

# 运动控制器编程手册

---

## 二维位置比较输出

### R1.2

2019 年 7 月

© 1999-2019 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](mailto: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](mailto: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 樓

電話：+886-4-23588245

傳真：+886-4-23586495

電子郵件：[twinfo@googoltech.com](mailto:twinfo@googoltech.com)

# 文档版本

版本号	修订日期
1.0	2015 年 11 月 30 日
1.1	2018 年 07 月 23 日
1.2	2019 年 07 月 08 日

# 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](mailto: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

本手册由六章内容组成，详细介绍了运动控制器的二维位置比较输出功能及编程实现。

## 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》。

关于扩展模块的使用，请参见随产品配套的扩展模块编程手册。



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](http://www.googoltech.com.cn/pro_view-3.html)

# 目录

前言 .....	3
目录 .....	4
索引 .....	5
1. 指令索引 .....	5
2. 图片索引 .....	5
3. 表格索引 .....	5
4. 例程索引 .....	5
一、原理 .....	6
二、指令列表 .....	6
三、重点说明 .....	6
四、例程 .....	7
例程 1 输出指定电平或脉冲 .....	7
例程 2 设置位置比较输出数据 .....	7
例程 3 设置位置比较输出口 .....	8
例程 4 二维位置比较的一维模式 .....	8
例程 5 二维位置比较 FIFO 切换 .....	9
五、常见问题及检查 .....	10
1. 设置二维位置比较参数出错 .....	10
2. 压入比较数据出错 .....	10
3. 指令无返回错误，位置比较没有输出 .....	10
六、指令详细说明 .....	11

# 索引

## 1. 指令索引

指令 1	GT_2DCompareClear .....	11
指令 2	GT_2DCompareClearData .....	11
指令 3	GT_2DCompareData .....	11
指令 4	GT_2DCompareMode .....	12
指令 5	GT_2DComparePulse .....	12
指令 6	GT_2DCompareSetPrm .....	12
指令 7	GT_2DCompareStart .....	13
指令 8	GT_2DCompareStatus .....	13
指令 9	GT_2DCompareStop .....	13
指令 10	GT_SetComparePort .....	14

## 2. 图片索引

图 1	二维位置比较输出原理示意图 1 .....	6
-----	-----------------------	---

## 3. 表格索引

表 1	二维位置比较输出指令列表 .....	6
-----	--------------------	---

## 4. 例程索引

例程 1	输出指定电平或脉冲 .....	7
例程 2	设置位置比较输出数据 .....	7
例程 3	设置位置比较输出口 .....	8
例程 4	二维位置比较的一维模式 .....	8
例程 5	二维位置比较 FIFO 切换 .....	9

# 一、原理

设  $T(T_x, T_y)$  为平面上的目标点， $M(M_x, M_y)$  为最大允许位置误差（误差区）。如图 1 所示，当系统运行至蓝色区域，则认为已经进入目标区域。进入目标区后，控制器会通过寻找最优算法找到最接近  $T$  点，并输出 IO 控制。

