

插补运动控制 指令介绍





四、插补运动指令列表

二维插补运动需要使用的指令

| 指令 | 说明 |
|--------------|----------------------------------|
| GT_SetCrdPrm | 设置坐标系参数, 确立坐标系映射, 建立坐标系 |
| GT_LnXY | 缓存区指令, 两维直线插补 |
| GT_ArcXYR | 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) |
| GT_ArcXYC | 缓存区指令, XY平面圆弧插补(以终点位置和圆心位置为输入参数) |
| GT_BufIO | 缓存区指令, 缓存区内数字量IO输出设置指令 |
| GT_CrdSpace | 查询插补缓存区剩余空间 |
| GT_CrdClear | 清除插补缓存区内的插补数据 |
| GT_CrdStart | 启动插补运动 |
| GT_CrdStatus | 查询插补运动坐标系状态 |





四、插补运动指令列表

1. GT_SetCrdPrm

| | |
|------|--|
| 指令原型 | GT_SetCrdPrm(short crd, TCrdPrm *pCrdPrm) |
| 指令说明 | 设置坐标系参数，确立坐标系映射，建立坐标系。 |
| 指令类型 | 立即指令，调用后立即生效。 |
| 指令参数 | <p>该指令共有2个参数。crd，坐标系号，取值范围：[1, 2]。 pCrdPrm，设置坐标系的相关参数。</p> <pre>typedef struct CrdPrm { short dimension; short profile[8]; double synVelMax; double synAccMax; short evenTime; short setOriginFlag; long originPos[8]; }</pre> <p>TCrdPrm; dimension：坐标系的维数。取值范围：[1, 4]。 Profile[8]：坐标系与规划器的映射关系。 Profile[0..7]对应规划轴1~8，如果规划轴没有对应到该坐标系，则profile[x]的值为0；如果对应到了X轴，则profile[x]为1，Y轴对应为2，Z轴对应为3，A轴对应为4。不允许多个规划轴映射到相同坐标系的相同坐标轴，也不允许把相同规划轴对应到不同的坐标系，否则该指令将会返回错误值。每个元素的取值范围：[0, 4]。</p> |





四、插补运动指令列表

1. GT_SetCrdPrm

指令参数

synVelMax: 该坐标系的最大合成速度。如果用户在输入插补段的时候所设置的目标速度大于了该速度, 则将会被限制为该速度。取值范围: (0, 32767)。单位: pulse/ms。

synAccMax: 该坐标系的最大合成加速度。如果用户在输入插补段的时候所设置的加速度大于了该加速度, 则将会被限制为该加速度。取值范围: (0, 32767)。单位: pulse/ms²。

evenTime: 每个插补段的最小匀速段时间。取值范围: [0, 32767)。单位: ms。

setOriginFlag: 表示是否需要指定坐标系的原点坐标的规划位置, 该参数可以方便用户建立区别于机床坐标系的加工坐标系。0: 不需要指定原点坐标值, 则坐标系的原点在当前规划位置上。1: 需要指定原点坐标值, 坐标系的原点在originPos指定的规划位置上。

originPos[8]: 指定的坐标系原点的规划位置值。

指令返回值

若返回值为1:

(1) 若坐标系下各轴在规划运动, 请调用GT_Stop()停止运动再调用该指令。

(2) 请检查映射到Profile有没被激活, 若无, 则返回错误。

(3) 请见检查相应轴是否在坐标系下。其他返回值: 请参照指令返回值列表。





四、插补运动指令列表

2.GT_LnXY

| | |
|-------|---|
| 指令原型 | GT_LnXY (short crd, long x, long y, double synVel, double synAcc, double velEnd=0, short fifo=0) |
| 指令说明 | XY平面二维直线插补。 |
| 指令类型 | 缓存区指令。 |
| 指令参数 | <p>该指令共有7个参数。</p> <p>crd, 坐标系号。正整数, 取值范围: [1, 2]。</p> <p>x, 插补段x轴终点坐标值。取值范围: [-1073741823, 1073741823], 单位: pulse。</p> <p>y, 插补段y轴终点坐标值。取值范围: [-1073741823, 1073741823], 单位: pulse。</p> <p>synVel, 插补段的目标合成速度。取值范围: (0, 32767), 单位: pulse/ms。</p> <p>synAcc, 插补段的合成加速度。取值范围: (0, 32767), 单位: pulse/ms²。</p> <p>velEnd, 插补段的终点速度。取值范围: [0, 32767), 单位: pulse/ms。该值只有在没有使用前瞻预处理功能时才有意义, 否则该值无效。默认值为: 0。</p> <p>fifo, 插补缓存区号。取值范围: [0, 1]。默认值为: 0。</p> |
| 指令返回值 | <p>若返回值为1:</p> <ol style="list-style-type: none">(1) 检查当前坐标系是否映射了相关轴。(2) 检查是否向fifo1中传递数据, 若是, 则检查fifo0是否使用并运动, 若运动, 则返回错误。(3) 检查相应的fifo是否已满。 |





