

GXN 系列运动控制器编程手册

Watch 功能

R1.0

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 樓
電 話：+886-4-23588245
傳 真：+886-4-23586495
電子郵件：tinfo@googoltech.com

文档版本

版本号	修订日期
1.0	2018 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；

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

本手册由五章内容组成，详细介绍了运动控制器的激光控制功能及编程实现。

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《GXN 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《GXN 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。



注意

相关手册及控制器适用文档列表见于光盘的 manual 目录下。

目录

版权申明.....	1
联系我们.....	1
文档版本.....	2
前言.....	3
目录.....	4
索引.....	5
1. 指令索引.....	5
2. 表格索引.....	5
3. 例程索引.....	5
一、 指令汇总表.....	6
二、 功能简介.....	6
三、 配置文件.....	6
四、 例程.....	8
五、 指令详细说明.....	9
附录 I.....	15

索引

1. 指令索引

指令 1	GTN_AddWatchEvent.....	9
指令 2	GTN_AddWatchVar.....	10
指令 3	GTN_ClearWatch.....	11
指令 4	GTN_GetWatchInfo.....	11
指令 5	GTN_LoadWatchConfig.....	12
指令 6	GTN_PrintWatch.....	12
指令 7	GTN_ProgramWatch.....	12
指令 8	GTN_ReadWatch.....	13
指令 9	GTN_WatchOff.....	13
指令 10	GTN_WatchOn.....	13

2. 表格索引

表 1	激光控制功能指令汇总表.....	6
-----	------------------	---

3. 例程索引

例程 1	通过下载配置文件采集规划位置、规划速度等数据.....	8
例程 2	通过下载配置文件采集指令执行情况.....	8
例程 3	通过烧录配置文件至运动控制器并动执行.....	9

一、 指令汇总表



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

表 1 激光控制功能指令汇总表

指令	说明	页码
GTN_ClearWatch	清除已设置的 Watch 变量和 Watch 事件	11
GTN_AddWatchVar	添加 Watch 变量	10
GTN_AddWatchEvent	添加 Watch 事件	9
GTN_WatchOn	使能 Watch	13
GTN_WatchOff	关闭 Watch	13
GTN_PrintWatch	打印 Watch 数据到文件	12
GTN_LoadWatchConfig	下载 Watch 配置信息至运动控制器	12
GTN_ProgramWatch	设置 Watch 配置文件烧录格式	12
GTN_ReadWatch	读取 Watch 数据	13
GTN_GetWatchInfo	读取 Watch 状态	11

二、 功能简介

通过 Watch 功能可实时并准确地监测某些变量值（变量类型详见附录 I）和指令执行情况，详细介绍见例程。

Watch 功能模块既可以通过加载配置文件后执行，还可以通过烧录至 Flash 后控制器自动执行。采集数据或指令执行情况可以通过相关指令打印至文件也可以实时地通过指令读取。

