固高科技
GOOGOL TECHNOLOGY

扩展功能

运动控制器编程手册

多轴前瞻模块

R1.3

2020 年 1 月

© 1999-2022 固高科技 版权所有

版权申明

固高科技有限公司

保留所有权力

固高科技有限公司（以下简称固高科技）保留在不事先通知的情况下，修改本手册中的产品和产品规格等文件的权力。

固高科技不承担由于使用本手册或本产品不当，所造成直接的、间接的、特殊的、附带的或相应产生的损失或责任。

固高科技具有本产品及其软件的专利权、版权和其它知识产权。未经授权，不得直接或者间接地复制、制造、加工、使用本产品及其相关部分。

运动中的机器有危险！使用者有责任在机器中设计有效的出错处理和安全保护机制，固高科技没有义务或责任对由此造成的附带的或相应产生的损失负责。

联系我们

固高科技（深圳）有限公司

地址：深圳市高新技术产业园南区深港产学研基地西座二楼 W211 室

电话：0755-26970817　26737236　26970824

传真：0755-26970821

电子邮件：support@googoltech.com

网址：http://www.googoltech.com.cn

固高科技（香港）有限公司

地址：香港九龍觀塘偉業街 108 號絲寶國際大廈 10 樓 1008-09 室

電話：+(852) 2358-1033

傳真：+(852) 2719-8399

電子郵件：info@googoltech.com

網址：http://www.googoltech.com

臺灣固高科技股份有限公司

地址：台中市西屯區工業區三十二路 86 號 3 楼

電話：+886-4-23588245

傳真：+886-4-23586495

電子郵件：twinfo@googoltech.com

© 1999-2020 固高科技版权所有

文档版本

版本号	修订日期
1.0	2018 年 07 月 23 日
1.1	2019 年 07 月 08 日
1.2	2019 年 11 月 15 日
1.3	2020 年 01 月 05 日

© 1999-2020 固高科技版权所有

前言

感谢选用固高运动控制器

为回报客户，我们将以品质一流的运动控制器、完善的售后服务、高效的技术支持，帮助您建立自己的控制系统。

固高产品的更多信息

固高科技的网址是 http://www.googoltech.com.cn。在我们的网页上可以得到更多关于公司和产品的信息，包括：公司简介、产品介绍、技术支持、产品最新发布等等。

您也可以通过电话（0755－26970817）咨询关于公司和产品的更多信息。

技术支持和售后服务

您可以通过以下途径获得我们的技术支持和售后服务：

电子邮件：support@googoltech.com；
电话：0755－26970843
发函至：深圳市高新技术产业园南区园深港产学研基地西座二楼 W211 室
固高科技（深圳）有限公司
邮编：518057

编程手册的用途

用户通过阅读本手册，能够了解运动控制器的功能，掌握函数的用法，熟悉编程实现。最终，用户可以根据自己特定的控制系统，编制用户应用程序，实现控制要求。

编程手册的使用对象

本编程手册适用于具有C语言编程基础或Windows环境下使用动态链接库的基础，同时具有一定运动控制工作经验，对伺服或步进控制的基本结构有一定了解的工程开发人员。

编程手册的主要内容

本手册由二章内容组成，详细介绍了运动控制器的多轴前瞻模块功能。

相关文件

关于控制器的调试和安装，请参见随产品配套的运动控制器用户手册。
关于控制器的基本功能，请参见随产品配套的《运动控制器编程手册之基本功能》。
关于更复杂的控制器功能，请参见随产品配套的《运动控制器编程手册之高级功能》。
关于其他扩展模块的使用，请参见随产品配套的相关扩展模块编程手册。

通过固高科技公司网站可下载如驱动程序、dll 文件、例程、Demo 等相关文件，网址为：www.googoltech.com.cn/pro_view-3.html

© 1999-2020 固高科技版权所有

目录

© 1999-2020 固高科技版权所有

索引

1. 指令索引

2. 图片索引

© 1999-2020 固高科技版权所有

3. 表格索引

4. 例程索引

© 1999-2020 固高科技版权所有

一、多轴前瞻模块

本手册中所有字体为蓝色的指令（如 GT_SetupLookAheadCrd）均带有超级链接，点击可跳转至指令说明。

多轴前瞻模块是一个独立的功能模块，在连续轨迹领域能够根据轨迹线段的特点自动地对插补运动速度进行优化和限制，能够解决小线段轨迹下由于频繁加减速造成的机械振动。

1. 指令列表

表 1　多轴前瞻模块指令列表

指令列表		页码
GT_SetupLookAheadCrd	建立前瞻坐标系，设置机床类型	19
GT_SetFollowAxisParaLa	设置跟随轴参数	33
GT_SetVelDefineModeLa	设置速度定义模式	35
GT_SetAxisLimitModeLa	设置各个轴运动能力限制模式	32
GT_SetWorkLimitModeLa	设置工件坐标系限制模式	36
GT_SetAxisVelValidModeLa	设置合成速度对哪些轴生效	33
GT_InitLookAheadEx	初始化前瞻预处理模块的参数	27
GT_InitLookAheadPara	多轴前瞻模块初始化功能简化指令	27
GT_LnXYEx	缓存区指令，二维直线插补	29
GT_LnXYG0Ex	缓存区指令，二维直线插补，且终点速度始终为 0	30
GT_LnXYZEx	缓存区指令，三维直线插补	错误! 未定义书签。
GT_LnXYZG0Ex	缓存区指令，三维直线插补，且终点速度始终为 0	32
GT_LnXYZAEx	缓存区指令，四维直线插补	30
GT_LnXYZAG0Ex	缓存区指令，四维直线插补，且终点速度始终为 0	31
GT_ArcXYREx	缓存区指令，XY 平面圆弧插补，以终点位置和半径为输入参数	19
GT_ArcYZREx	缓存区指令，YZ 平面圆弧插补，以终点位置和半径为输入参数	21
GT_ArcZXREx	缓存区指令，ZX 平面圆弧插补，以终点位置和半径为输入参数	22
GT_ArcXYCEx	缓存区指令，XY 平面圆弧插补，以终点位置和圆心为输入参数	19
GT_ArcYZCEx	缓存区指令，YZ 平面圆弧插补，以终点位置和圆心为输入参数	21
GT_ArcZXCEx	缓存区指令，ZX 平面圆弧插补，以终点位置和圆心为输入参数	22
GT_CrdDataEx	缓存区指令，向插补缓冲区增加插补数据	26
GT_BufDelayEx	缓存区指令，缓存区内延时设置指令	24
GT_BufGearEx	缓存区指令，实现刀向跟随功能，启动某个轴跟随运动	24
GT_BufMoveEx	缓存区指令，实现刀向跟随功能，启动某个轴点位运动	25
GT_BufIOEx	缓存区指令，缓存区内数字量 IO 输出设置指令	25
GT_BufDAEx	缓存区指令，缓存区内输出 DA 值	23
GT_SetUserSegNumEx	缓存区指令，获取用户段号	35

© 1999-2020 固高科技版权所有

错误!未找到引用源。	缓存区指令，空间圆弧插补	20
GT_HelixXYRZEx	缓存区指令，XY 平面螺旋线插补，以终点位置和半径为输入参数	26
GT_HelixXYCZEx	缓存区指令，XY 平面螺旋线插补，以终点位置和圆心为输入参数	26

2. 重点说明

(1) 功能说明

下面用一个实例来说明前瞻机制优势。假设机床要加工一个长方形的零件，刀具所走的轨迹如图 1 图 1 使用前瞻与不使用前瞻的速度规划区别所示。假设 m 点到 n 点距离 3000 个单位长度，有 30 段规划。n 点到 p 点距离 2000 个单位长度，有 20 段规划。每段规划 100 个单位长度。

