Softlink

Powered By Googoltech

iDEABOX 3

智能控制器编程手册
基本功能

V1.6

2018.12

www.softlinkcloud.cn

版权申明

上海固高欧辰智能科技有限公司

保留所有权力

上海固高欧辰智能科技有限公司（以下简称固高欧辰）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。

固高欧辰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。

固高欧辰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，固高欧辰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

上海固高欧辰智能科技有限公司

地　　址：上海闵行区东川路 555 号 4 号楼 1 层

客户服务：4006 300 321

电　　话：021-54708386　　54708786

传　　真：021-54708386

电子邮件：info@softlinkcloud.cn

网　　址：http://www.softlinkcloud.cn

文档版本

版本号	修订内容	修订日期
1.0		2016 年 05 月 16 日
1.1		2017 年 03 月 07 日
1.2		2017 年 08 月 21 日
1.3		2018 年 01 月 31 日
1.4		2018 年 03 月 21 日
1.5		2018 年 04 月 16 日
1.6		2018 年 12 月 28 日
1.7	邮件，网站信息更新	2020 年 02 月 19 日

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

前言

感谢选用固高欧辰控制器

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。

固高欧辰产品的更多信息

固高欧辰的网址是 http://www.softlinkcloud.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（4006 300 321）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： info@softlinkcloud.cn
电　　话： 4006 300 321
发 函 至：上海闵行区东川路 555 号 4 号楼 1 层
　　上海固高欧辰智能科技有限公司
邮　　编： 200241

编程手册的用途

用户通过阅读本手册，能够了解 iDEABOX 3 智能控制器的控制功能，掌握函数的用法，熟悉特定控制功能的编程实现。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

本手册由十四章内容组成。详细介绍了 iDEABOX 3 智能控制器的控制功能及编程实现。

相关文件

关于 iDEABOX 3 智能控制器调试和安装，请参见随产品配套的《iDEABOX 3 智能控制器用户手册》。

关于 iDEABOX 3 智能控制器配置文件及配置工具，请参见随产品配套的《EtherCAT 配置文件&配置工具 EthercatConfig 使用说明》。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

目录

版权申明 **1**

联系我们 **1**

文档版本 **2**

前言 **3**

目录 **4**

第 1 章 指令列表 **7**

第 2 章 OtoStudio 中运动函数库的使用 **12**

2.1 OtoStudio 软件库的使用 12

2.1.1 OtoStudio 平台中库的使用 12

第 3 章 指令返回值及其意义 **13**

3.1 本章简介 13

3.2 指令返回值 13

3.3 例程 13

第 4 章 系统配置 **15**

4.1 本章简介 15

4.2 系统配置基本概念 15

4.2.1 硬件资源 15

4.2.2 软件资源 15

4.2.3 资源组合 15

4.3 系统配置工具 17

4.3.1 配置 axis 18

4.3.2 配置 encoder 21

4.3.3 配置 profile 23

4.4 配置文件生成和下载 24

4.5 配置信息修改指令 25

4.5.1 指令列表 25

4.5.2 重点说明 26

4.5.3 例程 27

4.6 控制器配置初始化状态 27

第 5 章 EtherCAT 指令说明 **29**

5.1 本章简介 29

5.2 EtherCAT 库指令 29

5.2.1 指令列表 29

5.2.2 重点说明 30

5.2.2.1 EtherCAT 通讯 30

5.2.2.2 回零 30

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

5.2.2.3 探针功能 31
5.2.2.4 PDO 操作 31
5.2.2.5 EtherCAT IO 的使用 33
5.2.3 例程 34

第 6 章 运动状态检测 38

6.1 本章简介 38
6.2 指令列表 38
6.3 重点说明 39
6.3.1 轴状态定义 39
6.3.2 轴的运动参数 40
6.4 例程 40

第 7 章 运动模式 43

7.1 本章简介 43
7.2 点位运动模式 43
7.2.1 指令列表 43
7.2.2 重点说明 44
7.2.3 例程 44
7.3 Jog 运动模式 46
7.3.1 指令列表 46
7.3.2 重点说明 47
7.3.3 例程 47
7.4 PT 运动模式 49
7.4.1 指令列表 49
7.4.2 重点说明 50
7.4.3 例程 52
7.5 电子齿轮（Gear）运动模式 57
7.5.1 指令列表 57
7.5.2 重点说明 57
7.5.3 例程 59
7.6 Follow 运动模式 61
7.6.1 指令列表 61
7.6.2 重点说明 61
7.6.3 例程 65

第 8 章 访问硬件资源 76

8.1 本章简介 76
8.2 访问编码器 76
8.2.1 指令列表 76
8.2.2 例程 76

第 9 章 安全机制 78

9.1 本章简介 78
9.2 限位 78
9.2.1 指令列表 78

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

9.2.2 重点说明 79
9.2.3 例程 79
9.3 报警 81
9.4 平滑停止和急停 81

第 10 章 运动程序 82

10.1 本章简介 82
10.2 运动程序概述 82
10.3 运动程序的使用 83
10.3.1 编写运动程序 83
10.3.2 编译 83
10.3.3 指令列表 84
10.3.4 下载 84
10.3.5 绑定线程、函数和数据页 85
10.3.6 启动、停止、暂停线程 85
10.3.7 查询线程状态 85
10.3.8 例程 86
10.4 如何编写运动程序 89
10.4.1 语言元素 89
10.4.2 运算指令 90
10.4.3 流程控制与标准 C 语言的流程控制对比 90
10.5 可在运动程序中使用的指令 92

第 11 章 其它指令 94

11.1 本章简介 94
11.2 复位运动控制器 94
11.3 读取固件版本号 94
11.4 读取系统时钟 95
11.5 打开/关闭电机使能信号 95
11.6 维护位置值 95
11.7 电机到位检测 95
11.8 铁电功能 99
11.8.1 写数据至存储器 100
11.8.2 从存储器读数据 100
11.8.3 例程 100

第 12 章 加密机制 102

第 13 章 指令详细说明 103

第 14 章 索引 151

14.1 指令索引 151
14.2 例程索引 154
14.3 表格索引 154
14.4 图片索引 155

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第1章 指令列表

本章表格中右侧的数字为“页码”，其中指令右侧的为“**第 13 章 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 GT_AlarmOff）均带有超级链接，点击可跳转至指令说明。

表 1-1 指令列表

第 4 章 系统配置		15
4.4 配置文件生成和下载		**24**
GT_LoadConfig	下载配置信息到运动控制器	130
4.5 配置信息修改指令		**25**
GT_AlarmOff	控制轴驱动报警信号无效	103
GT_AlarmOn	控制轴驱动报警信号有效	103
GT_LmtsOn	控制轴限位信号有效	130
GT_LmtsOff	控制轴限位信号无效	129
GT_ProfileScale	设置控制轴的规划器当量变换值	133
GT_EncScale	设置控制轴的编码器当量变换值	108
GT_EncSns	设置编码器的计数方向	108
GT_EncOn	设置为“外部编码器”计数方式	107
GT_EncOff	设置为“脉冲计数器”计数方式	107
GT_SetPosErr	设置跟随误差极限值	142
GT_GetPosErr	读取跟随误差极限值	123
GT_SetStopDec	设置平滑停止减速度和急停减速度	144
GT_GetStopDec	读取平滑停止减速度和急停减速度	126

第 5 章 EtherCAT 指令说明		29
5.2 EtherCAT 库		**29**
GT_InitEcatComm	EtherCAT 初始化（OtoStudio 程序不需要执行）	129
GT_StartEcatComm	启动 EtherCAT 通讯（OtoStudio 程序不需要执行）	147
GT_TerminateEcatComm	结束 EtherCAT 通讯（OtoStudio 程序不需要执行）	149
GT_IsEcatReady	查询 GUC EtherCAT 通讯状态	129
GT_EcatSDODownload	通用 SDO 下载（Service Data Object，参考 IEC 61800）	106
GT_EcatSDOUpLoad	通用 SDO 上传（Service Data Object，参考 IEC 61800）	107
GT_SetEcatHomingPrm	设置 EtherCAT 轴回零的参数	136
GT_SetHomingMode	切换 EtherCAT 轴的回零模式	141
GT_StartEcatHoming	启动 EtherCAT 轴回零	147
GT_StopEcatHoming	停止 EtherCAT 轴回零	148
GT_GetEcatHomingStatus	查询 EtherCAT 轴的回零状态	116
GT_SetTouchProbeFunction	设置探针功能的参数（参数方式）	144

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

GT_SetTouchProbeFunctionEx	设置探针功能的参数（功能描述方式）	145
GT_GetTouchProbeStatus	查询 EtherCAT 轴的探针状态和捕获值	127
GT_EcatIOReadInput	读取 EtherCAT IO 模块数字量输入	106
GT_EcatIOWriteOutput	写入 EtherCAT IO 模块数字量输出	106
GT_GetEcatAxisAI	读取 EtherCAT 轴的模拟量输入	114
GT_GetEcatAxisDI	读取 EtherCAT 轴的数字量输入	114
GT_SetEcatAxisDO	设置 EtherCAT 轴的数字量输出	135
GT_GetEcatAxisDO	读取 EtherCAT 轴的数字量输出	114
GT_SetPosScale	设置编码器倍率值	142
GT_GetPosScale	读取编码器倍率值	123
GT_GetEcatEncPos	读取 EtherCAT 轴的编码器位置	116
GT_GetEcatEncVel	读取 EtherCAT 轴的编码器速度	116
GT_GetEcatAxisPE	读取 EtherCAT 轴的规划和编码器误差	115
GT_SetEcatAxisMode	设置 EtherCAT 轴的操作模式	136
GT_GetEcatAxisMode	读取 EtherCAT 轴的操作模式	115
GT_SetEcatAxisPT	设置 EtherCAT 轴的力矩	136
GT_GetEcatAxisAtlTorque	读取 EtherCAT 轴的力矩值	114
GT_GetEcatAxisAtlCurrent	读取 EtherCAT 轴的电流值	114
GT_SetEcatAxisPV	设置 EtherCAT 轴的速度	137
GT_GetEcatSlaves	读取 EtherCAT 总线的在线站数目	117
GT_GetMcEcatAxisNum	读取 EtherCAT 伺服轴数	122
GT_GetEcatRawData	读取 EtherCAT 总线的 PDO 数据	117
GT_SetEcatRawData	设置 EtherCAT 总线的 PDO 数据	137

第 6 章 运动状态检测		38
GT_GetSts	读取轴状态	126
GT_ClrSts	清除驱动器报警标志、跟随误差越限标志、限位触发标志 1. 只有当驱动器没有报警时才能清除轴状态字的报警标志 2. 只有当跟随误差正常以后，才能清除跟随误差越限标志 3. 只有当离开限位开关，或者规划位置在软限位行程以内时才能清除轴状态字的限位触发标志	104
GT_GetPrfMode	读取轴运动模式	123
GT_GetPrfPos	读取规划位置	124
GT_GetPrfVel	读取规划速度	124
GT_GetPrfAcc	读取规划加速度	123
GT_GetAxisPrfPos	读取轴(axis)的规划位置值	113
GT_GetAxisPrfVel	读取轴(axis)的规划速度值	113
GT_GetAxisPrfAcc	读取轴(axis)的规划加速度值	112
GT_GetAxisEncPos	读取轴(axis)的编码器位置值	111
GT_GetAxisEncVel	读取轴(axis)的编码器速度值	112
GT_GetAxisEncAcc	读取轴(axis)的编码器加速度值	111
GT_GetAxisError	读取轴(axis)的规划位置值和编码器位置值的差值	112
GT_Stop	停止一个或多个轴的规划运动	148