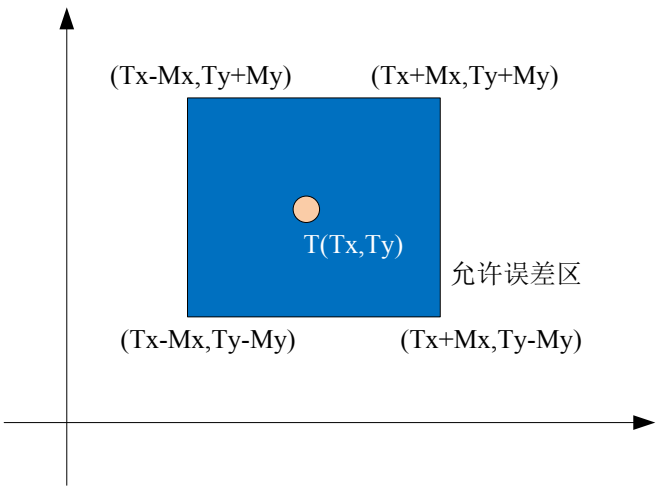


图 1 二维位置比较输出原理示意图 1

# 二、指令列表



提示

本手册中所有字体为蓝色的指令（如 [GT\\_2DCompareMode](#)）均带有超级链接，点击可跳转至指令说明。

表 1 二维位置比较输出指令列表

指令	说明	页码
<a href="#">GT_2DCompareMode</a>	设置为 2 维位置比较输出模式	12
<a href="#">GT_2DComparePulse</a>	输出指定电平或脉冲	12
<a href="#">GT_2DCompareSetPrm</a>	设置二维位置比较输出参数	12
<a href="#">GT_2DCompareData</a>	设置位置数据	11
<a href="#">GT_2DCompareClear</a>	复位二维位置比较模块，清空位置比较数据	11
<a href="#">GT_2DCompareClearData</a>	清空位置比较数据	11
<a href="#">GT_2DCompareStatus</a>	查询位置比较状态	13
<a href="#">GT_2DCompareStart</a>	启动位置比较输出	13
<a href="#">GT_2DCompareStop</a>	停止位置比较输出	13
<a href="#">GT_SetComparePort</a>	设置位置比较输出接口	14

# 三、重点说明

运动控制器有 2 个二维位置比较模块，可用于将编码器位置或内部脉冲计数器位置与设定位置比较，2 个二维位置比较模块可以挂接不同的比较源（编码器或脉冲计数器）。当编码器位置或内部

脉冲计数器位置到达设定位置时，在高速 IO（HSIO）口输出脉冲或反转电平。

通过 `GT_2DCompareData` 指令以数组方式向控制器传递比较数据，这些比较数据为相对位置，即相对于调用 `GT_2DCompareStart` 时控制器位置的偏移量。支持非单调方向数据输入，但用户需保证位置数组中的数据在运动过程中均能达到触发位置，否则系统将处于等待触发状态。用户也可以设置为一维比较方式，这时比较模块只比较 Y 比较轴上的位置数据。

启动位置比较以后，当外部编码器或者内部脉冲计数器到达设定的比较位置时，若选择定宽脉冲方式输出，立即从设置的输出接口输出一个脉冲，脉冲宽度可以设置为 1~32767us；若选择电平方式输出，立即从设置的输出接口反转电平。当所有比较位置都触发以后，位置比较结束。按电平方式输出时，可以设置比较输出的起始电平，但脉冲方式不能设置输出的初始电平。

位置比较过程中，可调用 `GT_CompareStatus` 查询位置比较输出是否已经完成，已输出的上升沿个数，当前可输入的 FIFO 及剩余空间。每个二维位置比较输出模块分配有两个 2048 深度的 FIFO，用户可以在输出的过程中在未使用 FIFO 中压入新的数据。在启动二维位置比较输出之前，FIFO0 为当前 FIFO 但是仍可以在 FIFO0 中压入数据。

端子板共有 2 个位置比较输出通道，CN14 的 1、6 脚差分输出 HSIO0，2、7 脚差分输出 HSIO1。5 脚是 5V，9 脚是 GND。位置比较输出通道输出的电压为 5V，如果用户需要 24V 位置比较输出，可以调用 `GT_SetComparePort` 指令把位置比较输出关联到通用输出 GPO0-GPO15，这时位置比较输出在指定的 GPO 接口输出（位置比较输出和 GPO 指令相与输出）。

二维位置比较输出与原一维比较模块共存，使用相同的接口。设置二维比较模块后，就无法使用一维位置比较功能，如果需要使用一维位置比较功能需要先关闭把二维位置功能。



一维位置比较和二维位置比较都是以启动位置比较后当前位置点为基准点，进行比较输出的。

## 四、例程

### 例程 1 输出指定电平或脉冲

```
// 在 HSIO 通道 0 输出一个宽度为 500 微秒的脉冲。
GT_2DCompareMode(0, COMPARE2D_MODE_2D); //启动二维比较输出模块
GT_2DComparePulse(0, // 通道 0
    1, //输出有效
    0, // 脉冲方式
    500); // 脉冲宽度为 500us
// HSIO 通道 0 引脚输出电平取反
GT_2DComparePulse(0, // HSIO 通道 0
    1, // 引脚输出电平取反
    1, // 电平方式
    0); // 无效参数
```

