axis_Line_V
启动指定模块的直线插补
MT_Set_Axis_Line_X_Run_Rel
MT_Set_Axis_Line_X_Run_Abs
或者
MT_Set_Axis_Line_X_Target_Rel
MT_Set_Axis_Line_X_Target_Abs
直线插补完成?
MT_Get_Axis_Line_Status
Y
N
继续下一段?
Y
N
结束

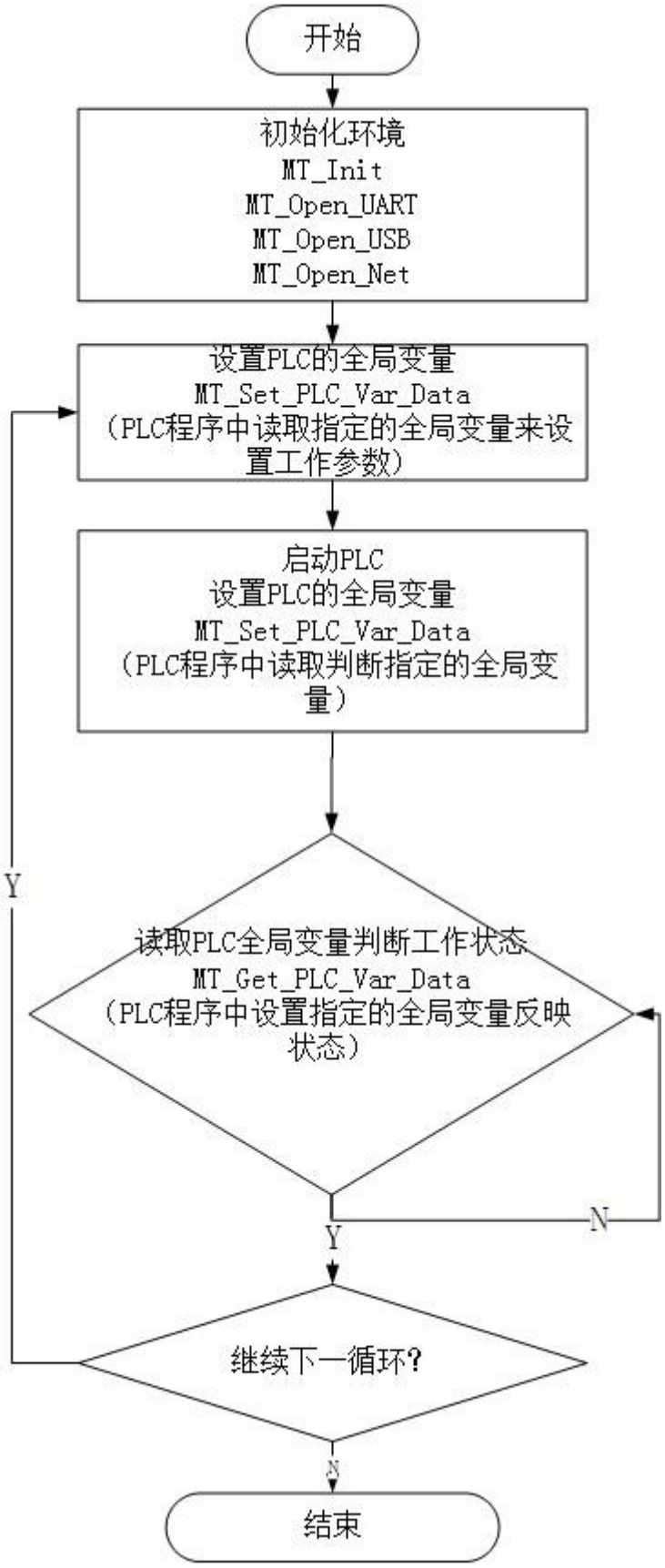
1.4.3 圆弧插补模式


开始
初始化环境
MT_Init
MT_Open_UART
MT_Open_USB
MT_Open_Net
设置指定插补模块中参与圆弧插补的轴
MT_Set_Axis_Circle_Axis
设置指定插补模块的运动参数
MT_Set_Axis_Circle_Acc
MT_Set_Axis_Circle_Dec
MT_Set_Axis_Circle_V
启动指定模块的顺时针圆弧插补
MT_Set_Axis_Circle_R_CW_Run_Abs
MT_Set_Axis_Circle_R_CW_Run_Rel
或者逆时针
MT_Set_Axis_Circle_R_CCW_Run_Abs
MT_Set_Axis_Circle_R_CCW_Run_Rel
圆弧插补完成?
MT_Get_Axis_Circle_Status
Y
N
继续下一段?
Y
N
结束

1.4.4 流指令模式



1.4.5 PLC 模式


开始
初始化环境
MT_Init
MT_Open_UART
MT_Open_USB
MT_Open_Net
设置PLC的全局变量
MT_Set_PLC_Var_Data
（PLC程序中读取指定的全局变量来设置工作参数）
启动PLC
设置PLC的全局变量
MT_Set_PLC_Var_Data
（PLC程序中读取判断指定的全局变量）
读取PLC全局变量判断工作状态
MT_Get_PLC_Var_Data
（PLC程序中设置指定的全局变量反映状态）
Y
N
Y
继续下一循环?
N
结束

2 SDK 的注意事项

2.1 线程安全性

一般提供的版本是线程不安全的，请保证在调用本 SDK 的函数是在一个线程里面，或者通过定时器来实现定时调度功能。
如果有特殊需要，V3.10(含)后的版本可以提供线程安全的，Dll 内部实现了多线程保护功能，请用户注意使用，避免出现死锁，多单线程应用没有区别；单线程模式下，如果非常频繁的调用（例如非常多个定时器发生同时调用），有非常小的概率发生通信冲突（通信必须串行），V3.10(后)的版本可以提供了通信口冲突检测功能，遇到冲突会返回错误，请注意判断返回值，推荐在单线程时使用单个定时器状态机的方式进行调用。

一般情况下提供的 dll 是线程不安全的，但是性能和可靠性好，一般情况下推荐使用。多线程和保护功能的由于 dll 是受控于应用程序，不能主动做动作，用户使用时需要注意，有些情况下多线程环境比较复杂，使用不当会有死锁，调试也不是很方便。

一般推荐用户的使用模式：单定时器状态机调度模式（部分例程有这个功能）；一个独立的线程操作 MT 相关的函数，其它线程不操作。

2.2 SET 类型函数执行有效性

0 只是表示本函数执行成功（通信成功），设置的参数有可能本丢弃或者采用默认值，具体的参数限制请注意不同板卡的不同参数指标，不清楚的情况下可以联系我司或者通过对应的 GET 函数来获取

2.3 GET 类型函数的返回值

只有在函数返回值为 0 的情况下，返回的参数才是有效值，否则为随机值

2.4 函数的适应性

某些函数只在特定的模式下有效，请注意查看相关的函数说明，可能函数执行返回为 0，但是未设置成功。

2.5 光栅尺和编码器的计数

对于光栅尺和编码器，控制器计数时是进行电子 4 倍频的，也就是说，计数值是脉冲的 4 倍，在计算和设置时需要注意。本 SDK 提供的辅助计算函数已经考虑 4 倍的关系，用户自己计算时需要考虑 4 倍频。

3 SDK 函数说明

3.1 初始化

3.1.1 MT_Init

功能描述：创建 Dll 工作需要的资源，在使用其它函数前必须调用本函数，一般放在软件的初始化部分执行

VC	INT32 MT_Init(void)	
VB	Function MT_Init () As Long	
Delphi	function MT_Init():Integer;	
C#	public static extern int MT_Init();	
输入参数	无	
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.1.2 MT_DeInit

功能描述：释放 Dll 工作需要的资源，在软件退出时执行

VC	INT32 MT_DeInit (void)	
VB	Function MT_DeInit () As Long	
Delphi	function MT_DeInit ():Integer;	
C#	public static extern int MT_DeInit ();	
输入参数	无	
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.1.3 MT_Get_Dll_Version

功能描述：读取本 Dll 的版本号，也可以通过 Windows 资源管理器查看 Dll 版本号,一般不使用，做兼容性时的判断依据

VC	INT32 MT_Get_Dll_Version(char** sVer)

VB	Function MT_Get_Dll_Version (ByRef sVer As String) As Long	
Delphi	function MT_Get_Dll_Version(sVer:PPChar):Integer;	
C#	public static extern int MT_Get_Dll_Version(ref String sVer);	
输入参数	无	
输出参数	sVer	版本号字符串
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.2 通信端口

3.2.1 MT_Close_UART

功能描述：关闭串口通信口，在需要切换通信端口、退出软件或者释放占用的串口时调用

VC	INT32 MT_Close_UART (void)	
VB	Function MT_Close_UART () As Long	
Delphi	function MT_Close_UART():Integer;	
C#	public static extern int MT_Close_UART();	
输入参数	无	
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.2.2 MT_Open_UART

功能描述：打开串口通信口，在需要通过串口和板卡通信前调用

VC	INT32 MT_Open_UART (char* sCOM)	
VB	Function MT_Open_UART (ByVal sCOM As String) As Long	
Delphi	function MT_Open_UART(sCOM:PAnsiChar):Integer;	
C#	public static extern int MT_Open_UART(string sCOM);	
输入参数	sCOM	字符串，串口名称，例如 COM1，COM2
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	函数执行成功不代表串口已经能工作，只是完成了相关的通信初始化工作	

3.2.3 MT_Close_USB

功能描述：关闭 USB 通信口，在需要切换通信端口或者退出软件时调用,

VC	INT32 MT_Close_USB (void)	
VB	Function MT_Close_USB () As Long	
Delphi	function MT_Close_USB():Integer;	
C#	public static extern int MT_Close_USB();	
输入参数	无	
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.2.4 MT_Open_USB

功能描述：打开 USB 通信口，在需要通过 USB 和板卡通信前调用

VC	INT32 MT_Open_USB (void)	
VB	Function MT_Open_USB () As Long	
Delphi	function MT_Open_USB():Integer;	
C#	public static extern int MT_Open_USB();	
输入参数	无	
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	函数执行成功不代表 USB 已经能工作，只是完成了相关的通信初始化工作	

3.2.5 MT_Close_Net

功能描述：关闭网络通信口，在需要切换通信端口或者退出软件时调用,

VC	INT32 MT_Close_Net (void)	
VB	Function MT_Close_Net () As Long	
Delphi	function MT_Close_Net():Integer;	
C#	public static extern int MT_Close_Net();	
输入参数	无	
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.2.6 MT_Open_Net

功能描述：打开网络通信口，在需要通过网络和板卡通信前调用

VC	INT32 MT_Open_USB (BYTE IP0,BYTE IP1,BYTE IP2,BYTE IP3,WORD IPPort)	
VB	Function MT_Open_USB (ByVal IP0 As Byte, ByVal IP1 As Byte, ByVal IP2 As Byte, ByVal IP3 As Byte, ByVal IPPort As Integer) As Long	
Delphi	function MT_Open_USB(IP0,IP1,IP2,IP3:Byte;IPPort:Word):Integer;	
C#	public static extern int MT_Open_USB(BYTE IP0,BYTE IP1,BYTE IP2,BYTE IP3,WORD IPPort);	
输入参数	IP0,IP1,IP2,IP3	IP 地址,例如 192.168.0.2
	IPPort	端口地址,例如 8888
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	函数执行成功不代表网络已经能工作，只是完成了相关的通信初始化工作	

3.3 通信握手

3.3.1 MT_Check

功能描述：在通信端口打开以后，检测是否存在 MT 系列运动控制板卡

VC	INT32 MT_Check (void)	
VB	Function MT_Check () As Long	
Delphi	function MT_Check ():Integer;	
C#	public static extern int MT_Check ();	
输入参数	无	
输出参数	无	
函数返回	0	函数执行成功，检测到板卡，可以进行通信
	非 0	函数执行失败
备注		

3.4 硬件信息

3.4.1 MT_Get_Product_Resource

功能描述：检测硬件中拥有的资源类型，包括控制的电机轴数、输入、输出等信息，再配合不同类型资源的读数量的函数，来完成最终具体资源的判断。不同的板卡，所带的资源不一样，

如果已经了解所用板卡的资源，无需调用本函数。考虑到软件兼容性的时候，可以调用本函数。也可通过官方工具来查看

VC	INT32 MT_Get_Product_Resource (INT32* pValue)	
VB	Function MT_Get_Product_Resource (ByRef Value As Long) As Long	
Delphi	function MT_Get_Product_Resource (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Product_Resource (ref Int32 Value);	
输入参数	无	
输出参数	Value	输出硬件中资源类型，由以下资源类型进行或操作后的结果 目前定义的资源 #define RESOURCE_MOTOR 0x00000001 //电机 #define RESOURCE_TTL_IN 0x00000002 //TTL 输入 #define RESOURCE_TTL_OUT 0x00000004 //TTL 输出 #define RESOURCE_OPTIC_IN 0x00000008 //光隔输入 #define RESOURCE_OPTIC_OUT 0x00000010 //光隔输出 #define RESOURCE_AD 0x00000020 //AD 输入 #define RESOURCE_OC 0x00000040 //OC 输出 #define RESOURCE_LINE 0x00000080 //直线插补 #define RESOURCE_CIRCLE 0x00000100 //圆弧插补 #define RESOURCE_PLC 0x00000200 //PLC 功能支持 #define RESOURCE_STREAM 0x00000400 //流模式支持 #define RESOURCE_ENCODER 0x00000800 //编码器光栅尺接口
函数返回	0	函数执行成功，读取到的板卡资源有效
	非 0	函数执行失败，读取到的板卡资源无效
备注		

3.4.2 MT_Get_Product_ID

功能描述：读取板卡的型号。已了解无需调用本函数。也可通过官方工具来查看

VC	INT32 MT_Get_Product_ID (char* sID)	
VB	Function MT_Get_Product_ID (ByRef sID As String) As Long	
Delphi	function MT_Get_Product_ID(sID:PAnsiChar):Integer;	
C#	public static extern int MT_Get_Product_ID (ref String sID);	
输入参数	无	
输出参数	sID	16 长度的字符缓冲区
函数返回	0	函数执行成功，读取到的型号有效
	非 0	函数执行失败，读取到的型号无效
备注	**缓冲区需用户先申请好内存**	

3.4.3 MT_Get_Product_SN

功能描述：读取板卡的序列号。也可通过官方的工具来查看。序列号本司用来确认产品的相关

信息的判据。用户也可以用本唯一序列号完成用户产品的确认等工作。

VC	INT32 MT_Get_Product_SN (char* sSN)	
VB	Function MT_Get_Product_SN (ByRef sSN As String) As Long	
Delphi	function MT_Get_Product_SN(sID:PAnsiChar):Integer;	
C#	public static extern int MT_Get_Product_SN (ref String sSN);	
输入参数	无	
输出参数	sSN	12 长度的字符缓冲区
函数返回	0	函数执行成功，读取到的序列号有效
	非 0	函数执行失败，读取到的序列号无效
备注	缓冲区需用户先申请好内存	