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第 7 章 运动模式		43
GT_PrfTrap	设置指定轴为点位模式	132
GT_PrfJog	设置指定轴为 Jog 模式	132
GT_PrfPt	设置指定轴为 PT 模式	132
GT_PrfGear	设置指定轴为电子齿轮模式	131
GT_PrfFollow	设置指定轴为 Follow 模式	131
GT_GetPrfMode	查询指定轴的运动模式	123
7.2 点位运动模式		**43**
GT_PrfTrap	设置指定轴为点位运动模式	132
GT_SetTrapPrm	设置点位运动模式下的运动参数	146
GT_GetTrapPrm	读取点位运动模式下的运动参数	127
GT_SetPos	设置目标位置	142
GT_GetPos	读取目标位置	122
GT_SetVel	设置目标速度	147
GT_GetVel	读取目标速度	128
GT_Update	启动点位运动	149
7.3 Jog 运动模式		**46**
GT_PrfJog	设置指定轴为 Jog 运动模式	132
GT_SetJogPrm	设置 Jog 运动模式下的运动参数	141
GT_GetJogPrm	读取 Jog 运动模式下的运动参数	121
GT_SetVel	设置目标速度	147
GT_GetVel	读取目标速度	128
GT_Update	启动 Jog 运动	149
7.4 PT 运动模式		**49**
GT_PrfPt	设置指定轴为 PT 运动模式	132
GT_PtSpace	查询 PT 运动模式指定 FIFO 的剩余空间	134
GT_PtData	向 PT 运动模式指定 FIFO 增加数据	133
GT_PtClear	清除 PT 运动模式指定 FIFO 中的数据 运动状态下该指令无效 动态模式下该指令无效	133
GT_SetPtLoop	设置 PT 运动模式循环执行的次数 动态模式下该指令无效	143
GT_GetPtLoop	查询 PT 运动模式循环执行的次数 动态模式下该指令无效	125
GT_PtStart	启动 PT 运动	134
GT_SetPtMemory	设置 PT 运动模式的缓存区大小	143
GT_GetPtMemory	读取 PT 运动模式的缓存区大小	125
7.5 电子齿轮（Gear）运动模式		**57**
GT_PrfGear	设置指定轴为电子齿轮运动模式	131
GT_SetGearMaster	设置电子齿轮运动模式跟随主轴	140
GT_GetGearMaster	读取电子齿轮运动模式跟随主轴	120
GT_SetGearRatio	设置电子齿轮比	140

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

GT_GetGearRatio	读取电子齿轮比	121
GT_GearStart	启动电子齿轮运动	110
7.6 Follow 运动模式		**61**
GT_PrfFollow	设置指定轴为 Follow 运动模式	131
GT_SetFollowMaster	设置 Follow 运动模式跟随主轴	139
GT_GetFollowMaster	读取 Follow 运动模式跟随主轴	119
GT_SetFollowLoop	设置 Follow 运动模式循环次数	138
GT_GetFollowLoop	读取 Follow 运动模式循环次数	119
GT_SetFollowEvent	设置 Follow 运动模式启动跟随条件	138
GT_GetFollowEvent	读取 Follow 运动模式启动跟随条件	118
GT_FollowSpace	查询 Follow 运动模式指定 FIFO 的剩余空间	109
GT_FollowData	向 Follow 运动模式指定 FIFO 增加数据	109
GT_FollowClear	清除 Follow 运动模式指定 FIFO 中的数据 运动状态下该指令无效	108
GT_FollowStart	启动 Follow 运动	109
GT_FollowSwitch	切换 Follow 运动模式所使用的 FIFO	110
GT_SetFollowMemory	设置 Follow 运动模式的缓存区大小	139
GT_GetFollowMemory	读取 Follow 运动模式的缓存区大小	120

第 8 章 访问硬件资源		**76**
8.2 访问编码器		**76**
GT_GetEncPos	读取编码器位置	118
GT_GetEncVel	读取编码器速度	118
GT_SetEncPos	修改编码器位置	137

第 9 章 安全机制		**78**
9.2 限位		**78**
GT_SetSoftLimit	设置轴正向软限位和负向软限位	144
GT_GetSoftLimit	读取轴正向软限位和负向软限位	125

第 10 章 运动程序		**82**
GT_Download	下载运动程序到运动控制器	105
GT_GetFunId	读取运动程序中函数的标识	120
GT_GetVarId	读取运动程序中变量的标识	128
GT_Bind	绑定线程、函数、数据页	104
GT_RunThread	启动线程	135
GT_StopThread	停止正在运行的线程	149
GT_PauseThread	暂停正在运行的线程	130
GT_GetThreadSts	读取线程的状态	126
GT_SetVarValue	设置运动程序中变量的值	146
GT_GetVarValue	读取运动程序中变量的值	128

第 11 章 其它指令	**94**

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

11.2 复位运动控制器		**94**
GT_Reset	复位运动控制器	135
11.3 读取固件版本号		**94**
GT_GetVersion	读取的运动控制器的固件版本号	129
11.4 读取系统时钟		**95**
GT_GetClock	读取的运动控制器的时钟	113
11.5 打开/关闭电机使能信号		**95**
GT_AxisOn	打开伺服使能的轴的编号	104
GT_AxisOff	关闭伺服使能的轴的编号	103
11.6 维护位置值		**95**
GT_SetPrfPos	修改指定轴的规划位置	142
GT_SynchAxisPos	axis 合成规划位置和所关联的 profile 同步 axis 合成编码器位置和所关联的 encoder 同步	149
GT_ZeroPos	清零规划位置和实际位置，并进行零漂补偿	150
11.7 电机到位检测		**95**
GT_SetAxisBand	设置轴到位误差带 规划器静止，规划位置和实际位置的误差小于设定误差带，并且在误差带内保持设定时间后，置起到位标志	135
GT_GetAxisBand	读取轴到位误差带	111
11.8 铁电功能		**99**
GT_SetMRAMData	传输数据到控制器	142
GT_GetMRAMData	从控制器读取数据	122
GT_StartAccessMRAM	启动存储器读/写操作	147
GT_StopAccessMRAM	停止存储器读/写操作	148
GT_GetMRAMStatus	获取存储器读/写完成状态	122

第 12 章 加密机制		**102**
GetMacAddress	读取网卡的物理地址	103

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第2章 OtoStudio 中运动函数库的使用

2.1 OtoStudio 软件库的使用

使用 CPAC 系列运动控制器，首先需要安装 OtoStudio 开发环境，运动控制器指令函数库将存放在默认路径下。iDEABOX 3 控制器的基础运动控制库文件名为 IDB3_Standard.lib。由于运动控制器要使用 EtherCAT 总线，因此还需要调用 EtherCAT 专用库，库文件名为 IDB3_ECAT.lib。

2.1.1 OtoStudio 平台中库的使用

(1) 启动OtoStudio.exe，新建一个工程；

(2) 选择目标平台：iDeabox3 PAC Controller。如果目标系统中没有该选项，请从官网上下载IDB3 Target，并参阅《ReadMe》安装ideabox3.trg；

(3) 选择正确的目标系统后，库文件管理器自动添加IDB3_Standard.lib和IDB3_ECAT.lib；

至此，用户就可以在 OtoStudio 中调用函数库中的任何函数，开始编写应用程序。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第3章 指令返回值及其意义

3.1 本章简介

本章主要介绍了运动控制器指令的所有返回值及其意义。在'第 13 章 指令详细说明'，每一条指令介绍的'指令返回值'一项中，都有更加详细的关于返回值意义和操作的介绍。

3.2 指令返回值

CPAC 控制器按照主机发送的指令工作，相关指令封装在动态链接库中。用户在编写应用程序时，通过调用运动控制库 IDB3_Standard.lib 指令来操纵 CPAC 控制器的运动控制。

CPAC 控制器在接收到主机发送的指令时，将执行结果反馈到主机，指示当前指令是否正确执行。指令返回值的定义如下。

表 3-1 运动控制器指令返回值定义

返回值	意义	处理方法
0	指令执行成功	无
1	指令执行错误	1. 检查当前指令的执行条件是否满足
7	指令参数错误	1. 检查当前指令输入参数的取值
-1	主机和运动控制器通讯失败	1. 是否正确安装运动控制器驱动程序 2. 检查运动控制器是否接插牢靠 3. 更换主机 4. 更换控制器
-6	打开控制器失败	1. 是否正确安装运动控制器驱动程序 2. 是否调用了 2 次 GT_Open 指令 3. 其他程序是否已经打开运动控制器
-7	运动控制器没有响应	1. 更换运动控制器

3.3 例程

例程 3-1 检测 GTN 指令是否正常执行

检测 GTN 指令是否正常执行，该程序在 OtoStudio 工程下运行。

```
FUNCTION commandhandler : INT
VAR_INPUT
    command:STRING;
    error:INT;
END_VAR
VAR
    errInfo:STRING;
END_VAR
(* @END_DECLARATION := '0' *)
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
IF error<>0 THEN
    errInfo:= CONCAT(command,' = ');
    errInfo:= CONCAT(errInfo,INT_TO_STRING(error));
END_IF
END_FUNCTION

PROGRAM PLC_PRG
VAR
    rtn:INT;  (*指令返回值变量*)
    sts:DWORD;
END_VAR
(* @END_DECLARATION := '0' *)
rtn:= GT_GetSts(1,1,ADR(sts),1,0);
commandhandler('GT_GetSts',rtn); (*指令返回值校验*)
END_PROGRAM
```

 注意	建议在用户程序中，检测每条指令的返回值，以判断指令的执行状态。并建立必要的错误处理机制，保证程序安全可靠地运行。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第4章 系统配置

4.1 本章简介

在使用运动控制器进行各种操作之前，需要对运动控制器进行配置，使运动控制器的状态和各种工作模式能够满足客户的要求。这个过程，叫做系统配置。

在运动控制器管理软件 Motion Controller Toolkit 2008 中包括一个系统配置的组件，用户可以利用该组件来对运动控制器进行配置，配置完成之后，生成相应的配置文件*.cfg，用户在编程时，调用相关的指令，将配置信息传递给运动控制器，即可完成整个运动控制器的配置工作。用户也可以利用相关的指令完成运动控制器的配置。

本章表格中右侧的数字为“页码”，其中指令右侧的为“**第 13 章 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 GT_AlarmOff）均带有超级链接，点击可跳转至指令说明。

4.2 系统配置基本概念

运动控制器内部包含了各种软硬件资源，各种软硬件资源之间相互组合，即可实现运动控制器的各种应用。

4.2.1 硬件资源

编码器计数资源(encoder)：用来对外部编码器的脉冲输出进行计数。

4.2.2 软件资源

规划器资源(profile)：根据运动模式和运动参数实时计算规划位置和规划速度，生成所需的速度曲线，实时地输出规划位置。

伺服控制器资源(control)：根据伺服控制算法、控制参数、跟随误差实时地计算控制量。

轴资源(axis)：将软件资源、硬件资源进行组合，作为整体进行操作。其中包括驱动报警信号、正限位信号、负限位信号、平滑停止信号、紧急停止信号的管理；规划器输出的规划位置的当量变换；编码器计数位置的当量变换等功能。

4.2.3 资源组合

系统配置就是将上述的硬件资源和软件资源相互组合，并对各个资源的基本属性进行配置的过程。下面的两个例子描述了资源组合的基本概念。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

1. 开环控制模式（脉冲控制）

使用步进电机，或使用伺服电机的位置控制模式的运动控制系统，其基本配置如图 4-1 所示。

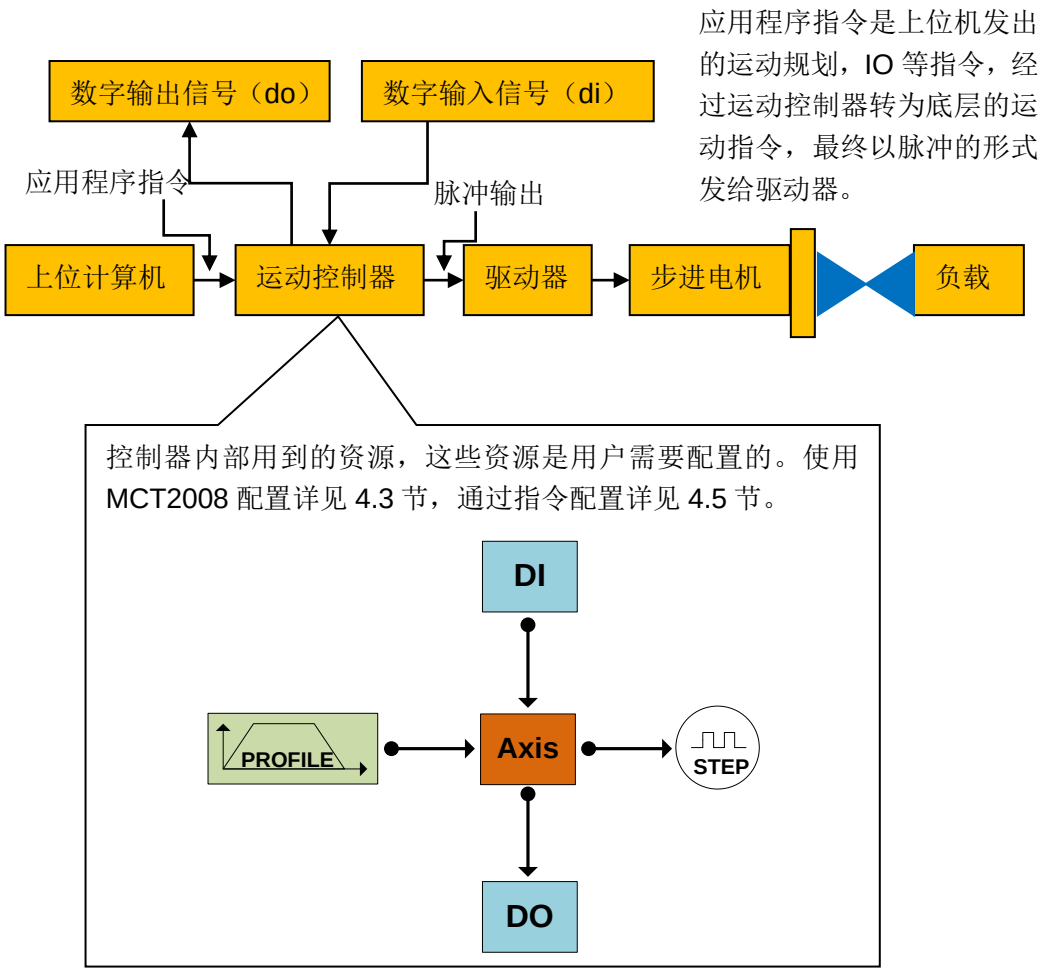

图 4-1 开环运动控制系统的配置

该实例中，profile 输出的规划位置进入 axis 中，在 axis 中进行当量变换的处理后，输出到 step，由 step 产生控制脉冲，驱动电机运动。axis 需要驱动报警、正向限位信号、负向限位信号、平滑停止信号、紧急停止信号等一些数字量输入信号来对运动进行管理；同时，axis 需要输出伺服使能信号，让电机使能。