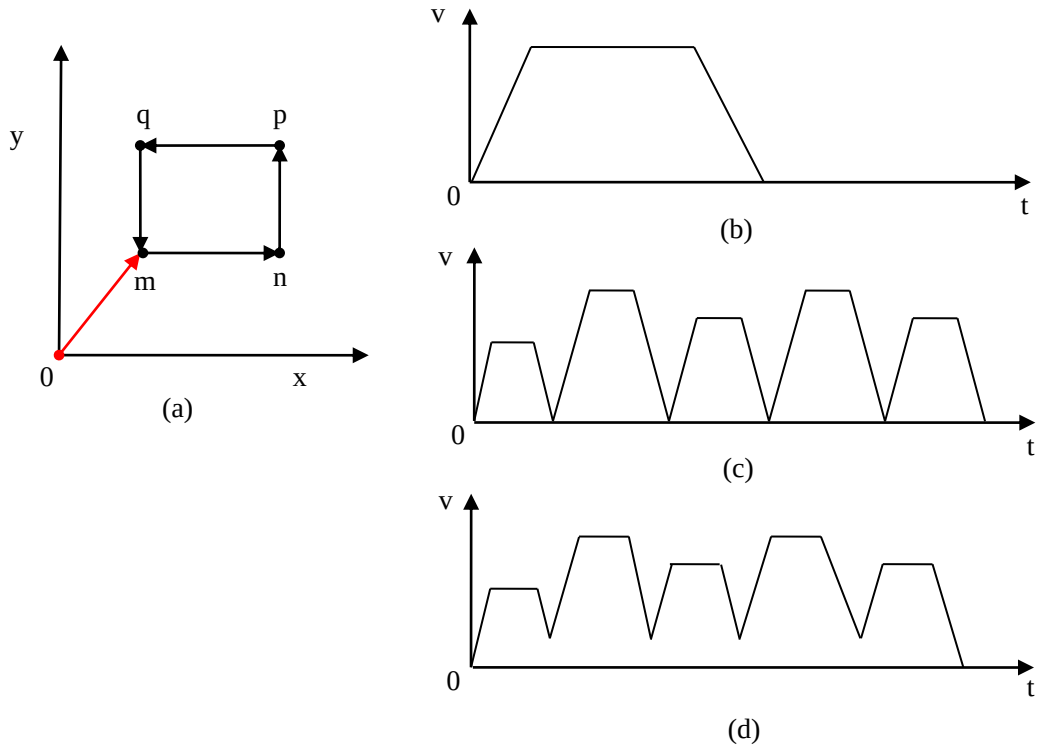

图 1 使用前瞻与不使用前瞻的速度规划区别

如果按照图 1(b)所示的速度规划，即在拐角处不减速，则加工精度一定会较低，而且可能在拐弯时对刀具和零件造成较大冲击。如果按照图 1(c)所示的速度规划，即在拐角处减速为 0，可以最大限度保证加工精度，但加工速度就会慢下来。如果按照图 1(d)所示的速度规划，在拐角处将速度减到一个合理值，既可以满足加工精度又能提高加工速度，就是一个好的速度规划。

为了实现类似图 1(d)所示的好的速度规划，前瞻模块不仅要知道当前运动的位置参数，还要提前知道后面若干段运动的位置参数，这就是所谓的前瞻。例如在对图 1 (a)中的轨迹做前瞻处理时，我们设定控制器预先读取 50 段运动轨迹到缓存区中，则它会自动分析出在第 30 段将会出现拐点，并依据用户设定的拐弯时间计算在拐弯处的终点速度。前瞻处理模块也会依照用户设定的最大加速度值计算速度规划，使任何加减速过程都不会超过这个值，防止对机械部分产生破坏性冲击力。

(2) 前瞻流程

运动缓存区：插补缓存区是运动控制器内部专门用于插补运动的缓存区资源，大小 4096 段，每一段可以放一条指令。

© 1999-2020 固高科技版权所有

当前瞻缓存区的段数不为 0 时，用户调用缓存区指令传递的插补数据先进入前瞻缓存区，当前瞻缓存区放满之后，如果再有新的数据传入，最先进入前瞻缓存区的数据，则会进入插补缓存区。

如果用户所有的插补数据已经输入完毕，前瞻缓存区中还有数据没有进入插补缓存区，这时，需要调用 GT_CrdDataEx(1, NULL, 0)，运动控制器会将前瞻缓存区的数据依次传递给插补缓存区，直到前瞻缓存区被清空为止。

在数据量比较大的时候，用户需要配合 GT_CrdSpace 指令查询插补缓存区的剩余空间，在有空间的时候再调用缓存区指令传递数据，如果插补缓存区已满，调用缓存区指令将会返回错误，说明该段插补数据没有输入成功，需要再次输入该段插补数据。

(3) 与前瞻预处理模块的区别

《运动控制器编程手册之基本功能》第 6 章的插补运动包含了对前瞻预处理模块的说明，此处的多轴前瞻模块（新前瞻）要实现的目的与前瞻预处理模块（老前瞻）是一致的。多轴前瞻模块（新前瞻）是对前瞻预处理模块（老前瞻）的扩展和优化，除了包含原有功能外，增加了：

1) 算法优化

2) 对插补轨迹曲率趋势的考虑

3) 单轴速度、加速度和速度跳变限制

4) 旋转轴速度限制

5) 3 轴以上的速度前瞻

与前瞻预处理模块（老前瞻）相比，多轴前瞻模块（新前瞻）的缓冲区指令使用的单位不再是脉冲而是毫米，并且指令带 Ex 后缀，例如二维直线插补：

- 前瞻预处理模块（老前瞻）：

```
GT_LnXY(1,          //插补坐标系号为 1
        1000,       //x 位置值 1000 脉冲
        2000,       //y 位置值 2000 脉冲
        100,        //插补合成速度 100 脉冲/ms
        1,          //插补合成加速度 1 脉冲/ms^2
        0,          //终点速度（使用前瞻此值无效，实际值有前瞻决定）
        0)          //插补缓冲区 FIFO 号
```

- 多轴前瞻模块（新前瞻）：

```
GT_LnXYEx(1,        //插补坐标系号为 1
          1,        //x 位置值为 1 毫米
          2,        //y 位置值为 2 毫米
          100,      //插补合成速度 100mm/s
          1000,     //插补合成加速度 1000mm/s^2
          0,        //此段的段号被标记为 0
          0,        //第二倍率为 0，不使用第二倍率
          0)        //插补缓冲区 FIFO 号
```


© 1999-2020 固高科技版权所有

鉴于多轴前瞻模块（新前瞻）和前瞻预处理模块（老前瞻）的差异，用户程序应避免两种混合。例如：不能既有 GT_LnXYEx 指令又有 GT_LnXY 指令。

(4) 动态库说明

多轴前瞻模块需要引入的动态库和头文件：

LookAheadEx.h
LAFunc.dll
PIFunc.dll
VFunc.dll

使用时应当把这些动态库和头文件与 gts.dll 放在同一目录下。

(5) 注意事项

标准 XYZ 三轴模式并不意味着必须建立三轴插补坐标系，它只是说明插补合成速度按照笛卡尔坐标系计算，用户完全可以建立满足笛卡尔坐标系的二轴、三轴、四轴插补。

连续调用缓冲区辅助运动指令问题：目前支持最多连续调用 16 条缓冲区辅助运动指令（例如：GT_BufIOEx，GT_BufGearEx……）。

起始点问题：新前瞻初始化时会构造一个起始点，因此可以直接调用缓冲区辅助运动指令（例如：GT_BufGearEx，GT_BufIOEx 等），而一旦调用了 GT_CrdDataEx 后，就不能直接调用缓冲区辅助运动指令，而应该先构造一个主运动指令作为起始点，再调用辅助运动指令：

```
GT_InitLookAheadEx;      //新前瞻初始化
GT_BufIOEx;              //缓冲区辅助运动指令
GT_LnXYEx;               //主运动指令
……
GT_CrdDataEx
```

错误用法：

```
GT_BufIOEx;              //（这条指令将不起作用，它被当作起始点）
GT_LnXYEx;
```

正确用法：

```
GT_LnXYEx;               //（构造起始点的指令）该指令不会压入缓冲区执行，目标位置 x 和
                         y 可以是上一段插补运动指令的终点位置，插补速度和加速度与上
                         一段一致即可
GT_BufIOEx;
GT_LnXYEx;
```