3.4.4 MT_Get_Product_ Version[兼容]

功能描述：读取板卡固件的版本号。也可通过官方的工具来查看。版本号本司用来确认产品的相关信息的判据。

VC	INT32 MT_Get_Product_ Version (INT32* pMajor,INT32* pMinor)	
VB	Function MT_Get_Product_ Version (ByRef Major As Long, ByRef Minor As Long) As Long	
Delphi	function MT_Get_Product_ Version(pMajor:PInteger;pMinor:PInteger):Integer;	
C#	public static extern int MT_Get_Product_ Version (ref Int32 Major,ref Int32 Minor);	
输入参数	无	
输出参数	Major	大版本号
	Minor	小版本号
函数返回	0	函数执行成功，读取到的版本号有效
	非 0	函数执行失败，读取到的版本号无效
备注		

3.4.5 MT_Get_Product_ Version2

功能描述：读取板卡固件的版本号。也可通过官方的工具来查看。版本号本司用来确认产品的相关信息的判据。

VC	INT32 MT_Get_Product_Version2(INT32* pMajor,INT32* pMinor,INT32* pRelease,INT32* pBuild)	
VB	Function MT_Get_Product_ Version2 (ByRef pMajor As Long,ByRef pMinor As Long,ByRef pRelease As Long,ByRef pBuild As Long) As Long	
Delphi	function MT_Get_Product_ Version2(pMajor:PInteger;pMinor:PInteger; pRelease:PInteger;pBuild:PInteger):Integer;	
C#	public static extern int MT_Get_Product_ Version 2(ref Int32 pMajor,ref Int32 pMinor,ref Int32 pRelease,ref Int32 pBuild);	
输入参数	无	
输出参数	pMajor	大版本号

	pMinor	小版本号
	pRelease	发布次数
	pBuild	编译次数
函数返回	0	函数执行成功，读取到的版本号有效
	非 0	函数执行失败，读取到的版本号无效
备注		

3.5 零位模式(开环和闭环)

3.5.1 MT_Set_Axis_Mode_Home[兼容]

功能描述：设置指定的轴为开环零位模式

VC	INT32 MT_Set_Axis_Mode_Home(WORD AObj)	
VB	Function MT_Set_Axis_Mode_Home (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Mode_Home(AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Mode_Home(UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.5.2 MT_Set_Axis_Mode_Home_Home_Switch

功能描述：设置指定的轴为开环零位模式，查找零位以零位开关和限位开关为判断依据。

VC	INT32 MT_Set_Axis_Mode_Home_Home_Switch (WORD AObj)	
VB	Function MT_Set_Axis_Mode_Home_Home_Switch (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Mode_Home_Home_Switch (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Mode_Home_Home_Switch (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.5.3 MT_Set_Axis_Mode_Home_ Encoder_Index

功能描述：设置指定的轴为闭环零位模式，查找零位以光栅尺零位和编码器零位为判断依据。

VC	INT32 MT_Set_Axis_Mode_Home_ Encoder_Index (WORD AObj)	
VB	Function MT_Set_Axis_Mode_Home_ Encoder_Index (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Mode_Home_ Encoder_Index (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Mode_Home_ Encoder_Index (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.5.4 MT_Set_Axis_Mode_Home_ Encoder_Home_Switch

功能描述：设置指定的轴为闭环零位模式，查找零位以零位开关和限位开关为判断依据。

VC	INT32 MT_Set_Axis_Mode_Home_ Encoder_Home_Switch (WORD AObj)	
VB	Function MT_Set_Axis_Mode_Home_ Encoder_Home_Switch (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Mode_Home_ Encoder_Home_Switch (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Mode_Home_ Encoder_Home_Switch (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.5.5 MT_Set_Axis_Home_V

功能描述：设置指定轴以指定速度查找零位，速度值为正，则正向查找 0 位，速度为负，则负向查找零位

VC	INT32 MT_Set_Axis_Home_V (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Home_V (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Home_V (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Home_V (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的查找零位速度 Hz/S 可正可负
函数返回	0	函数执行成功

	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界 较慢的速度有利于查找的精度	

3.5.6 MT_Set_Axis_Home_Stop

功能描述：停止指定轴的零位运动模式

VC	INT32 MT_Set_Axis_Home_Stop (WORD AObj)	
VB	Function MT_Set_Axis_Home_Stop (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Home_Stop (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Home_Stop (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.5.7 MT_Set_Axis_Home_V_Start

功能描述：设置指定轴的零位模式启动初始速度，默认为 50Hz/s,一般无需修改。

VC	INT32 MT_Set_Axis_Home_V_Start (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Home_V _Start(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Home_V_Start (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Home_V_Start (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的启动速度 Hz/S 只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.5.8 MT_Set_Axis_Home_Acc

功能描述：设置指定轴的零位模式的加速度值，默认为 500Hz/s^2 。

VC	INT32 MT_Set_Axis_Home_Acc (WORD AObj,INT32 pValue);
VB	Function MT_Set_Axis_Home_Acc(ByVal AObj As Integer, ByVal Value As Long) As Long

Delphi	function MT_Set_Axis_Home_Acc (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Home_Acc (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的加速度 Hz/S² 只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.5.9 MT_Set_Axis_Home_Dec

功能描述：设置指定轴的零位模式的减速度值，默认为 500Hz/s² 。

VC	INT32 MT_Set_Axis_Home_Dec (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Home_Dec(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Home_Dec (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Home_Dec (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的减速度 Hz/S² 只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.5.10 MT_Get_Axis_Home_Acc

功能描述：读取指定控制轴零位模式的加速度

VC	INT32 MT_Get_Axis_Home_Acc(WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_ Home_Acc (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_ Home_Acc (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_ Home_Acc (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的当前加速度 Hz/S²
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.5.11 MT_Get_Axis_Home_Dec

功能描述：读取指定控制轴零位模式的减速度

VC	INT32 MT_Get_Axis_Home_Dec(WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_ Home_Dec (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_ Home_Dec (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_ Home_Decc (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的当前减速度 Hz/S^2
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.6 速度模式

3.6.1 MT_Set_Axis_Mode_Velocity

功能描述：设置指定的轴为速度模式

VC	INT32 MT_Set_Axis_Mode_Velocity(WORD AObj)	
VB	Function MT_Set_Axis_Mode_Velocity (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Mode_Velocity(AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Mode_Velocity(UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.6.2 MT_Get_Axis_Velocity_V_Target

功能描述：读取速度模式下的目标速度

VC	INT32 MT_Get_Axis_Velocity_V_Target (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Velocity_V_Target (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Velocity_V_Target (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Velocity_V_Target (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号

输出参数	Value	指定轴的目标速度 Hz/S
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.6.3 MT_Set_Axis_Velocity_V_Target_Abs

功能描述：设置速度模式下的绝对目标速度

VC	INT32 MT_Set_Axis_Velocity_V_Target_Abs (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Velocity_V_Target_Abs (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Velocity_V_Target_Abs (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Velocity_V_Target_Abs (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的绝对目标速度 Hz/S 可正可负
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.6.4 MT_Set_Axis_Velocity_V_Target_Rel

功能描述：设置速度模式下的相对当前速度的相对速度

VC	INT32 MT_Set_Axis_Velocity_V_Target_Rel (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Velocity_V_Target_Rel (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Velocity_V_Target_Rel(AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Velocity_V_Target_Rel(UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的相对当前速度的速度 Hz/S 可正可负
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.6.5 MT_Set_Axis_Velocity_Stop

功能描述：停止指定轴的速度运动模式

VC	INT32 MT_Set_Axis_Velocity_Stop (WORD AObj)	
VB	Function MT_Set_Axis_Velocity_Stop (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Velocity_Stop (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Velocity_Stop (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.6.6 MT_Set_Axis_Velocity_V_Start

功能描述：设置指定轴的速度模式启动初始速度，默认为 50Hz/s,一般无需修改。

VC	INT32 MT_Set_Axis_Velocity_V_Start (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Velocity_V _Start(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Velocity_V_Start (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Velocity_V_Start (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的启动速度 Hz/S 只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.6.7 MT_Set_Axis_Velocity_Acc

功能描述：设置指定轴的速度模式的加速度值，默认为 500Hz/s^2 。

VC	INT32 MT_Set_Axis_Velocity_Acc (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Velocity_Acc(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Velocity_Acc (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Velocity_Acc (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的加速度 Hz/S^2 只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.6.8 MT_Set_Axis_Velocity_Dec

功能描述：设置指定轴的速度模式的减速度值，默认为 500Hz/s² 。

VC	INT32 MT_Set_Axis_Velocity_Dec (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Velocity_Dec(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Velocity_Dec (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Velocity_Dec (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的减速度 Hz/S² 只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.6.9 MT_Get_Axis_Velocity_Acc

功能描述：读取指定控制轴速度模式的加速度

VC	INT32 MT_Get_Axis_Velocity_Acc(WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_ Velocity_Acc (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_ Velocity_Acc (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_ Velocity_Acc (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的当前加速度 Hz/S²
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.6.10 MT_Get_Axis_Velocity_Dec

功能描述：读取指定控制轴速度模式的减速度

VC	INT32 MT_Get_Axis_Velocity_Dec(WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_ Velocity_Dec (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_ Velocity_Dec (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_ Velocity_Decc (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号

输出参数	Value	指定轴的当前减速度 Hz/S^2
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7 位置模式(开环和闭环)

3.7.1 MT_Set_Axis_Mode_Position[兼容]

功能描述：设置指定的轴为开环位置模式

VC	INT32 MT_Set_Axis_Mode_Position (WORD AObj)	
VB	Function MT_Set_Axis_Mode_Position (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Mode_Position (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Mode_Position (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.7.2 MT_Set_Axis_Mode_Position_Open

功能描述：设置指定的轴为开环位置模式

VC	INT32 MT_Set_Axis_Mode_Position_Open (WORD AObj)	
VB	Function MT_Set_Axis_Mode_Position _Open(ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Mode_Position _Open(AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Mode_Position_Open (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.7.3 MT_Set_Axis_Mode_Position_Close

功能描述：设置指定的轴为闭环位置模式

VC	INT32 MT_Set_Axis_Mode_Position_Close (WORD AObj)	
VB	Function MT_Set_Axis_Mode_Position _Close(ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Mode_Position _Close(AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Mode_Position_Close (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.7.4 MT_Get_Axis_Position_V_Max

功能描述：读取位置模式下的最大速度

VC	INT32 MT_Get_Axis_Position_V_Max (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Position_V_Max (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Position_V_Max (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Position_V_Max (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的位置模式下最大速度 Hz/S
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.5 MT_Set_Axis_Position_V_Max

功能描述：设置位置模式下的最大速度

VC	INT32 MT_Set_Axis_Position_V_Max (WORD AObj,INT32 Value);	
VB	Function MT_Set_Axis_Position_V_Max (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Position_V_Max (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Position_V_Max (UInt16 AObj, Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的位置模式下最大速度 Hz/S
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.6 MT_Get_Axis_Position_P_Target

功能描述：读取位置模式下的绝对目标位置,开环模式下为目标开环脉冲数，闭环模式下为目标闭环电子脉冲数（由于电路上进行了 4 倍频，单位为实际物理光栅或者编码器脉冲数*4，目标为 4000 的话，对应的物理上为 1000 脉冲）

VC	INT32 MT_Get_Axis_Position_P_Target (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Position_P_Target (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Position_P_Target (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Position_P_Target (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的位置模式下绝对目标位置 单位为 电机步数 可正可负
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.7 MT_Set_Axis_Position_P_Target_Abs

功能描述：设置位置模式下的绝对目标位置，开环模式下为目标开环脉冲数，闭环模式下为目标闭环电子脉冲数（由于电路上进行了 4 倍频，单位为实际物理光栅或者编码器脉冲数*4，目标为 4000 的话，对应的物理上为 1000 脉冲）

VC	INT32 MT_Set_Axis_Position_P_Target_Abs (WORD AObj,INT32 Value);	
VB	Function MT_Set_Axis_Position_P_Target_Abs (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Position_P_Target_Abs (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Position_P_Target_Abs (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的位置模式下绝对目标位置 单位为 电机步数 可正可负
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.8 MT_Set_Axis_Position_P_Target_Rel

功能描述：设置位置模式下的相对当前位置下运动量，开环模式下为目标开环脉冲数，闭环模

式下为目标闭环电子脉冲数（由于电路上进行了 4 倍频，单位为实际物理光栅或者编码器脉冲数*4，目标为 4000 的话，对应的物理上为 1000 脉冲）

VC	INT32 MT_Set_Axis_Position_P_Target_Rel (WORD AObj,INT32 Value);	
VB	Function MT_Set_Axis_Position_P_Target_Rel (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Position_P_Target_Rel (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Position_P_Target_Rel (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的位置模式下相对当前位置的目标移动量 单位为 电机步数 可正可负
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.9 MT_Set_Axis_Position_Stop

功能描述：停止指定轴的位置运动模式

VC	INT32 MT_Set_Axis_Position_Stop (WORD AObj)	
VB	Function MT_Set_Axis_Position_Stop (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Position_Stop (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Position_Stop (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.7.10 MT_Set_Axis_Position_V_Start

功能描述：设置指定轴的位置模式启动初始速度，默认为 50Hz/s,一般无需修改。

VC	INT32 MT_Set_Axis_Position_V_Start (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Position_V _Start(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Position_V_Start (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Position_V_Start (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号

输出参数	Value	设置指定轴的启动速度 Hz/S 只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界 较慢的速度有利于查找的精度	

3.7.11 MT_Set_Axis_Position_Acc

功能描述：设置指定轴的位置模式的加速度值，默认为 500Hz/s^2 。

VC	INT32 MT_Set_Axis_Position_Acc (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Position_Acc(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Position_Acc (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Position_Acc (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的加速度 Hz/S^2 只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.12 MT_Set_Axis_Position_Dec

功能描述：设置指定轴的位置模式的减速度值，默认为 500Hz/s^2 。

VC	INT32 MT_Set_Axis_Position_Dec (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Axis_Position_Dec(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Position_Dec (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Position_Dec (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的减速度 Hz/S^2 只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.13 MT_Get_Axis_Position_Acc

功能描述：读取指定控制轴位置模式的加速度

VC	INT32 MT_Get_Axis_Position_Acc(WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_ Position_Acc (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_ Position_Acc (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_ Position_Acc (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的当前加速度 Hz/S²
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.14 MT_Get_Axis_Position_Dec

功能描述：读取指定控制轴位置模式的减速度

VC	INT32 MT_Get_Axis_Position_Dec(WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_ Position_Dec (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_ Position_Dec (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_ Position_Decc (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的当前减速度 Hz/S²
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.15 MT_Set_Axis_Position_Close_Dec_Factor

功能描述：设置指定轴的闭环位置模式的减速系数，默认为 1。根据负载等各种情况进行调整。可以用后面章节的估算函数进行估算。

VC	INT32 MT_Set_Axis_Position_Close_Dec_Factor (WORD AObj,float pValue);	
VB	Function MT_Set_Axis_Position_Close_Dec_Factor(ByVal AObj As Integer, ByVal Value As Single) As Long	
Delphi	function MT_Set_Axis_Position_Dec (AObj:Word;Value:Single):Integer;	
C#	public static extern int MT_Set_Axis_Position_Dec (UInt16 AObj,float Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的闭环减速系数　只能正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1	