四、插补运动指令列表

3.GT_ArcXYC

| | |
|-------|---|
| 指令原型 | GT_ArcXYC(short crd, long x, long y, double xCenter, double yCenter, short circleDir, double synVel, double synAcc, double velEnd=0, short fifo=0) |
| 指令说明 | XY平面圆弧插补。使用圆心描述方法描述圆弧。 |
| 指令类型 | 缓存区指令。 |
| 指令参数 | <p>该指令共有10个参数。</p> <p>crd, 坐标系号。正整数, 取值范围: [1, 2]。</p> <p>x, 圆弧插补x轴的终点坐标值。取值范围: [-1073741823, 1073741823], 单位: pulse。</p> <p>y, 圆弧插补y轴的终点坐标值。取值范围: [-1073741823, 1073741823], 单位: pulse。</p> <p>xCenter, 圆弧插补的圆心x方向相对于起点位置的偏移量。</p> <p>yCenter, 圆弧插补的圆心y方向相对于起点位置的偏移量。</p> <p>circleDir, 圆弧的旋转方向。0: 顺时针圆弧。1: 逆时针圆弧。</p> <p>synVel, 插补段的目标合成速度。取值范围: (0, 32767), 单位: pulse/ms。</p> <p>synAcc, 插补段的合成加速度。取值范围: (0, 32767), 单位: pulse/ms²。</p> <p>velEnd, 插补段的终点速度。取值范围: [0, 32767), 单位: pulse/ms。该值只有在没有使用前瞻预处理功能时才有意义, 否则该值无效。默认值为: 0。</p> <p>fifo, 插补缓存区号。取值范围: [0, 1]。默认值为: 0。</p> <p>若返回值为1:</p> |
| 指令返回值 | <ol style="list-style-type: none">(1) 检查当前坐标系是否映射了相关轴。(2) 检查是否向fifo1中传递数据, 若是, 则检查fifo0是否使用并运动, 若运动, 则返回错误。(3) 检查相应的fifo是否已满。其他返回值: 请参照指令返回值列表 |





四、插补运动指令列表

4. GT_ArcXYR

| | |
|------|---|
| 指令原型 | GT_ArcXYR (short crd, long x, long y, double radius, short circleDir, double synVel, double synAcc, double velEnd=0, 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 | 缓存区指令。 |
| 指令参数 | <p>该指令共有9个参数。</p> <p>crd, 坐标系号。正整数, 取值范围: [1, 2]。</p> <p>x, 圆弧插补x轴的终点坐标值。取值范围: [-1073741823, 1073741823], 单位: pulse。</p> <p>y, 圆弧插补y轴的终点坐标值。取值范围: [-1073741823, 1073741823], 单位: pulse。</p> <p>radius, 圆弧插补的圆弧半径值。取值范围: [-1073741823, 1073741823], 单位: pulse。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。</p> <p>circleDir, 圆弧的旋转方向。0: 顺时针圆弧。1: 逆时针圆弧。</p> <p>synVel, 插补段的目标合成速度。取值范围: (0, 32767), 单位: pulse/ms。</p> |





四、插补运动指令列表

4. GT_ArcXYR

指令参数

synAcc, 插补段的合成加速度。取值范围: (0, 32767), 单位: pulse/ms²。

velEnd, 插补段的终点速度。取值范围: [0, 32767), 单位: pulse/ms。该值只有在没有使用前瞻预处理功能时才有意义, 否则该值无效。默认值为: 0

fifo, 插补缓存区号。正整数, 取值范围: [0, 1], 默认值为: 0。

指令返回值

若返回值为1:

(1) 检查当前坐标系是否映射了相关轴。

(2) 检查是否向fifo1中传递数据, 若是, 则检查fifo0是否使用并运动, 若运动, 则返回错误。

(3) 检查相应的fifo是否已满。其他返回值: 请参照指令返回值列表。

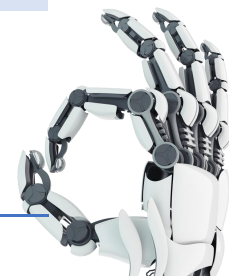




四、插补运动指令列表

5. GT_BufIO

| | |
|-------|--|
| 指令原型 | GT_BufIO (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 | 缓存区指令。 |
| 指令参数 | <p>该指令共有5个参数。</p> <p>crd, 坐标系号。正整数, 取值范围: [1, 2]。</p> <p>doType, 数字量输出的类型。</p> <p>MC_ENABLE(该宏定义为10): 输出驱动器使能。</p> <p>MC_CLEAR(该宏定义为11): 输出驱动器报警清除。</p> <p>MC_GPO(该宏定义为12): 输出通用输出。</p> <p>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。</p> <p>doValue, 从bit0~bit15按位表示指定的数字量输出的值。</p> <p>fifo, 插补缓存区号。正整数, 取值范围: [0, 1]。默认值为: 0。</p> <p>若返回值为1:</p> |
| 指令返回值 | <p>(1) 检查当前坐标系是否映射了相关轴。</p> <p>(2) 检查是否向fifo1中传递数据, 若是, 则检查fifo0是否使用并运动, 若运动, 则返回错误。</p> <p>(3) 检查相应的fifo是否已满。 其他返回值: 请参照指令返回值列表。</p> |





四、插补运动指令列表

6. GT_CrdSpace

| | |
|-------|---|
| 指令原型 | GT_CrdSpace (short crd, long *pSpace, short fifo=0) |
| 指令说明 | 查询插补缓存区剩余空间。 |
| 指令类型 | 立即指令，调用后立即生效。 |
| 指令参数 | 该指令共有3个参数。 crd ，坐标系号。正整数，取值范围：[1, 2]。 pSpace ，读取插补缓存区中的剩余空间。 fifo ，插补缓存区号。正整数，取值范围：[0, 1]。默认值为：0。 |
| 指令返回值 | 若返回值为1： 检查当前坐标系是否映射了相关轴。其他返回值：请参照指令返回值列表。 |





四、插补运动指令列表

7. GT_CrdClear

| | |
|-------|---|
| 指令原型 | GT_CrdClear(short crd, short fifo) |
| 指令说明 | 清除插补缓存区内的插补数据。 |
| 指令类型 | 立即指令，调用后立即生效。 |
| 指令参数 | 该指令共有2个参数。 crd ，坐标系号。正整数，取值范围：[1, 2]。 fifo ，插补缓存区号。正整数，取值范围：[0, 1]。默认值为：0。 |
| 指令返回值 | 若返回值为1： (1) 检查当前坐标系是否映射了相关轴。 (2) 检查是否向fifo1中传递数据，若是，则检查fifo0是否使用并运动，若运动，则返回错误。其他返回值：请参照指令返回值列表。 |





四、插补运动指令列表

8. GT_CrdStart

| | |
|-------|--|
| 指令原型 | GT_CrdStart (short mask, short option) |
| 指令说明 | 启动插补运动。 |
| 指令类型 | 立即指令，调用后立即生效。 |
| 指令参数 | <p>该指令共有2个参数。</p> <p>mask，从bit0~bit1按位表示需要启动的坐标系。 bit0对应坐标系1，bit1对应坐标系2。 0：不启动该坐标系，1：启动该坐标系。</p> <p>option，从bit0~bit1按位表示坐标系需要启动的缓存区的编号。 bit0对应坐标系1，bit1对应坐标系2。 0：启动坐标系中FIFO0的运动，1：启动坐标系中FIFO1的运动。</p> |
| 指令返回值 | <p>若返回值为1：</p> <ol style="list-style-type: none">(1) 检查当前坐标系是否映射了相关轴。(2) 若使用了辅助fifo1运动，检查当前坐标系位置没有恢复到fifo0断点坐标系位置。(3) 检查参数设置是否启动了坐标系。(4) 检查坐标系是否在运动。 其他返回值：请参照指令返回值列表。 |





四、插补运动指令列表

9. GT_CrdStatus

| | |
|-------|--|
| 指令原型 | GT_CrdStatus (short crd, short *pRun, long *pSegment, short fifo=0) |
| 指令说明 | 查询插补运动坐标系状态。 |
| 指令类型 | 立即指令，调用后立即生效。 |
| 指令参数 | <p>该指令共有4个参数。</p> <p>crd，坐标系号。正整数，取值范围：[1, 2]。</p> <p>pRun，读取插补运动状态。</p> <p>0：该坐标系的该FIFO没有在运动；1：该坐标系的该FIFO正在进行插补运动。</p> <p>pSegment，读取当前已经完成的插补段数。当重新建立坐标系或者调用GT_CrdClear指令后，该值会被清零。</p> <p>fifo，所要查询运动状态的插补缓存区号。正整数，取值范围：[0, 1]，默认值为：0。</p> |
| 指令返回值 | <p>若返回值为1：检查当前坐标系是否映射了相关轴。</p> <p>其他返回值：请参照指令返回值列表。</p> |



谢谢观看