3. 例程

例程 1

标准 XYZ 三轴加工，无轴运动限制，无跟随轴。

© 1999-2020 固高科技版权所有

假设机床加工过程中，需要走一长直线，该直线由 300 条小直线段组成，现对这段路径进行前瞻预处理。其轨迹如图 2 所示。红色线段为起始轨迹。

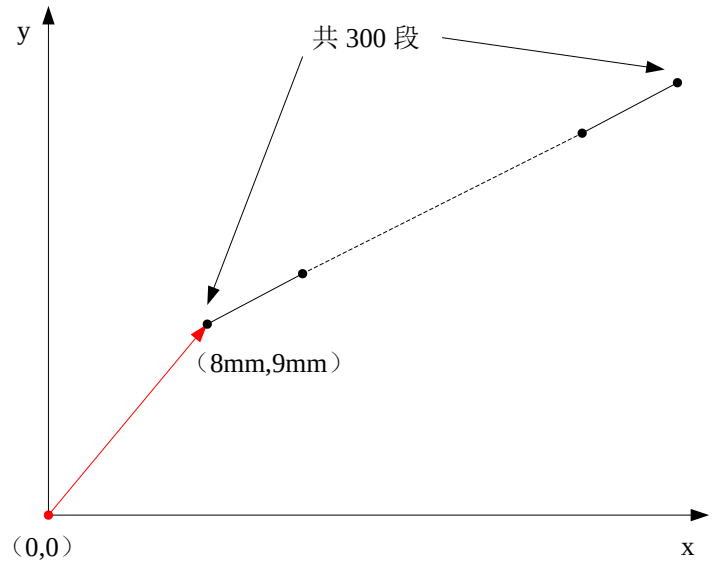

图 2 多轴前瞻模块例程 1

```
int i;
short sRtn;
short crd = 1, fifo = 0;

…… …… ……     // 初始化控制器

// 建立号坐标系，设置坐标系参数
TCrdPrm crdPrm;
memset(&crdPrm, 0, sizeof(crdPrm));
sRtn = GT_GetCrdPrm(crd,&crdPrm);
crdPrm.dimension = 2;              // 坐标系为二维坐标系
crdPrm.synVelMax = 500;                // 最大合成速度：pulse/ms
crdPrm.synAccMax = 1;              // 最大加速度：pulse/ms^2
crdPrm.evenTime = 50;              // 最小匀速时间：ms
crdPrm.profile[0] = 1;                 // 规划器对应到X轴
crdPrm.profile[1] = 2;                 // 规划器对应到Y轴
crdPrm.setOriginFlag = 0;              // 不需要指定坐标系的原点坐标的规划位置
crdPrm.originPos[0] = 0;               // 坐标系的原点坐标的规划位置为(, 0)
crdPrm.originPos[1] = 0;
sRtn = GT_SetCrdPrm(crd, &crdPrm);

// 初始化多轴前瞻模块
EMachineMode machineMode = NORMAL_THREE_AXIS;          // 机床类型
TLookAheadParameter lookAheadPara;                     // 前瞻参数
lookAheadPara.lookAheadNum = 200;                      // 前瞻缓冲区大小段
lookAheadPara.time = 0.1;                                  // 时间常数
lookAheadPara.radiusRatio = 0.01;                  //曲率参数
```


© 1999-2020 固高科技版权所有

```
for (i=0;i<8;++i)
{
    lookAheadPara.vMax[i] = 100;                //轴最大速度限制
    lookAheadPara.aMax[i] = 500;                //轴最大加速度限制
    lookAheadPara.DVMax[i] = 500;          //轴速度跳变限制
    lookAheadPara.axisRelation[i] = i+1;    //坐标系轴与前瞻轴一一映射
    lookAheadPara.scale[i] = 1000;              //脉冲当量
}

sRtn = GT_SetupLookAheadCrd(crd, machineMode);          //加载前瞻，设置机床模型
sRtn = GT_InitLookAheadEx(crd, & lookAheadPara, fifo, 0);    //初始化前瞻

double posTest[2];
double sycVel = 100;          //插补合成速度
double sycAcc = 800;          //插补合成加速度
long space;
long segNum = 0;              //用户段号

// 压插补数据：小线段加工
posTest[0] = 0;
posTest[1] = 0;
for(i=0;i<300;++i)
{
    sRtn = GT_LnXYEx(crd, 8+posTest[0], 9+posTest[1], sycVel, sycAcc, segNum, 0, fifo);
    // 查询返回值是否成功
    if(0 != sRtn)
    {
        do
        {
            // 查询运动缓存区空间，直至空间不为
            sRtn = GT_CrdSpace(crd, &space, 0);
        }while(0 == space);
        // 重新调用上次失败的插补指令
        sRtn = GT_LnXYEx(crd, 8+posTest[0], 9+posTest[1], sycVel, sycAcc, segNum, 0, fifo);
    }
    posTest[0] += 1.6;
    posTest[1] += 1.852;
}

// 将前瞻缓存区中的数据压入控制器
while(1)
{
    sRtn = GT_CrdDataEx(crd, NULL, fifo);
    if(!sRtn)
    {
        break;    //确认GT_CrdDataEx指令返回值为，表示所有数据都压入控制器
```


© 1999-2020 固高科技版权所有

```
    }
}
// 启动运动
sRtn = GT_CrdStart(1, 0);
…… …… ……
```

例程 2

标准 XYZ 三轴加工，轴运动限制生效，带跟随轴。

如下图 3 所示，需要给平面上一个矩形框点胶，每当转换到矩形框的下一个边框时，点胶头需要旋转 90 度，以保持点胶头复合喷胶方向。首先，从坐标系原点运动到（20mm，20mm）位置再开始矩形框的点胶，矩形长 30mm，宽 20mm，点胶头旋转 90 度等价的当量位移为 10mm。有两种方式可以实现上述动作：(1) 使用四维插补，A 轴作为旋转轴；(2) 使用二维插补，结合 BufGear 功能。

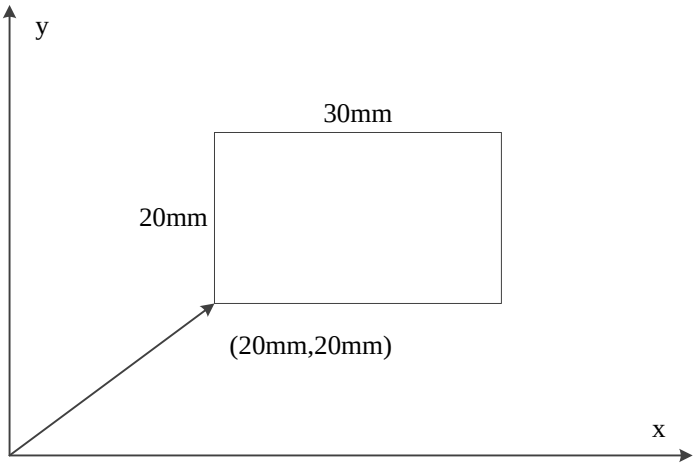

图 3 多轴前瞻模块例程 2

1) 四维插补 A 轴为旋转轴

```
int i;
short sRtn;
short crd = 1, fifo = 0;

…… …… ……     //初始化控制器

// 建立号坐标系，设置坐标系参数
TCrdPrm crdPrm;
memset(&crdPrm, 0, sizeof(crdPrm));
sRtn = GT_GetCrdPrm(crd,&crdPrm);
crdPrm.dimension = 4;              // 坐标系为二维坐标系
crdPrm.synVelMax = 500;                // 最大合成速度：pulse/ms
crdPrm.synAccMax = 1;              // 最大加速度：pulse/ms^2
crdPrm.evenTime = 50;              // 最小匀速时间：ms
crdPrm.profile[0] = 1;                 // 规划器对应到X轴
crdPrm.profile[1] = 2;                 // 规划器对应到Y轴
crdPrm.profile[2] = 3;                 // 规划器对应到Z轴
```


© 1999-2020 固高科技版权所有

```
    crdPrm.profile[3] = 4;                  // 规划器对应到A轴
    crdPrm.setOriginFlag = 0;               // 不需要指定坐标系的原点坐标的规划位置
    crdPrm.originPos[0] = 0;                // 坐标系的原点坐标的规划位置为（, 0）
    crdPrm.originPos[1] = 0;
    sRtn = GT_SetCrdPrm(crd, &crdPrm);

    //初始化多轴前瞻模块
    EMachineMode machineMode = NORMAL_THREE_AXIS;          //机床类型
    int axisLimitMode[8];                                  //轴限制模式
    for (i=0;i<8;i++)
    {
        axisLimitMode[i] = AXIS_LIMIT_NONE;
    }
    axisLimitMode[0] = AXIS_LIMIT_MAX_VEL | AXIS_LIMIT_MAX_ACC;//x轴限制速度和加速度
    axisLimitMode[1] = AXIS_LIMIT_MAX_VEL | AXIS_LIMIT_MAX_ACC;//y轴限制速度和加速度
    axisLimitMode[2] = AXIS_LIMIT_MAX_VEL | AXIS_LIMIT_MAX_ACC;//z轴限制速度和加速度
    axisLimitMode[3] = AXIS_LIMIT_MAX_VEL | AXIS_LIMIT_MAX_ACC |
AXIS_LIMIT_MAX_DV;    //a轴限制速度，加速度和速度跳变

    TLookAheadParameter lookAheadPara;        // 前瞻参数
    lookAheadPara.lookAheadNum = 200;         // 前瞻缓冲区大小段
    lookAheadPara.time = 0.1;                 // 时间常数
    lookAheadPara.radiusRatio = 0.01;         // 曲率参数
    for (i=0;i<8;++i)
    {
        lookAheadPara.vMax[i] = 100;          // 轴最大速度限制
        lookAheadPara.aMax[i] = 500;          // 轴最大加速度限制
        lookAheadPara.DVMax[i] = 500;         // 轴速度跳变限制
        lookAheadPara.axisRelation[i] = i+1;  // 坐标系轴与前瞻轴一一映射
        lookAheadPara.scale[i] = 1000;        // 脉冲当量
    }

    sRtn = GT_SetupLookAheadCrd(crd, machineMode);      // 加载前瞻，设置机床模型
    sRtn = GT_SetAxisLimitModeLa(crd, axisLimitMode);          // 设置轴限制
    sRtn = GT_SetAxisVelValidModeLa(crd, 0xF);       // 设置轴速度有效模式，按位设置，前个轴
                                                        均有效
    sRtn = GT_InitLookAheadEx(crd, & lookAheadPara, fifo, 0);    // 初始化前瞻

    double sycVel = 100;                // 补合成速度
    double sycAcc = 1000;         // 插补合成加速度
    long segNum = 0;                    // 用户段号

    // 压插补数据
    sRtn = GT_LnXYEx(crd, 20, 20, sycVel, sycAcc, segNum, 0, fifo);
    segNum++;
    sRtn = GT_LnXYZAEx(crd, 20, 40, 0, 10, sycVel, sycAcc, segNum, 0, fifo);
```