	电机轴序号的函数需要注意不要越界

3.7.16 MT_Set_Encoder_Over_Enable

功能描述：设置指定轴在闭环运动时是否进行过冲补偿，默认不开启。在有些情况下，闭环控制可能会有几个脉冲误差的定位情况。过冲补偿可以在定位误差几个脉冲的情况下进行补偿到指定的位置。如果闭环能正确到达，无需使用本功能

VC	INT32 MT_Set_Encoder_Over_Enable (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Encoder_Over_Enable(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Encoder_Over_Enable (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Encoder_Over_Enable (UInt16 AObj,Int32 Value);	
输入参数	AObj	闭环控制电机控制轴序号
输出参数	Value	1：开启补偿功能 0：不开启补偿功能
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.17 MT_Set_Encoder_Over_Max

功能描述：在开启闭环补偿功能后，本参数决定补偿的最大步数，默认为 100

VC	INT32 MT_Set_Encoder_Over_Max (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Encoder_Over_Max(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Encoder_Over_Max (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Encoder_Over_Max (UInt16 AObj,Int32 Value);	
输入参数	AObj	闭环控制电机控制轴序号
输出参数	Value	补偿步数，必须为正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.7.18 MT_Set_Encoder_Over_Stable

功能描述：在开启闭环补偿功能后，本参数决定补偿的稳定判据，默认为 50

VC	INT32 MT_Set_Encoder_Over_Stable (WORD AObj,INT32 pValue);
VB	Function MT_Set_Encoder_Over_Stable(ByVal AObj As Integer, ByVal Value As Long)

	As Long	
Delphi	function MT_Set_Encoder_Over_Stable (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Encoder_Over_Stable (UInt16 AObj,Int32 Value);	
输入参数	AObj	闭环控制电机控制轴序号
输出参数	Value	补偿步数，必须为正
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.8 编码器/光栅尺接口配置

3.8.1 MT_Set_Encoder_Z_Polarity

功能描述：设置编码器/光栅尺接口 Z 信号的电平定义,一般情况下无需设置

VC	INT32 MT_Set_Encoder_Z_Polarity (WORD AObj,INT32 Value);	
VB	Function MT_Set_Encoder_Z_Polarity (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Encoder_Z_Polarity (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Encoder_Z_Polarity (UInt16 AObj,Int32 Value);	
输入参数	AObj	编码器/光栅尺接口序号
输出参数	Value	0:正常电平 1：反向电平
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	编码器/光栅尺接口的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 编码器/光栅尺接口序号的函数需要注意不要越界	

3.8.2 MT_Set_Encoder_Dir_Polarity

功能描述：设置编码器/光栅尺接口计数方向，一般情况下无需设置

VC	INT32 MT_Set_Encoder_Dir_Polarity (WORD AObj,INT32 Value);	
VB	Function MT_Set_Encoder_Dir_Polarity (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Encoder_Z_Polarity (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Encoder_Z_Polarity (UInt16 AObj,Int32 Value);	
输入参数	AObj	编码器/光栅尺接口序号
输出参数	Value	0：正常方向 1：反向方向

函数返回	0	函数执行成功
	非 0	函数执行失败
备注	编码器/光栅尺接口的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 编码器/光栅尺接口序号的函数需要注意不要越界	

3.9 软件限位

3.9.1 MT_Set_Axis_Software_Limit_Neg_Value

功能描述：设置位置模式下的软件限位功能负限位的值

VC	INT32 MT_Set_Axis_Software_Limit_Neg_Value (WORD AObj,INT32 Value);	
VB	Function MT_Set_Axis_Software_Limit_Neg_Value (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Software_Limit_Neg_Value (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Software_Limit_Neg_Value (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的位置模式下软件负限位值 单位为 电机步数 可正可负
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界 在软件限位有效的情况下本值才起作用 负软限位值要小于正软限位值	

3.9.2 MT_Set_Axis_Software_Limit_Pos_Value

功能描述：设置位置模式下的软件限位功能正限位的值

VC	INT32 MT_Set_Axis_Software_Limit_Pos_Value (WORD AObj,INT32 Value);	
VB	Function MT_Set_Axis_Software_Limit_Pos_Value (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Software_Limit_Pos_Value (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Software_Limit_Pos_Value (UInt16 AObj,Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置指定轴的位置模式下软件正限位值 单位为 电机步数

		可正可负
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界 在软件限位有效的情况下本值才起作用 负软限位值要小于正软限位值	

3.9.3 MT_Set_Axis_Software_Limit_Enable

功能描述：使能指定轴的软件限位模式

VC	INT32 MT_Set_Axis_Software_Limit_Enable (WORD AObj)	
VB	Function MT_Set_Axis_Software_Limit_Enable (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Software_Limit_Enable (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Software_Limit_Enable (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.9.4 MT_Set_Axis_Software_Limit_Disable

功能描述：禁止指定轴的软件限位模式

VC	INT32 MT_Set_Axis_Software_Limit_Disable (WORD AObj)	
VB	Function MT_Set_Axis_Software_Limit_Disable (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Software_Limit_Disable (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Software_Limit_Disable (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界	

3.10 状态相关函数

3.10.1 MT_Get_Axis_Num

功能描述：读取板卡的电机控制轴数

VC	INT32 MT_Get_Axis_Num (INT32* pValue)	
VB	Function MT_Get_Axis_Num (ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Num (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Num (ref Int32 Value);	
输入参数	无	
输出参数	Value	板卡支持的电机控制的数量 0-N
函数返回	0	函数执行成功，读取到的轴数有效
	非 0	函数执行失败，读取到的轴数无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 后续需要用到电机轴序号的函数需要注意不要越界	

3.10.2 MT_Get_Encoder_Num

功能描述：读取板卡的闭环接口轴数（光栅尺或者编码器）

VC	INT32 MT_Get_Encoder_Num (INT32* pValue)	
VB	Function MT_Get_Encoder_Num (ByRef Value As Long) As Long	
Delphi	function MT_Get_Encoder_Num (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Encoder_Num (ref Int32 Value);	
输入参数	无	
输出参数	Value	板卡支持的闭环接口的数量 0-N
函数返回	0	函数执行成功，读取到的轴数有效
	非 0	函数执行失败，读取到的轴数无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 后续需要用到闭环接口序号的函数需要注意不要越界	

3.10.3 MT_Get_Axis_Mode

功能描述：读取板卡指定的工作模式

VC	INT32 MT_Get_Axis_Mode(WORD AObj,INT32* pValue)	
VB	Function MT_Get_Axis_Mode(ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Mode(AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Mode(UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	板卡指定轴的当前工作模式 0:零位模式 1:速度模式 2：位置模式（默认）

		3：直线插补 4：圆弧插补
函数返回	0	函数执行成功，读取到的加速度有效
	非 0	函数执行失败，读取到的加速度无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界 序号越界后本次操作无效	

3.10.4 MT_Get_Axis_Acc[兼容]

功能描述：读取板卡指定的电机控制轴的当前加速度

VC	INT32 MT_Get_Axis_Acc(WORD AObj,INT32* pValue)	
VB	Function MT_Get_Axis_Acc(ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Acc(AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Acc(UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	板卡指定轴的当前加速度值，单位为 Hz/S^2 >=0
函数返回	0	函数执行成功，读取到的加速度有效
	非 0	函数执行失败，读取到的加速度无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界 序号越界后本次操作无效	

3.10.5 MT_Set_Axis_Acc[兼容]

功能描述：设置板卡指定的电机控制轴的加速度

VC	INT32 MT_Set_Axis_Acc(WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Acc(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Acc(AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Acc(UInt16 AObj, Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置板卡指定轴的加速度值，单位为 Hz/S^2 >=0
函数返回	0	函数执行成功，本次设置操作成功
	非 0	函数执行失败，本次设置操作失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界 序号越界后本次操作虽然成，但是数据也是被丢弃无效	

3.10.6 MT_Get_Axis_Dec[兼容]

功能描述：读取板卡指定的电机控制轴的当前减速度

VC	INT32 MT_Get_Axis_Dec(WORD AObj,INT32* pValue)	
VB	Function MT_Get_Axis_Dec(ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Dec(AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Dec(UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	板卡指定轴的当前减速度值，单位为 Hz/S^2 >=0
函数返回	0	函数执行成功，读取到的减速度有效
	非 0	函数执行失败，读取到的减速度无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界 序号越界后本次操作无效	

3.10.7 MT_Set_Axis_Dec[兼容]

功能描述：设置板卡指定的电机控制轴的加速度

VC	INT32 MT_Set_Axis_Dec(WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Dec(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Dec(AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Dec(UInt16 AObj, Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	设置板卡指定轴的减速度值，单位为 Hz/S^2 >=0
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界 序号越界后本次操作虽然成，但是数据也是被丢弃无效	

3.10.8 MT_Get_Axis_V_Now

功能描述：读取指定控制轴当前的速度

VC	INT32 MT_Get_Axis_V_Now(WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_V_Now (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_V_Now(AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_V_Now (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号

输出参数	Value	指定轴的当前速度 Hz/S
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.9 MT_Get_Axis_Software_P

功能描述：读取指定控制轴当前的软件位置

VC	INT32 MT_Get_Axis_Software_P_Now(WORD AObj,INT32* pValue)	
VB	Function MT_Get_Axis_Software_P_Now (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Software_P (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Software_P (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的当前软件位置 单位为步数
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.10 MT_Set_Axis_Software_P

功能描述：设置指定控制轴当前的软件位置

VC	INT32 MT_Set_Axis_Software_P_Now(WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Software_P_Now (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Software_P (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Software_P (UInt16 AObj, Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的当前软件位置 单位为步数
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.11 MT_Get_Axis_Status[兼容]

功能描述：读取指定控制轴当前的状态信息

VC	INT32 MT_Get_Axis_Status(WORD AObj, BYTE* pRun, BYTE* pDir, BYTE* pNeg, BYTE* pPos, BYTE* pZero, BYTE* pMode);	
VB	Function MT_Get_Axis_Status Lib "MT_API.dll" _ (ByVal AObj As Integer, ByRef Run As Byte, ByRef Dir As Byte, _ ByRef Neg As Byte, ByRef Pos As Byte, ByRef Zero As Byte, ByRef Mode As Byte) As Long	
Delphi	function MT_Get_Axis_Status(AObj:Word; pRun:PByte; pDir:PByte; pNeg:PByte; pPos:PByte; pZero:PByte; pMode:PByte):Integer;	
C#	public static extern int MT_Get_Axis_Status(WORD AObj, ref BYTE Run,ref BYTE Dir,ref BYTE Neg,ref BYTE Pos,ref BYTE Zero,ref BYTE Mode);	
输入参数	AObj	电机控制轴序号
输出参数	Run	指定轴的当前运动状态，1 为运动，0 为不运动
	Dir	指定轴当前的方向，1 为正方向，0 为负方向
	Neg	指定轴当前的负限位,1 为负限位有效，0 为负限位无效
	Pos	指定轴当前的正限位,1 为正限位有效，0 为正限位无效
	Zero	指定轴当前的零位,1 为零位有效，0 为零位无效
	Mode	工作模式，0=零位模式，1=速度模式，2=位置模式
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.12 MT_Get_Axis_Status2

功能描述：读取指定控制轴当前的状态信息

VC	INT32 MT_Get_Axis_Status2(WORD AObj, INT32* pRun, INT32* pDir, INT32* pNeg, INT32* pPos, INT32* pZero,

	INT32* pMode);	
VB	Function MT_Get_Axis_Status2 Lib "MT_API.dll" _ (ByVal AObj As Integer, ByRef Run As Integer, ByRef Dir As Integer, _ ByRef Neg As Integer, ByRef Pos As Integer, ByRef Zero As Integer, ByRef Mode As Integer) As Long	
Delphi	function MT_Get_Axis_Status2(AObj:Word; pRun:PInteger; pDir: PInteger; pNeg: PInteger; pPos: PInteger; pZero: PInteger; pMode: PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Status2(WORD AObj, ref Int32 Run,ref Int32 Dir,ref Int32 Neg,ref Int32 Pos,ref Int32 Zero,ref Int32 Mode);	
输入参数	AObj	电机控制轴序号
输出参数	Run	指定轴的当前运动状态，1 为运动，0 为不运动
	Dir	指定轴当前的方向，1 为正方向，0 为负方向
	Neg	指定轴当前的负限位,1 为负限位有效，0 为负限位无效
	Pos	指定轴当前的正限位,1 为正限位有效，0 为正限位无效
	Zero	指定轴当前的零位,1 为零位有效，0 为零位无效
	Mode	工作模式，0=零位模式，1=速度模式，2=位置模式
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.13 MT_Get_Axis_Status_Run

功能描述：读取指定控制轴当前的运行状态

VC	INT32 MT_Get_Axis_Status_Run(WORD AObj, INT32* pRun);	
VB	Function MT_Get_Axis_Status_Run Lib "MT_API.dll" _ (ByVal AObj As Integer, ByRef Run As Integer) As Long	
Delphi	function MT_Get_Axis_Status_Run(AObj:Word; pRun:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Status_Run(WORD AObj, ref Int32 Run);	
输入参数	AObj	电机控制轴序号
输出参数	Run	指定轴的当前运动状态，1 为运动，0 为不运动
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效

备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界

3.10.14 MT_Get_Axis_Status_Dir

功能描述：读取指定控制轴当前的运行方向

VC	INT32 MT_Get_Axis_Status_Dir(WORD AObj, INT32* pDir);	
VB	Function MT_Get_Axis_Status_Dir Lib "MT_API.dll" _ (ByVal AObj As Integer, ByRef Dir As Integer) As Long	
Delphi	function MT_Get_Axis_Status_Dir(AObj:Word; pDir:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Status_Dir(WORD AObj, ref Int32 Dir);	
输入参数	AObj	电机控制轴序号
	Dir	指定轴当前的方向，1 为正方向，0 为负方向
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.15 MT_Get_Axis_Status_Neg

功能描述：读取指定控制轴当前的负限位状态

VC	INT32 MT_Get_Axis_Status_Neg(WORD AObj, INT32* pNeg);	
VB	Function MT_Get_Axis_Status_Neg Lib "MT_API.dll" _ (ByVal AObj As Integer, ByRef Neg As Integer) As Long	
Delphi	function MT_Get_Axis_Status_Neg(AObj:Word; pNeg:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Status_Neg(WORD AObj, ref Int32 Neg);	
输入参数	AObj	电机控制轴序号
	Neg	指定轴当前的负限位,1 为负限位有效，0 为负限位无效
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.16 MT_Get_Axis_Status_Pos