### 例程 2 设置位置比较输出数据

```
T2Dcompare Datadatabuf[3]; //二维位置比较输出数据
```



```

T2DcomparePrm Prm; //二维位置比较输出参数
GT_2DCompareMode(0, COMPARE2D_MODE_2D); //启动二维比较输出模块，为二维位置比较
方式
//设置参数
Prm.encx = 1; // X 轴为轴
Prm.ency = 2; // Y 轴为轴
Prm.maxerr = 300; // 最大误差Pulse
Prm.outputType = 0; // 输出类型脉冲
Prm.source = 0; // 比较源规划器
Prm.startLevel = 0; // 无效参数
Prm.threshold = 0; //该参数设置为0
Prm.time = 500; // 脉冲宽度 us
GT_2DCompareSetPrm(0,&Prm);
//设置数据在XY轴位置达到(,1000), (,2000), (,3500)时通道HSIO0输出脉冲。
Databuf[0].px = 1000;
Databuf[0].py = 1000;
Databuf[1].px = 2000;
Databuf[1].py = 2000;
Databuf[2].px = 3500;
Databuf[2].py = 3500;
GT_2DCompareData(0, // 通道 0
3, // 需要进行比较的数据个数
databuf, // 数据数组地址
0, // 使用 FIFO0
);
GT_2DCompareStart(0); //启动位置比较输出

```

### 例程 3 设置位置比较输出口

```

GT_SetComparePort(COMPARE_PORT_GPO,1,2); //位置比较输出通道 0,1 分别输出
GPO1 ,GPO2 上

```

### 例程 4 二维位置比较的一维模式

由于二维位置比较模块对压入的数据没有单调性要求，其一维模式也同样对比较数据没有单调性要求。

注意：黑色字体部分显示了与二维模式的区别

```

short rtn;
T2DComparePrm prm;
rtn = GT_2DCompareMode(0, COMPARE2D_MODE_1D); //使用二维位置比较的 1 维模式
prm.encx = 1; //1 维模式时，这个值没有用，但是必须是一个非 0 值
prm.ency = 1; //以 1 轴作为比较源
prm.maxerr = 3; //最大误差 pulse
prm.outputType = 0; //输出脉冲
prm.source = 0; //内部脉冲计数器

```

```

prm.startLevel = 0;    //设置初始电平
prm.threshold = 0;    //最优点计算阈值，用户无需关心，取 0
prm.time = 500;      //脉冲宽度 us
rtn =GT_2DCompareSetPrm(0,&prm);
rtn =GT_2DCompareMode(1,COMPARE2D_MODE_1D); //使用二维位置比较的 1 维模式

T2DCompareData databuf[5];
databuf[0].px = 0;
databuf[0].py = 10000;
databuf[1].px = 0;
databuf[1].py = 20000;
rtn =GT_2DCompareData(0,2,databuf,0); //设置比较数据，只需设置 y 的数据，x 的数据取 0

rtn =GT_2DCompareStart(0);

```

### 例程 5 二维位置比较 FIFO 切换

二维位置比较模块有两级 FIFO，第一级包含 1 个 FIFO，大小为 512，也称为 FPGA 的 FIFO；第二级包含 2 个 FIFO（分别为 FIFO0 和 FIFO1），每个大小为 2048，称为用户级 FIFO。

如果先压入数据而没有启动二维位置比较，那么第二级 FIFO 没有启用，压入的数据先进入 FPGA 的 FIFO，对于该 FIFO 来说，最多只能压入 512 个数据（512 个数据以后就无效了）；如果先启动位置比较，第二级 FIFO 启用，用户可以操作第二级 FIFO，压入的数据不再受限于 512；控制器默认 FIFO0 为被占用的用户级 FIFO，启动位置比较后，不能向被占用的用户级 FIFO 压入数据，但可以向空闲的用户级 FIFO 压入比较数据。