2. 闭环控制模式（模拟量控制）

一个闭环伺服电机运动控制系统的常见组成部分如图 4-2 所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

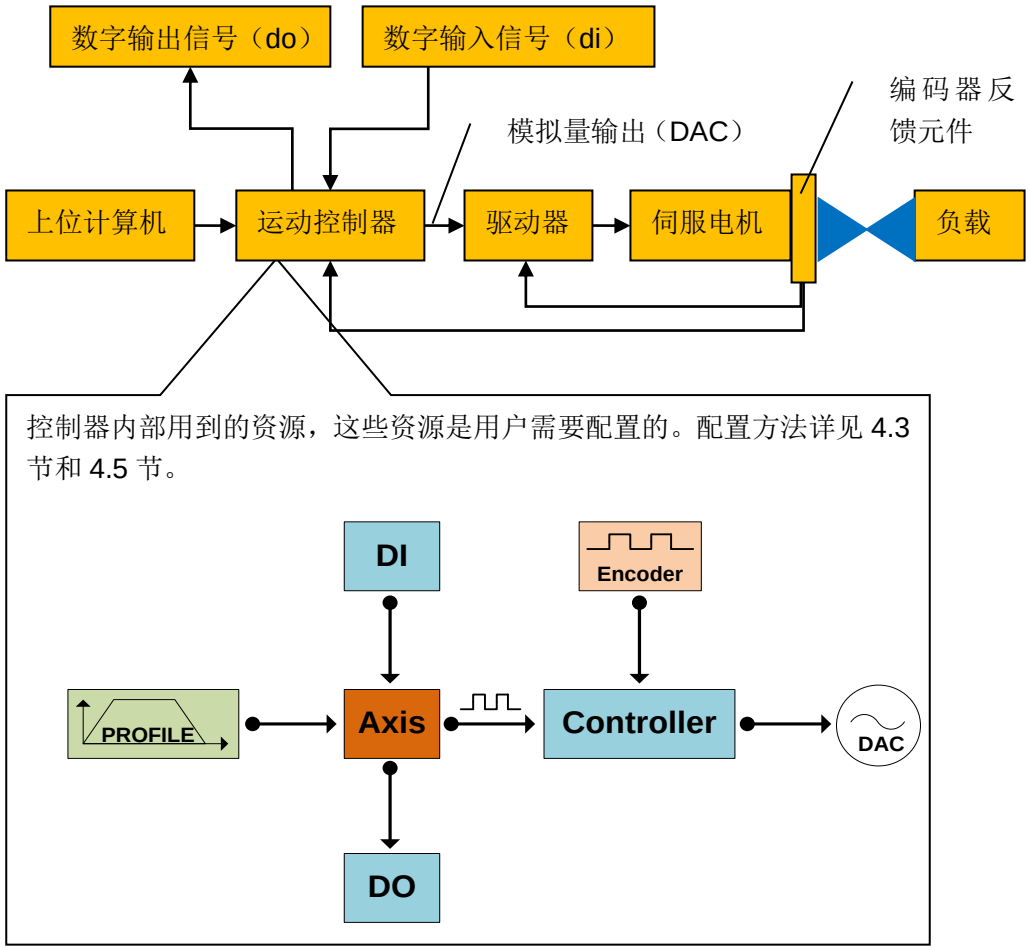

图 4-2 闭环运动控制系统的配置

该实例中，profile 输出的规划位置进入 axis 中，在 axis 中进行当量变换的处理后，输出到伺服控制器中，伺服控制器将规划位置与 encoder 的计数位置进行比较，获得跟随误差，并通过一定的伺服控制算法，得到实时的控制量，将控制量传递给 dac，由 dac 转换成控制电压来控制电机的运动。axis 需要驱动报警、正向限位信号、负向限位信号、平滑停止信号、紧急停止信号等一些数字量输入信号来对运动进行管理；同时，axis 需要输出伺服使能信号，让电机使能。

4.3 系统配置工具

使用固高欧辰提供的 Motion Controller Toolkit 2008 运动控制器管理软件能够方便地对系统进行配置，启动软件以后显示如图 4-3 所示界面。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

图 4-3 MCT2008 运动控制器管理软件界面

选择“工具”菜单，点击“控制器配置”，打开运动控制器配置面板就可以对系统进行配置。如图 4-4 所示。

图 4-4 打开控制器配置

4.3.1 配置 axis

Axis 配置界面如图 4-5 所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

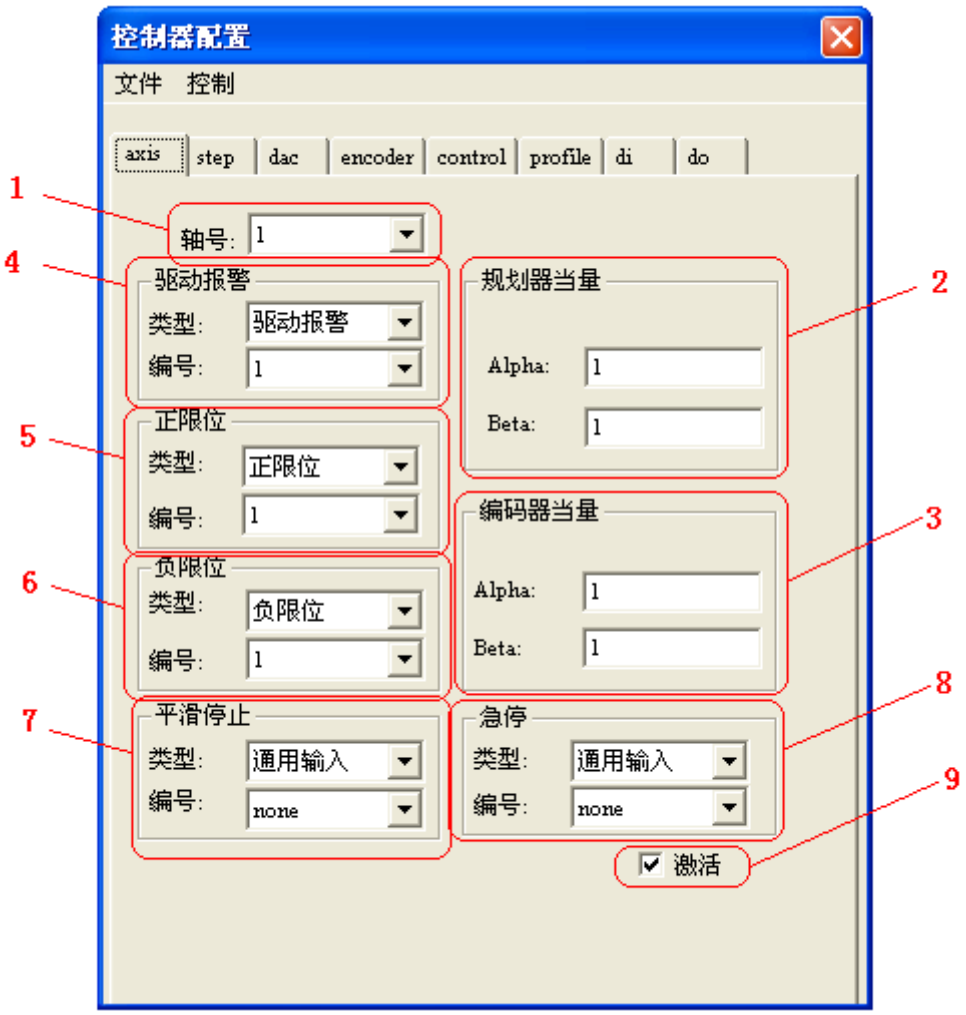

图 4-5 axis 配置界面

配置说明：'axis'选项主要用来配置轴控制的相关信息。配置后对控制系统可能产生的影响如图 4-6 所示。

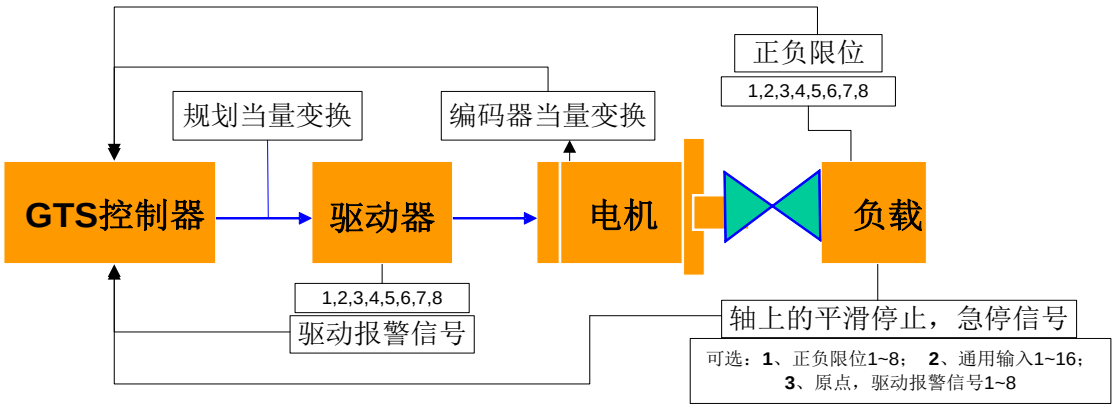

图 4-6 axis 配置对控制系统的影响

一般情况下，驱动报警、正负限位信号均保持默认设置状态，即'驱动报警'，'正限位'和'负限位'的'编号'同'轴号'相对应。为了帮助用户提高配置的灵活性，可选用其他配置，如将轴 2 的限位信号配给轴 1 使用。用户设置时，一般情况下，该选项保持默认设置，除非出现特殊需求。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

1. axis编号选择：选择需要进行配置的axis的编号。

2. 规划器当量变换参数：如果需要在axis中对规划器输出的规划位置进行当量变换，则可以对该项的参数进行设置，当量变换的关系如下：

$$\frac{\Delta P_{profile}}{\Delta P_{axis}} = \frac{Alpha}{Beta}$$

其中：

$\Delta P_{profile}$ ——规划器输出的规划位置的变化量

ΔP_{axis} ——axis 输出的规划位置的变化量

系统默认的 Alpha 和 Beta 都为 1，因此，规划器输出的规划位置在经过 axis 之后没有经过任何变化。Alpha 的取值范围：(-32767, 0)和(0, 32767)；Beta 的取值范围：(-32767, 0)和(0, 32767)。该项可以通过指令 GT_ProfileScale 来设置。

3. 编码器当量变换参数：如果需要在axis中对编码器计数的位置值进行当量变换，则可以对该项的参数进行设置，当量变换的关系如下：

$$\frac{\Delta E_{enc}}{\Delta E_{axis}} = \frac{Alpha}{Beta}$$

其中：

ΔE_{enc} ——编码器计数的位置值的变化量

ΔE_{axis} ——axis 输出的编码器位置值的变化量

系统默认的 Alpha 和 Beta 都为 1，因此，编码器计数的位置值在经过 axis 之后没有经过任何变化。Alpha 的取值范围：(-32767, 0)和(0, 32767)；Beta 的取值范围：(-32767, 0)和(0, 32767)。该项可以通过指令 GT_EncScale 来设置。

