

Config 相关指令详细说明

指令 1 GT_SetAxisConfig

指令原型

指令说明

指令类型

指令参数

axis

pAxis

GT_API GT_SetAxisConfig(short axis,TAxisConfig *pAxis);

设置轴的配置参数。

立即指令，调用后立即生效。

该指令共有 2 个参数，参数的详细信息如下。

轴号，正整数，取值范围根据卡的不同取[1,4]或[1,8]。

设置轴状态参数，该参数为一个结构体，包含17个参数，详细的参数说明及定义如下：

#define MC_LIMIT_POSITIVE0//正限位

#define MC_LIMIT_NEGATIVE1//负限位

#define MC_ALARM2//驱动报警

#define MC_HOME3//原点

#define MC_GPI4//通用输入

#define MC_NONE-1//none

typedef struct AxisConfig

{

short active;//状态激活标志，默认等于1

short alarmType;//驱动轴报警标志位，相关宏定义如下：

short alarmIndex;//驱动报警对应的轴索引，取-1或[1-8]

如不需要报警，则取-1。

short limitPositiveType;//正限位标志位，宏定义同上alarmType，取0；

short limitPositiveIndex; //正限位报警对应的轴索引，取-1或[1，8]

如不需要报警，则取-1。

short limitNegativeType; //负限位标志位，宏定义同上alarmType，取1；

short limitNegativeIndex; //负限位报警对应的轴索引，取-1或[1，8]

如不需要报警，则取-1。

short smoothStopType;//平滑停止设置位，默认取4：通用输入

short smoothStopIndex;//平滑停止索引，取-1或[1，8]，如不需报警，

则取-1。

short abruptStopType;//急停设置位，默认取4：通用输入

short abruptStopIndex; //急停索引，取-1或[1，8]，如不需报警，则取-1。

long prfMap;//规划器设置，默认与轴号对应，16位数，取值示例：0x1。

long encMap; //编码器设置，默认与轴号对应，16位数，取值示例：0x1。

Bit	7	6	5	4	3	2	1	0
对应轴	8轴	7轴	6轴	5轴	4轴	3轴	2轴	1轴

按位指示轴号与规划器与编码器的映射关系，

	<p>[axis1]//轴1的相关参数配置示例</p> <p>prfMap=0x1//规划器映射设置，16位数（0000 0001）</p> <p>encMap=0x1//编码器映射设置，16位数</p> <p>0x1:将规划器1及编码器值1的值映射到轴1；</p> <p>prfMap=0x2//规划器映射设置，16位数（0000 0010）</p> <p>encMap=0x2//编码器映射设置，16位数</p> <p>0x2:将规划器2与编码器2的值映射到轴1</p> <p>prfMap=0x3//规划器映射设置，16位数（0000 0011）</p> <p>encMap=0x3//编码器映射设置，16位数</p> <p>0x3:将规划器1、2与编码器1、2的值都映射到轴1（轴1的实际规划位置为两规划器位置之和，编码器反馈值为两编码器反馈值之和）</p> <p>prfMap=0x4//规划器映射设置，16位数（0000 0100）</p> <p>encMap=0x4//编码器映射设置，16位数</p> <p>0x4:将规划器3与编码器3的值映射到轴1</p> <p>prfMap=0x5//规划器映射设置，16位数（0000 0101）</p> <p>encMap=0x5//编码器映射设置，16位数</p> <p>0x5:将规划器1、3与编码器值1、3的值都映射到轴1</p> <p>prfMap=0x6//规划器映射设置，16位数（0000 0110）</p> <p>encMap=0x6//编码器映射设置，16位数</p> <p>0x6:将规划器2、3与编码器值2、3的值都映射到轴1</p> <p>prfMap=0x7//规划器映射设置，16位数（0000 0111）</p> <p>encMap=0x7//编码器映射设置，16位数</p> <p>0x7:将规划器1、2、3与编码器值1、2、3的值都映射到轴1</p> <p>*****等*****等*****</p> <p>short prfMapAlpha[PRF_MAP_MAX];//规划器当量值设置，Alpha，默认值1。</p> <p>short prfMapBeta[PRF_MAP_MAX]; //规划器当量值设置，Beta，默认值1。</p> <p>short encMapAlpha[ENC_MAP_MAX];//编码器当量值设置，Alpha，默认值1。</p> <p>short encMapBeta[ENC_MAP_MAX]; //编码器当量值设置，Beta，默认值1。</p> <p>} TAxisConfig;</p>
指令返回值	若返回值为 1：请检查相应轴的配置是否正确。 其他返回值：请参照指令返回值列表。
相关指令	GT_SetAxisConfig(short axis,TAxisConfig *pAxis)
指令示例	<p>[axis1]//轴1的相关参数配置示例</p> <p>active=1</p> <p>alarmType=2</p> <p>alarmIndex=-1</p> <p>limitPositiveType=0</p> <p>limitPositiveIndex=-1</p> <p>limitNegativeType=1</p>