Step1: 二维位置比较初始化及二维位置比较参数设置

Step2: 启动位置比较，此时用户级 FIFO0 被占用，不能压入数据

Step3: 向 FIFO1 压入数据

```

T2DCompareData databuf[1000];
for(inti=0;i<1000;i++)
{
    databuf[i].px = (i+1)*20;
    databuf[i].py = 0;
}
rtn =GT_2DCompareData(0,1000,databuf,1); //向 FIFO1 压入 1000 条数据，此时 FIFO0 空闲
for(inti=0;i<800;i++)
{
    databuf[i].px = 20000+(i+1)*20;
    databuf[i].py = 0;
}
rtn =GT_2DCompareData(0,800,databuf,0);
//向 FIFO0 压入 800 条数据，此时 FIFO0 仍然空闲，剩余空间为 1248

```

Step4.启动运动，FIFO1 被占用，会先比较 FIFO1 的数据，FIFO1 的数据比较完后会被清空并且变成空闲状态，可以向 FIFO1 压入数据，而 FIFO0 此时被占用

Step5.向 FIFO1 压入数据。

```

T2DCompareData databuf[1000];
for(int i=0;i<1000;i++)
{
    databuf[i].px = 40000+(i+1)*20;
    databuf[i].py = 0;
}
rtn = GT_2DCompareData(0,1000,databuf,1);

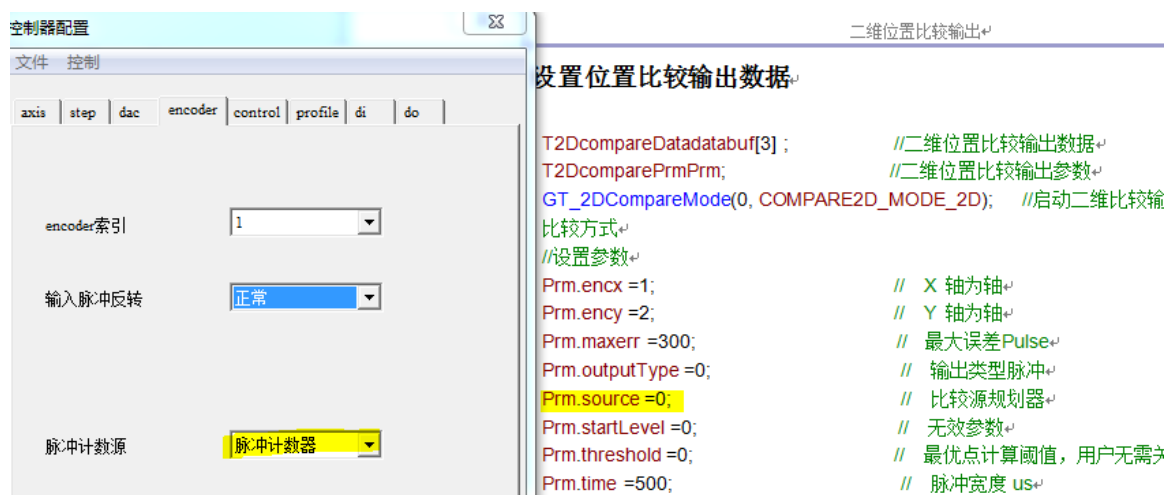
```

使用建议：先调用 **GT\_2DCompareStart** 启动二维位置比较，再压入数据，压入数据的过程中要调用 **GT\_2DCompareStatus** 查看空闲 FIFO，只能向空闲 FIFO 压入数据。

## 五、常见问题及检查

### 1. 设置二维位置比较参数出错

这个问题一般都是由于指令 **GT\_2DCompareSetPrm** 设置的比较源和控制器配置的“脉冲计数源”不一致造成的，比较源为规划器，则脉冲计数源应该为脉冲计数器；比较源为编码器，则脉冲计数源应该为“编码器”。