注意	规划器当量和编码器当量的设置参数要满足 Alpha≥Beta。

4. 驱动报警信号数字量输入选择：选择驱动报警信号的数字量输入的来源，运动控制器支持将任何数字量输入信号配置为驱动报警信号，增加用户进行硬件接线的自由性。该项的第一个下拉列表选择数字量输入的类型，默认为选择驱动报警数字量输入；第二个下拉列表选择数字量输入的编号，在第二个下拉列表中如果选择“none”，则表示该axis的驱动报警信号无效。驱动报警无效可以通过指令GT_AlarmOff设置，驱动报警有效可以通过指令GT_AlarmOn设置。

5. 正限位信号数字量输入选择：选择正限位信号的数字量输入的来源，运动控制器支持将任何数字量输入信号配置为正限位信号，增加用户进行硬件接线的自由性。该项的第一个下拉列表选择数字量输入的类型，默认为选择正限位数字量输入；第二个下拉列表选择数字量输入的编号，在第二个下拉列表中如果选择“none”，则表示该axis的正限位信号无效。限位开关无效可以通过指令

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

GT_LmtsOff设置，限位开关有效可以通过指令GT_LmtsOn设置。

6. 负限位信号数字量输入选择：选择负限位信号的数字量输入的来源，运动控制器支持将任何数字量输入信号配置为负限位信号，增加用户进行硬件接线的自由性。该项的第一个下拉列表选择数字量输入的类型，默认为选择负限位数字量输入；第二个下拉列表选择数字量输入的编号，在第二个下拉列表中如果选择“none”，则表示该axis的负限位信号无效。限位开关无效可以通过指令GT_LmtsOff设置，限位开关有效可以通过指令GT_LmtsOn设置。

驱动报警信号、正负限位信号在控制器复位状态下，都默认各轴对应相应的报警和限位信号，如 1 轴的驱动报警以及正负限位的编号都是 1 号。

但若有特殊情况，可以通过设置将不同轴上的限位和报警信号挂接到本轴上，如将 1 轴的正限位设置为 2 号。这种情况下，若 2 号正限位触发了，该信号将传递给 1 轴。

7. 平滑停止信号数字量输入选择：配置无效。

8. 紧急停止信号数字量输入选择：配置无效。

9. Axis 激活选项：如果 axis 不被激活，则与该 axis 相关的所有计算和管理任务将会无效。默认 axis 都是激活的。但是如果没有用到某个 axis 的相关功能，则可以不把该 axis 激活，这样可以节约运动控制器的处理资源。

4.3.2 配置 encoder

Encoder 配置界面如图 4-7 所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

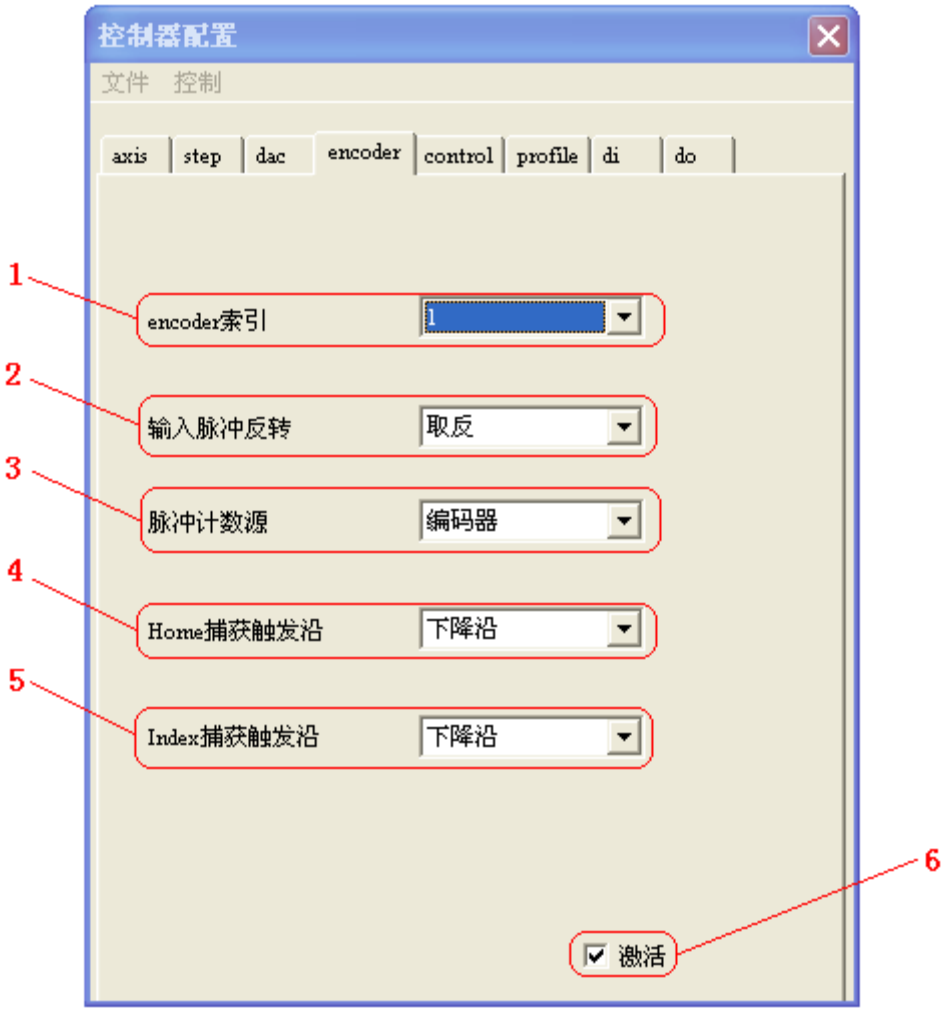

图 4-7 encoder 配置界面

配置说明：'encoder'选项主要配置与编码器有关的参数。配置后对控制系统可能产生的影响如图 4-8 所示。

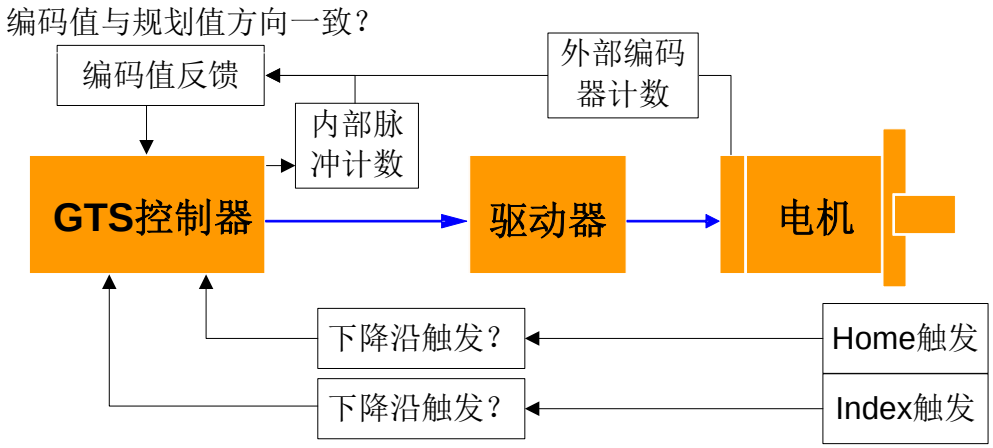

图 4-8 encoder 配置对控制系统的影响

当编码器值与规划值方向相反时，可以通过修改'输入脉冲反转'来校正。在闭环控制模式下，若出现'飞车'现象，也可通过修改该选项来校正。脉冲计数源的设置，一般情况下，保持默认设置。若没安装编码器，则可通过设置该选项为'脉冲计数器'。对于'Home 触发沿'和'Index 触发沿'的设置，取决于传感器的安

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

装，若不能触发，可以通过修改这部分。

1. Encoder 索引：选择需要进行配置的 encoder 的编号。
2. 输入脉冲反转：运动控制器可以接收正交编码器信号，该项选项与反馈脉冲方向以及编码器计数方向的关系如图 4-9 所示，该项可以通过指令 GT_EncSns 来修改。

	正常		取反	
A 相				
B 相				
编码器	计数增加	计数减少	计数增加	计数减少

图 4-9 encoder 输入脉冲反转项的影响

3. 脉冲计数源：表示编码器计数来源，默认情况下是外部编码器计数。如果没有外接编码器，则可以将其设置为脉冲计数器，encoder 将会对 step 输出的脉冲个数进行计数。设置为外部编码器，可以调用指令 GT_EncOn 来实现；设置为脉冲计数器，可以调用指令 GT_EncOff 来实现。

   设置为外部编码器是指通过外部安装的编码器计数值来计算，而脉冲计数器则是指通过控制器内部硬件来计算发出去的脉冲个数。在闭环控制方式下，必须设置成外部编码器计数方式。

4. Home 捕获触发沿：配置无效。
5. Index 捕获触发沿：配置无效。
6. Encoder 激活选项：如果 encoder 不被激活，则将不会对输入脉冲进行计数。默认 encoder 都是激活的。但是如果没有用到某个 encoder，则可以不把该 encoder 激活，这样可以节约运动控制器的处理资源。

4.3.3 配置 profile

Profile 配置界面如图 4-10 所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

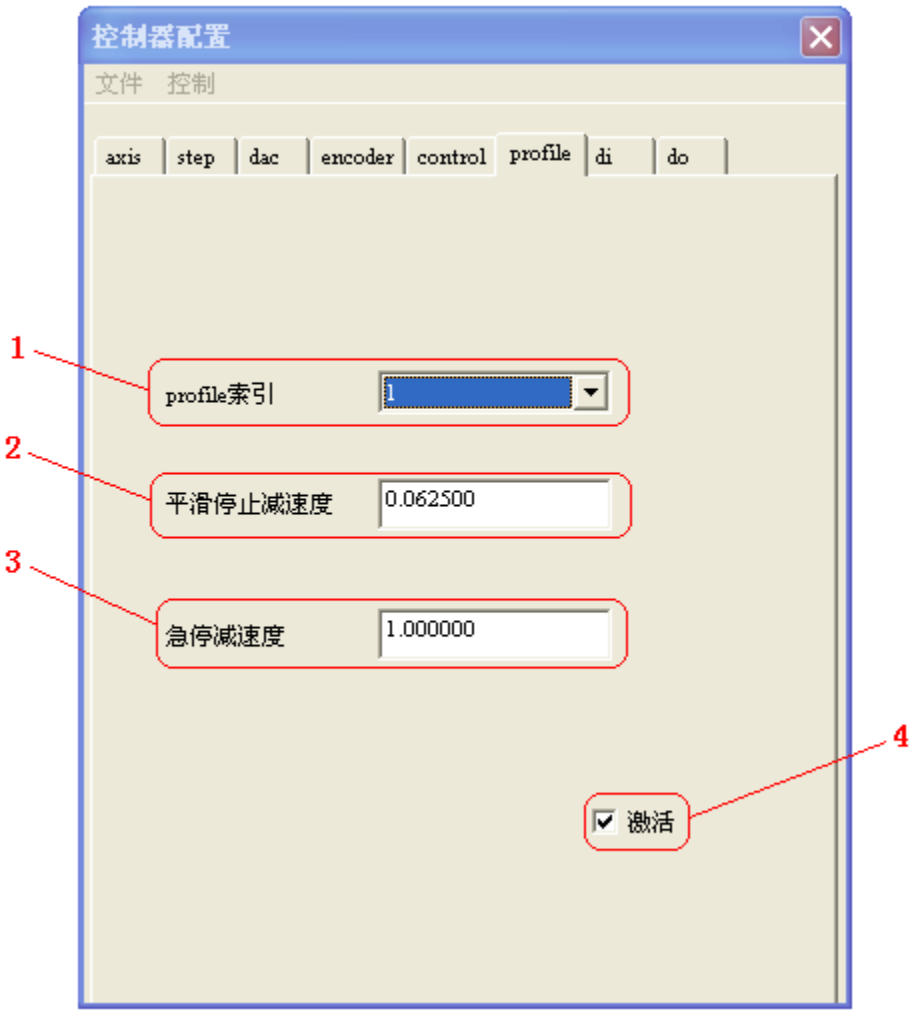

图 4-10 profile 配置界面

1. Profile 索引：选择需要进行配置的 profile 的编号。
2. 平滑停止减速度：表示调用指令 GT_Stop 平滑停止时所使用的减速度，默认为 1pulse/ms2。该值可以通过调用指令 GT_SetStopDec 来修改。
3. 紧急停止减速度：表示调用 GT_Stop 指令急停时所使用的减速度，默认为 1000pulse/ms2。该值可以通过调用指令 GT_SetStopDec 来修改。
4. Profile 激活选项：如果 profile 不被激活，则将不会进行运动规划。默认 profile 都是激活的。但是如果没有用到某个 profile，则可以不把该 profile 激活，这样可以节约运动控制器的处理资源。