程序示例

```

limitNegativeIndex=-1
smoothStopType=4
smoothStopIndex=-1
abruptStopType=4
abruptStopIndex=-1
prfMap=0x1//规划器映射设置，16位数
encMap=0x1//编码器映射设置，16位数
prfMapAlpha1=1
prfMapBeta1=1
prfMapAlpha2=1
prfMapBeta2=1
encMapAlpha1=1
encMapBeta1=1
encMapAlpha2=1
encMapBeta2=1

```

备注：第一、二轴做插补运动，并一二轴的规划位置及编码器的反馈位置映射到第四轴，第三轴跟随第四轴的规划位置运动

```

// 省略建立坐标系及卡初始化等操作
long space;
//1、2轴插补运动
sRtn=GT_CrdClear(1,0);
sRtn=GT_LnXY(1,100000,50000,20,1,0,0);
//sRtn=GT_LnXY(1,0,0,10,1,0,0);
sRtn=GT_CrdSpace(1,&space,0);
sRtn=GT_CrdStart(1,0);
//修改配置文件，将1、2轴的轴规划位置，编码器位置分别映射到第四轴，
然后使用第三轴跟随第四轴的做Gear运动
sRtn=GT_PrftGear(3,0);//设置第三轴为Gear运动模式
sRtn=GT_SetGearMaster(3,4,3,0);//第三轴跟随第四轴的规划位置：0：
跟随规划，1：跟随编码器
sRtn=GT_SetGearRatio(3,5,1,10000);//跟随比例设置
sRtn=GT_GearStart(1<<(3-1));//启动第三轴位置跟随

轴[axis4]的相关参数配置
当sRtn=GT_SetGearMaster(3,4,3,0);//第三轴跟随第四轴的规划位置
此时设置第四轴：prfMap=0x3，即将1、2轴的规划位置映射到第四轴，
encMap=0x8，编码器的参数不变；
*****//其他参数不变
prfMap=0x3
encMap=0x8
*****//其他参数不变
当sRtn=GT_SetGearMaster(3,4,3,1);//第三轴跟随第四轴的编码器位置
此时设置第四轴：prfMap=0x8，
encMap=0x3，即将1、2轴的编码器的参数映射到第四轴

```

```
*****//其他参数不变
prfMap=0x8
encMap=0x3
*****//其他参数不变

另外：规划期当量及编码器当量变化设置
prfMapAlpha1=1
prfMapBeta1=1
prfMapAlpha2=1
prfMapBeta2=1
encMapAlpha1=1
encMapBeta1=1
encMapAlpha2=1
encMapBeta2=1
```

可通过修改规划期或者编码器的当量变换的值，对当前轴的规划值及编码器反馈值进行修改，具体关系参考GTS编程手册，
特别说明：当量变换只对当前设置轴起作用，对映射轴的规划及编码器值没影响，
使用：sRtn=GT_GetAxisPrfPos(Axis,&prfpos);
sRtn=GT_GetAxisEncPos(Axis,&encpos);读取对应轴的规划位置及实际位置

指令 2 GT_GetAxisConfig

指令原型	GT_API GT_GetAxisConfig(short axis,TAxisConfig *pAxis)		
指令说明	读取当前轴的配置参数		
指令类型	立即指令，调用后立即生效。	章节页码	
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
axis	轴号，正整数，取值范围根据卡的不同分为[1,4]或[1,8]。		
pAxis	设置轴状态参数，该参数为一个结构体，包含17个参数，定义参考GT_SetAxisConfig		
指令返回值	若返回值为 1：请检查相应轴的配置是否正确。 其他返回值：请参照指令返回值列表。		
相关指令	GT_GetAxisConfig		
指令示例	参考 GT_SetAxisConfig		

指令 3 GT_SetStepConfig

指令原型	GT_API GT_SetStepConfig(short step,TStepConfig *pStep)		
指令说明	轴通道脉冲输出方式设置		
指令类型	立即指令，调用后立即生效。	章节页码	
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
Step	Step索引，与控制轴对应，取值范围[1,8]		
pStep	设置脉冲输出的方式，该参数是一个结构体，包含 5 个参数，详细的参数说明及定义如下：		

