

# GXN 系列运动控制器编程手册

---

# GANTRY

R1.1

# 版权申明

## 固高科技股份有限公司

### 保留所有权力

固高科技股份有限公司（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

## 固高科技股份有限公司

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：[googol@googoltech.com](mailto:googol@googoltech.com)

网址：<http://www.googoltech.com.cn>

## 固高科技（海外）有限公司

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

電話：+(852) 2358-1033

傳真：+(852) 2719-8399

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

[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：台中市西屯區福中二路 10 巷 22 號 2 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	2018年10月16日
1.1	2020年11月1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

固高科技股份有限公司

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

本手册由四章内容组成，详细介绍了运动控制器的Gantry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。

关于控制器基本功能使用，请参见随产品配套的《GSN 系列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《GSN 系列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。



产品相关手册及安装文件如驱动程序、dll 文件、例程、Demo 等，请登录固高科技公司网站下载，网址为：[www.googoltech.com.cn/pro\\_view-53.html](http://www.googoltech.com.cn/pro_view-53.html)

---

# 目录

版权申明 .....	1
联系我们 .....	1
文档版本 .....	2
前言 .....	3
目录 .....	4
一、 指令列表 .....	5
二、 重点说明 .....	5
三、 例程 .....	6
例程 1 龙门闭环同步模式 .....	6
例程 2 龙门开环同步模式 .....	7
例程 3 平面插补例程 .....	7
例程 4 驱动器龙门模式 .....	8
四、 指令详细说明 .....	10

## 一、 指令列表



本章表格中指令右侧的页码为“四、指令详细说明”中的对应页码，其他为章节页码，均可以使用“超级链接”进行索引。

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

表 1 Gantry 功能指令列表

指令	说明
<a href="#">GTN_SetGantryMode</a>	设置龙门配置信息
<a href="#">GTN_GetGantryMode</a>	获取龙门配置信息。
<a href="#">GTN_SetGantryPid</a>	配置主轴 PID 和同步 PID 参数
<a href="#">GTN_GetGantryPid</a>	获取主轴 PID 和同步 PID 参数

## 二、 重点说明

1、龙门功能是针对两个电机同步驱动一个轴的应用而设计。控制卡将该双驱轴看成一个整体进行控制，包括解耦龙门闭环同步模式和龙门开环同步模式。

龙门闭环同步模式通过两个独立的 PID 控制器（Control1 和 Control2）实现龙门闭环同步控制。其中 Control1 控制龙门轴整体运动，Control1 控制器的刚性越高，对位置命令跟踪越好。Control2 控制龙门轴之间的差动，防止偏差过大导致电机发热性能下降，Control2 刚性越高，两电机间偏差越小，但注意 Control2 刚性过高时容易产生震荡，反而加大电机间拉扯。

龙门开环同步模式下只是让两个龙门轴发出一样的脉冲。

### 2、使用注意事项

- 1) 进行龙门同步运动时，需要使用指令 [GTN\\_SetGantryMode](#) 将两个电机轴配置为同步控制。配置完毕后，任意一个轴上伺服时另一个轴也会自动上伺服，实现轴同步。
- 2) 切换模式时需要先下使能，切换到 NONE 模式后，才可以切换到另一种同步模式。
- 3) 进入解耦同步模式前，需配置驱动器进入模拟量速度模式，测试 DAC 与编码器对应方向，保证两轴 DAC 输出与编码器增量方向相同，防止出现反馈异常。
- 4) 龙门开环同步模式，驱动器设置为位置环模式。