© 1999-2020 固高科技版权所有

```
segNum++;
sRtn = GT_LnXYZAEx(crd, 50, 40, 0, 20, sycVel, sycAcc, segNum, 0, fifo);
segNum++;
sRtn = GT_LnXYZAEx(crd, 50, 20, 0, 30, sycVel, sycAcc, segNum, 0, fifo);
segNum++;
sRtn = GT_LnXYZAEx(crd, 20, 20, 0, 40, sycVel, sycAcc, segNum, 0, fifo);

// 将前瞻缓存区中的数据压入控制器
while(1)
{
    sRtn = GT_CrdDataEx(crd, NULL, fifo);
     if(!sRtn)
    {
        break;    // 确认GT_CrdDataEx指令返回值为，表示所有数据都压入控制器
    }
}

// 启动运动
sRtn = GT_CrdStart(1, 0);
…… …… ……
```

2)　二维插补结合 BufGear

```
int i;
short sRtn;
short crd = 1, fifo = 0;

…… …… ……    // 初始化控制器

// 建立号坐标系，设置坐标系参数
TCrdPrm crdPrm;
memset(&crdPrm, 0, sizeof(crdPrm));
sRtn = GT_GetCrdPrm(crd,&crdPrm);
crdPrm.dimension = 2;            // 坐标系为二维坐标系
crdPrm.synVelMax = 500;              // 最大合成速度：pulse/ms
crdPrm.synAccMax = 1;            // 最大加速度：pulse/ms^2
crdPrm.evenTime = 50;            // 最小匀速时间：ms
crdPrm.profile[0] = 1;               // 规划器对应到X轴
crdPrm.profile[1] = 2;               // 规划器对应到Y轴
crdPrm.setOriginFlag = 0;            // 不需要指定坐标系的原点坐标的规划位置
crdPrm.originPos[0] = 0;             // 坐标系的原点坐标的规划位置为（, 0）
crdPrm.originPos[1] = 0;
sRtn = GT_SetCrdPrm(crd, &crdPrm);

//初始化多轴前瞻模块
EMachineMode machineMode = NORMAL_THREE_AXIS;          //机床类型
```


© 1999-2020 固高科技版权所有

```
    double vMax[8],aMax[8],dvMax[8];
    short axisLimitMode[8];                    //轴限制模式
    for (i=0;i<8;i++)
    {
        vMax[i] = 100;
        aMax[i] = 500;
        dvMax[i] = 500;
        axisLimitMode[i] = AXIS_LIMIT_NONE;
    }
    axisLimitMode[2] = AXIS_LIMIT_MAX_VEL | AXIS_LIMIT_MAX_ACC |
AXIS_LIMIT_MAX_DV;          //跟随轴限制速度，加速度和速度跳变

    TLookAheadParameter lookAheadPara;          //前瞻参数
    lookAheadPara.lookAheadNum = 200;          //前瞻缓冲区大小段
    lookAheadPara.time = 0.1;                        //时间常数
    lookAheadPara.radiusRatio = 0.01;                //曲率参数
    for (i=0;i<8;++i)
    {
        lookAheadPara.vMax[i] = 100;                //轴最大速度限制
        lookAheadPara.aMax[i] = 500;                //轴最大加速度限制
        lookAheadPara.DVMax[i] = 500;          //轴速度跳变限制
        lookAheadPara.axisRelation[i] = i+1;   //坐标系轴与前瞻轴一一映射
        lookAheadPara.scale[i] = 1000;              //脉冲当量
    }

    sRtn = GT_SetupLookAheadCrd(crd, machineMode);       //加载前瞻，设置机床模型
    sRtn = GT_SetFollowAxisParaLa(crd, axisLimitMode, vMax, aMax, dvMax);//设置跟随轴参数
    sRtn = GT_InitLookAheadEx(crd, & lookAheadPara, fifo, 0);    //初始化前瞻

    double sycVel = 100;                //插补合成速度
    double sycAcc = 1000;          //插补合成加速度
    long segNum = 0;                    //用户段号

    // 压插补数据
    sRtn = GT_LnXYEx(crd, 20, 20, sycVel, sycAcc, segNum, 0, fifo);
    segNum++;
    sRtn = GT_BufGearEx(crd, 3, 10, fifo);
    sRtn = GT_LnXYZAEx(crd, 20, 40, 0, 10, sycVel, sycAcc, segNum, 0, fifo);
    segNum++;
    sRtn = GT_BufGearEx(crd, 3, 10, fifo);
    sRtn = GT_LnXYZAEx(crd, 50, 40, 0, 20, sycVel, sycAcc, segNum, 0, fifo);
    segNum++;
    sRtn = GT_BufGearEx(crd, 3, 10, fifo);
    sRtn = GT_LnXYZAEx(crd, 50, 20, 0, 30, sycVel, sycAcc, segNum, 0, fifo);
    segNum++;
```


© 1999-2020 固高科技版权所有

```
sRtn = GT_BufGearEx(crd, 3, 10, fifo);
sRtn = GT_LnXYZAEx(crd, 20, 20, 0, 40, sycVel, sycAcc, segNum, 0, fifo);

// 将前瞻缓存区中的数据压入控制器
while(1)
{
    sRtn = GT_CrdDataEx(crd, NULL, fifo);
    if(!sRtn)
    {
        break;    //确认GT_CrdDataEx指令返回值为，表示所有数据都压入控制器
    }
}

// 启动运动
sRtn = GT_CrdStart(1, 0);
…… …… ……
```

例程 3

多轴前瞻模块兼容使用方式，为了让原来使用老前瞻的客户代码尽可能少变更，可以通过采用下面例子的方式在调用老前瞻插补指令的时候仍然可以达到使用多轴前瞻模块性能。必要的变更包括：（1）拷贝多轴前瞻库到 gts.dll 同一路径，且 gts.dll 支持该功能，（2）把老前瞻的前瞻初始化指令（GT_InitLookAhead）变更为多轴前瞻初始化简化指令（GT_InitLookAheadPara）。下面的例子是把 GTS 编程手册前瞻那部分的例子基础上把前瞻初始化指令替换为多轴前瞻模块初始化简化指令（加粗字体所示）。

```
int i;
short sRtn;
short crd = 1, fifo = 0;

…… …… ……    // 初始化控制器

// 建立号坐标系，设置坐标系参数
TCrdPrm crdPrm;
memset(&crdPrm, 0, sizeof(crdPrm));
sRtn = GT_GetCrdPrm(crd,&crdPrm);
crdPrm.dimension = 2;            // 坐标系为二维坐标系
crdPrm.synVelMax = 500;              // 最大合成速度：pulse/ms
crdPrm.synAccMax = 1;            // 最大加速度：pulse/ms^2
crdPrm.evenTime = 50;            // 最小匀速时间：ms
crdPrm.profile[0] = 1;               // 规划器对应到X轴
crdPrm.profile[1] = 2;               // 规划器对应到Y轴
crdPrm.setOriginFlag = 0;            // 不需要指定坐标系的原点坐标的规划位置
crdPrm.originPos[0] = 0;             // 坐标系的原点坐标的规划位置为(0, 0)
crdPrm.originPos[1] = 0;
sRtn = GT_SetCrdPrm(crd, &crdPrm);

// 初始化多轴前瞻模块
```