	<pre>typedef struct StepConfig { short active;//状态激活标志位，默认设置为1； short axis;//轴号 short mode;//脉冲输出方式，0：脉冲+方向，1：CCW/CW short parameter;//无效参数 short reverse;//反转设置，默认取0：不反转 } TStepConfig;</pre>
指令返回值	若返回值为 1：检查脉冲输出参数是否正确 其他返回值：请参照指令返回值列表。
相关指令	GT_SetStepConfig
指令示例	<pre>[step1]//轴 1 对应的脉冲输出参数设置 active=1 axis=1 mode=0 parameter=0 reverse=0</pre>

指令 4 GT_GetStepConfig

指令原型	GT_API GT_GetStepConfig(short step,TStepConfig *pStep)		
指令说明	获取当前轴通道的脉冲输出方式		
指令类型	立即指令，调用后立即生效。	章节页码	
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
Step	Step索引，与控制轴号对应，取值范围[1,8]		
pStep	存储当前轴的脉冲输出方式，该参数是一个结构体，包含 5 个参数，参数定义请参考 GT_SetStepConfig		
指令返回值	若返回值为 1：检查脉冲输出参数是否正确 其他返回值：请参照指令返回值列表。		
相关指令	无。		
指令示例	参考 GT_SetStepConfig		

指令 5 GT_SetDacConfig

指令原型	GT_API GT_SetDacConfig(short dac,TDacConfig *pDac)		
指令说明	设置轴通道的模拟量输出相关参数		
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，参数的详细信息如下		
dac	模拟量输出通道选择，包括轴通道[1,8]和非轴通道。		
	通道模拟量输出参数设置，该参数是一个结构体，包含 5 个参数，参数的详细说明如下：		
pDac	typedef struct DacConfig { short active;//激活标志位，默认取 1； short control;//关联 control,取值-1 或[1,8],不关联则取-1，关联则取对应		

指令返回值	轴号（宏定义#define MC_NONE -1） short reverse;//输出电压反转设置位，0：正常，1：取反； short bias;//零漂补偿，根据实际零漂情况设置，默认取 0； short limit;//输出电压饱和极限设置，默认取 32767。 } TDacConfig;
	若返回值为 1：检查 DAC 输出参数是否正确 其他返回值：请参照指令返回值列表。
	GT_SetDacConfig
	[dac1]//轴通道 1 的 DAC 输出设置 active=1 control=-1 reverse=0 bias=0 limit=32767

指令 6 GT_GetDacConfig

指令原型	GT_API GT_GetDacConfig(short dac,TDacConfig *pDac)
指令说明	获取当前通道的DAC设置参数
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，参数的详细信息如下
dac	模拟量输出通道选择，包括轴通道[1,8]和非轴通道。
pDac	参考 GT_SetDacConfig 相关参数定义
指令返回值	若返回值为 1：检查 DAC 输出参数是否正确 其他返回值：请参照指令返回值列表。
相关指令	GT_GetDacConfig
指令示例	参考 GT_SetDacConfig

指令 7 GT_SetControlConfig

指令原型	GT_API GT_SetControlConfig(short control,TControlConfig *pControl);
指令说明	用于闭环控制的 control 参数设置
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，参数的详细信息如下
control	Control索引，取值与轴号对应
pControl	轴对应的 control 参数设置，该参数是一个结构体，包含 8 个参数，具体参数定义如下（#define MC_NONE -1） typedef struct ControlConfig { short active;//激活状态标志位，默认取 1； short axis;//关联轴号，不关联取-1，关联则取对应轴号[1,8] short encoder1;// 关联编码器 1 ，不关联取-1，关联则取对应轴号[1,8] short encoder2;// 关联编码器 2 ，不关联取-1，关联则取对应轴号[1,8] long errorLimit;//跟随误差极限，默认取 32767 short filterType[3];// PID 算法类型，暂时不起作用

指令返回值	short encoderSmooth;//反馈平滑系数，暂时不起作用，默认取 8 short controlSmooth; //控制器平滑系数，暂时不起作用，默认取 8 } TControlConfig;
	若返回值为 1：检查 control 参数是否正确 其他返回值：请参照指令返回值列表。
相关指令	GT_SetControlConfig
指令示例	[control1]/轴通道不关联参数设置 active=0 axis=-1 encoder1=-1 encoder2=-1 errorLimit=32767 filterType1=0 filterType2=0 filterType3=0 encoderSmooth=8 controlSmooth=8