### 三、 例程

#### 例程 1 龙门闭环同步模式

```

int main(int argc, char* argv[])
{
    short rtn,core=1;
    short group = 1;
    short axis1 = 2;
    short axis2 = 4;
    short ctlMode=0; // 闭环模式
    short gantryMode;
    long synErrLmt =1000; // 最大同步误差为1000个脉冲
    TrapPrm trap;
    TPid pid1, pid2;
    rtn = GTN_Open();//打开运动控制卡
    rtn = GTN_Reset(core);
    rtn = GTN_AlarmOff(core,axis1); //关闭轴 axis1 报警
    rtn = GTN_AlarmOff(core,axis2);
    rtn = GTN_LmtsOn(core,axis1,-1); //开轴 axis1 限位，正负限位无效
    rtn = GTN_LmtsOn(core,axis2,-1);
    rtn = GTN_ClrSts(core,axis1); //清状态
    rtn = GTN_ClrSts(core,axis2); //清状态
    rtn = GTN_CtrlMode(core,axis1,ctlMode); //设置为闭环模式
    rtn = GTN_CtrlMode(core,axis2,ctlMode); //设置为闭环模式

    // 配置 1 轴 2 轴进入位置环解耦同步模式，最大同步误差为 1000
    gantryMode = GANTRY_MODE_DECOUPLE_POSITION_LOOP;
    rtn = GTN_SetGantryMode(core,group,axis1,axis2, gantryMode,synErrLmt);
    rtn = GTN_GetGantryPid(core,group, &pid1, &pid2); //获取龙门运动 PID 参数
    pid1.kp = 1; // pid1 控制龙门轴整体运动
    pid2.kp = 0.5; // pid2 控制龙门轴之间的差动
    rtn = GTN_SetGantryPid(core,group, &pid1, &pid2); //设置龙门运动 PID 参数

    rtn = GTN_AxisOn(core,axis1); // 使能任意龙门轴，控制器会自动使能另一轴开始同步
    rtn = GTN_PrflTrap(core,axis1);//点位运动
    rtn = GTN_GetTrapPrm(core,axis1,&trap);
    trap.acc = 0.5; //加速度
    trap.dec =0.5; //减速度
    trap.smoothTime = 20; //平滑时间
    rtn = GTN_SetTrapPrm(core,axis1, &trap);
    rtn = GTN_SetVel(core,axis1,10); //运动速度
    rtn = GTN_SetPos(core,axis1,20000); //运动距离
    rtn = GTN_Update(core,1<<(axis1-1)); //启动运动

```

```

return 0;
}

```

## 例程 2 龙门开环同步模式

```

int main(int argc, char* argv[])
{
    short rtn,core=1;
    short group = 1;
    short axis1 = 2;
    short axis2 = 4;
    short ctlMode=1;// 脉冲模式
    short gantryMode;
    long synErrLmt=1000;// 最大同步误差为1000个脉冲
    TrapPrm trap;
    rtn = GTN_Open();//打开运动控制卡
    rtn = GTN_Reset(core);
    rtn = GTN_AlarmOff(core,axis1); //关闭轴 axis1 报警
    rtn = GTN_AlarmOff(core,axis2);
    rtn = GTN_LmtsOn(core,axis1,-1); //开轴 axis1 限位，正负限位无效
    rtn = GTN_LmtsOn(core,axis2,-1);
    rtn = GTN_ClrSts(core,axis1); //清状态
    rtn = GTN_ClrSts(core,axis2); //清状态
    rtn = GTN_CtrlMode (core,axis1, ctlMode); //设置为开环模式
    rtn = GTN_CtrlMode (core,axis2, ctlMode); //设置为开环模式

    //配置轴轴进入开环同步模式，最大同步误差为 1000
    gantryMode = GANTRY_MODE_OPEN_LOOP_GANTRY;
    rtn = GTN_SetGantryMode(core,group,axis1,axis2, gantryMode, synErrLmt);
    rtn = GTN_AxisOn(core,axis1); // 使能任意龙门轴，控制器会自动使能另一轴开始同步
    rtn = GTN_PrflTrap(core,axis1);//点位运动
    rtn = GTN_GetTrapPrm(core,axis1,&trap);
    trap.acc = 0.5; //加速度
    trap.dec =0.5; //减速度
    trap.smoothTime = 20; //平滑时间
    rtn = GTN_SetTrapPrm(core,axis1, &trap);
    rtn = GTN_SetVel(core,axis1,10); //运动速度
    rtn = GTN_SetPos(core,axis1,20000); //运动距离
    rtn = GTN_Update(core,1<<(axis1-1)); //启动运动
return 0;
}

```

## 例程 3 平面插补例程

(假设 1 轴为 X 轴，2 轴为 Y1 轴，3 轴为 Y2 轴)



```

short rtn,core = 1;
short group = 1;
short axisX = 1;
short axisY1 = 2;
short axisY2 = 3;
short crd = 1;
short ctlMode=0; // 闭环模式
short gantryMode;
long synErrLmt;
TCrdPrm crdPrm;
TPid pid1;
TPid pid2;
// ... 配置 X 轴及其他配置
rtn = GTN_AxisOn(core,axisX);
rtn = GTN_CtrlMode (core,axisY1, ctlMode); //设置为开环模式
rtn = GTN_CtrlMode (core,axisY2, ctlMode); //设置为开环模式

// 配置 2 轴 3 轴进入位置环解耦同步模式，最大同步误差为 1000
gantryMode = GANTRY_MODE_DECOUPLE_POSITION_LOOP;
rtn = GTN_SetGantryMode(core,group, axisY1, axisY2, gantryMode,synErrLmt);
rtn = GTN_GetGantryPid(core,group, &pid1, &pid2);
pid1.kp = 1;
pid2.kp = 0.5;
rtn = GTN_SetGantryPid(core,group, &pid1, &pid2);
rtn = GTN_AxisOn(core,axisY1); // 使能任意龙门轴，控制器会自动使能另一轴开始同步
rtn = GTN_GetCrdPrm(core,crd, &crdPrm); //查询坐标系参数
crdPrm.dimension = 2; //2 维坐标
crdPrm.profile[0] = axisX;
crdPrm.profile[1] = axisY1;
crdPrm.synVelMax= 50;
crdPrm.synAccMax = 1;
rtn = GTN_SetCrdPrm(core,crd, & crdPrm); //设置坐标系参数
rtn = GTN_LnXY(core,crd,10000,0,50,5,0); //XY 平面 2 维直线插补
rtn = GTN_CrdStart(core,crd,0); //启动坐标系 1