Watch 最多可以同时采集 20 组数据，最多可以添加 8 个相互独立的触发条件。

三、 配置文件

配置文件详细解析：

```
[setting]
/*数据所在区域，目前分为 WATCH_GROUP_TIME(0) 中断和
WATCH_GROUP_BACKGROUND(1)背景*/
group=WATCH_GROUP_TIME
/*mode 设置 watch 模式，WATCH_MODE_STATIC(0)：静态模式，
WATCH_MODE_LOOP (1)：循环模式*/
/*静态模式下 Watch 缓冲区放满以后自动停止；循环模式下 Watch 缓冲区放满以后返回到缓冲
区头部重新开始；*/
mode=WATCH_MODE_LOOP
/*采集数据间隔。如果是背景任务模式下，该参数无实际意义；中断模式下，数据间隔为
(interval+1)*250us*/
interval=0
```

```

/*小数点位数，采集数据精度*/
precision=3
/*下载模式，详细说明请参照指令 GTN_ProgramWatch 中参数 2*/
loadMode=-1
[var1]
/*设置变量类型，根据变量地址表选择需要 Watch 的变量,常用变量,详见附录（最多可同时观测
20 组变量）*/
type=WATCH_VAR_PRF_POS
/*变量索引*/
index=1
/*变量二级索引*/
id=1
[var2]
type=WATCH_VAR_PRF_VEL
index=1//可选
id=1 //可选
...
[event1]
/* type: 设置事件类型，可以设置为以下值（最多可同时增加 8 个触发事件）
WATCH_EVENT_RUN 当条件成立时采集数据，不成立时停止采集数据。
WATCH_EVENT_START 当条件成立时启动数据采集。
WATCH_EVENT_STOP 当条件成立时停止数据采集。
WATCH_EVENT_OFF 当条件成立时关闭数据采集。*/
type=WATCH_EVENT_START
/*设置事件触发次数，为 0 时表示不限制事件的触发次数。*/
loop=0
/* watchCount: 设置事件触发以后采集数据的次数。
对于不同的事件类型，有相应的处理方式
WATCH_EVENT_RUN: 设置停止以后采集的次数，0 表示不采集。
WATCH_EVENT_START: 设置启动以后采集次数，0 表示一直采集(条件不成立状态下，继续
采集的个数)。
WATCH_EVENT_STOP: 设置停止以后采集次数，0 表示不采集。*/
watchCount=0
/*设置事件的条件变量，根据设置变量地址表选择需要的条件变量。*/
varType=WATCH_VAR_PRF_POS
/*变量索引*/
varIndex=1
/*变量二级索引*/
varId=1
/* condition: 设置事件条件
WATCH_CONDITION_EQ 条件变量的值等于设定值
WATCH_CONDITION_NE 条件变量的值不等于设定值
WATCH_CONDITION_GE 条件变量的值大于等于设定值
WATCH_CONDITION_LE 条件变量的值小于等于设定值
WATCH_CONDITION_CHANGE_TO 条件变量的值改变为设定值*/
condition=WATCH_CONDITION_GE

```



```
/*可选，设置事件比较值*/
value=0
```

四、 例程

例程 1 通过下载配置文件采集规划位置、规划速度等数据

// 当轴 1 规划位置大于等于 100 时，开始采集轴 1 的规划位置和规划速度

(1) watchTimer.ini 配置文件

```
[setting]
group=WATCH_GROUP_TIME
mode=WATCH_MODE_LOOP
interval=0
precision=3 // 小数点精度
loadMode=-1
[var1]
type=WATCH_VAR_PRF_POS
index=1
[var2]
type=WATCH_VAR_PRF_VEL
index=1
...
[event1]
type=WATCH_EVENT_START
loop=0
watchCount=0
varType=WATCH_VAR_PRF_POS
varIndex=1
condition=WATCH_CONDITION_GE
value=100
```

(2) 程序

```
short rtn;
rtn=GTN_LoadWatchConfig(1,"watchTimer.ini");
```

例程 2 通过下载配置文件采集指令执行情况

// 采集所有写类型的指令

(1) watchBackground.ini 配置文件

```
[setting]
group=WATCH_GROUP_BACKGROUND
mode=WATCH_MODE_LOOP
interval=0
precision=3
loadMode=2
```

```

[var1]
type=WATCH_VAR_COMMAND_DATA

[var2]
type=WATCH_VAR_COMMAND_CODE
...
[event1]
type=WATCH_EVENT_STOP
loop=0
watchCount=1
varType=WATCH_VAR_COMMAND_COUNT
condition=WATCH_CONDITION_CHANGE
[event2]
type=WATCH_EVENT_START
loop=0
watchCount=0
varType=WATCH_VAR_COMMAND_READ_FLAG
condition=WATCH_CONDITION_EQ
value=1

```

(2) 程序

```

short rtn;
rtn=GTN_LoadWatchConfig(1,"watchBackground.ini");