4.4 配置文件生成和下载

表 4-1 下载配置文件指令

指令	说明	页码
GT_LoadConfig	下载配置信息到运动控制器，调用该指令后需再调用 GT_ClrSts 才能使该指令生效	130

按照 4.3 节中所描述进行运动控制器的配置之后，如图 4-11 所示，在“控制器配置”界面的“文件”菜单，点击“写入到文件”，即可对配置信息进行保存，生成配置文件*.cfg。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

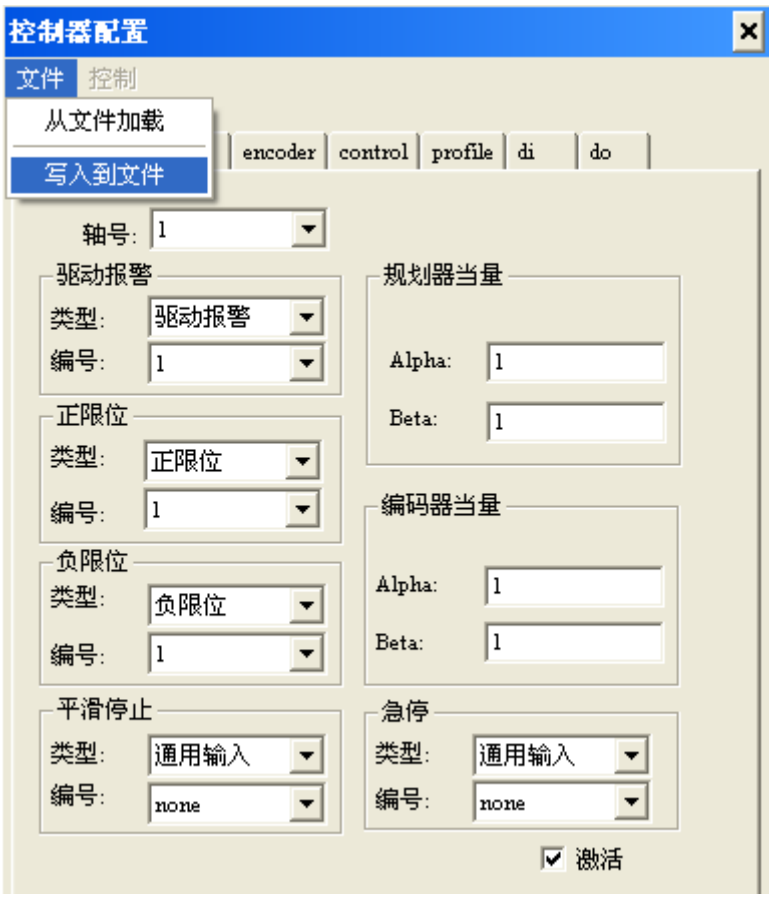

图 4-11 生成配置文件界面

用户可以调用 GT_LoadConfig 指令将配置文件里的配置信息下载到运动控制器中，需要注意的是，如果配置文件与可执行文件不在同一目录下，在调用 GT_LoadConfig 指令时，参数需要包含配置文件的绝对路径。

注意	GT_LoadConfig 中的文件名长度不允许超过 125 个字，否则调用指令将会失败。

4.5 配置信息修改指令

用户除了可以使用上面所述的配置文件的方式实现运动控制器的初始化配置外，还可以使用指令的方式来实现初始化配置。

4.5.1 指令列表

表 4-2 配置信息指令列表

指令	说明	页码
GT_AlarmOff	控制轴驱动报警信号无效	103
GT_AlarmOn	控制轴驱动报警信号有效	103
GT_EncSns	设置编码器的计数方向	108
GT_EncOn	设置为“外部编码器”计数方式	107

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

GT_EncOff	设置为“脉冲计数器”计数方式	107
GT_LmtsOn	控制轴限位信号有效	130
GT_LmtsOff	控制轴限位信号无效	129
GT_ProfileScale	设置控制轴的规划器当量变换值	133
GT_EncScale	设置控制轴的编码器当量变换值	108
GT_SetPosErr	设置跟随误差极限值	142
GT_GetPosErr	读取跟随误差极限值	123
GT_SetStopDec	设置平滑停止减速度和急停减速度	144
GT_GetStopDec	读取平滑停止减速度和急停减速度	126

4.5.2 重点说明

1. 编码器计数方向设置

指令 GT_EncSns 可以修改运动控制器各编码器的计数方向，当指令参数的某个状态位为 1 时，将所对应的控制轴的编码器计数方向取反，指令参数的状态位定义如表 4-3 所示：

表 4-3 编码器计数方向设置

状态位	11	10	9	8	7	6	5	4	3	2	1	0
编码器	Enc12	Enc11	Enc10	Enc9	Enc8	Enc7	Enc6	Enc5	Enc4	Enc3	Enc2	Enc1

例程 4-1 设置编码器计数方向

设置编码器 1 计数方向取反，其余编码器计数方向不取反。

```
PROGRAM PLC_PRG
VAR
    rtn:INT;  (*指令返回值变量*)
END_VAR
(* @END_DECLARATION := '0' *)
rtn:= GT_EncSns(16#01);
END_PROGRAM
```

2. 设置限位触发报警无效

运动控制器默认的限位报警是有效的，在测试阶段，可通过指令 GTN_LmtsOff 将限位报警设置为无效。

例程 4-2 设置限位触发报警无效

设置轴 1 限位报警无效。

```
PROGRAM PLC_PRG
VAR
    rtn:INT;  (*指令返回值变量*)
END_VAR
(* @END_DECLARATION := '0' *)
rtn:= GT_LmtsOff(1, -1);          (*设置限位报警无效*)
END_PROGRAM
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

4.5.3 例程

(1) 复位运动控制器，GT_Reset；

(2) 检查相关轴驱动器报警信号有没有连接。（一般若采用步进电机，可能没有驱动器报警信号），若没有连接，则应该调用 GT_AlarmOff 指令，使驱动器报警无效，默认是有效的；

(3) 检查相关轴的限位开关，若没有连接，则需要通过调用 GT_LmtsOff，使限位无效，默认是有效的；若有连接，则要检查触发电平是否设置正确，需通过驱动器调试软件修改；

(4) 在确认前面两步操作之后，调用 GT_ClrSts，更新设置的状态；

(5) 调用 GT_AxisOn，使能驱动器，这样相应的电机便能工作了；

(6) 使轴运动，运动后，若出现编码器位置和规划位置方向不一致，则可通过调用 GT_EncSns 改变编码器的计数方向。

例程 4-3 初始化配置控制器