功能描述：读取指定控制轴当前的正限位状态

VC	INT32 MT_Get_Axis_Status_Pos(WORD AObj, INT32* pPos);	
VB	Function MT_Get_Axis_Status_Pos Lib "MT_API.dll" _ (ByVal AObj As Integer, ByRef Pos As Integer) As Long	
Delphi	function MT_Get_Axis_Status_Pos(AObj:Word; pPos:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Status_Pos(WORD AObj, ref Int32 Pos);	
输入参数	AObj	电机控制轴序号
	Pos	指定轴当前的正限位,1 为正限位有效，0 为正限位无效
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.17 MT_Get_Axis_Status_Zero

功能描述：读取指定控制轴当前的零位状态

VC	INT32 MT_Get_Axis_Status_Zero(WORD AObj, INT32* pZero);	
VB	Function MT_Get_Axis_Status_Zero Lib "MT_API.dll" _ (ByVal AObj As Integer, ByRef Zero As Integer) As Long	
Delphi	function MT_Get_Axis_Status_Zero(AObj:Word; pZero:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Status_Zero(WORD AObj, ref Int32 Zero);	
输入参数	AObj	电机控制轴序号
	Zero	指定轴当前的零位,1 为零位有效，0 为零位无效
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.18 MT_Get_Axis_Status_Mode

功能描述：读取指定控制轴当前的工作模式

VC	INT32 MT_Get_Axis_Status_Mode(WORD AObj, INT32* pMode);
VB	Function MT_Get_Axis_Status_Mode Lib "MT_API.dll" _ (ByVal AObj As Integer, ByRef Mode As Integer) As Long
Delphi	function MT_Get_Axis_Status_Mode(AObj:Word; pMode:PInteger):Integer;
C#	public static extern int MT_Get_Axis_Status_Mode(WORD AObj, ref Int32 Mode);

输入参数	AObj	电机控制轴序号
输出参数	Mode	工作模式，0=零位模式，1=速度模式，2=位置模式
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 电机轴序号的函数需要注意不要越界	

3.10.19 MT_Get_Axis_Encoder_Pos

功能描述：读取指定控制轴当前的编码器/光栅尺位置(电子 4 倍频，为实际物理脉冲*4)

VC	INT32 MT_Get_Axis_Encoder_Pos(WORD AObj,INT32* pValue)	
VB	Function MT_Get_Axis_Encoder_Pos(ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Encoder_Pos (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Encoder_Pos (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的当前软件位置 单位为脉冲数
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	闭环电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 闭环电机轴序号的函数需要注意不要越界	

3.10.20 MT_Set_Axis_Encoder_Pos

功能描述：设置指定控制轴当前的编码器/光栅尺位置(电子 4 倍频，为实际物理脉冲*4)

VC	INT32 MT_Set_Axis_Encoder_Pos_Now(WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Encoder_Pos_Now (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Encoder_Pos (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Encoder_Pos (UInt16 AObj, Int32 Value);	
输入参数	AObj	电机控制轴序号
输出参数	Value	指定轴的当前软件位置 单位为脉冲数
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	闭环电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 闭环电机轴序号的函数需要注意不要越界	

3.10.21 MT_Set_Axis_Halt

功能描述：立即停止指定轴的运动，没有减速过程，对所有运动模式有效

VC	INT32 MT_Set_Axis_Halt (WORD AObj)	
VB	Function MT_Set_Axis_Halt (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Halt (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Halt (UInt16 AObj);	
输入参数	AObj	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界 紧急停止惯性较大的运动体时会对驱动器造成一定的冲击，紧急情况下使用，每个模式的停止是减速过程，冲击较小	

3.10.22 MT_Set_Axis_Halt_All

功能描述：立即停止所有轴的运动，没有减速过程，对所有运动模式有效

VC	INT32 MT_Set_Axis_Halt_All (void)	
VB	Function MT_Set_Axis_Halt_All () As Long	
Delphi	function MT_Set_Axis_Halt_All ():Integer;	
C#	public static extern int MT_Set_Axis_Halt_All ();	
输入参数	无	电机控制轴序号
输出参数	无	
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	电机轴的序号从 0 开始，N 轴的卡，序号为 0,1,2,3…N-1 用到电机轴序号的函数需要注意不要越界 紧急停止惯性较大的运动体时会对驱动器造成一定的冲击，紧急情况下使用，每个模式的停止是减速过程，冲击较小	

3.11 存储器操作

备注:当前版本如果有 PLC 模式，则存储器会被 PLC 模式占用，用户不可使用

3.11.1 MT_Get_Param_Mem_Len

功能描述：读取板卡上存储器的容量，单位为字节(Byte)

VC	INT32 MT_Get_Param_Mem_Len (INT32* pValue)
VB	Function MT_Get_Param_Mem_Len (ByRef Value As Long) As Long

Delphi	function MT_Get_Param_Mem_Len (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Num (ref Int32 Value);	
输入参数	无	
输出参数	Value	存储器的容量
函数返回	0	函数执行成功，读取到的容量有效
	非 0	函数执行失败，读取到的容量无效
备注	存储器地址的序号从 0 开始，N 的存储器，序号为 0,1,2,3…N-1	

3.11.2 MT_Get_Param_Mem_Data

功能描述：读取存储器指定地址上的数据，每次一个字节

VC	INT32 MT_Get_Param_Mem_Data (WORD AObj,BYTE* pValue);	
VB	Function MT_Get_Param_Mem_Data (ByVal AObj As Integer, ByRef Value As BYTE) As Long	
Delphi	function MT_Get_Param_Mem_Data (AObj:Word;pValue:PByte):Integer;	
C#	public static extern int MT_Get_Param_Mem_Data (UInt16 AObj, ref Byte Value);	
输入参数	AObj	存储器地址
输出参数	Value	指定地址上的字节数据
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	存储器地址的序号从 0 开始，N 的存储器，序号为 0,1,2,3…N-1	

3.11.3 MT_Set_Param_Mem_Data

功能描述：设置存储器指定地址上的数据，每次一个字节

VC	INT32 MT_Set_Param_Mem_Data (WORD AObj,BYTE Value);	
VB	Function MT_Set_Param_Mem_Data (ByVal AObj As Integer, ByVal Value As BYTE) As Long	
Delphi	function MT_Set_Param_Mem_Data (AObj:Word;Value:BYTE):Integer;	
C#	public static extern int MT_Set_Param_Mem_Data (UInt16 AObj, BYTE Value);	
输入参数	AObj	存储器地址
输出参数	Value	指定地址上的字节数据
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	存储器地址的序号从 0 开始，N 的存储器，序号为 0,1,2,3…N-1	

3.12 光电隔离输入

3.12.1 MT_Get_Optic_In_Num

功能描述：读取板卡上光电隔离输入通道的数量

VC	INT32 MT_Get_Optic_In_Num (INT32* pValue)	
VB	Function MT_Get_Optic_In_Num (ByRef Value As Long) As Long	
Delphi	function MT_Get_Optic_In_Num (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Optic_In_Num (ref Int32 Value);	
输入参数	无	
输出参数	Value	光电隔离输入通道的数量
函数返回	0	函数执行成功，读取到通道数有效
	非 0	函数执行失败，读取到通道数无效
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.12.2 MT_Get_Optic_In_Single

功能描述：读取指定光电隔离输入通道上的电平状态

VC	INT32 MT_Get_Optic_In_Single (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Optic_In_Single (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Optic_In_Single (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Optic_In_Single (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	光电隔离输入通道号
输出参数	Value	指定通道上的电平状态，1 为高电平，0 为低电平
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.12.3 MT_Get_Optic_In_All

功能描述：读取光电隔离输入所有通道上的电平状态

VC	INT32 MT_Get_Optic_In_All (INT32* pValue);	
VB	Function MT_Get_Optic_In_All (ByRef Value As Long) As Long	
Delphi	function MT_Get_Optic_In_All (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Optic_In_All (ref Int32 Value);	
输入参数	无	
输出参数	Value	所有通道上的电平状态，1 为高电平，0 为低电平 通道 0 为 LSB(最低位) 通道 N 对应第 N-1 位
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效

备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1

3.13 光电隔离输出

3.13.1 MT_Get_Optic_Out_Num

功能描述：读取板卡上光电隔离输出通道的数量

VC	INT32 MT_Get_Optic_Out_Num (INT32* pValue)	
VB	Function MT_Get_Optic_Out_Num (ByRef Value As Long) As Long	
Delphi	function MT_Get_Optic_Out_Num (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Optic_Out_Num (ref Int32 Value);	
输入参数	无	
输出参数	Value	光电隔离输出通道的数量
函数返回	0	函数执行成功，读取到通道数有效
	非 0	函数执行失败，读取到通道数无效
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.13.2 MT_Set_Optic_Out_Single

功能描述：设置指定光电隔离输出通道上的电平状态

VC	INT32 MT_Set_Optic_Out_Single (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Optic_Out_Single (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Optic_Out_Single (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Optic_Out_Single (UInt16 AObj, Int32 Value);	
输入参数	AObj	光电隔离输出通道号
输出参数	Value	指定通道上的电平状态，1 为高电平，0 为低电平
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.13.3 MT_Set_Optic_Out_All

功能描述：设置光电隔离输出所有通道上的电平状态

VC	INT32 MT_Set_Optic_Out_All (INT32 Value);
VB	Function MT_Set_Optic_Out_All (ByVal Value As Long) As Long
Delphi	function MT_Set_Optic_Out_All (Value:Integer):Integer;
C#	public static extern int MT_Set_Optic_Out_All (Int32 Value);

输入参数	无	
输出参数	Value	所有通道上的电平状态，1 为高电平，0 为低电平 通道 0 为 LSB(最低位) 通道 N 对应第 N-1 位
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.14 OC 输出

3.14.1 MT_Get_OC_Out_Num

功能描述：读取板卡上 OC 输出通道的数量

VC	INT32 MT_Get_OC_Out_Num (INT32* pValue)	
VB	Function MT_Get_OC_Out_Num (ByRef Value As Long) As Long	
Delphi	function MT_Get_OC_Out_Num (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_OC_Out_Num (ref Int32 Value);	
输入参数	无	
输出参数	Value	光电隔离输出通道的数量
函数返回	0	函数执行成功，读取到通道数有效
	非 0	函数执行失败，读取到通道数无效
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.14.2 MT_Set_OC_Out_Single

功能描述：设置指定 OC 输出通道上的电平状态

VC	INT32 MT_Set_OC_Out_Single (WORD AObj,INT32 pValue);	
VB	Function MT_Set_OC_Out_Single (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_OC_Out_Single (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_OC_Out_Single (UInt16 AObj, Int32 Value);	
输入参数	AObj	光电隔离输出通道号
输出参数	Value	指定通道上的电平状态，1 为高电平，0 为低电平
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.14.3 MT_Set_OC_Out_All

功能描述：设置 OC 输出所有通道上的电平状态

VC	INT32 MT_Set_OC_Out_All (INT32 Value);	
VB	Function MT_Set_OC_Out_All (ByVal Value As Long) As Long	
Delphi	function MT_Set_OC_Out_All (Value:Integer):Integer;	
C#	public static extern int MT_Set_OC_Out_All (Int32 Value);	
输入参数	无	
输出参数	Value	所有通道上的电平状态，1 为高电平，0 为低电平 通道 0 为 LSB(最低位) 通道 N 对应第 N-1 位
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.15 多轴联动插补(包括直线插补)

备注：推荐优先使用带 X 的多轴插补函数

3.15.1 MT_Set_Axis_Line_Axis

功能描述：设置指定插补模块中参与插补的运动轴。不同的板卡有不同数量的插补模块，多个模块可以同时进行多个直线插补，每个插补模块可以随意指定任意 2 轴参与插补，同时进行插补的模块不可以有相同的插补轴。例如，模块 0 用轴 2，轴 3 插补，模块 1 用轴 1 轴 2 插补，这两个模块独立动作时没有问题，不可以同时动作。

VC	INT32 MT_Set_Axis_Line_Axis (WORD AObj,INT32 Axis_ID0,INT32 Axis_ID1)	
VB	Function MT_Set_Axis_Line_Axis (ByVal AObj As Integer, ByVal Axis_ID0 As Long, ByVal Axis_ID1 As Long) As Long	
Delphi	function MT_Set_Axis_Line_Axis (AObj:Word;Axis_ID0,Axis_ID1:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Line_Axis (UInt16 AObj, Int32 Axis_ID0, Int32 Axis_ID1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Axis_ID0	本插补模块的第一个插补轴序号
	Axis_ID1	本插补模块的第二个插补轴序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1 插补轴的序号从 0 开始，N 的轴，序号为 0,1,2,3…N-1	

3.15.2 MT_Set_Axis_Line_Acc

功能描述：设置指定插补模块的加速度。

VC	INT32 MT_Set_Axis_Line_Acc (WORD AObj,INT32 Value)
VB	Function MT_Set_Axis_Line_Acc (ByVal AObj As Integer, ByVal Value As Long) As Long
Delphi	function MT_Set_Axis_Line_Acc (AObj:Word;Value:Integer):Integer;

C#	public static extern int MT_Set_Axis_Line_Acc (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的加速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.3 MT_Set_Axis_Line_Dec

功能描述：设置指定插补模块的加速度。

VC	INT32 MT_Set_Axis_Line_Dec (WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Line_Dec (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Line_Dec (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Line_Dec (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的减速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.4 MT_Set_Axis_Line_V

功能描述：设置指定插补模块的速度。

VC	INT32 MT_Set_Axis_Line_V (WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Line_V (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Line_V (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Line_V (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.5 MT_Set_Axis_Line_Run

功能描述：在设置插补的参数后，启动指定的插补模块，在 MT_Set_Axis_Line_Rel 和 MT_Set_Axis_Line_Abs 后执行，可用组合指令 MT_Set_Axis_Line_Run_Rel 和 MT_Set_Axis_Line_Run_Abs 代替

VC	INT32 MT_Set_Axis_Line_Run (WORD AObj)	
VB	Function MT_Set_Axis_Line_Run (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Line_Run (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Line_Run (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号

函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.6 MT_Set_Axis_Line_Stop

功能描述：减速停止插补轴

VC	INT32 MT_Set_Axis_Line_Stop (WORD AObj)	
VB	Function MT_Set_Axis_Line_Stop (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Line_Stop (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Line_Stop (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.7 MT_Set_Axis_Line_Halt

功能描述：立即停止插补轴

VC	INT32 MT_Set_Axis_Line_Halt (WORD AObj)	
VB	Function MT_Set_Axis_Line_Halt (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Line_Halt (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Line_Halt (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.8 MT_Set_Axis_Line_Rel

功能描述：相对当前的位置设置直线插补的目标位置，设置目标后插补轴不动作，需要用 MT_Set_Axis_Line_Run 启动。

VC	INT32 MT_Set_Axis_Line_Rel (WORD AObj,INT32 Axis_Target0,INT32 Axis_Target1)	
VB	Function MT_Set_Axis_Line_Rel (ByVal AObj As Integer, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Axis_Line_Rel(AObj:Word;Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Axis_Line_Rel(UInt16 AObj, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Axis_Target0	本插补模块的第一个插补轴相对移动的距离
	Axis_Target1	本插补模块的第二个插补轴相对移动的距离
函数返回	0	函数执行成功