### 2. 压入比较数据出错

这个问题一般是把数据压入被占用的 FIFO，请参考例程“二维位置比较 FIFO 切换”。

### 3. 指令无返回错误，位置比较没有输出

- (1) 获取实际位置，实际位置是否有变化，是否与规划位置方向一致。
- (2) 如果使用的比较源为编码器，那么先看看比较为规划器方式下是否会输出比较，如果此方式下有输出，则很有可能是电机跟随性不好，可以重新调节电机跟随性或者是“最大误差允许范围”设置得太小了，而实际机械系统无法进入该误差范围。

## 六、指令详细说明

### 指令 1GT\_2DCompareClear

指令原型	short GT_2DCompareClear(short chn);		
指令说明	复位二维位置比较模块，清空二维位置比较数据。		
指令类型	立即指令，调用后立即生效。	章节页码	6
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
chn	通道号 0: HSIO0, 通道号 1: HSIO1		
指令返回值	请参照指令返回值列表。		
相关指令	无		
指令示例	无		

### 指令 2GT\_2DCompareClearData

指令原型	short GT_2DCompareClearData(short chn);		
指令说明	清空二维位置比较数据。仅清空比较数据，不会复位二维位置比较模块，因此不会清除二维位置比较起始点。		
指令类型	立即指令，调用后立即生效。	章节页码	6
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
chn	通道号 0: HSIO0, 通道号 1: HSIO1		
指令返回值	请参照指令返回值列表。		
相关指令	无		
指令示例	无		

### 指令 3GT\_2DCompareData

指令原型	short GT_2DCompareData(short chn,short count,T2DCompareData *pBuf,shortfifo);		
指令说明	设置二维位置比较输出数据。		
指令类型	立即指令，调用后立即生效。	章节页码	6
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
chn	通道号 0: HSIO0, 通道号 1: HSIO1		
count	位置比较数据个数		
pBuf	比较数据点输入 typedefstruct { long px;                    //X轴坐标 long py;                    //Y 轴坐标 } T2DCompareData;		
fifo	数据压入的 FIFO 取值范围: [0,1]		
指令返回值	请参照指令返回值列表。		
相关指令	无		
指令示例	例程 2 设置位置比较输出数据		

## 指令 4GT\_2DCompareMode

指令原型	short GT_2DCompareMode(short chn,short mode);		
指令说明	设置二维位置比较输出模式		
指令类型	立即指令，调用后立即生效。	章节页码	6
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
chn	通道号 0: HSIO0, 通道号 1: HSIO1		
mode	位置比较模式 COMPARE2D_MODE_1D 表示一维比较模式 COMPARE2D_MODE_2D 表示二维比较模式		
指令返回值	请参照指令返回值列表。		
相关指令	无		
指令示例	例程 1 输出指定电平或脉冲		

## 指令 5GT\_2DComparePulse

指令原型	short GT_2DComparePulse(short chn,shortlevel,shortoutputType,short time);		
指令说明	二维位置比较输出单次输出。		
指令类型	立即指令，调用后立即生效。	章节页码	6
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
chn	通道号 0: HSIO0, 通道号 1: HSIO1		
level	位置比较输出电平方式时起始电平（当 outputType=0 时，level 应该为 1） level=0, 位置比较输出引脚电平复位（恢复到初始化后的电平状态） level=1, 位置比较输出引脚电平取反（与初始化后的电平状态相反）		
outputType	输出类型 0: 脉冲方式, 1: 电平方式		
time	脉冲方式时的脉冲时间，单位: us		
指令返回值	请参照指令返回值列表。		
相关指令	无		
指令示例	例程 1 输出指定电平或脉冲		