© 1999-2020 固高科技版权所有

```
sRtn = GT_InitLookAheadPara (crd, 200, 0.1, 0.01, 1000);

double posTest[2];
long space;
// 压插补数据：小线段加工
posTest[0] = 0;
posTest[1] = 0;
for(i=0;i<300;++i)
{
    sRtn = GT_LnXY(crd, 8000+posTest[0], 9000+posTest[1], 100, 0.8, 0, fifo);
    // 查询返回值是否成功
    if(0 != sRtn)
    {
        do
        {
            // 查询运动缓存区空间，直至空间不为
            sRtn = GT_CrdSpace(crd, &space, 0);
        }while(0 == space);
        // 重新调用上次失败的插补指令
        sRtn = GT_LnXYEx(crd, 8000+posTest[0], 9000+posTest[1], 100, 0.8, 0, fifo);
    }
    posTest[0] += 1600;
    posTest[1] += 1852;
}

// 将前瞻缓存区中的数据压入控制器
sRtn = GT_CrdData(1, NULL, 0);
// 启动运动
sRtn = GT_CrdStart(1, 0);
…… …… ……
```


© 1999-2020 固高科技版权所有

二、指令详细说明

指令 1　GT_ArcXYCEx

指令原型	short GT_ArcXYCEx(short crd,double x,double y,double xCenter,double yCenter,short circleDir,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	XY平面圆弧插补(以终点位置和圆心位置为输入参数)。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 11 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
x	插补段 x 轴终点坐标值，单位：mm。		
y	插补段 y 轴终点坐标值，单位：mm。		
xCenter	圆弧插补的圆心 x 方向相对于起点位置的偏移量，单位：mm。		
yCenter	圆弧插补的圆心 y 方向相对于起点位置的偏移量，单位：mm。		
circleDir	圆弧的旋转方向，0：顺时针圆弧，1：逆时针圆弧。		
synVel	插补段的目标合成速度，单位：mm/s。		
synAcc	插补段的目标合成加速度，单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。 五轴模式下不允许圆弧插补，若为圆弧插补段，返回 107。 圆弧参数不合理。平面圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm(当偏差处于 0.001mm 到 0.01mm 之间时会自动进行圆弧校正)或者弦长大于直径；若起点和终点相同或者弦长大于直径且偏差大于 0.01mm。空间圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm 或者弦长大于直径，或者圆弧为整圆或半圆，上述圆弧参数不合理情况均会返回 107。		
相关指令	无。		
指令示例	无。		

指令 2　GT_ArcXYREx

指令原型	short GT_ArcXYREx(short crd,double x,double y,double radius,short circleDir,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	XY 平面圆弧插补(以终点位置和半径为输入参数)。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 10 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
x	插补段 x 轴终点坐标值，单位：mm。		
y	插补段 y 轴终点坐标值，单位：mm。		
radius	圆弧插补的圆弧半径值，单位：mm。		

© 1999-2020 固高科技版权所有

circleDir	圆弧的旋转方向，0：顺时针圆弧，1：逆时针圆弧。
synVel	插补段的目标合成速度，单位：mm/s。
synAcc	插补段的目标合成加速度，单位：mm/s^2。
segNum	插补段段号标识。
override2	0：使用第一倍率；1：使用第二倍率。
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。 五轴模式下不允许圆弧插补，若为圆弧插补段，返回 107。 圆弧参数不合理。平面圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm(当偏差处于 0.001mm 到 0.01mm 之间时会自动进行圆弧校正)或者弦长大于直径；若起点和终点相同或者弦长大于直径且偏差大于 0.01mm。空间圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm 或者弦长大于直径，或者圆弧为整圆或半圆，上述圆弧参数不合理情况均会返回 107。
相关指令	无。
指令示例	无。

指令 3　GT_ArcXYZEx

指令原型	short GT_ArcXYZEx(short crd,double x, double y, double z,double interX,double interY,double interZ,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	空间圆弧插补，根据前一个点和该指令参数设置的终点和中间点，由三个点确定圆弧，并实现圆弧插补运动。		
指令类型	立即指令，调用后立即生效。	章节页码	7
指令参数	该指令共有 12 个参数，参数的详细信息如下。		
crd	插补坐标系，取值范围：[1,2]。		
x	圆弧终点坐标 x 值，单位：mm。		
y	圆弧终点坐标 y 值，单位：mm。		
z	圆弧终点坐标 z 值，单位：　mm。		
interX	圆弧经过的中间点 x 值，单位：mm。		
interY	圆弧经过的中间点 y 值，单位：mm。		
interZ	圆弧经过的中间点 z 值，单位：mm。		
synVel	目标合成速度，单位：mm/s。		
syneAcc	合成加速度，单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号。取值范围：[0, 1]，　默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。 五轴模式下不允许圆弧插补，若为圆弧插补段，返回 107。		

© 1999-2020 固高科技版权所有

	圆弧参数不合理。平面圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm(当偏差处于 0.001mm 到 0.01mm 之间时会自动进行圆弧校正)或者弦长大于直径；若起点和终点相同或者弦长大于直径且偏差大于 0.01mm。空间圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm 或者弦长大于直径，或者圆弧为整圆或半圆，上述圆弧参数不合理情况均会返回 107。
相关指令	无。
指令示例	无。

指令 4　GT_ArcYZCEx

指令原型	short GT_ArcYZCEx(short crd,double y,double z,double yCenter,double zCenter,short circleDir,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	YZ平面圆弧插补(以终点位置和圆心位置为输入参数)。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 11 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
y	插补段 y 轴终点坐标值，单位：mm。		
z	插补段 z 轴终点坐标值，单位：mm。		
yCenter	圆弧插补的圆心 y 方向相对于起点位置的偏移量，单位：mm。		
zCenter	圆弧插补的圆心 z 方向相对于起点位置的偏移量，单位：mm。		
circleDir	圆弧的旋转方向，0：顺时针圆弧，1：逆时针圆弧。		
synVel	插补段的目标合成速度，单位：mm/s。		
synAcc	插补段的目标合成加速度，单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。 五轴模式下不允许圆弧插补，若为圆弧插补段，返回 107。 圆弧参数不合理。平面圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm(当偏差处于 0.001mm 到 0.01mm 之间时会自动进行圆弧校正)或者弦长大于直径；若起点和终点相同或者弦长大于直径且偏差大于 0.01mm。空间圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm 或者弦长大于直径，或者圆弧为整圆或半圆，上述圆弧参数不合理情况均会返回 107。		
相关指令	无。		
指令示例	无。		

指令 5　GT_ArcYZREx

指令原型	short GT_ArcYZREx(short crd,double y,double z,double radius,short circleDir,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	YZ 平面圆弧插补(以终点位置和半径为输入参数)。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 10 个参数，参数的详细信息如下。		

© 1999-2020 固高科技版权所有

crd	坐标系号，取值范围：[1, 2]。
y	插补段 y 轴终点坐标值，单位：mm。
z	插补段 z 轴终点坐标值，单位：mm。
radius	圆弧插补的圆弧半径值，单位：mm。
circleDir	圆弧的旋转方向，0：顺时针圆弧，1：逆时针圆弧。
synVel	插补段的目标合成速度，单位：mm/s。
synAcc	插补段的目标合成加速度，单位：mm/s^2。
segNum	插补段段号标识。
override2	0：使用第一倍率；1：使用第二倍率。
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。 五轴模式下不允许圆弧插补，若为圆弧插补段，返回 107。 圆弧参数不合理。平面圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm(当偏差处于 0.001mm 到 0.01mm 之间时会自动进行圆弧校正)或者弦长大于直径；若起点和终点相同或者弦长大于直径且偏差大于 0.01mm。空间圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm 或者弦长大于直径，或者圆弧为整圆或半圆，上述圆弧参数不合理情况均会返回 107。
相关指令	无。
指令示例	无。

指令 6　GT_ArcZXCEx

指令原型	short GT_ArcZXCEx(short crd,double z,double x,double zCenter,double xCenter,short circleDir,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	ZX平面圆弧插补(以终点位置和圆心位置为输入参数)。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 11 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
z	插补段 z 轴终点坐标值，单位：mm。		
x	插补段 x 轴终点坐标值，单位：mm。		
zCenter	圆弧插补的圆心 z 方向相对于起点位置的偏移量，单位：mm。		
xCenter	圆弧插补的圆心 x 方向相对于起点位置的偏移量，单位：mm。		
circleDir	圆弧的旋转方向，0：顺时针圆弧，1：逆时针圆弧。		
synVel	插补段的目标合成速度，单位：mm/s。		
synAcc	插补段的目标合成加速度，单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。		

© 1999-2020 固高科技版权所有

	五轴模式下不允许圆弧插补，若为圆弧插补段，返回 107。 圆弧参数不合理。平面圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm(当偏差处于 0.001mm 到 0.01mm 之间时会自动进行圆弧校正)或者弦长大于直径；若起点和终点相同或者弦长大于直径且偏差大于 0.01mm。空间圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm 或者弦长大于直径，或者圆弧为整圆或半圆，上述圆弧参数不合理情况均会返回 107。
相关指令	无。
指令示例	无。