```

例程 3 通过烧录配置文件至运动控制器并动执行

// 控制器上电后自动采集所有写类型的指令

(1) 配置文件详见例程 2 中 watchBackground.ini**(2) 程序**

```

short rtn;
rtn=GTN_ProgramWatch(1,WATCH_LOAD_MODE_RUN);

```

五、指令详细说明

指令 1 GTN_AddWatchEvent

指令原型	short GTN_AddWatchEvent(short core,const TWatchEvent *pEvent)		
指令说明	添加 Watch 事件		
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。		
core	内核，正整数，取值范围[1,2]		
pEvent	typedef struct { unsigned short type;		

```

unsigned short loop;
unsigned short watchCount;
TWatchVar var;
unsigned short condition;
double value;
} TWatchEvent;
    
```

type: 设置事件类型, 可以设置为以下值 (最多可同时增加 8 个触发事件)
 WATCH_EVENT_RUN 当条件成立时采集数据, 不成立时停止采集数据。
 WATCH_EVENT_START 当条件成立时启动数据采集。
 WATCH_EVENT_STOP 当条件成立时停止数据采集。
 WATCH_EVENT_OFF 当条件成立时关闭数据采集。

loop: 设置事件触发次数, 为 0 时表示不限制事件的触发次数。

watchCount: 设置事件触发以后采集数据的次数。

对于不同的事件类型, 有相应的处理方式

WATCH_EVENT_RUN: 设置停止以后采集的次数, 0 表示不采集。

WATCH_EVENT_START: 设置启动以后采集次数, 0 表示一直采集(条件不成立状态下, 继续采集的个数)。

WATCH_EVENT_STOP: 设置停止以后采集次数, 0 表示不采集。

var: 设置事件的条件变量, 根据设置变量地址表选择需要的条件变量。

condition: 设置事件条件

WATCH_CONDITION_EQ 条件变量的值等于设定值

WATCH_CONDITION_NE 条件变量的值不等于设定值

WATCH_CONDITION_GE 条件变量的值大于等于设定值

WATCH_CONDITION_LE 条件变量的值小于等于设定值

WATCH_CONDITION_CHANGE_TO 条件变量的值改变为设定值

WATCH_CONDITION_CHANGE 条件变量的值改变

WATCH_CONDITION_UP 条件变量的值变大

WATCH_CONDITION_DOWN 条件变量的值变小

WATCH_CONDITION_REMAIN_AT 条件变量的值保持设定值

WATCH_CONDITION_REMAIN 条件变量的值保持不变

value: 设置事件比较值

指令返回值

请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章

相关指令

无。

指令示例

无。

指令 2 GTN_AddWatchVar

指令原型

short GTN_AddWatchVar(short core,const TWatchVar *pVar)

指令说明

添加 Watch 变量

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。		
core	内核，正整数，取值范围[1,2]		
pVar	<pre>typedef struct { unsigned short type; unsigned short index; unsigned short id; } TWatchVar</pre> <p>type: 设置变量类型, 根据变量地址表选择需要 Watch 的变量, 常用变量详见附录 (最多可同时观测 20 组变量)。</p> <p>index: 变量索引</p> <p>id: 变量二级索引</p>		
指令返回值	请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章		
相关指令	无。		
指令示例	无		

指令 3 GTN_ClearWatch

指令原型	short GTN_ClearWatch(short core, short mode=0)		
指令说明	清除已设置的 Watch 变量和 Watch 事件。		
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。		
core	内核，正整数，取值范围[1,2]		
mode	<p>mode 设置清除方式</p> <p>0: 全部清除。</p> <p>1: 只清除变量。</p> <p>2: 只清除事件。</p>		
指令返回值	请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章		
相关指令	无		
指令示例	无		