## 指令 6GT\_2DCompareSetPrm

指令原型	short GT_2DCompareSetPrm(short chn,T2DComparePrm *pPrm);		
指令说明	设置二维位置比较输出参数。		
指令类型	立即指令，调用后立即生效。	章节页码	6
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
chn	通道号 0: HSIO0, 通道号 1: HSIO1		
pPrm	<pre>typedefstruct {     short encx;          //X比较轴选择取值范围，4轴卡则为[1,4]，8轴卡则为[1,8]     short ency;          //Y比较轴选择取值范围，4轴卡则为[1,4]，8轴卡则为[1,8]     short source;        //比较源0: 规划，1: 编码器     short outputType;    //输出方式0: 脉冲，1: 电平     short startLevel;    //起始电平方式0: 位置比较输出引脚电平复位                         1: 位置比较输出引脚电平取反 }</pre>		

## 二维位置比较输出

相关指令 指令示例	<pre>short time;      //脉冲方式的脉冲宽度us short maxerr;    //比较位置到位的最大允许误差，取值范围[0,512)，单位：pulse short threshold; //该参数设置为0 } T2DComparePrm;</pre> <p>source=0 时，比较源为规划器，配置控制器时，“encoder”选型的“脉冲计数源”必须设置为“内部脉冲计数器”；source=1 时，“脉冲计数源必须设置为“编码器”。否则会报参数错误。</p> <p>请参照指令返回值列表。</p>
	无
	例程 2 设置位置比较输出数据

### 指令 7GT\_2DCompareStart

指令原型	short GT_API GT_2DCompareStart(short chn);		
指令说明	启动二维位置比较输出。		
指令类型	立即指令，调用后立即生效。	章节页码	6
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
chn	通道号 0: HSIO0, 通道号 1: HSIO1		
指令返回值	请参照指令返回值列表。		
相关指令	无		
指令示例	例程 3 设置位置比较输出口		

### 指令 8GT\_2DCompareStatus

指令原型	short GT_2DCompareStatus(short chn,short *pStatus,long *pCount,short *pFifo,short *pFifoCount,short *pBufCount);		
指令说明	读取二维位置比较输出状态。		
指令类型	立即指令，调用后立即生效。	章节页码	6
指令参数	该指令共有 6 个参数，参数的详细信息如下。		
chn	通道号 0: HSIO0, 通道号 1: HSIO1		
pStatus	0: 正在进行比较输出 1: 比较输出完成		
pCount	位置比较已输出次数		
pFifo	当前空闲 FIFO		
pFifoCount	当前空闲 FIFO 剩余空间		
pBufCount	FPGA 中 FIFO 剩余空间，FPGA 的 FIFO 总大小为 512，启动位置比较之前，压入的数据先进入 FPGA 的 FIFO		
指令返回值	请参照指令返回值列表。		
相关指令	无		
指令示例	无		

### 指令 9GT\_2DCompareStop

指令原型	short GT_2DCompareStop(short chn);		
指令说明	停止二维位置比较输出。停止后位置比较输出端口的控制权移交给原一维位置比较输出模块。		
指令类型	立即指令，调用后立即生效。	章节页码	6

指令参数	该指令共有 1 个参数，参数的详细信息如下。
chn	通道号 0: HSIO0, 通道号 1: HSIO1
指令返回值	请参照指令返回值列表。
相关指令	无
指令示例	无

## 指令 10GT\_SetComparePort

指令原型	short GT_SetComparePort(short channel,short hsio0,short hsio1);		
指令说明	设置位置比较输出端口。		
指令类型	立即指令，调用后立即生效。	章节页码	6
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
channel	COMPARE_PORT_GPO: 位置比较输出到 GPO 端口		
	COMPARE_PORT_HSIO: 位置比较输出到 HSIO 端口		
hsio0	位置比较输出到 GPO 端口时有效，指定通道 0 输出端口		
hsio1	位置比较输出到 GPO 端口时有效，指定通道 1 输出端口		
补充说明	hsio0 和 hsio1 对应于通用输出口，取值范围为 0~15。 调用该指令把位置比较输出配置到通用输出口之前应该先把该通用输出口设置为高电平（因为最终的比较输出值为比较输出和通用输出值相与）。		
指令返回值	请参照指令返回值列表。		
相关指令	无		
指令示例	例程 3 设置位置比较输出口		