```
PROGRAM PLC_PRG
VAR
    rtn:INT;  (*指令返回值变量*)
END_VAR
(* @END_DECLARATION := '0' *)
rtn:= GT_Reset();            (*系统复位*)
commandhandler('GT_Reset', rtn);
rtn:= GT_AlarmOff(1);        (*配置轴1报警输出无效*)
commandhandler ('GT_AlarmOff', rtn);
rtn:= GT_LmtsOff(1,-1);          (*配置轴1正负限位无效*)
commandhandler('GT_LmtsOff', rtn);
rtn:= GT_ClrSts(1,1);        (*配置完成后要更新状态*)
commandhandler('GT_ClrSts', rtn);
rtn:= GT_AxisOn(1);          (*轴1伺服使能*)
commandhandler('GT_AxisOn', rtn);
END_PROGRAM
```

注意	通过指令配置控制器之后，用户必须调用指令 GT_ClrSts 更新状态，使得配置生效。很多初次使用的客户会容易忘记这一点。

4.6 控制器配置初始化状态

控制器初始化状态，是指调用指令 GT_Open 成功后，调用指令 GT_Reset 复位后的状态。此时所有的状态都为默认状态。想要让控制器回到初始状态，只要调用 GT_Reset 指令即可。

表 4-4 所示为复位后，控制器配置选项的默认状态，调用第四栏'相关指令'中的指令可以修改对应选项的状态。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

表 4-4 控制器配置初始化状态

资源	配置选项	默认状态	相关指令
axis	该资源是否激活	激活	/
	规划器当量	Alpha 和 Beta 都为 1	GT_ProfileScale
	编码器当量	Alpha 和 Beta 都为 1	GT_EncScale
	正负限位输入	有效，编号与轴号对应	GT_LmtsOn GT_LmtsOff
	驱动报警输出	有效，编号与轴号对应	GT_AlarmOn GT_AlarmOff
encoder	该资源是否激活	激活	/
	脉冲计数源	外部编码器	GT_EncOn GT_EncOff
	输入脉冲反转	取反	GT_EncSns
profile	该资源是否激活	激活	/
	平滑停止减速度	1 pulse/ms^2	GT_SetStopDec
	急停的减速度	1000 pulse/ms^2	

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第5章 EtherCAT 指令说明

5.1 本章简介

本章重点介绍 iDEABOX 3 控制器使用 EtherCAT 总线相关的指令，库文件名为 IDB3_ECAT.lib。以及在非 OtoStudio 软件平台下使用 EtherCAT 总线需要额外调用的指令。

本章表格中右侧的数字为“页码”，其中指令右侧的为“**第 13 章 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 GT_AlarmOff）均带有超级链接，点击可跳转至指令说明。

5.2 EtherCAT 库指令

5.2.1 指令列表

表 5-1 EtherCAT 库指令列表

指令	说明	页码
GT_InitEcatComm	EtherCAT 初始化（OtoStudio 程序不需要执行）	129
GT_StartEcatComm	启动 EtherCAT 通讯（OtoStudio 程序不需要执行）	147
GT_TerminateEcatComm	结束 EtherCAT 通讯（OtoStudio 程序不需要执行）	149
GT_IsEcatReady	查询 GUC EtherCAT 通讯状态	129
GT_EcatSDODownload	通用 SDO 下载（Service Data Object，参考 IEC 61800）	106
GT_EcatSDOUpLoad	通用 SDO 上传（Service Data Object，参考 IEC 61800）	107
GT_SetEcatHomingPrm	设置 EtherCAT 轴回零的参数	136
GT_SetHomingMode	切换 EtherCAT 轴的回零模式	141
GT_StartEcatHoming	启动 EtherCAT 轴回零	147
GT_StopEcatHoming	停止 EtherCAT 轴回零	148
GT_GetEcatHomingStatus	查询 EtherCAT 轴的回零状态	116
GT_SetTouchProbeFunction	设置探针功能的参数（参数方式）	144
GT_SetTouchProbeFunctionEx	设置探针功能的参数（功能描述方式）	145
GT_GetTouchProbeStatus	查询 EtherCAT 轴的探针状态和捕获值	127
GT_EcatIOReadInput	读取 EtherCAT IO 模块数字量输入	106
GT_EcatIOWriteOutput	写入 EtherCAT IO 模块数字量输出	106
GT_GetEcatAxisAI	读取 EtherCAT 轴的模拟量输入	114
GT_GetEcatAxisDI	读取 EtherCAT 轴的数字量输入	114
GT_SetEcatAxisDO	设置 EtherCAT 轴的数字量输出	135
GT_GetEcatAxisDO	读取 EtherCAT 轴的数字量输出	114
GT_SetPosScale	设置编码器倍率值	142
GT_GetPosScale	读取编码器倍率值	123

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

GT_GetEcatEncPos	读取 EtherCAT 轴的编码器位置	116
GT_GetEcatEncVel	读取 EtherCAT 轴的编码器速度	116
GT_GetEcatAxisPE	读取 EtherCAT 轴的规划和编码器误差	115
GT_SetEcatAxisMode	设置 EtherCAT 轴的操作模式	136
GT_GetEcatAxisMode	读取 EtherCAT 轴的操作模式	115
GT_SetEcatAxisPT	设置 EtherCAT 轴的力矩	136
GT_GetEcatAxisAtlTorque	读取 EtherCAT 轴的力矩值	114
GT_GetEcatAxisAtlCurrent	读取 EtherCAT 轴的电流值	114
GT_SetEcatAxisPV	设置 EtherCAT 轴的速度	137
GT_GetEcatSlaves	读取 EtherCAT 总线的在线站数目	117
GT_GetMcEcatAxisNum	读取 EtherCAT 伺服轴数	122
GT_GetEcatRawData	读取 EtherCAT 总线的 PDO 数据	117
GT_SetEcatRawData	设置 EtherCAT 总线的 PDO 数据	137

5.2.2 重点说明

5.2.2.1 EtherCAT 通讯

首先使用 EtherCATConfig 工具配置正确的 eni 文件，并将此 eni 文件存放在程序所在目录中，具体配置方法详见 EtherCATConfigTool 相关资料。

执行下载程序的动作时，eni 文件会同时下载到目标控制器，并进行 EtherCAT 初始化，程序必须首先调用 GT_IsEcatReady 指令，pStatus 返回值为 1 时表示 EtherCAT 从站和运动控制器实现正常通讯，返回其他值表示通讯未建立。

5.2.2.2 回零

EtherCAT 轴的回零，控制器只对驱动器做设置回零参数、切换回零模式、启动回零、读取回零状态、停止回零操作，除此之外回零的整个过程由驱动器完成。

控制器可以通过 PDO 或 SDO 发送相关指令给驱动器。如果通过 PDO 方式发送指令，则驱动器的对象字 6060、6061 必须要支持 PDO 操作，即在配置驱动器 PDO 时，将 6060、6061 勾选上，如图 5-1 所示。如果驱动器的对象字 6060 和 6061 不能配置成 PDO，则按默认方式，即 SDO 通讯。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

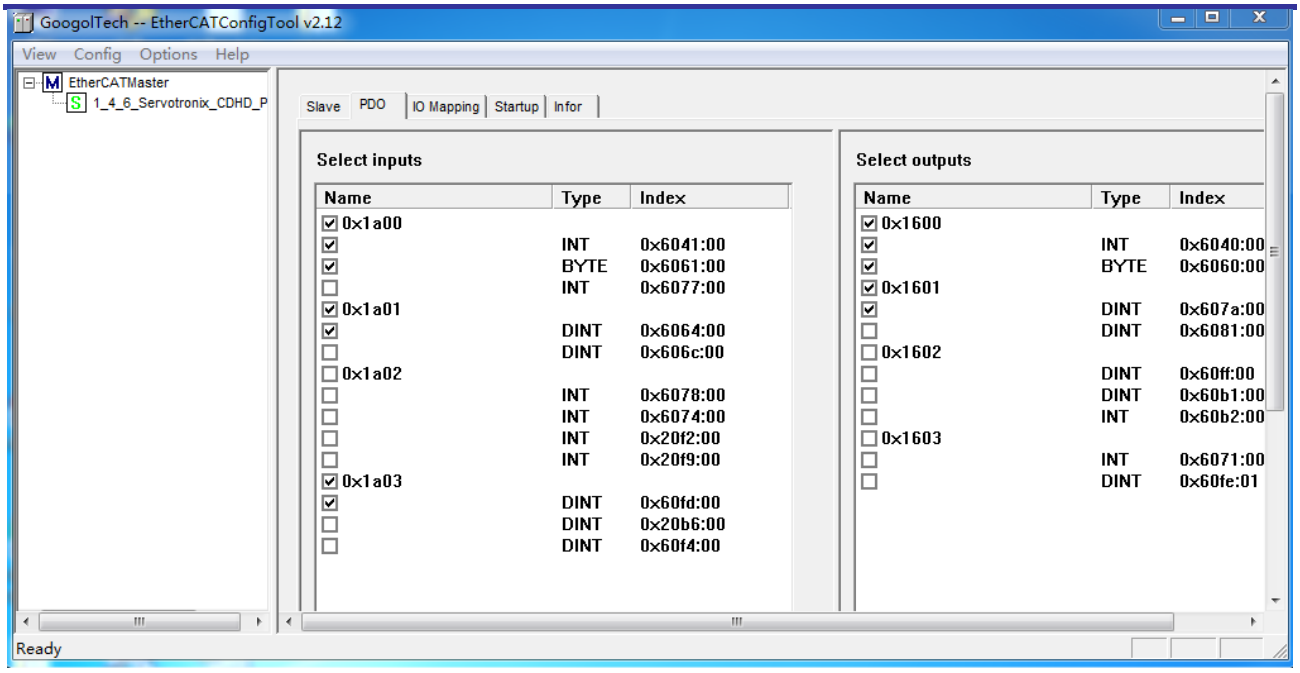

图 5-1 PDO 配置界面

GT_SetEcatHomingPrm 指令设置回零参数，具体参数说明见指令详细列表，其中回零方式由驱动器决定，不同驱动器支持的回零方式不同，要根据具体驱动器来选择回零方式。其中的 probeFunction 参数只有在使用探针相关回零方式时才有效。

注意	启动回零的三个必要条件： 1. 目标驱动器处于伺服使能状态； 2. 目标驱动器的操作模式是回零模式； 3. 已成功设置回零参数。 如果以上三个条件之一未满足，指令 GT_StartEcatHoming 将返回 1，并不做任何动作。

5.2.2.3 探针功能

控制器的探针捕获功能通过驱动器实现，控制器负责设置探针捕获方式和读取探针捕获状态。

控制器支持使用指令 GT_SetTouchProbeFunction 和 GT_SetTouchProbeFunctionEx 两种方式设置探针功能，区别只在于探针功能参数是按位表示还是具体描述方式。一般驱动器的探针功能支持单次捕获和连续捕获，单次即只记录一次位置值，连续捕获则自动刷新捕获位置值。捕获沿分为上升沿和下降沿，捕获的位置值分别存储于不同的参数中，使用指令 GT_GetTouchProbeStatus 可读取捕获状态和捕获位置值。

注意	探针捕获状态置位后，需要关闭探针功能才能复位。因此，连续捕获模式下，需要借助用户自定义状态位来判断是否触发新的捕获，可查阅驱动器手册。例如，GTHD 将第 6 位作为上升沿捕获状态刷新标识。

5.2.2.4 PDO 操作

控制器支持使用指令 GT_GetEcatRawData 和 GT_SetEcatRawData 直接对 EtherCAT 总线的 PDO 数据进行读写操作，但是要求用户熟知各对象字典的含义以及数据长度。PDO 偏移地址的计算方法如下：

(1) 如图 5-2 所示连接 4 个 EtherCAT 从站，其中 Slave0 和 Slave1 是 GTHD 伺服从站，Slave2 和

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

Slave3 是 IO 从站。各从站的 PDO 信息可在【IO Mapping】项目中查看。

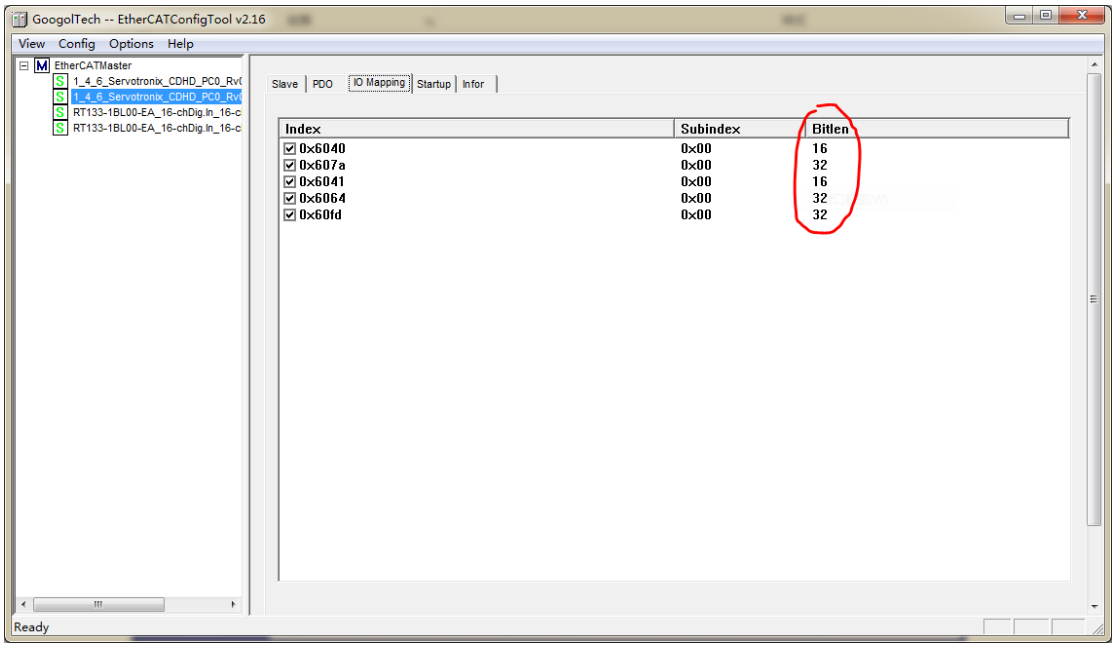

图 5-2 GTHD 默认配置之 PDO 信息

(2) Slave0 的 PDO 首地址是 0，之后的 PDO 偏移地址依次增加。首先假设 Slave0 和 Slave1 的 PDO 配置相同，都如上图所示。则 Slave0 的 PDO 0x6040 的偏移地址是 0 byte；0x6041 的偏移地址是 6 bytes；Slave1 的 PDO 0x6040 的偏移地址是 16 bytes；0x6041 的偏移地址是 22 bytes；Slave2 的 PDO 首地址是 32 bytes；Slave3 的 PDO 首地址是 36 bytes。

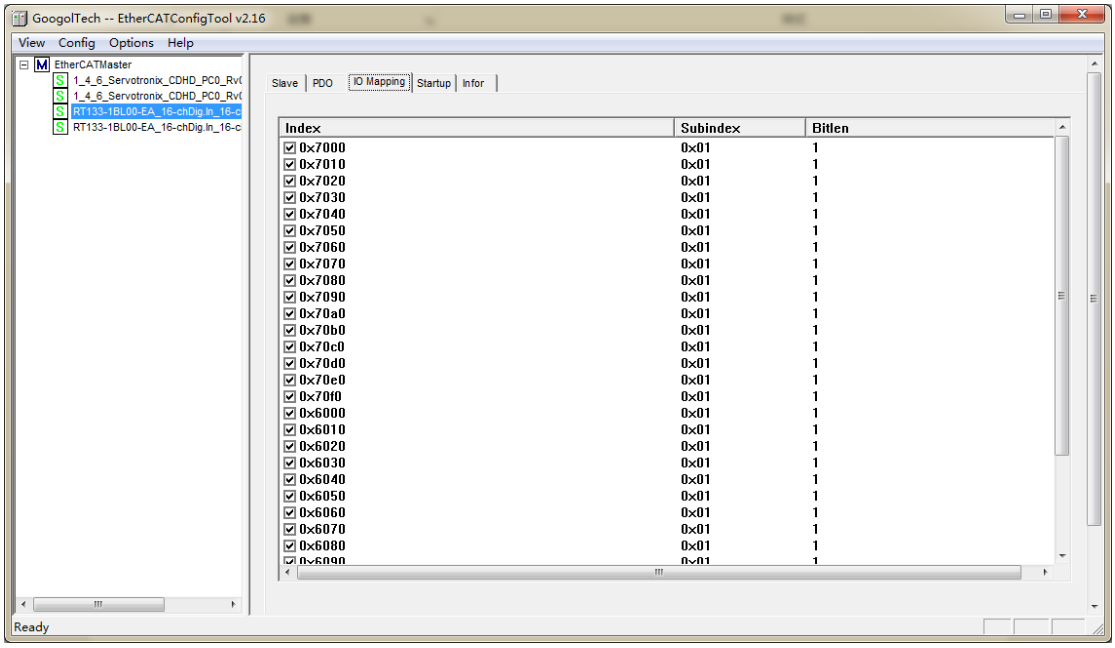

图 5-3 EtherCAT IO 默认配置之 PDO 信息

(3) 读取 Slave2（第一个 IO 模块）的 16 位输入，可调用指令 GT_GetEcatRawData(34, 2, ADR(wInput))；设置 Slave2 的 16 位输出，可调用指令 GT_SetEcatRawData(32, 2, ADR(wOutput))；读取 Slave3 的 16 位输入，可调用指令 GT_GetEcatRawData(38, 2, ADR(wInput))；设置 Slave2 的 16 位输出，可调用指令 GT_SetEcatRawData(36, 2,

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

ADR(wOutput))。

5.2.2.5 EtherCAT IO 的使用

EtherCAT IO 的数值可通过 5.2.2.4 章节描述的 PDO 的方式读取，也可以通过专用的 GT_EcatIOReadInput 和 GT_EcatIOWriteOutput 指令读取。以 EtherCAT 耦合器（IBM337-001-EC）、数字量输入模块（IBM311-160）、数字量输出模块（IBM321-160）、模拟量输入模块（IBM312-041）和模拟量输出模块（IBM322-041）顺序连接为例，PDO 配置信息如图 5-4 所示。可见耦合器连接的所有模块共同组成一个 EtherCAT 从站，并且从站内的 PDO 配置和实际连接的模块顺序相匹配。

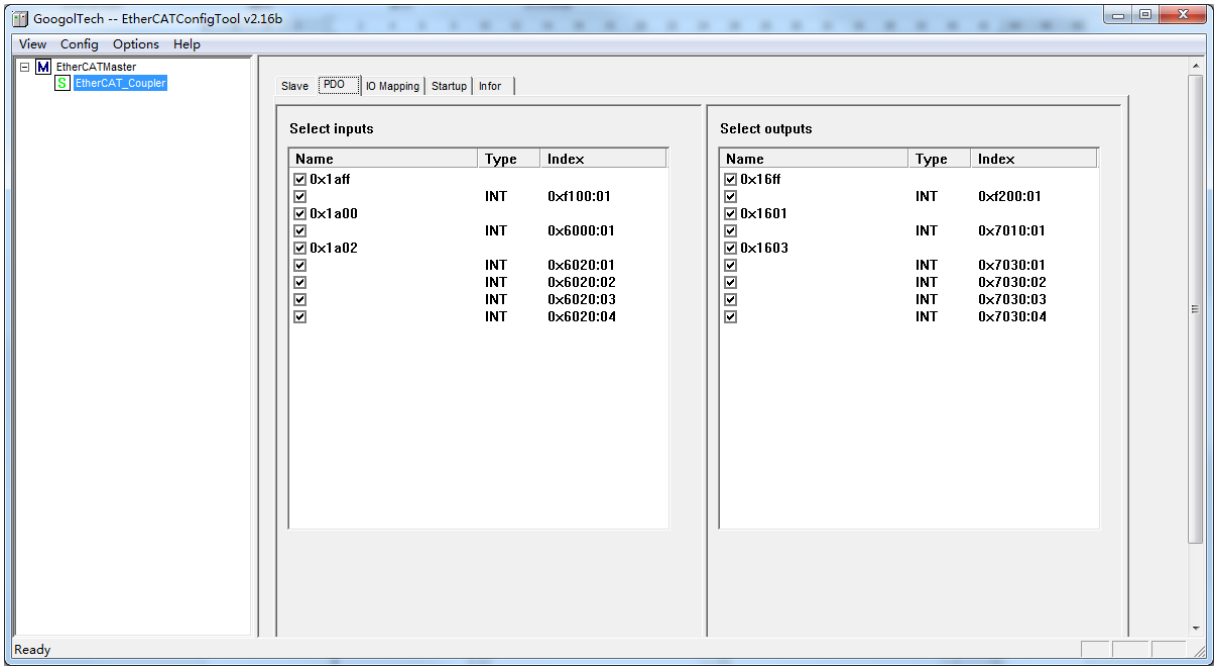

图 5-4 EtherCAT 耦合器和 IO 的 PDO 配置信息

首先，0x6000:01 和 0x6020:01~04 分别是数字量输入模块（IBM311-160）和模拟量输入模块（IBM312-041）的对象字典，0x7010:01 和 0x7030:01~04 分别是数字量输出模块（IBM321-160）和模拟量输出模块（IBM322-041）的对象字典。再看 IO Mapping 信息，如图 5-5 所示，所有的 output 数据都列在前面，input 数据都在后面，此时计算各自的偏移量就需要根据 IO Mapping 的信息进行。可见，0x7010:01 的偏移地址为 2bytes，0x7030:01~04 的偏移地址为 4bytes，0x6000:01 的偏移地址为 14bytes，0x6020:01~04 的偏移地址为 16bytes。

如果读取 IBM311-160 的输入值，调用指令 GT_EcatIOReadInput(1, 0, 14, 2, ADR(wDInput))；如果写入 IBM321-160 的输出值，则调用指令 GT_EcatIOWriteOutput(1, 0, 2, 2, ADR(wDOutput))；如果读取 IBM312-041 的第一个通道的模拟量输入值，调用指令 GT_EcatIOReadInput(1, 0, 16, 2, ADR(wAInput1))；如果读取 IBM312-041 的第二个通道的模拟量输入值，调用指令 GT_EcatIOReadInput(1, 0, 18, 2, ADR(wAInput2))；同时读取 IBM312-041 的第三和第四通道的模拟量输入值，调用指令 GT_EcatIOReadInput(1, 0, 20, 4, ADR(arrAInput34))；同时写入 IBM322-041 四个通道的模拟量输出值，调用指令 GT_EcatIOWriteOutput(1, 0, 4, 8, ADR(arrAOutput1234))。具体指令调用参考例程 5-4。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

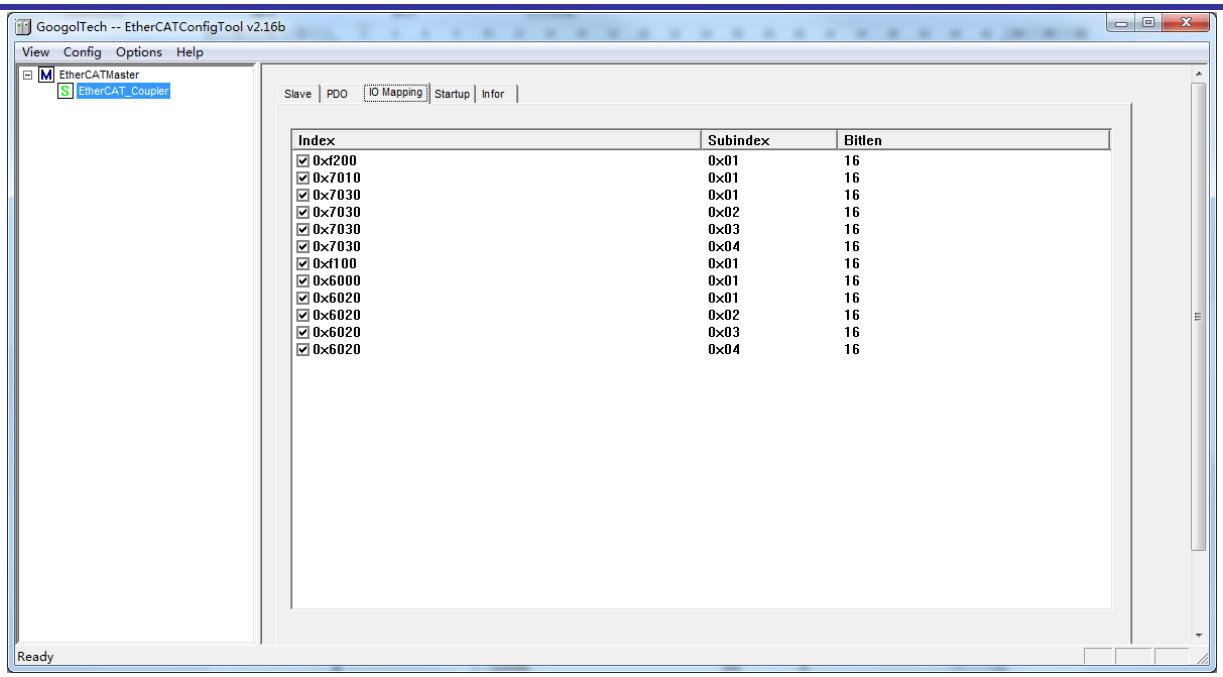

图 5-5 EtherCAT 耦合器和 IO 的 IO Mapping 信息

5.2.3 例程

例程 5-1 读取总线通讯状态