指令 8 GT_GetControlConfig

指令原型	GT_API GT_GetControlConfig(short control,TControlConfig *pControl);
指令说明	用于获取闭环控制的 control 参数设置
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，参数的详细信息如下
control	Control索引，取值与轴号对应
pControl	存储轴对应的 control 参数设置，相关定义参考 GT_SetControlConfig
指令返回值	若返回值为 1：检查 control 参数是否正确 其他返回值：请参照指令返回值列表。
相关指令	GT_GetControlConfig
指令示例	

指令 9 GT_SetCountConfig

指令原型	GT_SetCountConfig(short countType,short countIndex,TCountConfig *pCount);
指令说明	设置编码器相关配置参数
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，参数的详细信息如下
countType	MC_ENCODER (#define MC_ENCODER 23)
countIndex	encoder 索引，包括轴通道[1,8]和非轴通道
pCount	typedef struct CountConfig { short active;//状态激活标志位，默认为 1 short reverse;//输入脉冲取反设置，0：正常，1：取反 short filterType;// 编码器滤波算法：

	<p>0: 保留上次的值, 用上次增量估算本次计数值 1: 抛弃上次的值, 增量清零</p> <p>short captureSource;//脉冲记数源, 0: 外部编码器, 1: 脉冲计数器 short captureHomeSense;//Home 捕获触发沿, 0: 下降沿, 1: 上升沿 short captureIndexSense;//index 捕获触发沿, 0: 下降沿, 1: 上升沿 } TCountConfig;</p>
指令返回值	若返回值为 1: 检查 count 参数是否设置正确 其他返回值: 请参照指令返回值列表。
相关指令	
指令示例	<pre>[encoder1]//encoder1 相关参数设置 active=1 reverse=1 filterType=0 captureSource=0 captureHomeSense=0 captureIndexSense=0</pre>

指令 10 GT_GetCountConfig

指令原型	GT_GetCountConfig(short countType,short countIndex,TCountConfig *pCount);
指令说明	获取编码器相关配置参数
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, 参数的详细信息如下
countType	
countIndex	encoder 索引, 包括轴通道[1,8]和非轴通道
pControl	相关定义参考 GT_SetControlConfig
指令返回值	若返回值为 1: 检查 count 参数是否设置正确 其他返回值: 请参照指令返回值列表。
相关指令	GT_SetCountConfig
指令示例	

指令 11 GT_SetProfileConfig

指令原型	GT_API GT_SetProfileConfig(short profile,TProfileConfig *pProfile);
指令说明	规划器配置
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, 参数的详细信息如下
profile	Profile索引, 与轴号对应, 取值范围[1,8]
pProfile	<p>Profile 配置文件, 该参数为结构体, 包含 3 个参数, 参数的详细说明如下</p> <pre>typedef struct ProfileConfig { short active;//状态激活标志位, 默认取 1; double decSmoothStop;//平滑停止减速度; double decAbruptStop;//急停减速度</pre>

	} TProfileConfig;
指令返回值	若返回值为 1：检查 profile 参数是否设置正确 其他返回值：请参照指令返回值列表。
相关指令	
指令示例	[profile1] active=1 decSmoothStop=1.000000 decAbruptStop=1000.000000

指令 12 GT_GetProfileConfig

指令原型	GT_API GT_GetProfileConfig(short profile,TProfileConfig *pProfile);
指令说明	读取当前规划器配置
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，参数的详细信息如下
profile	Profile索引，与轴号对应，取值范围[1,8]
pProfile	Profile 配置文件，该参数为结构体，包含 3 个参数，参数的详细说明如下 typedef struct ProfileConfig { short active;//状态激活标志位，默认取 1; double decSmoothStop;//平滑停止减速度; double decAbruptStop;//急停减速度 } TProfileConfig;
指令返回值	若返回值为 1：检查 profile 参数是否设置正确 其他返回值：请参照指令返回值列表。
相关指令	
指令示例	参考 GT_GetProfileConfig 示例

指令 13 GT_SetDiConfig

指令原型	GT_API GT_SetDiConfig(short diType,short diIndex,TDiConfig *pDi);
指令说明	设置 Di 输出的类型
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，参数的详细信息如下
diType	Di 输入的信号类型,相关宏定义如下: #define MC_LIMIT_POSITIVE 0//正限位 #define MC_LIMIT_NEGATIVE 1//负限位 #define MC_ALARM 2//驱动报警 #define MC_HOME 3//原点 #define MC_GPI 4//通用输入 #define MC_ARRIVE 5//电机到位 #define MC_MPG 6//手轮
diIndex	Di 索引,与轴号对应
pDi	typedef struct DiConfig