	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.9 MT_Set_Axis_Line_Abs

功能描述：设置直线插补的绝对目标位置，设置目标后插补轴不动作，需要用 MT_Set_Axis_Line_Run 启动。

VC	INT32 MT_Set_Axis_Line_Abs (WORD AObj,INT32 Axis_Target0,INT32 Axis_Target1)	
VB	Function MT_Set_Axis_Line_Abs (ByVal AObj As Integer, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Axis_Line_Abs(AObj:Word;Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Axis_Line_Abs(UInt16 AObj, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Axis_Target0	本插补模块的第一个插补轴需要移动到的绝对位置
	Axis_Target1	本插补模块的第二个插补轴需要移动到的绝对位置
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.10 MT_Set_Axis_Line_Run_Rel

功能描述：相对当前的位置设置直线插补的目标位置,插补轴立即动作。

VC	INT32 MT_Set_Axis_Line_Run_Rel (WORD AObj,INT32 Axis_Target0,INT32 Axis_Target1)	
VB	Function MT_Set_Axis_Line_Run_Rel (ByVal AObj As Integer, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Axis_Line_Run_Rel(AObj:Word;Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Axis_Line_Run_Rel(UInt16 AObj, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Axis_Target0	本插补模块的第一个插补轴相对移动的距离
	Axis_Target1	本插补模块的第二个插补轴相对移动的距离
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.11 MT_Set_Axis_Line_Run_Abs

功能描述：设置直线插补的绝对目标位置,插补轴立即动作。

VC	INT32 MT_Set_Axis_Line_Run_Abs (WORD AObj,INT32 Axis_Target0,INT32

	Axis_Target1)	
VB	Function MT_Set_Axis_Line_Run_Abs (ByVal AObj As Integer, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Axis_Line_Run_Abs(AObj:Word;Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Axis_Line_Run_Abs(UInt16 AObj, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Axis_Target0	本插补模块的第一个插补轴需要移动到的绝对位置
	Axis_Target1	本插补模块的第二个插补轴需要移动到的绝对位置
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.12 MT_Get_Axis_Line_Num

功能描述：读取板卡上直线插补模块的数量

VC	INT32 MT_Get_Axis_Line_Num (INT32* pValue)	
VB	Function MT_Get_Axis_Line_Num (ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Line_Num (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Line_Num (ref Int32 Value);	
输入参数	无	
输出参数	Value	直线插补模块的数量
函数返回	0	函数执行成功，读取到通道数有效
	非 0	函数执行失败，读取到通道数无效
备注	模块的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.15.13 MT_Get_Axis_Line_Status

功能描述：读取指定直线插补模块的当前插补状态

VC	INT32 MT_Get_Axis_Line_Status (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Line_Status (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Line_Status (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Line_Status (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	Value	指定模块的插补状态，0 为插补结束，1 为正在插补中
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.14 MT_Get_Axis_Line_Acc

功能描述：读取指定直线插补模块的当前加速度

VC	INT32 MT_Get_Axis_Line_Acc (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Line_Acc (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Line_Acc (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Line_Acc (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	Value	指定模块的加速度
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.15 MT_Get_Axis_Line_Dec

功能描述：读取指定直线插补模块的当前减速度

VC	INT32 MT_Get_Axis_Line_Dec (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Line_Dec (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Line_Dec (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Line_Dec (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	Value	指定模块的减速度
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.16 MT_Get_Axis_Line_V

功能描述：读取指定直线插补模块的当前速度

VC	INT32 MT_Get_Axis_Line_V (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Line_V (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Line_V (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Line_V (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	Value	指定模块的速度
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.17 MT_Get_Axis_Line_Axis

功能描述：读取指定直线插补模块的当前的插补轴

VC	INT32 MT_Get_Axis_Line_Axis (WORD AObj,INT32* pID0,INT32* pID1);	
VB	Function MT_Get_Axis_Line_Axis (ByVal AObj As Integer, ByRef pID0 As Long, ByRef pID1 As Long) As Long	
Delphi	function MT_Get_Axis_Line_Axis (AObj:Word; pID0,pID1:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Line_Axis (UInt16 AObj, ref Int32 ID0, ref Int32 ID1);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	pID0	指定模块的第一个插补轴
	pID1	指定模块的第二个插补轴
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1 插补轴的序号从 0 开始，N 的运动轴，序号为 0,1,2,3…N-1	

3.15.18 MT_Set_Axis_Line_V_Start

功能描述：设置指定插补模块的起始速度(最快轴速度)。

VC	INT32 MT_Set_Axis_Line_V_Start (WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Line_V _Start(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Line_V _Start(AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Line_V_Start (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.19 MT_Set_Axis_Line_X_Count

功能描述：设置参与联动插补的轴数。

VC	INT32 MT_Set_Axis_Line_X_Count (WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Line_X_Count (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Line_X_Count(AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Line_X_Count(UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号

	Value	参数插补的轴的数量，范围 2-X，X 为控制卡的轴数
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.20 MT_Set_Axis_Line_X_Axis

功能描述：设置参与联动插补的轴和在插补组中的序号,设置完序号和插补目标后，用 MT_Set_Axis_Line_Run 启动。

VC	INT32 MT_Set_Axis_Line_X_Axis(WORD AObj,INT32 AAxisID,INT32 AAxis)	
VB	Function MT_Set_Axis_Line_X_Axis (ByVal AObj As Integer,ByVal AAxisID As Long,ByVal AAxis As Long) As Long	
Delphi	function MT_Set_Axis_Line_X_ Axis (AObj:Word;AAxisID:Integer;AAxis:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Line_X_Axis(UInt16 AObj,Int32 AAxisID,Int32 AAxis)	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AAxisID	插补模式的参与轴在插补模式中的序号，不是插补轴的序号，范围为 MT_Set_Axis_Line_X_Count 中设置的 0..Count-1
	AAxis	插补轴的序号，0..M-1　M 为轴数
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.21 MT_Set_Axis_Line_X_Target_Rel

功能描述：设置参与联动插补的轴相对的插补目标，设置完序号和插补目标后，用 MT_Set_Axis_Line_Run 启动。

VC	INT32 MT_Set_Axis_Line_X_ Target_Rel (WORD AObj,INT32 AAxisID,INT32 ATarget)	
VB	Function MT_Set_Axis_Line_X_ Target_Rel (ByVal AObj As Integer,ByVal AAxisID As Long,ByVal ATarget As Long) As Long	
Delphi	function MT_Set_Axis_Line_X_ Target_Rel (AObj:Word; AAxisID:Integer;ATarget:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Line_X_ Target_Rel (UInt16 AObj,Int32 AAxisID,Int32 ATarget)	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AAxisID	插补模式的参与轴在插补模式中的序号，不是插补轴的序号，范围为 MT_Set_Axis_Line_X_Count 中设置的 0..Count-1
	ATarget	相对插补位置
函数返回	0	函数执行成功

	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.22 MT_Set_Axis_Line_X_Target_Abs

功能描述：设置参与联动插补的轴绝对的插补目标，设置完序号和插补目标后，用 MT_Set_Axis_Line_Run 启动。

VC	INT32 MT_Set_Axis_Line_X_ Target_Abs (WORD AObj,INT32 AAxisID,INT32 ATarget)	
VB	Function MT_Set_Axis_Line_X_ Target_Abs (ByVal AObj As Integer,ByVal AAxisID As Long,ByVal ATarget As Long) As Long	
Delphi	function MT_Set_Axis_Line_X_ Target_Abs (AObj:Word; AAxisID:Integer;ATarget:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Line_X_ Target_Abs (UInt16 AObj,Int32 AAxisID,Int32 ATarget)	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AAxisID	插补模式的参与轴在插补模式中的序号，不是插补轴的序号，范围为 MT_Set_Axis_Line_X_Count 中设置的 0..Count-1
	ATarget	绝对插补位置
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.23 MT_Set_Axis_Line_X_Run_Rel

功能描述：相对方式启动联动插补，相当于 MT_Set_Axis_Line_X_Count,MT_Set_Axis_Line_X_Axis,MT_Set_Axis_Line_X_Target_Rel 和 MT_Set_Axis_Line_Run 的组合。

VC	INT32 MT_Set_Axis_Line_X_Run_Rel(WORD AObj,INT32 AAxis_Num,INT32* pAxis,INT32* pTarget)	
VB	Function MT_Set_Axis_Line_X_Run_Rel (ByVal AObj As Integer,ByVal AAxis_Num As Long,ByRef pAxis As Long,ByRef pTarget As Long) As Long	
Delphi	function MT_Set_Axis_Line_X_Run_Rel (AObj:Word; AAxis_Num:Integer;pAxis:PInteger;pTarget:PInteger):Integer;	
C#	public static extern int MT_Set_Axis_Line_X_Run_Rel(UInt16 AObj,Int32 AAxis_Num,ref Int32 pAxis,ref Int32 pTarget)	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AAxis_Num	参与插补的轴的数量 2-X
	pAxis	参与插补轴的序号的**数组**，需要传入一个数组，这个数组的长度要大于等于参与插补轴的数量
	ATarget	参与插补轴的相对目标的**数组**，需要传入一个数组，这个数组

		的长度要大于等于参与插补轴的数量
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.15.24 MT_Set_Axis_Line_X_Run_Abs

功能描述：绝对方式启动联动插补，相当于 MT_Set_Axis_Line_X_Count,MT_Set_Axis_Line_X_Axis,MT_Set_Axis_Line_X_Target_Abs 和 MT_Set_Axis_Line_Run 的组合。

VC	INT32 MT_Set_Axis_Line_X_Run_Abs(WORD AObj,INT32 AAxis_Num,INT32* pAxis,INT32* pTarget)	
VB	Function MT_Set_Axis_Line_X_Run_Abs (ByVal AObj As Integer,ByVal AAxis_Num As Long,ByRef pAxis As Long,ByRef pTarget As Long) As Long	
Delphi	function MT_Set_Axis_Line_X_Run_Abs (AObj:Word; AAxis_Num:Integer;pAxis:PInteger;pTarget:PInteger):Integer;	
C#	public static extern int MT_Set_Axis_Line_X_Run_Abs(UInt16 AObj,Int32 AAxis_Num,ref Int32 pAxis,ref Int32 pTarget)	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AAxis_Num	参与插补的轴的数量 2-X
	pAxis	参与插补轴的序号的**数组**，需要传入一个数组，这个数组的长度要大于等于参与插补轴的数量
	ATarget	参与插补轴的绝对目标的**数组**，需要传入一个数组，这个数组的长度要大于等于参与插补轴的数量
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16 圆弧插补

3.16.1 MT_Set_Axis_Circle_Axis

功能描述：设置指定插补模块中参与插补的运动轴。不同的板卡有不同数量的插补模块，多个模块可以同时进行多个圆弧插补，每个插补模块可以随意指定任意 2 轴参与插补，同时进行插补的模块不可以有相同的插补轴。例如，模块 0 用轴 2，轴 3 插补，模块 1 用轴 1 轴 2 插补，这两个模块独立动作时没有问题，不可以同时动作。

VC	INT32 MT_Set_Axis_Circle_Axis (WORD AObj,INT32 Axis_ID0,INT32 Axis_ID1)
VB	Function MT_Set_Axis_Circle_Axis (ByVal AObj As Integer, ByVal Axis_ID0 As Long,

	ByVal Axis_ID1 As Long) As Long	
Delphi	function MT_Set_Axis_Circle_Axis (AObj:Word;Axis_ID0,Axis_ID1:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Circle_Axis (UInt16 AObj, Int32 Axis_ID0, Int32 Axis_ID1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Axis_ID0	本插补模块的第一个插补轴序号
	Axis_ID1	本插补模块的第二个插补轴序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1 插补轴的序号从 0 开始，N 的轴，序号为 0,1,2,3…N-1	

3.16.2 MT_Set_Axis_Circle_Acc

功能描述：设置指定插补模块的加速度。

VC	INT32 MT_Set_Axis_Circle_Acc (WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Circle_Acc (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Circle_Acc (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Circle_Acc (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的加速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.3 MT_Set_Axis_Circle_Dec

功能描述：设置指定插补模块的加速度。

VC	INT32 MT_Set_Axis_Circle_Dec (WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Circle_Dec (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Circle_Dec (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Circle_Dec (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的减速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.4 MT_Set_Axis_Circle_V

功能描述：设置指定插补模块的速度。

VC	INT32 MT_Set_Axis_Circle_V (WORD AObj,INT32 Value)	
VB	Function MT_Set_Axis_Circle_V (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Axis_Circle_V (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Axis_Circle_V (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.5 MT_Set_Axis_Circle_Stop

功能描述：减速停止插补轴

VC	INT32 MT_Set_Axis_Circle_Stop (WORD AObj)	
VB	Function MT_Set_Axis_Circle_Stop (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Circle_Stop (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Circle_Stop (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.6 MT_Set_Axis_Circle_Halt

功能描述：立即停止插补轴

VC	INT32 MT_Set_Axis_Circle_Halt (WORD AObj)	
VB	Function MT_Set_Axis_Circle_Halt (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Axis_Circle_Halt (AObj:Word):Integer;	
C#	public static extern int MT_Set_Axis_Circle_Halt (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.7 MT_Set_Axis_Circle_R_CW_Run_Rel

功能描述：以相对当前的位置的参数顺时针圆弧插补，立即启动。

VC	INT32 MT_Set_Axis_Circle_R_CW_Run_Rel (WORD AObj,INT32 AR,INT32 Axis_Target0,INT32 Axis_Target1)
VB	Function MT_Set_Axis_Circle_R_CW_Run_Rel (ByVal AObj As Integer,ByVal AR As Long, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long
Delphi	Function MT_Set_Axis_Circle_R_CW_Run_Rel

	(AObj:Word;AR,Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Axis_Circle_R_CW_Run_Rel (UInt16 AObj,Int32 AR, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AR	圆弧插补的半径，正数为劣弧，负数为优弧，半径为圆弧的半径，无相对绝对之分
	Axis_Target0	本插补模块的第一个插补轴相对移动的距离
	Axis_Target1	本插补模块的第二个插补轴相对移动的距离
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.8 MT_Set_Axis_Circle_R_CW_Run_Abs

功能描述：以绝对的位置的参数顺时针圆弧插补，立即启动。

VC	INT32 MT_Set_Axis_Circle_R_CW_Run_Abs (WORD AObj,INT32 AR,INT32 Axis_Target0,INT32 Axis_Target1)	
VB	Function MT_Set_Axis_Circle_R_CW_Run_Abs (ByVal AObj As Integer,ByVal AR As Long, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Axis_Circle_R_CW_Run_Abs (AObj:Word;AR,Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Axis_Circle_R_CW_Run_Abs (UInt16 AObj,Int32 AR, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AR	圆弧插补的半径，正数为劣弧，负数为优弧，半径为圆弧的半径，无相对绝对之分
	Axis_Target0	本插补模块的第一个插补轴需要移动到的绝对位置
	Axis_Target1	本插补模块的第二个插补轴需要移动到的绝对位置
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.9 MT_Set_Axis_Circle_R_CCW_Run_Rel