指令 7　GT_ArcZXREx

指令原型	short GT_ArcZXREx(short crd,double z,double x,double radius,short circleDir,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	YZ 平面圆弧插补(以终点位置和半径为输入参数)。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 10 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
z	插补段 z 轴终点坐标值，单位：mm。		
x	插补段 x 轴终点坐标值，单位：mm。		
radius	圆弧插补的圆弧半径值，单位：mm。		
circleDir	圆弧的旋转方向，0：顺时针圆弧，1：逆时针圆弧。		
synVel	插补段的目标合成速度，单位：mm/s。		
synAcc	插补段的目标合成加速度，单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。 五轴模式下不允许圆弧插补，若为圆弧插补段，返回 107。 圆弧参数不合理。平面圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm(当偏差处于 0.001mm 到 0.01mm 之间时会自动进行圆弧校正)或者弦长大于直径；若起点和终点相同或者弦长大于直径且偏差大于 0.01mm。空间圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm 或者弦长大于直径，或者圆弧为整圆或半圆，上述圆弧参数不合理情况均会返回 107。		
相关指令	无。		
指令示例	无。		

指令 8　GT_BufDAEx

指令原型	short GT_BufDAEx(short crd,short chn,short daValue,short fifo=0)		
指令说明	缓存区内输出 DA 值。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		

© 1999-2020 固高科技版权所有

chn	模拟量输出的通道号。取值范围：[1, 8]。
daValue	模拟量输出的值。取值范围：[-32768, 32767]，其中：-32768 对应-10V，32767 对应+10V。
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。
相关指令	无。
指令示例	无。

指令 9　GT_BufDelayEx

指令原型	short GT_BufDelayEx(short crd,unsigned short delayTime,short fifo=0)		
指令说明	缓存区内延时设置指令。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
delayTime	延时时间，单位：ms。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。		
相关指令	无。		
指令示例	无。		

指令 10　GT_BufGearEx

指令原型	short GT_BufGearEx(short crd,short gearAxis,double deltaPos,short fifo=0)		
指令说明	缓冲区实现刀向跟随功能，启动某个轴跟随运动。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
gearAxis	需要进行跟随运动的轴号，该轴不能处于坐标系中。		
deltaPos	跟随运动的位移量，单位：mm。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。		
相关指令	无。		
指令示例	**例程 2**		

© 1999-2020 固高科技版权所有

指令 11　GT_BufIOEx

指令原型	short GT_BufIOEx(short crd,unsigned short doType,unsigned short doMask,unsigned short doValue,short fifo=0)		
指令说明	缓存区内数字量 IO 输出设置指令。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 5 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
doType	数字量输出的类型： MC_ENABLE(该宏定义为 10)：输出驱动器使能。 MC_CLEAR(该宏定义为 11)：输出驱动器报警清除。 MC_GPO(该宏定义为 12)：输出通用输出。		
doMask	从 bit0~bit15 按位表示指定的数字量输出是否有操作。 0：该路数字量输出无操作。1：该路数字量输出有操作。		
doValue	从 bit0~bit15 按位表示指定的数字量输出的值。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。		
相关指令	无。		
指令示例	无。		

指令 12　GT_BufMoveEx

指令原型	short GT_BufMoveEx(short crd,short moveAxis,double pos,double vel,double acc,short modal,short fifo=0)		
指令说明	实现刀向跟随功能，启动某个轴点位运动。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 7 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
moveAxis	需要进行点位运动的轴号，该轴不能处于坐标系中。		
pos	点位运动的目标位置，单位：mm。		
vel	点位运动的目标速度，单位：mm/s。		
acc	点位运动的加速度，单位：mm/s^2。		
modal	点位运动的模式： 0：该指令为非模态指令，即不阻塞后续的插补缓存区指令的执行。 1：该指令为模态指令，将会阻塞后续的插补缓存区指令的执行。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。		
相关指令	无。		
指令示例	无。		

© 1999-2020 固高科技版权所有

指令 13　GT_CrdDataEx

指令原型	short GT_CrdDataEx(short crd,TCrdData *pCrdData,short fifo=0)		
指令说明	用于在使用前瞻时。调用该指令表示后续没有新的数据，将会一次性把前瞻缓存区的数据压入运动缓存区。		
指令类型	立即指令。	章节页码	7
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
pCrdData	只能设置为：NULL。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。 缓冲区空间不足或使用旧版本的多轴前瞻模块库，返回 1。		
相关指令	无。		
指令示例	**例程 1，例程 2**		

指令 14　GT_HelixXYCZEx

指令原型	short GT_HelixXYCZEx(short crd,double x,double y,double z,double xCenter, double yCenter,short circleDir,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	XY 方向做圆弧运动同时 Z 方向做直线运动的螺旋线插补，使用圆心描述投影圆。		
指令类型	立即指令，调用后立即生效。	章节页码	7
指令参数	该指令共有 12 个参数，参数的详细信息如下。		
crd	插补坐标系，取值范围：[1,2]。		
x	螺旋线插补 x 轴终端位置值，单位：mm。		
y	螺旋线插补 y 轴终端位置值，单位：mm。		
z	螺旋线插补 z 轴终端位置值，单位：mm。		
xCenter	圆弧插补的圆心 x 方向相对于起点位置的偏移量。		
yCenter	圆弧插补的圆心 y 方向相对于起点位置的偏移量。		
circleDir	圆弧的旋转方向。0：顺时针圆弧。1：逆时针圆弧。		
synVel	目标合成速度。取值范围：(0, 32767)， 单位：mm/s。		
syneAcc	合成加速度。取值范围：(0, 32767)， 单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号。 取值范围：[0, 1]， 默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。 五轴模式下不允许圆弧插补，若为圆弧插补段，返回 107。 圆弧参数不合理。平面圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm(当偏差处于 0.001mm 到 0.01mm 之间时会自动进行圆弧校正)或者弦长大		

© 1999-2020 固高科技版权所有

	于直径；若起点和终点相同或者弦长大于直径且偏差大于 0.01mm。空间圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm 或者弦长大于直径，或者圆弧为整圆或半圆，上述圆弧参数不合理情况均会返回 107。
相关指令	无。
指令示例	无。

指令 15　GT_HelixXYRZEx

指令原型	short GT_HelixXYRZEx(short crd,double x, double y, double z,double radius,short circleDir,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	XY 方向做圆弧运动同时 Z 方向做直线运动的螺旋线插补，使用半径描述投影圆弧运动。		
指令类型	立即指令，调用后立即生效。	章节页码	7
指令参数	该指令共有 11 个参数，参数的详细信息如下。		
crd	插补坐标系，取值范围：[1,2]。		
x	螺旋线插补 x 轴终端位置值，单位：mm。		
y	螺旋线插补 y 轴终端位置值，单位：mm。		
z	螺旋线插补 z 轴终端位置值，单位：mm。		
radius	圆弧插补的圆弧半径值，单位：mm。半径为正时，表示圆弧为小于等于 180 °圆弧。半径为负时，表示圆弧为大于 180 °圆弧。半径描述方式不能用来描述整圆。		
circleDir	圆弧的旋转方向。0：顺时针圆弧。1：逆时针圆弧。		
synVel	目标合成速度， 单位：mm/s。		
syneAcc	合成加速度， 单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号。取值范围：[0, 1]， 默认值为： 0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。 五轴模式下不允许圆弧插补，若为圆弧插补段，返回 107。 圆弧参数不合理。平面圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm(当偏差处于 0.001mm 到 0.01mm 之间时会自动进行圆弧校正)或者弦长大于直径；若起点和终点相同或者弦长大于直径且偏差大于 0.01mm。空间圆弧插补模式下，若起点和终点到圆心的距离偏差大于 0.01mm 或者弦长大于直径，或者圆弧为整圆或半圆，上述圆弧参数不合理情况均会返回 107。		
相关指令	无。		
指令示例	无。		

指令 16　GT_InitLookAheadPara

指令原型	short GT_InitLookAheadPara(short crd, long lookAheadNum, double time, double radiusRatio, double scale, short fifo)		
指令说明	多轴前瞻模块初始化简化指令。		
指令类型	立即指令，调用后立即生效。	章节页码	7
指令参数	该指令共有 6 个参数，参数的详细信息如下。		