	<pre> { short active;//状态激活标志位，默认值 1; short reverse;//输入取反，0：正常，1：取反 short filterTime;//滤波时间 } TDiConfig; </pre>
指令返回值	若返回值为 1：检查 Di 参数是否正确 其他返回值：请参照指令返回值列表。
相关指令	
指令示例	<pre> [gpi1] active=1 reverse=0 filterTime=3 </pre>

指令 14 GT_GetDiConfig

指令原型	GT_API GT_GetDiConfig(short diType,short diIndex,TDiConfig *pDi);														
指令说明	获取当前的 Di 设置参数														
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，参数的详细信息如下														
diType	Di 输入的信号类型： <table> <tr><td>#define MC_LIMIT_POSITIVE</td><td>0//正限位</td></tr> <tr><td>#define MC_LIMIT_NEGATIVE</td><td>1//负限位</td></tr> <tr><td>#define MC_ALARM</td><td>2//驱动报警</td></tr> <tr><td>#define MC_HOME</td><td>3//原点</td></tr> <tr><td>#define MC_GPI</td><td>4//通用输入</td></tr> <tr><td>#define MC_ARRIVE</td><td>5//电机到位</td></tr> <tr><td>#define MC_MPG</td><td>6//手轮</td></tr> </table>	#define MC_LIMIT_POSITIVE	0//正限位	#define MC_LIMIT_NEGATIVE	1//负限位	#define MC_ALARM	2//驱动报警	#define MC_HOME	3//原点	#define MC_GPI	4//通用输入	#define MC_ARRIVE	5//电机到位	#define MC_MPG	6//手轮
#define MC_LIMIT_POSITIVE	0//正限位														
#define MC_LIMIT_NEGATIVE	1//负限位														
#define MC_ALARM	2//驱动报警														
#define MC_HOME	3//原点														
#define MC_GPI	4//通用输入														
#define MC_ARRIVE	5//电机到位														
#define MC_MPG	6//手轮														
diIndex	Di 索引,与轴号对应														
pDi	<pre> typedef struct DiConfig { short active;//状态激活标志位，默认值 1; short reverse;//输入取反，0：正常，1：取反 short filterTime;//滤波时间 } TDiConfig; </pre>														
指令返回值	若返回值为 1：检查 Di 参数是否正确 其他返回值：请参照指令返回值列表。														
相关指令															
指令示例	参考 GT_SetDiConfig 示例														

指令 15 GT_SetDoConfig

指令原型	GT_API GT_SetDoConfig(short doType,short doIndex,TDoConfig *pDo);
指令说明	设置 Do 输出的类型
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，参数的详细信息如下

doType	Do 输出的类型
	<pre>#define MC_ENABLE 10//伺服使能 #define MC_CLEAR 11//清除报警 #define MC_GPO 12//通用输出</pre>
doIndex	Do 索引，与轴号对应，取值[1,8]
pDo	<pre>typedef struct DoConfig { short active;//状态激活标志位，默认取 1; short axis;//关联 axis,不关联取-1，关联取与轴号对应的值 short axisItem; //暂时无用 short reverse;//输出取反，0：正常，1：取反 } TDoConfig;</pre>
指令返回值	若返回值为 1：检查 Do 参数是否设置正确 其他返回值：请参照指令返回值列表。
相关指令	
指令示例	<pre>[gpo1] active=1 axis=-1 axisItem=-1 reverse=1</pre>

指令 16 GT_GetDoConfig

指令原型	GT_API GT_GetDoConfig(short doType,short doIndex,TDoConfig *pDo);
指令说明	读取当前的 Do 输出配置
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，参数的详细信息如下
doType	Do 输出的类型
	<pre>#define MC_ENABLE 10//伺服使能 #define MC_CLEAR 11//清除报警 #define MC_GPO 12//通用输出</pre>
doIndex	Do 索引，与轴号对应，取值[1,8]
pDo	<pre>typedef struct DoConfig { short active;//状态激活标志位，默认取 1; short axis;//关联 axis,不关联取-1，关联取与轴号对应的值 short axisItem; //暂时无用，默认值取-1 short reverse;//输出取反，0：正常，1：取反 } TDoConfig;</pre>
指令返回值	若返回值为 1：检查 Do 参数是否设置正确 其他返回值：请参照指令返回值列表。
指令示例	参考 GT_SetDoConfig 示例