功能描述：以相对当前的位置的参数逆时针圆弧插补，立即启动。

VC	INT32 MT_Set_Axis_Circle_R_CCW_Run_Rel (WORD AObj,INT32 AR,INT32 Axis_Target0,INT32 Axis_Target1)
VB	Function MT_Set_Axis_Circle_R_CCW_Run_Rel (ByVal AObj As Integer,ByVal AR As Long, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long
Delphi	Function MT_Set_Axis_Circle_R_CCW_Run_Rel (AObj:Word;AR,Axis_Target0,Axis_Target1:Integer):Integer
C#	public static extern int MT_Set_Axis_Circle_R_CCW_Run_Rel (UInt16 AObj,Int32 AR, Int32 Axis_Target0, Int32 Axis_Target1);

输入参数	AObj	插补模块序号，不是插补轴的序号
	AR	圆弧插补的半径，正数为劣弧，负数为优弧，半径为圆弧的半径，无相对绝对之分
	Axis_Target0	本插补模块的第一个插补轴相对移动的距离
	Axis_Target1	本插补模块的第二个插补轴相对移动的距离
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.10 MT_Set_Axis_Circle_R_CCW_Run_Abs

功能描述：以绝对的位置的参数逆时针圆弧插补，立即启动。

VC	INT32 MT_Set_Axis_Circle_R_CCW_Run_Abs (WORD AObj,INT32 AR,INT32 Axis_Target0,INT32 Axis_Target1)	
VB	Function MT_Set_Axis_Circle_R_CCW_Run_Abs (ByVal AObj As Integer,ByVal AR As Long, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Axis_Circle_R_CCW_Run_Abs (AObj:Word;AR,Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Axis_Circle_R_CCW_Run_Abs (UInt16 AObj,Int32 AR, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AR	圆弧插补的半径，正数为劣弧，负数为优弧，半径为圆弧的半径，无相对绝对之分
	Axis_Target0	本插补模块的第一个插补轴需要移动到的绝对位置
	Axis_Target1	本插补模块的第二个插补轴需要移动到的绝对位置
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.11 MT_Get_Axis_Circle_Num

功能描述：读取板卡上圆弧插补模块的数量

VC	INT32 MT_Get_Axis_Circle_Num (INT32* pValue)	
VB	Function MT_Get_Axis_Circle_Num (ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Circle_Num (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Circle_Num (ref Int32 Value);	
输入参数	无	
输出参数	Value	圆弧插补模块的数量
函数返回	0	函数执行成功，读取到通道数有效
	非 0	函数执行失败，读取到通道数无效
备注	模块的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.16.12 MT_Get_Axis_Circle_Status

功能描述：读取指定圆弧插补模块的当前插补状态

VC	INT32 MT_Get_Axis_Circle_Status (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Circle_Status (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Circle_Status (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Circle_Status (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	Value	指定模块的插补状态，0 为插补完后，1 为正在插补中
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.13 MT_Get_Axis_Circle_Acc

功能描述：读取指定圆弧插补模块的当前加速度

VC	INT32 MT_Get_Axis_Circle_Acc (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Circle_Acc (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Circle_Acc (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Circle_Acc (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	Value	指定模块的加速度
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.14 MT_Get_Axis_Circle_Dec

功能描述：读取指定圆弧插补模块的当前减速度

VC	INT32 MT_Get_Axis_Circle_Dec (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Circle_Dec (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Circle_Dec (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Circle_Dec (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	Value	指定模块的减速度
函数返回	0	函数执行成功，读取到的数据有效

	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.15 MT_Get_Axis_Circle_V

功能描述：读取指定圆弧插补模块的当前速度

VC	INT32 MT_Get_Axis_Circle_V (WORD AObj,INT32* pValue);	
VB	Function MT_Get_Axis_Circle_V (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_Axis_Circle_V (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Circle_V (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	Value	指定模块的速度
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.16.16 MT_Get_Axis_Circle_Axis

功能描述：读取指定圆弧插补模块的当前的插补轴

VC	INT32 MT_Get_Axis_Circle_Axis (WORD AObj,INT32* pID0,INT32* pID1);	
VB	Function MT_Get_Axis_Circle_Axis (ByVal AObj As Integer, ByRef pID0 As Long, ByRef pID1 As Long) As Long	
Delphi	function MT_Get_Axis_Circle_Axis (AObj:Word; pID0,pID1:PInteger):Integer;	
C#	public static extern int MT_Get_Axis_Circle_Axis (UInt16 AObj, ref Int32 ID0, ref Int32 ID1);	
输入参数	AObj	插补模块的序号，不是插补轴的序号
输出参数	pID0	指定模块的第一个插补轴
	pID1	指定模块的第二个插补轴
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1 插补轴的序号从 0 开始，N 的运动轴，序号为 0,1,2,3…N-1	

3.16.17 MT_Set_Axis_Circle_V_Start

功能描述：设置指定插补模块的起始速度。

VC	INT32 MT_Set_Axis_Circle_V_Start (WORD AObj,INT32 Value)
VB	Function MT_Set_Axis_Circle_V _Start(ByVal AObj As Integer, ByVal Value As Long) As Long
Delphi	function MT_Set_Axis_Circle_V _Start(AObj:Word;Value:Integer):Integer;
C#	public static extern int MT_Set_Axis_Circle_V_Start (UInt16 AObj, Int32 Value);

输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.17 PLC 模式

3.17.1 MT_Get_PLC_Var_Num

功能描述：读取板卡上 PLC 模式的全局变量的数量

VC	INT32 MT_Get_PLC_Var_Num (INT32* pValue)	
VB	Function MT_Get_PLC_Var_Num (ByRef Value As Long) As Long	
Delphi	function MT_Get_PLC_Var_Num (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_PLC_Var_Num (ref Int32 Value);	
输入参数	无	
输出参数	Value	PLC 的全局变量数量
函数返回	0	函数执行成功，读取到通道数有效
	非 0	函数执行失败，读取到通道数无效
备注	模块的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.17.2 MT_Get_PLC_Var_Data

功能描述：读取 PLC 模式指定序号的全局变量的值

VC	INT32 MT_Get_PLC_Var_Data (WORD AObj,INT32* pValue);	
VB	Function MT_Get_PLC_Var_Data (ByVal AObj As Integer, ByRef Value As Long) As Long	
Delphi	function MT_Get_PLC_Var_Data (AObj:Word;pValue:PInteger):Integer;	
C#	public static extern int MT_Get_PLC_Var_Data (UInt16 AObj, ref Int32 Value);	
输入参数	AObj	PLC 全局变量的序号，0..N-1
输出参数	Value	指定序号全局变量的值
函数返回	0	函数执行成功，读取到的数据有效
	非 0	函数执行失败，读取到的数据无效
备注	PLC 全局变量的序号从 0 开始，N 的全局变量，序号为 0,1,2,3…N-1	

3.17.3 MT_Set_PLC_Var_Data

功能描述：设置 PLC 模式指定序号的全局变量的值。

VC	INT32 MT_Set_PLC_Var_Data (WORD AObj,INT32 Value)
VB	Function MT_Set_PLC_Var_Data (ByVal AObj As Integer, ByVal Value As Long) As Long

Delphi	function MT_Set_PLC_Var_Data (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_PLC_Var_Data (UInt16 AObj, Int32 Value);	
输入参数	AObj	PLC 全局变量的序号，0..N-1
	Value	指定序号全局变量的值
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	PLC 全局变量的序号从 0 开始，N 的全局变量，序号为 0,1,2,3…N-1	

3.17.4 MT_Set_PLC_Pause

功能描述：暂停 PLC 模式的指令执行,使用 MT_Set_PLC_Run 后从停止的地方继续执行。

VC	INT32 MT_Set_PLC_Pause(void)	
VB	Function MT_Set_PLC_Pause () As Long	
Delphi	function MT_Set_PLC_Pause ():Integer;	
C#	public static extern int MT_Set_PLC_Pause();	
输入参数		
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.17.5 MT_Set_PLC_Stop

功能描述：停止 PLC 模式的指令执行,使用 MT_Set_PLC_Run 后从头开始执行。

VC	INT32 MT_Set_PLC_Stop(void)	
VB	Function MT_Set_PLC_Stop () As Long	
Delphi	function MT_Set_PLC_Stop ():Integer;	
C#	public static extern int MT_Set_PLC_Stop();	
输入参数		
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.17.6 MT_Set_PLC_Run

功能描述：启动 PLC 模式。

VC	INT32 MT_Set_PLC_Run(void)
VB	Function MT_Set_PLC_Run () As Long
Delphi	function MT_Set_PLC_Run ():Integer;

C#	public static extern int MT_Set_PLC_Run();	
输入参数		
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.18 Stream 流指令模式

3.18.1 MT_Get_Stream_Space

功能描述：读取板卡上 Stream 模式的指令空间空余量，有空余才能写入指令，否则会被丢弃。

VC	INT32 MT_Get_Stream_Space (INT32* pValue)	
VB	Function MT_Get_Stream_Space (ByRef Value As Long) As Long	
Delphi	function MT_Get_Stream_Space (pValue:PInteger):Integer;	
C#	public static extern int MT_Get_Stream_Space (ref Int32 Value);	
输入参数	无	
输出参数	Value	Stream 的指令空间余量
函数返回	0	函数执行成功，读取到通道数有效
	非 0	函数执行失败，读取到通道数无效
备注		

3.18.2 MT_Set_Stream_Pause

功能描述：暂停 Stream 模式的指令执行,使用 MT_Set_Stream_Run 后从停止的地方继续执行。

VC	INT32 MT_Set_Stream_Pause(void)	
VB	Function MT_Set_Stream_Pause () As Long	
Delphi	function MT_Set_Stream_Pause ():Integer;	
C#	public static extern int MT_Set_Stream_Pause();	
输入参数		
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.18.3 MT_Set_Stream_Stop

功能描述：停止 Stream 模式的指令执行,使用 MT_Set_Stream_Run 后从头开始执行。

VC	INT32 MT_Set_Stream_Stop(void)

VB	Function MT_Set_Stream_Stop () As Long	
Delphi	function MT_Set_Stream_Stop ():Integer;	
C#	public static extern int MT_Set_Stream_Stop();	
输入参数		
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.18.4 MT_Set_Stream_Run

功能描述：启动 Stream 模式。

VC	INT32 MT_Set_Stream_Run(void)	
VB	Function MT_Set_Stream_Run () As Long	
Delphi	function MT_Set_Stream_Run ():Integer;	
C#	public static extern int MT_Set_Stream_Run();	
输入参数		
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.18.5 MT_Set_Stream_Clear

功能描述：清除指令空间中还未执行的 Stream 指令。

VC	INT32 MT_Set_Stream_Clear(void)	
VB	Function MT_Set_Stream_Clear () As Long	
Delphi	function MT_Set_Stream_Clear ():Integer;	
C#	public static extern int MT_Set_Stream_Clear();	
输入参数		
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.18.6 MT_Set_Stream_Delay

功能描述：在 Stream 指令序列中插入延时等待，等待精度为 1ms。延时后才继续执行后续的指令,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Delay (INT32 Value);
VB	Function MT_Set_Stream_Delay (ByVal Value As Long) As Long

Delphi	function MT_Set_Stream_Delay (Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Delay (Int32 Value);	
输入参数	无	
输出参数	Value	等待的 ms 数
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.18.7 MT_Set_Stream_Line_Acc

功能描述：Stream 模式设置指定插补模块的加速度,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Line_Acc (WORD AObj,INT32 Value)	
VB	Function MT_Set_Stream_Line_Acc (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_Line_Acc (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Line_Acc (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的加速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.8 MT_Set_Stream_Line_Dec

功能描述：Stream 模式设置指定插补模块的加速度,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Line_Dec (WORD AObj,INT32 Value)	
VB	Function MT_Set_Stream_Line_Dec (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_Line_Dec (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Line_Dec (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的减速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.9 MT_Set_Stream_Line_V_Max

功能描述：Stream 模式设置指定插补模块的速度,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Line_V_Max (WORD AObj,INT32 Value)
VB	Function MT_Set_Stream_Line_V _Max(ByVal AObj As Integer, ByVal Value As Long) As Long

Delphi	function MT_Set_Stream_Line_V _Max(AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Line_V_Max (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.10 MT_Set_Stream_Circle_V_Start

功能描述：Stream 模式设置指定插补模块的起始速度,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_V_Start (WORD AObj,INT32 Value)	
VB	Function MT_Set_Stream_Circle_V _Start(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_Circle_V _Start(AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Circle_V_Start (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.11 MT_Set_Stream_Line_Stop

功能描述：Stream 模式减速停止插补轴,不阻塞后续指令

VC	INT32 MT_Set_Stream_Line_Stop (WORD AObj)	
VB	Function MT_Set_Stream_Line_Stop (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Stream_Line_Stop (AObj:Word):Integer;	
C#	public static extern int MT_Set_Stream_Line_Stop (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.12 MT_Set_Stream_Line_Halt

功能描述：Stream 模式立即停止插补轴,不阻塞后续指令

VC	INT32 MT_Set_Stream_Line_Halt (WORD AObj)
VB	Function MT_Set_Stream_Line_Halt (ByVal AObj As Integer) As Long
Delphi	function MT_Set_Stream_Line_Halt (AObj:Word):Integer;
C#	public static extern int MT_Set_Stream_Line_Halt (UInt16 AObj);

输入参数	AObj	插补模块序号，不是插补轴的序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.13 MT_Set_Stream_Line_X_Run_Rel

功能描述：Stream 模式相对方式启动联动插补,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Line_X_Run_Rel(WORD AObj,INT32 AStream_Num,INT32* pStream,INT32* pTarget)	
VB	Function MT_Set_Stream_Line_X_Run_Rel (ByVal AObj As Integer,ByVal AStream_Num As Long,ByRef pStream As Long,ByRef pTarget As Long) As Long	
Delphi	function MT_Set_Stream_Line_X_Run_Rel (AObj:Word; AStream_Num:Integer;pStream:PInteger;pTarget:PInteger):Integer;	
C#	public static extern int MT_Set_Stream_Line_X_Run_Rel(UInt16 AObj,Int32 AStream_Num,ref Int32 pStream,ref Int32 pTarget)	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AStream_Num	参与插补的轴的数量 2-X
	pStream	参与插补轴的序号的**数组**，需要传入一个数组，这个数组的长度要大于等于参与插补轴的数量
	ATarget	参与插补轴的相对目标的**数组**，需要传入一个数组，这个数组的长度要大于等于参与插补轴的数量
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.14 MT_Set_Stream_Line_X_Run_Abs

功能描述：Stream 模式绝对方式启动联动插补,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Line_X_Run_Abs(WORD AObj,INT32 AStream_Num,INT32* pStream,INT32* pTarget)	
VB	Function MT_Set_Stream_Line_X_Run_Abs (ByVal AObj As Integer,ByVal AStream_Num As Long,ByRef pStream As Long,ByRef pTarget As Long) As Long	
Delphi	function MT_Set_Stream_Line_X_Run_Abs (AObj:Word; AStream_Num:Integer;pStream:PInteger;pTarget:PInteger):Integer;	
C#	public static extern int MT_Set_Stream_Line_X_Run_Abs(UInt16 AObj,Int32 AStream_Num,ref Int32 pStream,ref Int32 pTarget)	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AStream_Num	参与插补的轴的数量 2-X

	pStream	参与插补轴的序号的**数组**，需要传入一个数组，这个数组的长度要大于等于参与插补轴的数量
	ATarget	参与插补轴的绝对目标的**数组**，需要传入一个数组，这个数组的长度要大于等于参与插补轴的数量
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.15 MT_Set_Stream_Wait_Line