指令 4 GTN_GetWatchInfo

指令原型	short GTN_GetWatchInfo (short core, TWatchInfo *pInfo)		
指令说明	读取 watch 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 2 个参数，参数的详细信息如下。		
core	内核，正整数，取值范围: [1,2]		
pInfo	<p>watch 信息</p> <pre>typedef struct { short enable; // 采集使能 short run; // 采集状态 unsigned long time; // 采集次数 unsigned long head; // 头指针</pre>		

Watch 功能

	<pre> unsigned long threshold; // 最多容纳采集次数 short interval; // 采集间隔 short mode; // 采集模式 unsigned short countBeforeEvent; // 事件触发之前的采集数量 unsigned short countAfterEvent; // 事件触发以后的采集数量 unsigned short varCount; // 采集变量数量 unsigned short eventCount; // 采集事件数量 } TWatchInfo; </pre>
指令返回值	请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章
相关指令	无
指令示例	无

指令 5 GTN_LoadWatchConfig

指令原型	short GTN_LoadWatchConfig (short core,char *pFile)		
指令说明	下载 Watch 配置信息至运动控制器		
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。		
core	内核，正整数，取值范围：[1,2]		
pFile	配置文件地址		
指令返回值	请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章		
相关指令	无		
指令示例	例程 1 通过下载配置文件采集规划位置、规划速度等数据		

指令 6 GTN_PrintWatch

指令原型	short GTN_PrintWatch(short core,const char *pFileName,long start=0,unsigned long printCount=0)		
指令说明	打印 Watch 数据到文件		
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。		
core	内核，正整数，取值范围[1,2]		
pFileName	打印 Watch 数据的文件名。 注意：在打印数据之前需要先停止数据采集		
start	采集数据地址偏移量 如果 start>=0，则表示数据起始地址为 start 值 如果 start<0，则表示数据起始地址为从最后停止采集时地址往上偏移 start 值		
printCount	从 start 地址开始总共打印数据个数。 如果 start=0，printCount=0，表示打印所有数据		
指令返回值	请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章		
相关指令	无。		
指令示例	无		

指令 7 GTN_ProgramWatch

指令原型	short GTN_ProgramWatch (short core,short loadMode)
------	--

指令说明	烧录 Watch 配置信息至运动控制器		
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。		
core	内核，正整数，取值范围：[1,2]		
loadMode	烧录 Watch 配置文件和执行方式 #define WATCH_LOAD_MODE_NONE 0 // 只烧录不加载 #define WATCH_LOAD_MODE_BOOT 2 // 烧录并加载不执行 #define WATCH_LOAD_MODE_RUN 3 // 烧录、加载并自动运行		
指令返回值	请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章		
相关指令	无		
指令示例	例程 3 通过烧录配置文件至运动控制器并动执行		

指令 8 GTN_ReadWatch

指令原型	short GTN_ReadWatch (short core,short varIndex,double *pBuffer,unsigned long bufSize,unsigned long *pReadCount)		
指令说明	读取 Watch 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 5 个参数，参数的详细信息如下。		
core	内核，正整数，取值范围：[1,2]		
varIndex	读取数据对应的序号，正整数，取值范围：[1, 20]		
pBuffer	读取 watch 数据		
bufSize	读取数据个数		
pReadCount	实际读取数据个数		
指令返回值	请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章		
相关指令	无		
指令示例	无		

指令 9 GTN_WatchOff

指令原型	short GTN_WatchOff(short core)		
指令说明	停止 Watch		
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。		
core	内核，正整数，取值范围：[1,2]		
指令返回值	请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章		
相关指令	无		
指令示例	无		

指令 10 GTN_WatchOn

指令原型	short GTN_WatchOn(short core,short interval=0,short mode=WATCH_MODE_STATIC,unsigned short count=0)		
指令说明	使能 Watch		
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。		