© 1999-2020 固高科技版权所有

crd	坐标系号，取值范围：[1, 2]
lookAheadNum	前瞻缓冲区大小，取值范围：正整数，一般建议200
time	时间常数，通过调节时间常数可以调节尖角处的减速幅度，一般 设置为0.01，时间常数越小，插补合成终点速度降的越低，时间常数越大，插补合成终点速度越高。
radiusRatio	曲率限制调节参数，通过调节参数可以调节曲率的限制幅度，初始可设置成1，比率越大，允许速度越高。
scale	脉冲当量，长度为8的数组，单位:pulse/mm，但目前只有数组第一个元素有效，所有轴都参考数组第一个元素，数组的其它元素的值都应该与第一个元素相同。
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。
指令返回值	前瞻坐标系号不在范围内，返回 107 前瞻坐标系未建立，返回 103
相关指令	无。
指令示例	设置指定坐标系下工件坐标系下的几何信息限制是否生效。 对应建立前瞻坐标系时指定的机床类型，有默认的工件坐标系限制模式，如果不需要修改，则不需要调用该指令

指令 17　GT_InitLookAheadEx

指令原型	short GT_InitLookAheadEx(short crd, TLookAheadParameter *pLookAheadPara,short fifo=0,short motionMode=0, TPreStartPos *pPreStartPos=NULL)		
指令说明	初始化指定坐标系前瞻预处理模块的参数。		
指令类型	立即指令，调用后立即生效。	**章节页码**	7
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]		

plookAheadPara	前瞻参数结构体数据指针，前瞻参数结构体描述如下
	struct TLookAheadParameter
	{
	int lookAheadNum; //前瞻段数，建议设置200段
	double time; //时间常数
	double radiusRatio; //曲率限制调节参数
	double vMax[LA_AXIS_NUM]; //各轴的最大速度
	double aMax[LA_AXIS_NUM]; //各轴的最大加速度
	double DVMax[LA_AXIS_NUM]; //各轴的最大速度变化量
	double scale[LA_AXIS_NUM]; //各轴的脉冲当量
	short axisRelation[LA_AXIS_NUM];//设置输入坐标轴号和内部前瞻规划坐标轴号的对应关系
	char machineCfgFileName[128]; //机床配置文件名
	};
	参数具体说明：
	lookAheadNum：前瞻段数，建议设置200段。
	time：时间常数，通过调节时间常数可以调节尖角处的减速幅度，一般 设置为0.01，时间常数越小，插补合成终点速度降的越低，时间常数越大，插补合成终点速度越高。
	radiusRatio：曲率限制调节参数，通过调节参数可以调节曲率的限制幅度，		

© 1999-2020 固高科技版权所有

	初始可设置成1，比率越大，允许速度越高。 vMax，aMax，DVMax：长度为8的数组，分别描述插补轴的最大速度，最大加速度，最大速度跳变量限制，单位：mm/s,mm/s^2,mm/s^2。它们是否生效由GT_SetAxisLimitModeLa来设定。在合成速度的终点速度不为零时，参与合成运动的其中某一轴所允许的最大速度跳变量ΔVel为：ΔVel = DVMax * time。 scale：脉冲当量，长度为8的数组，单位:pulse/mm，但目前只有数组第一个元素有效，所有轴都参考数组第一个元素，数组的其它元素的值都应该与第一个元素相同。 axisRelation：坐标系轴对应关系描述数组，数组长度为8，描述坐标系输入轴号（XYZAUVW）对应的内部速度前瞻轴号，若不对应内部规划轴，则设为0，内部轴号为1~8，一般使用时一一映射即可。
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。
motionMode	运动模式，0-加工模式，插补指令发到控制器，执行常规插补运动，1-虚拟加工模式，插补指令不会发到控制器，可以通过此模式进行加工时间的预估（调用GT_GetMotionTimeEx）。
pPreStartPos	预估加工时间时所采用的起始位置
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 前瞻段数小于等于 0，返回 107。 轴对应关系设置出错，例如对应内部轴号大于等于 8，或者没有任何轴有对应内部轴，即对应内部轴号全部为 0，则返回 107。 三轴模式下，前三个轴不允许为跟随轴，即跟随轴被映射到内部轴号小于 3 的轴，返回 107。 五轴模式下，缺少五轴机床配置库，返回 105。 五轴模式下，机床配置文件路径错误，返回 106。
相关指令	无。
指令示例	**例程 1**

指令 18　GT_LnXYEx

指令原型	short GT_LnXYEx(short crd,double x,double y,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	三维直线插补。		
指令类型	缓冲区指令。	**章节页码**	7
指令参数	该指令共有 8 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
x	插补段x轴终点坐标值，单位：mm。		
y	插补段x轴终点坐标值，单位：mm。		
synVel	插补段的目标合成速度，单位：mm/s。		
synAcc	插补段的目标合成加速度，单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。		

© 1999-2020 固高科技版权所有

	前瞻坐标系号不在范围内，返回 107。 指令压入的坐标系前 3 个轴的位置点数据与前一条指令的对应位置点完全一样，返回 102。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。
相关指令	无。
指令示例	**例程 1**

指令 19　GT_LnXYG0Ex

指令原型	short GT_LnXYG0Ex(short crd,double x,double y,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	三维直线插补，且终点速度始终为 0。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 8 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
x	插补段x轴终点坐标值，单位：mm。		
y	插补段x轴终点坐标值，单位：mm。		
synVel	插补段的目标合成速度，单位：mm/s。		
synAcc	插补段的目标合成加速度，单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 指令压入的坐标系前 3 个轴的位置点数据与前一条指令的对应位置点完全一样，返回 102。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。		
相关指令	无。		
指令示例	无。		

指令 20　GT_LnXYZAEx

指令原型	short GT_LnXYZAEx(short crd,double x,double y,double z,double a,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	四维直线插补。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 10 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
x	插补段x轴终点坐标值，单位：mm。		
y	插补段x轴终点坐标值，单位：mm。		
z	插补段z轴终点坐标值，单位：mm。		
a	插补段a轴终点坐标值，单位：mm。		
synVel	插补段的目标合成速度，单位：mm/s。		
synAcc	插补段的目标合成加速度，单位：mm/s^2。		

© 1999-2020 固高科技版权所有

segNum	插补段段号标识。
override2	0：使用第一倍率；1：使用第二倍率。
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 指令压入的坐标系前 3 个轴的位置点数据与前一条指令的对应位置点完全一样，返回 102。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。
相关指令	无。
指令示例	**例程 2**

指令 21　GT_LnXYZAG0Ex

指令原型	short GT_LnXYZAG0Ex(short crd,double x,double y,double z,double a,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	四维直线插补，且终点速度始终为 0。		
指令类型	缓冲区指令。	**章节页码**	7
指令参数	该指令共有 10 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
x	插补段x轴终点坐标值，单位：mm。		
y	插补段x轴终点坐标值，单位：mm。		
z	插补段z轴终点坐标值，单位：mm。		
a	插补段a轴终点坐标值，单位：mm。		
synVel	插补段的目标合成速度，单位：mm/s。		
synAcc	插补段的目标合成加速度，单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 指令压入的坐标系前 3 个轴的位置点数据与前一条指令的对应位置点完全一样，返回 102。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。		
相关指令	无。		
指令示例	无。		

指令 22　GT_LnXYZEx

指令原型	short GT_LnXYZEx(short crd,double x,double y,double z,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	三维直线插补。		
指令类型	缓冲区指令。	**章节页码**	7
指令参数	该指令共有 9 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		

© 1999-2020 固高科技版权所有

x	插补段x轴终点坐标值，单位：mm。
y	插补段x轴终点坐标值，单位：mm。
z	插补段z轴终点坐标值，单位：mm。
synVel	插补段的目标合成速度，单位：mm/s。
synAcc	插补段的目标合成加速度，单位：mm/s^2。
segNum	插补段段号标识。
override2	0：使用第一倍率；1：使用第二倍率。
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 指令压入的坐标系前 3 个轴的位置点数据与前一条指令的对应位置点完全一样，返回 102。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。
相关指令	无。
指令示例	无。

指令 23　GT_LnXYZG0Ex

指令原型	short GT_LnXYZG0Ex(short crd,double x,double y,double z,double synVel,double synAcc,long segNum,short override2,short fifo=0)		
指令说明	三维直线插补，且终点速度始终为 0。		
指令类型	缓冲区指令。	章节页码	7
指令参数	该指令共有 9 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
x	插补段x轴终点坐标值，单位：mm。		
y	插补段x轴终点坐标值，单位：mm。		
z	插补段z轴终点坐标值，单位：mm。		
synVel	插补段的目标合成速度，单位：mm/s。		
synAcc	插补段的目标合成加速度，单位：mm/s^2。		
segNum	插补段段号标识。		
override2	0：使用第一倍率；1：使用第二倍率。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 指令压入的坐标系前 3 个轴的位置点数据与前一条指令的对应位置点完全一样，返回 102。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。		
相关指令	无。		
指令示例	无。		