```
PROGRAM PLC_PRG
VAR
    rtn:INT;  (*指令返回值变量*)
    iEcatSts:INT; (*EtherCAT总线状态*)
END_VAR
(* @END_DECLARATION := '0' *)
rtn:= GT_IsEcatReady(ADR(iEcatSts));
IF iEcatSts <> 1 THEN
    RETURN;
END_IF
.....
END_PROGRAM
```

例程 5-2 采用 3 号回零方式

```
PROGRAM PLC_PRG
VAR
    iAxis:INT:= 1;
    dwAxisSts:DWORD;
    iHomeSts:INT;
    xStart:BOOL;
    xHoming:BOOL;
END_VAR
(* @END_DECLARATION := '0' *)
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
IF xStart THEN
    (*启动回零*)
    rtn:= GT_GetSts(iAxis, ADR(dwAxisSts), 1, 0);
    IF dwAxisSts.9 THEN        (*必须处于伺服使能状态*)
        (*切换到回零模式*)
        rtn:= GT_SetHomingMode(iAxis, 6);
        (*设置回零参数*)
        rtn:= GT_SetEcatHomingPrm(iAxis, 3, 5000, 3000, 100000, 0, 0);
        (*启动回零*)
        rtn:= GT_StartEcatHoming(iAxis);
        xHoming:= TRUE;
    END_IF
    xStart:= FALSE;
END_IF
(*检查回零状态*)
rtn:= GT_GetEcatHomingStatus(iAxis, ADR(iHomeSts));
IF iHomeSts = 3 THEN       (*回零完成*)
    rtn:= GT_SetHomingMode(iAxis, 8);       (*切换到位置控制模式*)
    xHoming:= FALSE;
END_IF
IF xStop THEN       (*中断、停止回零*)
    rtn:= GT_StopEcatHoming(iAxis);
    rtn:= GT_SetHomingMode(iAxis, 8);       (*切换到位置控制模式*)
    xHoming:= FALSE;
    xStop:= FALSE;
END_IF
END_PROGRAM
```

例程 5-3 探针 1 上升沿连续捕获（以 GTHD 为例）

```
PROGRAM PLC_PRG
VAR
    rtn:INT;  (*指令返回值变量*)
    iAxis:INT:= 1;
    xStartProbe:BOOL;
    xStopProbe:BOOL;
    iProbePrm:INT;
    uiProbeSts:UINT;       (*探针捕获状态*)
    diRiseValue1:DINT;     (*上升沿捕获值*)
    diFallValue1:DINT;     (*下降沿捕获值*)
    diRiseValue2:DINT;     (*上升沿捕获值*)
    diFallValue2:DINT;     (*下降沿捕获值*)
    diLastValue:DINT;      (*最新捕获值*)
    xPreSts:BOOL;
END_VAR
(* @END_DECLARATION := '0' *)
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
    IF xStartProbe THEN
        iProbePrm:= 16#0013;      (*探针1上升沿连续捕获，探针2无效*)
        rtn:= GT_SetTouchProbeFunction(iAxis, iProbePrm);
        xStartProbe:= FALSE;
    END_IF

    IF xStopProbe THEN
        iProbePrm:= 16#00;    (*终止探针捕获*)
        rtn:= GT_SetTouchProbeFunction(iAxis, iProbePrm);
        xStopProbe:= FALSE;
    END_IF

    rtn:= GT_GetTouchProbeStatus(iAxis, ADR(uiProbeSts), ADR(diRiseValue1),
ADR(diFallValue1), ADR(diRiseValue2), ADR(diFallValue2));
    IF uiProbeSts.0 THEN(*探针 1 有效*)
        IF uiProbeSts.1 THEN(*探针 1 上升沿捕获触发*)
            IF (uiProbeSts.6 <> xPreSts) THEN (*再次捕获触发，实际判断条件请查阅驱动器相关
            手册。GTHD 将第 6 位作为上升沿捕获状态刷新标识*)
                diLastValue:= diRiseValue1;
                xPreSts:= uiProbeSts.6;
            END_IF
        END_IF
        IF uiProbeSts.2 THEN(*探针 1 下降沿捕获触发*)
            ;
        END_IF
    END_IF
    IF uiProbeSts.8 THEN(*探针 2 有效*)
        ;
    END_IF
    END_PROGRAM
```

例程 5-4 EtherCAT IO 的使用

```
    PROGRAM PLC_PRG
    VAR
        rtn:INT;              (*指令返回值变量*)
        iSlave:INT:= 0;     (*从站号*)
        wDInput:WORD;
        wDOuput:WORD;
        wAInput1, wAInput2:WORD;
        arrAInput34:ARRAY [0..1] OF WORD;
        arrAOutput1234: ARRAY [0..3] OF WORD;
    END_VAR
    (* @END_DECLARATION := '0' *)
        ..... .....
        (*读取数字量模块的输入值*)
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        rtn:= GT_EcatIOReadInput(iSlave, 14, 2, ADR(wDInput));
        (*读取模拟量模块的输入值*)
        rtn:= GT_EcatIOReadInput(iSlave, 16, 2, ADR(wAInput1));
        rtn:= GT_EcatIOReadInput(iSlave, 18, 2, ADR(wAInput2));
        rtn:= GT_EcatIOReadInput(iSlave, 20, 4, ADR(wAInput34));

        (*写入数字量模块的输出值*)
        rtn:= GT_EcatIOWriteOutput(iSlave, 2, 2, ADR(wDOutput));
        (*写入模拟量模块的输出值*)
        rtn:= GT_EcatIOWriteOutput(iSlave, 4, 8, ADR(wAOutput1234));
        ..... .....
    END_PROGRAM
```

例程 5-5 C 或 C++环境 EtherCAT 初始化