功能描述：等待 Stream 模式联动插补停止，会阻塞后续指令的执行。

VC	INT32 MT_Set_Stream_Wait_Line (WORD AObj)	
VB	Function MT_Set_Stream_Wait_Line (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Stream_Wait_Line (AObj:Word):Integer;	
C#	public static extern int MT_Set_Stream_Wait_Line (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.16 MT_Set_Stream_Circle_Axis

功能描述：Stream 模式下设置指定插补模块中参与插补的运动轴。不同的板卡有不同数量的插补模块，多个模块可以同时进行多个圆弧插补，每个插补模块可以随意指定任意 2 轴参与插补，同时进行插补的模块不可以有相同的插补轴。例如，模块 0 用轴 2，轴 3 插补，模块 1 用轴 1 轴 2 插补，这两个模块独立动作时没有问题，不可以同时动作,不阻塞后续指令。。

VC	INT32 MT_Set_Stream_Circle_Axis (WORD AObj,INT32 Axis_ID0,INT32 Axis_ID1)	
VB	Function MT_Set_Stream_Circle_Axis (ByVal AObj As Integer, ByVal Axis_ID0 As Long, ByVal Axis_ID1 As Long) As Long	
Delphi	function MT_Set_Stream_Circle_Axis (AObj:Word;Axis_ID0,Axis_ID1:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Circle_Axis (UInt16 AObj, Int32 Axis_ID0, Int32 Axis_ID1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Axis_ID0	本插补模块的第一个插补轴序号
	Axis_ID1	本插补模块的第二个插补轴序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1 插补轴的序号从 0 开始，N 的轴，序号为 0,1,2,3…N-1	

3.18.17 MT_Set_Stream_Circle_Acc

功能描述：Stream 模式下设置指定插补模块的加速度,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_Acc (WORD AObj,INT32 Value)	
VB	Function MT_Set_Stream_Circle_Acc (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_Circle_Acc (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Circle_Acc (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的加速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.18 MT_Set_Stream_Circle_Dec

功能描述：Stream 模式下设置指定插补模块的加速度,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_Dec (WORD AObj,INT32 Value)	
VB	Function MT_Set_Stream_Circle_Dec (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_Circle_Dec (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Circle_Dec (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的减速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.19 MT_Set_Stream_Circle_V_Max

功能描述：Stream 模式下设置指定插补模块的速度,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_V_Max (WORD AObj,INT32 Value)	
VB	Function MT_Set_Stream_Circle_V _Max(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_Circle_V _Max(AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Circle_V_Max (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.20 MT_Set_Stream_Circle_V_Start

功能描述：Stream 模式下设置指定插补模块的起始速度,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_V_Start (WORD AObj,INT32 Value)	
VB	Function MT_Set_Stream_Circle_V _Start(ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_Circle_V _Start(AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Circle_V_Start (UInt16 AObj, Int32 Value);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	Value	本插补模块的起始速度
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.21 MT_Set_Stream_Circle_Stop

功能描述：Stream 模式下减速停止插补轴,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_Stop (WORD AObj)	
VB	Function MT_Set_Stream_Circle_Stop (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Stream_Circle_Stop (AObj:Word):Integer;	
C#	public static extern int MT_Set_Stream_Circle_Stop (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.22 MT_Set_Stream_Circle_Halt

功能描述：Stream 模式下立即停止插补轴,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_Halt (WORD AObj)	
VB	Function MT_Set_Stream_Circle_Halt (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Stream_Circle_Halt (AObj:Word):Integer;	
C#	public static extern int MT_Set_Stream_Circle_Halt (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.23 MT_Set_Stream_Circle_R_CW_Run_Rel

功能描述：Stream 模式下以相对当前的位置的参数顺时针圆弧插补，立即启动,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_R_CW_Run_Rel (WORD AObj,INT32 AR,INT32

	Axis_Target0,INT32 Axis_Target1)	
VB	Function MT_Set_Stream_Circle_R_CW_Run_Rel (ByVal AObj As Integer,ByVal AR As Long, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Stream_Circle_R_CW_Run_Rel (AObj:Word;AR,Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Stream_Circle_R_CW_Run_Rel (UInt16 AObj,Int32 AR, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AR	圆弧插补的半径，正数为劣弧，负数为优弧，半径为圆弧的半径，无相对绝对之分
	Axis_Target0	本插补模块的第一个插补轴相对移动的距离
	Axis_Target1	本插补模块的第二个插补轴相对移动的距离
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.24 MT_Set_Stream_Circle_R_CW_Run_Abs

功能描述：Stream 模式下以绝对的位置的参数顺时针圆弧插补，立即启动,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_R_CW_Run_Abs (WORD AObj,INT32 AR,INT32 Axis_Target0,INT32 Axis_Target1)	
VB	Function MT_Set_Stream_Circle_R_CW_Run_Abs (ByVal AObj As Integer,ByVal AR As Long, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Stream_Circle_R_CW_Run_Abs (AObj:Word;AR,Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Stream_Circle_R_CW_Run_Abs (UInt16 AObj,Int32 AR, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AR	圆弧插补的半径，正数为劣弧，负数为优弧，半径为圆弧的半径，无相对绝对之分
	Axis_Target0	本插补模块的第一个插补轴需要移动到的绝对位置
	Axis_Target1	本插补模块的第二个插补轴需要移动到的绝对位置
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.25 MT_Set_Stream_Circle_R_CCW_Run_Rel

功能描述：Stream 模式下以相对当前的位置的参数逆时针圆弧插补，立即启动,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_R_CCW_Run_Rel (WORD AObj,INT32 AR,INT32 Axis_Target0,INT32 Axis_Target1)
VB	Function MT_Set_Stream_Circle_R_CCW_Run_Rel (ByVal AObj As Integer,ByVal AR As

	Long, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Stream_Circle_R_CCW_Run_Rel (AObj:Word;AR,Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Stream_Circle_R_CCW_Run_Rel (UInt16 AObj,Int32 AR, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AR	圆弧插补的半径，正数为劣弧，负数为优弧，半径为圆弧的半径，无相对绝对之分
	Axis_Target0	本插补模块的第一个插补轴相对移动的距离
	Axis_Target1	本插补模块的第二个插补轴相对移动的距离
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.26 MT_Set_Stream_Circle_R_CCW_Run_Abs

功能描述：Stream 模式下以绝对的位置的参数逆时针圆弧插补，立即启动,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Circle_R_CCW_Run_Abs (WORD AObj,INT32 AR,INT32 Axis_Target0,INT32 Axis_Target1)	
VB	Function MT_Set_Stream_Circle_R_CCW_Run_Abs (ByVal AObj As Integer,ByVal AR As Long, ByVal Axis_Target0 As Long, ByVal Axis_Target1 As Long) As Long	
Delphi	Function MT_Set_Stream_Circle_R_CCW_Run_Abs (AObj:Word;AR,Axis_Target0,Axis_Target1:Integer):Integer	
C#	public static extern int MT_Set_Stream_Circle_R_CCW_Run_Abs (UInt16 AObj,Int32 AR, Int32 Axis_Target0, Int32 Axis_Target1);	
输入参数	AObj	插补模块序号，不是插补轴的序号
	AR	圆弧插补的半径，正数为劣弧，负数为优弧，半径为圆弧的半径，无相对绝对之分
	Axis_Target0	本插补模块的第一个插补轴需要移动到的绝对位置
	Axis_Target1	本插补模块的第二个插补轴需要移动到的绝对位置
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.27 MT_Set_Stream_Wait_Circle

功能描述：等待 Stream 模式圆弧插补停止，会阻塞后续指令的执行。

VC	INT32 MT_Set_Stream_Wait_Circle (WORD AObj)	
VB	Function MT_Set_Stream_Wait_Circle (ByVal AObj As Integer) As Long	
Delphi	function MT_Set_Stream_Wait_Circle(AObj:Word):Integer;	
C#	public static extern int MT_Set_Stream_Wait_Circle (UInt16 AObj);	
输入参数	AObj	插补模块序号，不是插补轴的序号

函数返回	0	函数执行成功
	非 0	函数执行失败
备注	插补模块的序号从 0 开始，N 的插补模块，序号为 0,1,2,3…N-1	

3.18.28 MT_Set_Stream_Optic_Out_Single

功能描述：Stream 模式下设置指定光电隔离输出通道上的电平状态,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Optic_Out_Single (WORD AObj,INT32 pValue);	
VB	Function MT_Set_Stream_Optic_Out_Single (ByVal AObj As Integer, ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_Optic_Out_Single (AObj:Word;Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Optic_Out_Single (UInt16 AObj, Int32 Value);	
输入参数	AObj	光电隔离输出通道号
输出参数	Value	指定通道上的电平状态，1 为高电平，0 为低电平
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.18.29 MT_Set_Stream_Optic_Out_All

功能描述：Stream 模式下设置光电隔离输出所有通道上的电平状态,不阻塞后续指令。

VC	INT32 MT_Set_Stream_Optic_Out_All (INT32 Value);	
VB	Function MT_Set_Stream_Optic_Out_All (ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_Optic_Out_All (Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_Optic_Out_All (Int32 Value);	
输入参数	无	
输出参数	Value	所有通道上的电平状态，1 为高电平，0 为低电平 通道 0 为 LSB(最低位) 通道 N 对应第 N-1 位
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.18.30 MT_Set_Stream_OC_Out_Single

功能描述：Stream 模式下设置指定 OC 输出通道上的电平状态,不阻塞后续指令

VC	INT32 MT_Set_Stream_OC_Out_Single (WORD AObj,INT32 pValue);
VB	Function MT_Set_Stream_OC_Out_Single (ByVal AObj As Integer, ByVal Value As Long) As Long
Delphi	function MT_Set_Stream_OC_Out_Single (AObj:Word;Value:Integer):Integer;

C#	public static extern int MT_Set_Stream_OC_Out_Single (UInt16 AObj, Int32 Value);	
输入参数	AObj	光电隔离输出通道号
输出参数	Value	指定通道上的电平状态，1 为高电平，0 为低电平
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.18.31 MT_Set_Stream_OC_Out_All

功能描述：Stream 模式下设置 OC 输出所有通道上的电平状态,不阻塞后续指令

VC	INT32 MT_Set_Stream_OC_Out_All (INT32 Value);	
VB	Function MT_Set_Stream_OC_Out_All (ByVal Value As Long) As Long	
Delphi	function MT_Set_Stream_OC_Out_All (Value:Integer):Integer;	
C#	public static extern int MT_Set_Stream_OC_Out_All (Int32 Value);	
输入参数	无	
输出参数	Value	所有通道上的电平状态，1 为高电平，0 为低电平 通道 0 为 LSB(最低位) 通道 N 对应第 N-1 位
函数返回	0	函数执行成功
	非 0	函数执行失败
备注	通道的序号从 0 开始，N 的通道，序号为 0,1,2,3…N-1	

3.18.32 MT_Set_Stream_Dec_Enable

功能描述：Stream 模式中运动（直线插补和圆弧插补）中连续两段中间允许减速后再加速。

VC	INT32 MT_Set_Stream_Dec_Enable (void)	
VB	Function MT_Set_Stream_Dec_Enable () As Long	
Delphi	function MT_Set_Stream_Dec_Enable ():Integer;	
C#	public static extern int MT_Set_Stream_Dec_Enable ();	
输入参数		
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.18.33 MT_Set_Stream_Dec_Disable

功能描述：Stream 模式中运动（直线插补和圆弧插补）中连续两段中间不允许有减速后再加速。但是可以通过速度设置指令来修改速度，建议在第一段后最后一段路径允许加减速

VC	INT32 MT_Set_Stream_Dec_Disable (void)

VB	Function MT_Set_Stream_Dec_Disable () As Long	
Delphi	function MT_Set_Stream_Dec_Disable ():Integer;	
C#	public static extern int MT_Set_Stream_Dec_Disable ();	
输入参数		
函数返回	0	函数执行成功
	非 0	函数执行失败
备注		

3.19 辅助计算

用户编程时辅助计算的实用公式，方便用户进行物理量和实际运动控制器的输入输出之间的转换计算。

3.19.1 MT_ Help_Step_Line_Real_To_Steps

功能描述：直线移动台物理量(速度、加速度、减速度等)到开环脉冲的转换计算

VC	INT32 MT_ Help_Step_Line_Real_To_Steps(double AStepAngle,INT32 ADiv,double APitch,double ALineRatio,double AValue)	
VB	Function MT_vStep_Line_Real_To_Steps (ByVal AStepAngle As Double,ByVal ADiv As Long,ByVal APitch As Double,ByVal ALineRatio As Double,ByVal AValue As Double) As Long	
Delphi	function MT_ Help_Step_Line_Real_To_Steps(AStepAngle:Double;ADiv:Integer; APitch:Double;ALineRatio:Double;AValue:Double):Integer	
C#	public static extern int MT_ Help_Step_Line_Real_To_Steps(double AStepAngle,Int32 ADiv,double APitch,double ALineRatio,double AValue)	
输入参数	AStepAngle	走一步电机旋转角度，例如两项步进电机一般为 1.8°
	ADiv	驱动器细分数，一般为 2,4,8,16 等等
	APitch	直线台的螺距，即电机旋转一圈的位移，单位 mm
	ALineRatio	直线台的传动比，没有传动为 1
	AValue	物理量，mm,mm/s,mm/s²
函数返回	32 位整形	转换后的步数
	非 0	函数执行失败
备注		

3.19.2 MT_ Help_Step_Circle_Real_To_Steps

功能描述：旋转台物理量(速度、加速度、减速度等)到开环脉冲的转换计算

VC	INT32 MT_ Help_Step_Circle_Real_To_Steps(double AStepAngle,INT32 ADiv,double ACircleRatio,double AValue)
VB	Function MT_ Help_Step_Circle_Real_To_Steps (ByVal AStepAngle As Double,ByVal

	ADiv As Long, ByVal ACircleRatio As Double,ByVal AValue As Double) As Long	
Delphi	function MT_vStep_Circle_Real_To_Steps(AStepAngle:Double;ADiv:Integer; ACircleRatio:Double;AValue:Double):Integer	
C#	public static extern int MT_ Help_Step_Circle_Real_To_Steps(double AStepAngle,Int32 ADiv, double ACircleRatio,double AValue)	
输入参数	AStepAngle	走一步电机旋转角度，例如两项步进电机一般为 1.8°
	ADiv	驱动器细分数，一般为 2,4,8,16 等等
	ACircleRatio	旋转台的传动比
	AValue	物理量，°,°/s,°/s²
函数返回	32 位整形	转换后的步数
	非 0	函数执行失败
备注		