指令 24　GT_SetAxisLimitModeLa

指令原型	short GT_SetAxisLimitModeLa(short crd,int *pAxisLimitMode)
指令说明	设置指定坐标系下各个轴运动能力的限制模式。

© 1999-2020 固高科技版权所有

	对应建立前瞻坐标系时指定的机床类型，有默认的轴限制模式，如果不需要修改，则不需要调用该指令。		
指令类型	立即指令，调用后立即生效。	章节页码	7
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
pAxisLimtMode	轴限制模式数组，数组长度为8，用于设置各个轴运动能力的限制模式。限制模式包括轴最大速度限制、最大加速度限制、最大速度跳变量限制，可通过位或操作得到每个轴需要进行限制的模式，如：需要对第1轴进行最大速度限制，则pAxisLimitMode[0] = AXIS_LIMIT_MAX_VEL；进行最大速度和最大速度变化量的限制，则pAxisLimitMode[0] = AXIS_LIMIT_MAX_VEL | AXIS_LIMIT_MAX_DV。 定义描述如下： #define AXIS_LIMIT_NONE 0 //轴无限制 #define AXIS_LIMIT_MAX_VEL 1 //轴最大速度限制 #define AXIS_LIMIT_MAX_ACC 2 //轴最大加速度限制 #define AXIS_LIMIT_MAX_DV 4 //轴最大速度跳变量限制		
指令返回值	前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。		
相关指令	无。		
指令示例	**例程 2**		

指令 25 GT_SetAxisVelValidModeLa

指令原型	short GT_SetAxisVelValidModeLa(short crd,int velValidAxis)		
指令说明	设置指定坐标系下，插补合成速度对于哪些轴生效。 默认为对坐标系8个轴都有效，如果不需要修改，则不需要调用该指令。		
指令类型	立即指令，调用后立即生效。	章节页码	7
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]		
velValidAxis	按位指定插补合成速度对于哪些轴生效。 Bit 7 6 5 4 3 2 1 0 对应轴或坐标系 8 轴 7 轴 6 轴 5 轴 4 轴 3 轴 2 轴 1 轴 例如：0xF表示设置的插补合成速度对坐标系前4个轴都有效，是前4个轴的速度的合成；而0x7表示设置的插补合成速度只对坐标系前3个轴有效。		
指令返回值	前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。		
相关指令	无。		
指令示例	**例程 2**		

指令 26 GT_SetFollowAxisParaLa

指令原型	short GT_SetFollowAxisParaLa (short crd,int *pAxisLimitMode, double *pVmax, double *pAmax, double *pDVmax)		
指令说明	设置指定坐标系下跟随轴（BufGear轴）的属性。		
指令类型	立即指令，调用后立即生效。	章节页码	7
指令参数	该指令共有 4 个参数，参数的详细信息如下。		

© 1999-2020 固高科技版权所有

crd	坐标系号，取值范围：[1, 2]
pAxisLimitMode	轴跟随属性数组，数组长度为8，对应1~8规划轴，用于标识某个规划轴是否为跟随轴；0：非跟随轴，1：跟随轴。
pVmax	速度限制数组，数组长度为8，对应1~8规划轴，单位：mm/s。
pAmax	加速度限制数组，数组长度为8，对应1~8规划轴，单位：mm/s^2。
pDVmax	速度跳变限制数组，数组长度为8，对应1~8规划轴，单位：mm/s^2。
指令返回值	前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 在非三轴模式下调用该指令，返回 107。 在三轴模式下，三轴联动插补轴不允许被设为跟随轴。
相关指令	GT_BufGearEx()
指令示例	设置指定坐标系下各个轴的跟随属性，默认没有跟随轴。 设置为跟随轴后，跟随轴的运动限制和约束会影响到合成轴。

指令 27　GT_SetupLookAheadCrd

指令原型	short GT_SetupLookAheadCrd(short crd,EMachineMode machineMode)		
指令说明	建立前瞻坐标系，并指定机床类型。 “前瞻坐标系”指的是多轴前瞻模块内部使用的坐标系，用户无需关心其具体实现及细节；“坐标系”指的是插补坐标系（即通过 GT_SetCrdPrm 设置的坐标系）		
指令类型	立即指令，调用后立即生效。	**章节页码**	7
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		

machineMode	机床类型，定义如下：
	enum EMachineMode
	{
	　　NORMAL_THREE_AXIS=0,　//标准三轴机床模式
	　　MULTI_AXES,　　//多轴联动模式
	　　FIVE_AXIS,　　//五轴机床模式,轴坐标系为主,工件坐标系为辅
	　　FIVE_AXIS_WORK,　　//五轴机床模式,工件坐标系为主,轴坐标系为辅
	};
	GTS通用版本只支持NORMAL_THREE_AXIS和MULTI_AXES，且MULTI_AXES模式下不超过4个轴。		

指令返回值	LAFun.dll 路径不对，加载不到动态库，返回 101。 平台校验出错，返回 110（例如，插补坐标系创建错误）。 前瞻坐标系号不在范围内，返回 107。		
相关指令	无。		
指令示例	用于建立前瞻坐标系，并指定坐标系内的机床类型。只有前瞻坐标系建立成功，才能进行后续的指令调用。否则，其他指令调用均会返回错误。 根据不同机床类型，会有默认的速度定义模式，轴限制模式，工件坐标系限制模式。 三轴模式下，默认速度定义为三轴合成速度，轴运动能力限制不生效，工件坐标系轨迹限制生效。 多轴模式下，默认速度定义为多轴合成速度，轴运动能力限制生效，工件坐标系限制不生效。		

© 1999-2020 固高科技版权所有

五轴模式下，默认速度定义为切削速度，轴运动能力限制生效，工件坐标系限制生效。

模式	默认速度定义	轴运动能力限制	工件坐标系轨迹限制
三轴	三轴合成速度	不生效	生效
多轴	多轴合成速度	生效	不生效
五轴	切削速度	生效	生效

指令 28　GT_SetUserSegNumEx

指令原型	short GT_ SetUserSegNumEx(short crd, long segNum, short fifo=0)		
指令说明	设置自定义插补段段号。		
指令类型	缓冲区指令。	**章节页码**	7
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		
segNum	设置用户自定义的插补段段号。		
fifo	插补缓存区号，取值范围：[0, 1]，默认值为：0。		
指令返回值	平台校验出错，返回 110。 前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 初始化前瞻函数没被调用或者调用失败时，调用该指令返回 104。		
相关指令	无。		
指令示例	无。		

指令 29　GT_SetVelDefineModeLa

指令原型	short GT_SetVelDefineModeLa(short crd,EVelSettingDef velDefMode)		
指令说明	设置坐标系输入速度的定义模式。		
指令类型	立即指令，调用后立即生效。	**章节页码**	7
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]。		

velDefMode	速度定义模式，定义如下：
	enum EVelSettingDef
	{
	NORMAL_DEF_VEL=0, //输入为轴坐标系所有轴的合成速度
	NUM_DEF_VEL, //以NUM系统的规则定义
	CUT_DEF_VEL, //速度为切削速度
	};		

指令返回值	前瞻坐标系号不在范围内，返回 107。 前瞻坐标系未建立，返回 103。 三轴模式下，设速度定义模式为 NUM_DEF_VEL，返回 107。 多轴模式下，设速度定义模式为 NUM_DEF_VEL 或者 CUT_DEF_VEL，返回 107。		
相关指令	无。		
指令示例	设置指定坐标系输入速度的定义模式。 对应建立前瞻坐标系时指定的机床类型，有默认的速度定义模式，如果不需要修改，则不需要调用该指令。		

© 1999-2020 固高科技版权所有

指令 30　GT_SetWorkLimitModeLa

指令原型	short GT_SetWorkLimitModeLa(short crd, EWorkLimitMode workLimitMode)		
指令说明	设置指定坐标系下工件坐标系限制模式。		
指令类型	立即指令，调用后立即生效。	**章节页码**	7
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
crd	坐标系号，取值范围：[1, 2]		

workLimitMode	工件坐标系限制模式，定义如下
	enum EWorkLimitMode
	{
	　　WORK_LIMIT_INVALID=0,　　//工件坐标系信息不限制
	　　WORK_LIMIT_VALID,　　//工件坐标系限制生效
	};		

指令返回值	前瞻坐标系号不在范围内，返回 107 前瞻坐标系未建立，返回 103		
相关指令	无。		
指令示例	设置指定坐标系下工件坐标系下的几何信息限制是否生效。 对应建立前瞻坐标系时指定的机床类型，有默认的工件坐标系限制模式，如果不需要修改，则不需要调用该指令		

© 1999-2020 固高科技版权所有