```

#### 例程 4 驱动器龙门模式

驱动器龙门模式需要配合固高自研驱动器使用。

##### (1) 使用步骤

- 1) 通过控制器调试软件 Motionstudio 进行配置, 打开 Motionstudio 调试软件点击“示波器”模块, 在“龙门调试”界面选择“驱动器龙门模式”, 选取对应的轴号后点击设置进行配置(配置完成后不需要重复配置, 配置一次即可)。

点位运动 触发设置 PID 龙门测试

核: 1 轴: 1 PID索引: 1

比例增益:	<input type="text" value="0"/>	速度前馈:	<input type="text" value="0"/>	积分饱和极限:	<input type="text" value="32767"/>	获取Pid
积分增益:	<input type="text" value="0"/>	加速度前馈:	<input type="text" value="0"/>	位置误差极限:	<input type="text" value="32767"/>	设置Pid
微分增益:	<input type="text" value="0"/>					

- 2) 同时选用驱动器龙门后 mode= GANTRY\_MODE\_NONE，表示只能对龙门主轴进行运动指令操作，从轴不响应任何指令。

```
int main(int argc, char* argv[])
{
    short rtn, core = 1;
    short group = 1;
    short axis1 = 2;//主轴
    short axis2 = 4;//从轴
    short gantryMode;
    long synErrLmt = 1000;// 最大同步误差为1000个脉冲
    TrapPrm trap;
    rtn = GTN_Open();//打开运动控制卡
    rtn = GTN_Reset(core);
    rtn = GTN_AlarmOff(core,axis1);    //关闭轴 axis1 报警
    rtn = GTN_AlarmOff(core,axis2);
    rtn = GTN_LmtsOn(core,axis1,-1);    //开轴 axis1 限位， 正负限位无效
    rtn = GTN_LmtsOn(core,axis2,-1);
    rtn = GTN_ClrSts(core,axis1);    //清状态
    rtn = GTN_ClrSts(core,axis2);    //清状态
    //配置两个轴进入到驱动器龙门模式 最大同步误差为 1000
    gantryMode = GANTRY_MODE_RINGNET_DRIVER;
    rtn = GTN_SetGantryMode(core,group,axis1,axis2, gantryMode, synErrLmt);
    rtn = GTN_AxisOn(core,axis1);    // 使能主轴 从轴会跟着上使能
    rtn = GTN_PrflTrap(core,axis1);//对主轴点位运动 就是驱动龙门运动
    rtn = GTN_GetTrapPrm(core,axis1,&trap);
    trap.acc = 0.5;    //加速度
    trap.dec = 0.5;    //减速度
    trap.smoothTime = 20;    //平滑时间
    rtn = GTN_SetTrapPrm(core,axis1, &trap);
    rtn = GTN_SetVel(core,axis1,10);    //运动速度
    rtn = GTN_SetPos(core,axis1,20000);    //运动距离
    rtn = GTN_Update(core,1<<(axis1-1));    //启动运动
    return 0;
}
```