Watch 功能

core	内核，正整数，取值范围[1,2]
interval	interval 设置 watch 采样间隔。
mode	mode 设置 watch 模式，0：静态模式，1：循环模式，2：动态模式。 静态模式下 Watch 缓冲区放满以后自动停止； 循环模式下 Watch 缓冲区放满以后返回到缓冲区头部重新开始；
count	循环模式下设置事件触发前采集数据的数量。
指令返回值	请参照《GTN 系列运动控制器编程手册之基本功能》中的第 3 章
相关指令	无。
指令示例	无

附录 I

// MC 时钟, 以 ms 为单位	
#define WATCH_VAR_CLOCK	(1200)
// MC 时钟, 以规划周期为单位 (250us)	
#define WATCH_VAR_PRF_LOOP	(1201)
// Profile 规划位置	
#define WATCH_VAR_PRF_POS	(6000)
// Profile 规划速度	
#define WATCH_VAR_PRF_VEL	(6001)
// Profile 规划加速度	
#define WATCH_VAR_PRF_ACC	(6002)
// Profile 运动状态	
#define WATCH_VAR_PRF_RUN	(6200)
// 插补运动合成规划位置	
#define WATCH_VAR_CRD_PRF_POS	(8000)
// 插补运动合成规划速度	
#define WATCH_VAR_CRD_PRF_VEL	(8001)
// 插补运动合成规划加速度	
#define WATCH_VAR_CRD_PRF_ACC	(8002)
// 插补运动状态	
#define WATCH_VAR_CRD_RUN	(8200)
// 插补段号	
#define WATCH_VAR_CRD_SEGMENT_NUMBER	(8202)
// 插补用户段号	
#define WATCH_VAR_CRD_SEGMENT_NUMBER_USER	(8203)
// 插补接收指令	
#define WATCH_VAR_CRD_COMMAND_RECEIVE	(8204)
// 插补执行指令	
#define WATCH_VAR_CRD_COMMAND_EXECUTE	(8205)
// SCAN 合成规划位置	
#define WATCH_VAR_SCAN_PRF_POS	(18000)
// SCAN 合成规划速度	
#define WATCH_VAR_SCAN_PRF_VEL	(18001)
// SCAN 合成规划加速度	
#define WATCH_VAR_SCAN_PRF_ACC	(18002)
// SCAN 的 X 轴规划位置	
#define WATCH_VAR_SCAN_PRF_POS_X	(18010)
// SCAN 的 Y 轴规划位置	
#define WATCH_VAR_SCAN_PRF_POS_Y	(18020)
// SCAN 运动状态	
#define WATCH_VAR_SCAN_RUN	(18200)
// SCAN 段号	
#define WATCH_VAR_SCAN_SEG_NUMBER	(18202)
// 激光 HSIO	

<code>#define WATCH_VAR_LASER_HSIO</code>	(18600)
<code>// 激光能量</code>	
<code>#define WATCH_VAR_LASER_POWER</code>	(18601)
<code>// AXIS 规划位置</code>	
<code>#define WATCH_VAR_AXIS_PRF_POS</code>	(20000)
<code>// AXIS 规划速度</code>	
<code>#define WATCH_VAR_AXIS_PRF_VEL</code>	(20001)
<code>// AXIS 规划加速度</code>	
<code>#define WATCH_VAR_AXIS_PRF_ACC</code>	(20002)
<code>// 编码器位置</code>	
<code>#define WATCH_VAR_ENC_POS</code>	(30000)
<code>// GPI</code>	
<code>#define WATCH_VAR_GPI</code>	(31000)
<code>// GPO</code>	
<code>#define WATCH_VAR_GPO</code>	(32000)
<code>// 捕获状态</code>	
<code>#define WATCH_VAR_TRIGGER_STATUS</code>	(38001)
<code>// 位置环跟随误差</code>	
<code>#define WATCH_VAR_POS_LOOP_ERROR</code>	(40000)
<code>// Watch 采集次数</code>	
<code>#define WATCH_VAR_WATCH_TIME</code>	