3.19.3 MT_ Help_Step_Line_Steps_To_Real

功能描述：直线移动台脉冲转物理量(速度、加速度、减速度等)

VC	double MT_ Help_Step_Line_Steps_To_Real(double AStepAngle,INT32 ADiv,double APitch,double ALineRatio,double AValue)	
VB	Function MT_ Help_Step_Line_Steps_To_Real (ByVal AStepAngle As Double,ByVal ADiv As Long,ByVal APitch As Double,ByVal ALineRatio As Double,ByVal AValue As Long) As Double	
Delphi	function MT_ Help_Step_Line_Steps_To_Real(AStepAngle:Double;ADiv:Integer; APitch:Double;ALineRatio:Double;AValue:Integer):Double	
C#	public static extern double MT_ Help_Step_Line_Steps_To_Real(double AStepAngle,Int32 ADiv,double APitch,double ALineRatio,INT32 AValue)	
输入参数	AStepAngle	走一步电机旋转角度，例如两项步进电机一般为 1.8°
	ADiv	驱动器细分数，一般为 2,4,8,16 等等
	APitch	直线台的螺距，即电机旋转一圈的位移，单位 mm
	ALineRatio	直线台的传动比，没有传动为 1
	AValue	脉冲数
函数返回	双精度	转换后的物理量，mm,mm/s,mm/s²
	非 0	函数执行失败
备注		

3.19.4 MT_ Help_Step_Circle_Steps_To_Real

功能描述：旋转台脉冲转物理量(速度、加速度、减速度等)

VC	double MT_ Help_Step_Circle_Steps_To_Real(double AStepAngle,INT32 ADiv,double ACircleRatio,INT32 AValue)
VB	Function MT_ Help_Step_Circle_Steps_To_Real (ByVal AStepAngle As Double,ByVal ADiv As Long, ByVal ACircleRatio As Double,ByVal AValue As Double) As Long

Delphi	function MT_ Help_Step_Circle_Steps_To_Real(AStepAngle:Double;ADiv:Integer; ACircleRatio:Double;AValue:Double):Integer	
C#	public static extern int MT_ Help_Step_Circle_Steps_To_Real(double AStepAngle,Int32 ADiv, double ACircleRatio,double AValue)	
输入参数	AStepAngle	走一步电机旋转角度，例如两项步进电机一般为 1.8°
	ADiv	驱动器细分数，一般为 2,4,8,16 等等
	ACircleRatio	旋转台的传动比
	AValue	脉冲数
函数返回	双精度浮点	转换后的物理量°,°/s,°/s²
	非 0	函数执行失败
备注		

3.19.5 MT_ Help_Encoder_Line_Real_To_Steps

功能描述：直线移动台物理量(速度、加速度、减速度等)到闭环编码器脉冲数的转换计算，编码器安装在电机上的用本转换函数，编码器的电子 4 倍频在转换时已经考虑，用户无需再考虑。

VC	INT32 MT_ Help_Encoder_Line_Real_To_Steps(double APitch,double ALineRatio,INT32 ALineCount,double AValue)	
VB	Function MT_ Help_Encoder_Line_Real_To_Steps (ByVal APitch As Double,ByVal ALineRatio As Double,ByVal ALineCount As Long,ByVal AValue As Double) As Long	
Delphi	function MT_ Help_Encoder_Line_Real_To_Steps(APitch:Double;ALineRatio:Double;ALineCount:Integer; AValue:Double):Integer	
C#	public static extern int MT_ Help_Encoder_Line_Real_To_Steps(double APitch,double ALineRatio,Int32 ALineCount,double AValue)	
	APitch	直线台的螺距，即电机旋转一圈的位移，单位 mm
	ALineRatio	直线台的传动比，没有传动为 1
	ALineCount	编码器线数，例如 1000,1024 等
	AValue	物理量，mm,mm/s,mm/s²
函数返回	32 位整形	转换后的编码器脉冲数
	非 0	函数执行失败
备注		

3.19.6 MT_ Help_Encoder_Circle_Real_To_Steps

功能描述：旋转台物理量(速度、加速度、减速度等)到闭环编码器脉冲数的转换计算，编码器安装在电机上的用本转换函数，编码器的电子 4 倍频在转换时已经考虑，用户无需再考虑。

VC	INT32 MT_ Help_Encoder_Circle_Real_To_Steps(double ACircleRatio,INT32 ALineCount,double AValue)
VB	Function MT_ Help_Encoder_Circle_Real_To_Steps(ByVal ACircleRatio As Double,ByVal ALineCount As Long,ByVal AValue As Double) As Long

Delphi	function MT_ Help_Encoder_Circle_Real_To_Steps(ACircleRatio:Double;ALineCount:Integer;AValue:Double):Integer	
C#	public static extern int MT_ Help_Encoder_Circle_Real_To_Steps(double ACircleRatio,Int32 ALineCount,double AValue)	
	ACircleRatio	旋转台的传动比
	ALineCount	编码器线数，例如 1000,1024 等
	AValue	物理量，°,°/s,°/s²
函数返回	32 位整形	转换后的步数
	非 0	函数执行失败
备注		

3.19.7 MT_ Help_Encoder_Line_Steps_To_Real

功能描述：直线移动台编码器脉冲转物理量(速度、加速度、减速度等)，编码器安装在电机上的用本转换函数，编码器的电子 4 倍频在转换时已经考虑，用户无需再考虑，输入 4 倍频后的脉冲数即可。

VC	double MT_ Help_Encoder_Line_Steps_To_Real(double APitch,double ALineRatio,INT32 ALineCount,INT32 AValue)	
VB	Function MT_ Help_Encoder_Line_Steps_To_Real (ByVal APitch As Double,ByVal ALineRatio As Double,ByVal ALineCount As Long,ByVal AValue As Long) As Long	
Delphi	function MT_ Help_Encoder_Line_Steps_To_Real(APitch:Double;ALineRatio:Double;ALineCount:Integer;AValue:Integer):Double	
C#	public static extern double MT_ Help_Encoder_Line_Steps_To_Real(double APitch,double ALineRatio,Int32 ALineCount,Int32 AValue)	
输入	APitch	直线台的螺距，即电机旋转一圈的位移，单位 mm
	ALineRatio	直线台的传动比，没有传动为 1
	ALineCount	编码器线数，例如 1000,1024 等
	AValue	脉冲数
函数返回	双精度	转换后的物理量，mm,mm/s,mm/s²
	非 0	函数执行失败
备注		

3.19.8 MT_ Help_Encoder_Circle_Steps_To_Real

功能描述：旋转台编码器脉冲转物理量(速度、加速度、减速度等) 编码器安装在电机上的用本转换函数，编码器的电子 4 倍频在转换时已经考虑，用户无需再考虑，输入 4 倍频后的脉冲数即可。

VC	double MT_ Help_Encoder_Circle_Steps_To_Real(double ACircleRatio,INT32 ALineCount,INT32 AValue)
VB	Function MT_ Help_Encoder_Circle_Steps_To_Real Lib "MT_API.dll" (ByVal ACircleRatio As Double,ByVal ALineCount As Long,ByVal AValue As Long) As Long

Delphi	function MT_ Help_Encoder_Circle_Steps_To_Real(ACircleRatio:Double;ALineCount:Integer;AValue:Integer):Double	
C#	public static extern int MT_ Help_Encoder_Circle_Steps_To_Real(double ACircleRatio,Int32 ALineCount,Int32 AValue)	
	ACircleRatio	旋转台的传动比
	ALineCount	编码器线数，例如 1000,1024 等
	AValue	脉冲数
函数返回	双精度浮点	转换后的物理量°,°/s,°/s²
	非 0	函数执行失败
备注		

3.19.9 MT_ Help_Grating_Line_Real_To_Steps

功能描述：直线移动台物理量(速度、加速度、减速度等)到闭环光栅尺脉冲数的转换计算，光栅尺安装在机械结构上，光栅尺的电子 4 倍频在转换时已经考虑，用户无需再考虑。

VC	INT32 MT_ Help_Grating_Line_Real_To_Steps(double AUnit_um,double AValue)	
VB	Function MT_ Help_Grating_Line_Real_To_Steps (ByVal AUnit_um As Double,ByVal AValue As Double) As Long	
Delphi	function MT_ Help_Grating_Line_Real_To_Steps(AUnit_um:Double;AValue:Double):Integer	
C#	public static extern int MT_ Help_Grating_Line_Real_To_Steps(double AUnit_um,double AValue)	
	AUnit_um	光栅尺的栅距，单位为 **um**
	AValue	物理量，mm,mm/s,mm/s²
函 数返回	32 位整形	转换后的光栅尺脉冲数
	非 0	函数执行失败
备注		

3.19.10 MT_ Help_Grating_Circle_Real_To_Steps

功能描述：旋转台物理量(速度、加速度、减速度等)到光栅尺脉冲数的转换计算，光栅尺安装在机械结构上，光栅尺的电子 4 倍频在转换时已经考虑，用户无需再考虑。

VC	INT32 MT_ Help_Grating_Circle_Real_To_Steps(INT32 ALineCount,double AValue)	
VB	Function MT_ Help_Grating_Circle_Real_To_Steps (ByVal ALineCount As Long,ByVal AValue As Double) As Long	
Delphi	function MT_ Help_Grating_Circle_Real_To_Steps(ALineCount:Integer;AValue:Double):Integer	
C#	public static extern int MT_ Help_Grating_Circle_Real_To_Steps(Int32 ALineCount,double AValue)	
	ALineCount	光栅尺整圈的光栅线数，例如 64800 等
	AValue	物理量，°,°/s,°/s²

函数返回	32 位整形	转换后的步数
	非 0	函数执行失败
备注		

3.19.11 MT_ Help_Grating_Line_Steps_To_Real

功能描述：直线移动台光栅尺脉冲转物理量(速度、加速度、减速度等)，光栅尺安装在机械上，光栅尺的电子 4 倍频在转换时已经考虑，用户无需再考虑，输入 4 倍频后的脉冲数即可。

VC	double MT_ Help_Grating_Line_Steps_To_Real(double AUnit_um,INT32 AValue)	
VB	Function MT_ Help_Grating_Line_Steps_To_Real (ByVal AUnit_um As Double,ByVal AValue As Long) As Long	
Delphi	function MT_ Help_Grating_Line_Steps_To_Real(AUnit_um:Double;AValue:Integer):Double	
C#	public static extern double MT_ Help_Grating_Line_Steps_To_Real(double AUnit_um,Int32 AValue)	
输入	AUnit_um	光栅尺的栅距，单位为 **um**
	AValue	光栅尺脉冲数
函数返回	双精度	转换后的物理量，mm,mm/s,mm/s²
	非 0	函数执行失败
备注		

3.19.12 MT_ Help_Grating_Circle_Steps_To_Real

功能描述：光栅尺脉冲转物理量(速度、加速度、减速度等) 光栅尺安装在机械结构用本转换函数，编码器的电子 4 倍频在转换时已经考虑，用户无需再考虑，输入 4 倍频后的脉冲数即可。

VC	double MT_ Help_Grating_Circle_Steps_To_Real(INT32 ALineCount,INT32 AValue)	
VB	Function MT_ Help_Grating_Circle_Steps_To_Real (ByVal ALineCount As Long,ByVal AValue As Long) As Long	
Delphi	function MT_ Help_Grating_Circle_Steps_To_Real(ALineCount:Integer;AValue:Integer):Double	
C#	public static extern int MT_ Help_Grating_Circle_Steps_To_Real(Int32 ALineCount,Int32 AValue)	
	ALineCount	光栅尺整圈的光栅线数，例如 64800 等
	AValue	脉冲数
函数返回	双精度浮点	转换后的物理量°,°/s,°/s²
	非 0	函数执行失败
备注		

3.19.13 MT_ Help_Encoder_Factor

功能描述：编码器闭环减速系数估算。

VC	float MT_ Help_Encoder_Factor(double AStepAngle,INT32 ADiv,INT32 ALineCount)	
VB	Function MT_ Help_Encoder_Factor(ByVal AStepAngle As Double,ByVal ADiv As Long,ByVal ALineCount As Long) As Single	
Delphi	function MT_ Help_Encoder_Factor(AStepAngle:Double;ADiv:Integer; ALineCount:Integer):Single	
C#	public static extern Single MT Help__Grating_Line_Steps_To_Real(double AUnit_um,Int32 AValue)	
输入	AStepAngle	走一步电机旋转角度，例如两项步进电机一般为 1.8°
	ADiv	驱动器细分数，一般为 2,4,8,16 等等
	ALineCount	编码器线数，例如 1000,1024 等
函数返回	单精度	减速系数
	非 0	函数执行失败
备注		

3.19.14 MT_ Help_Grating_Line_Factor

功能描述：光栅尺直线运动闭环减速系数估算。

VC	float MT_ Help_Grating_Line_Factor(double AStepAngle,INT32 ADiv,double APitch,double ALineRatio,double AUnit_um)	
VB	Function MT_ Help_Grating_Line_Factor (ByVal AStepAngle As Double,ByVal ADiv As Long,ByVal APitch As Double,ByVal ALineRatio As Double,ByVal AUnit_um As Double) As Single	
Delphi	function MT_ Help_Grating_Line_Factor(AStepAngle:Double;ADiv:Integer; APitch:Double;ALineRatio:Double;AUnit_um:Double):Single	
C#	public static extern Single MT_ Help_Grating_Line_Factor(double AStepAngle,Int32 ADiv,double APitch,double ALineRatio,double AUnit_um)	
输入	AStepAngle	走一步电机旋转角度，例如两项步进电机一般为 1.8°
	ADiv	驱动器细分数，一般为 2,4,8,16 等等
	APitch	直线台的螺距，即电机旋转一圈的位移，单位 mm
	ALineRatio	直线台的传动比，没有传动为 1
	AUnit_um	光栅尺的珊距，单位为 **um**
函数返回	单精度	减速系数
	非 0	函数执行失败
备注		

3.19.15 MT_Help_Grating_Circle_Factor

功能描述：光栅尺旋转运动闭环减速系数估算。

VC	float MT_ Help_Grating_Circle_Factor(double AStepAngle,INT32 ADiv,double ACircleRatio,INT32 ALineCount)

VB	Function MT_ Help_Grating_Circle_Factor (ByVal AStepAngle As Double,ByVal ADiv As Long,ByVal ACircleRatio As Double,ByVal ALineCount As Long) As Single	
Delphi	function MT_ Help_Grating_Circle_Factor(AStepAngle:Double;ADiv:Integer; ACircleRatio:Double;ALineCount:Integer):Single	
C#	public static extern Single MT_ Help_Grating_Circle_Factor(double AStepAngle,Int32 ADiv,double ACircleRatio,Int32 ALineCount)	
输入	AStepAngle	走一步电机旋转角度，例如两项步进电机一般为 1.8°
	ADiv	驱动器细分数，一般为 2,4,8,16 等等
	ACircleRatio	旋转台的传动比
	ALine_Count	光栅尺整圈的光栅数目
函数返回	单精度	减速系数
	非 0	函数执行失败
备注		