## 四、 指令详细说明

### 指令 1 GTN\_SetGantryMode

指令原型	GTN_SetGantryMode (short core,short group, short master, short slave, short mode, long syncErrorLimit)		
指令说明	配置两个电机轴为同步控制，使用同一个 IO 上伺服		
指令类型	立即指令，调用后立即生效。	章节页码	5
指令参数	该指令共有 6 个参数，参数的详细信息如下。		
core	内核，正整数，取值范围[1,2]		
group	龙门号，取值范围为[1,2]。		
master	主轴号，范围为[1,4]。		
slave	从轴号，范围为[1,4]。		
mode	设置龙门的同步模式，其宏定义如下： GANTRY_MODE_OPEN_LOOP_GANTRY (1): 脉冲模式 GANTRY_MODE_DECOUPLE_POSITION_LOOP (2): 位置环解耦补偿模式 GANTRY_MODE_RINGNET_DRIVER (10):驱动器龙门模式 GANTRY_MODE_NONE (-1): 解除同步 注意： 1.必须在下使能状态下才能切换同步模式，如果从模式2中解除同步，两轴保持闭环模式，两轴PID需重新设置。 2.驱动器龙门需要配合固高自研驱动器使用 选用驱动器龙门后： mode= GANTRY_MODE_RINGNET_DRIVER 进入驱动器龙门同步运动模式 mode= GANTRY_MODE_NONE 表示只能对龙门主轴进行运动指令操作 从轴不响应任何指令		
syncErrorLimit	主从轴的同步误差极限。		
指令返回值	7: 轴号超出范围，或两个轴为同一个轴，或模式错误。 1: 轴正在使能，或其中一个轴已经与别的轴建立同步。 其他请参照《GXN 系列运动控制器编程手册之基本功能》第 3 章指令返回值及其意义。		
相关指令	<a href="#">GTN_GetGantryMode</a>		
指令示例	例程 1 龙门闭环同步模式		

### 指令 2 GTN\_GetGantryMode

指令原型	GTN_GetGantryMode (short core, short group, short *master, short *slave, short *mode, long *syncErrorLimit)		
指令说明	读取一组龙门的配置信息。		
指令类型	立即指令，调用后立即生效。	章节页码	5
指令参数	该指令共有 6 个参数，参数的详细信息如下。		
core	内核，正整数，取值范围[1,2]		
group	龙门号，取值范围为[1,2]。		
master	用于返回主轴轴号的变量指针。		

slave	用于返回从轴轴号的变量指针。
mode	返回龙门正在使用的同步模式： GANTRY_MODE_OPEN_LOOP_GANTRY (1): 脉冲模式 GANTRY_MODE_DECOUPLE_POSITION_LOOP (2): 位置环解耦补偿模式 GANTRY_MODE_RINGNET_DRIVER (10): 驱动器龙门模式 GANTRY_MODE_NONE (-1): 解除同步，即未使用龙门同步模式
syncErrorLimit	用于返回龙门主从轴同步误差极限的变量指针。
指令返回值	其他请参照《GXN 系列运动控制器编程手册之基本功能》第 3 章指令返回值及其意义。
相关指令	<a href="#">GTN_SetGantryMode</a>
指令示例	例程 2 龙门开环同步模式

### 指令 3 GTN\_SetGantryPid

指令原型	GTN_SetGantryPid(short core, short group, TPid *pGantryPid, TPid *pYawPid);		
指令说明	配置主轴PID和同步PID参数。		
指令类型	立即指令，调用后立即生效。	章节页码	5
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
core	内核，正整数，取值范围[1,2]		
group	龙门号，取值范围为[1,2]。		
pGantryPid	龙门 PID。PID 参数是一个结构体，如下： <pre>typedef struct Pid {     double kp;     double ki;     double kd;     double kvff;     double kaff;     long integralLimit;     long derivativeLimit;     short limit; }TPid</pre> kp : 比例增益; ki : 积分增益; kd : 微分增益; Kvff : 速度前馈系数; kaff : 加速度前馈系数; integralLimit: 积分饱和极限 derivativeLimit: 微分饱和极限; limit: 控制量输出饱和极限;		
pYawPid	同步PID。		
指令返回值	请参照《GXN 系列运动控制器编程手册之基本功能》第 3 章指令返回值及其意义。		
相关指令	<a href="#">GTN_GetGantryPid</a>		
指令示例	例程 3 平面插补例程		

## 指令 4 GTN\_GetGantryPid

指令原型	GTN_SetGantryPid(short core,short group, TPid *pGantryPid, TPid *pYawPid);		
指令说明	获取主轴PID和同步PID参数。		
指令类型	立即指令，调用后立即生效。	章节页码	5
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
core	内核，正整数，取值范围[1,2]		
group	龙门号，取值范围为[1,2]。		
pGantryPid	<p>返回的龙门 PID。PID 参数是一个结构体，如下：</p> <pre>typedef struct Pid {     double kp;     double ki;     double kd;     double kvff;     double kaff;     long integralLimit;     long derivativeLimit;     short limit; }TPid</pre> <p>kp : 比例增益;  ki : 积分增益;  kd : 微分增益;  Kvff : 速度前馈系数;  kaff : 加速度前馈系数;  integralLimit: 积分饱和极限  derivativeLimit: 微分饱和极限;  limit: 控制量输出饱和极限;</p>		
pYawPid	返回的同步PID。		
指令返回值	请参照《GXN 系列运动控制器之基本功能》第 3 章指令返回值及其意义。		
相关指令	<a href="#">GTN_SetGantryPid</a>		
指令示例	例程 4 驱动器龙门模式		