```
#include <stdio.h>
#include "gts.h"
int main(int argc,char *argv[])
{
    short rtn;
    rtn = GT_Open(CHANNEL_PCI_ECAT);
    if(rtn){
        printf("Not find ecat device\n");
        return -1;
    }
    rtn = GT_InitEcatComm();
    if(rtn){
        printf("EtherCAT communication error\n");
            return -1;
    }
    //*wait untill EtherCAT comminication OK*/
    do {
        rtn = GT_IsEcatReady(&status);
    }
    while(status != 1 || rtn != 0);

    rtn = GT_StartEcatComm();
    rtn = GT_Reset();
    rtn = GT_LoadConfig("\\hard disk\\CPAC\\GTS800.cfg");
    //TODO: add other GT command below
    return 0;
}
```

注：程序的执行目录下需要包含 Gecat.eni 配置文件，此文件的生成请参考配置工具手册；同时 GTS800.cfg 可使用 MCT2008 生成。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第6章 运动状态检测

6.1 本章简介

当用户连接好一整套系统后（包括运动控制器，驱动器，电机），如何查看这套系统的运动状态？控制器可以帮助用户在应用程序中查看驱动器报警，当前运动位置，运动速度和加速度等一系列状态参数。本章将介绍用户可以检测到哪些状态以及如何检测。

本章表格中右侧的数字为“页码”，其中指令右侧的为“**第 13 章 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 GT_AlarmOff）均带有超级链接，点击可跳转至指令说明。

6.2 指令列表

表 6-1 运动状态检测指令列表

指令	说明	页码
GT_GetSts	读取轴状态	126
GT_ClrSts	清除驱动器报警标志、跟随误差越限标志、限位触发标志 4. 只有当驱动器没有报警时才能清除轴状态字的报警标志 5. 只有当跟随误差正常以后，才能清除跟随误差越限标志 6. 只有当离开限位开关，或者规划位置在软限位行程以内时才能清除轴状态字的限位触发标志	105
GT_GetPrfMode	读取轴运动模式	123
GT_GetPrfPos	读取规划位置	124
GT_GetPrfVel	读取规划速度	124
GT_GetPrfAcc	读取规划加速度	123
GT_GetAxisPrfPos	读取轴(axis)的规划位置值	113
GT_GetAxisPrfVel	读取轴(axis)的规划速度值	113
GT_GetAxisPrfAcc	读取轴(axis)的规划加速度值	112
GT_GetAxisEncPos	读取轴(axis)的编码器位置值	111
GT_GetAxisEncVel	读取轴(axis)的编码器速度值	112
GT_GetAxisEncAcc	读取轴(axis)的编码器加速度值	111
GT_GetAxisError	读取轴(axis)的规划位置值和编码器位置值的差值	112
GT_Stop	停止一个或多个轴的规划运动	148

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

6.3 重点说明

6.3.1 轴状态定义

用户可以从控制器的运动状态寄存器中读取轴状态。当调用 GT_GetSts 指令时，将返回一个 32 位的轴状态字，该轴状态字的定义如表 6-2 所示。

表 6-2 轴状态定义

位	定义
0	保留
1	驱动器报警标志 控制轴连接的驱动器报警时置 1
2	保留
3	保留
4	跟随误差越限标志 控制轴规划位置和实际位置的误差大于设定极限时置 1
5	正限位触发标志 正限位开关电平状态为限位触发电平时置 1 规划位置大于正向软限位时置 1
6	负限位触发标志 负限位开关电平状态为限位触发电平时置 1 规划位置小于负向软限位时置 1
7	IO 平滑停止触发标志 如果轴设置了平滑停止 IO，当其输入为触发电平时置 1，并自动平滑停止该轴
8	IO 急停触发标志 如果轴设置了急停 IO，当其输入为触发电平时置 1，并自动急停该轴
9	电机使能标志 电机使能时置 1
10	规划运动标志 规划器运动时置 1
11	电机到位标志 规划器静止，规划位置和实际位置的误差小于设定误差带，并且在误差带内保持设定时间后，置起到位标志
12~31	保留

驱动器报警标志、限位触发标志、IO 停止、跟随误差越限标志触发以后，不会自动清 0。只有当产生异常的原因已经消除以后，调用 GT_ClrSts 指令才能清除相应的异常标志。

规划运动状态(bit10)只表示理论上的运动状态。置 1 表示处于规划运动状态，清 0 表示处于规划静止状态。由于电机跟随滞后、机械系统震荡等原因，一般在规划静止一段时间以后，机械系统才能完全停止。

电机到位标志(bit11)表示实际到位状态。该标志位是电机到位检测功能的一部分。控制器复位后，默认该功能未开启。若要开启该功能，请参考 11.7 节的详细说明。该标志位置 1 表示已经处于规划静止状态(bit10=0)，并且规划位置和编码器位置的误差在设定的误差带内保持了设定时间。当规划运动或者规划位置和编码器位置的误差超出误差带时立即清 0。检测电机到位标志可以保证系统的定位精度，应当根据

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

机械系统的实际情况设置合适的到位误差带和误差带保持时间。如果到位误差带设置的太小，或者误差带保持时间太长，都会使到位时间增长，影响加工效率。

6.3.2 轴的运动参数

调用 GT_GetPrfMode 可以读取当前轴的运动模式。运动模式将在”第 7 章 运动模式”中详细介绍。控制器有如下几种运动模式：点位运动模式、Jog 模式、PT 模式、电子齿轮模式、Follow 模式、插补运动模式、PVT 模式，其中插补运动模式和 PVT 模式详见《iDEABOX 3 运动控制器编程手册之高级功能》。

“4.2.2 软件资源”中介绍过规划器的概念。规划器是位于控制器内部的，是根据用户所设置的运动模式和运动参数计算规划位置和规划速度的软件资源。它不代表电机系统的位置和速度。调用 GT_GetPrfPos，GT_GetPrfVel，GT_GetPrfAcc 可以读取规划器当前的位置，速度和加速度。

“4.2.2 软件资源”中介绍过轴的概念。轴亦是位于控制器内部的，是将规划器和编码器硬件资源整合起来的软件资源。轴的规划位置是规划器位置经过当量变换后的值，轴的编码器位置也是编码器硬件的位置经过当量变换后的值。默认情况下，规划器位置的当量与轴的规划位置的当量之比是 1:1，轴的编码器位置当量与编码器硬件的位置当量之比也是 1:1。调用 GT_GetAxisPrfPos，GT_GetAxisPrfVel，GT_GetAxisPrfAcc 可以读取轴的规划器当前的位置，速度和加速度。调用 GT_GetAxisEncPos，GT_GetAxisEncVel，GT_GetAxisEncAcc 可以读取轴的编码器当前的位置，速度和加速度。调用 GT_GetAxisError 读取轴的规划位置和轴的编码器位置的差值。

调用 GT_Stop 停止正在运动的轴。停止方式可以分为急停和平滑停止。具体介绍请参考”9.4 平滑停止和急停”。

6.4 例程

例程 6-1 获取轴 1 的轴状态、运动模式、位置、速度和加速度

```
PROGRAM PLC_PRG
VAR
    rtn:INT;                        (*指令返回值变量*)
    Axis:INT:=1;                    (*定义轴号: 1*)
    bFlagAlarm:BOOL:= FALSE;        (*伺服报警标志*)
    bFlagMError:BOOL:= FALSE;       (*跟随误差越限标志*)
    bFlagPosLimit:BOOL:= FALSE;     (*正限位触发标志*)
    bFlagNegLimit:BOOL:= FALSE;     (*负限位触发标志*)
    bFlagSmoothStop:BOOL:= FALSE;   (*平滑停止标志*)
    bFlagAbruptStop:BOOL:= FALSE;   (*急停标志*)
    bFlagServoOn:BOOL:= FALSE;      (*伺服使能标志*)
    bFlagMotion:BOOL:= FALSE;       (*规划器运动标志*)
    lrPrfPos:LREAL;                 (*规划位置*)
    lrPrfVel:LREAL;                 (*规划速度*)
    lrPrfAcc:LREAL;                 (*规划加速度*)
    diPrfMode:DINT;                 (*运动模式*)
    sPrfMode:STRING;                (*运动模式，字符串变量*)
    dwAxisStatus:DWORD;             (*轴状态*)
END_VAR
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
(* @END_DECLARATION := '0' *)
rtn:= GT_GetSts(Axis, ADR(dwAxisStatus),1,0);          (*读取轴状态*)
commandhandler('GT_GetSts', rtn);
(*伺服报警标志*)
IF dwAxisStatus.1 THEN
    bFlagAlarm = TRUE;
ELSE
    bFlagAlarm = FALSE;
END_IF
(*跟随误差越限标志*)
IF dwAxisStatus.4 THEN
    bFlagMError = TRUE;
ELSE
    bFlagMError = FALSE;
END_IF
(*正向限位*)
IF dwAxisStatus.5 THEN
    bFlagPosLimit = TRUE;
ELSE
    bFlagPosLimit = FALSE;
END_IF
(*负向限位*)
IF dwAxisStatus.6 THEN
    bFlagNegLimit = TRUE;
ELSE
    bFlagNegLimit = FALSE;
END_IF
(*平滑停止*)
IF dwAxisStatus.7 THEN
    bFlagSmoothStop = TRUE;
ELSE
    bFlagSmoothStop = FALSE;
END_IF
(*急停标志*)
IF dwAxisStatus.8 THEN
    bFlagAbruptStop = TRUE;
ELSE
    bFlagAbruptStop = FALSE;
END_IF
(*伺服使能标志*)
IF dwAxisStatus.9 THEN
    bFlagServoON = TRUE;
ELSE
    bFlagServoON = FALSE;
END_IF
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
(*规划器正在运动标志*)
IF dwAxisStatus.10 THEN
    bFlagMotion = TRUE;
ELSE
    bFlagMotion = FALSE;
END_IF

(*读取运动数据*)
rtn:= GT_GetPrfPos(Axis, ADR(lrPrfPos),1,0);
commandhandler('GT_GetPrfPos', rtn);
rtn:= GT_GetPrfVel(Axis, ADR(lrPrfVel),1,0);
commandhandler('GT_GetPrfVel', rtn);
rtn:= GT_GetPrfAcc(Axis, ADR(lrPrfAcc),1,0);
commandhandler('GT_GetPrfAcc', rtn);

(*读取运动模式*)
rtn:= GT_GetPrfMode(Axis, ADR(diPrfMode),1,0);
commandhandler('GT_GetPrfMode', rtn);
CASE diPrfMode OF
0:  sPrfMode:= 'Trap';
1:  sPrfMode:= 'Jog;
2:  sPrfMode:= 'PT';
3:  sPrfMode:= 'Gear;
4:  sPrfMode:= 'Follow;
5:  sPrfMode:= 'Interpolation;
6:  sPrfMode:= 'PVT';
END_CASE
END_PROGRAM
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第7章 运动模式

7.1 本章简介

运动模式是指规划一个或多个轴运动的方式。运动控制器支持的运动模式有点位运动模式、Jog 运动模式、PT 运动模式、电子齿轮（Gear）运动模式和电子凸轮（Follow）运动模式。

本章表格中右侧的数字为“页码”，其中指令右侧的为“**第 13 章 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 GT_AlarmOff）均带有超级链接，点击可跳转至指令说明。

用户需注意，每一个轴在任一时刻只能处在一种运动模式下。

表 7-1 设置运动模式指令列表

指令	说明	页码
GT_PrfTrap	设置指定轴为点位模式	132
GT_PrfJog	设置指定轴为 Jog 模式	132
GT_PrfPt	设置指定轴为 PT 模式	132
GT_PrfGear	设置指定轴为电子齿轮模式	131
GT_PrfFollow	设置指定轴为 Follow 模式	131
GT_GetPrfMode	查询指定轴的运动模式	123

在设置或切换运动模式时，要保证轴处于规划静止状态。如果轴正在运动，请先调用 GT_Stop 指令停止一个或多个轴的运动，然后再调用上表中的指令切换到想要的运动模式下。

7.2 点位运动模式

7.2.1 指令列表

表 7-2 点位运动模式指令列表

指令	说明	页码
GT_PrfTrap	设置指定轴为点位运动模式	132
GT_SetTrapPrm	设置点位运动模式下的运动参数	146
GT_GetTrapPrm	读取点位运动模式下的运动参数	127
GT_SetPos	设置目标位置	142
GT_GetPos	读取目标位置	122

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

GT_SetVel	设置目标速度	147
GT_GetVel	读取目标速度	128
GT_Update	启动点位运动	149

7.2.2 重点说明

每一个轴在规划静止时都可以设置为点位运动。在点位运动模式下，各轴可以独立设置目标位置、目标速度、加速度、减速段、起跳速度、平滑时间等运动参数，能够独立运动或停止。

调用 GT_Update 指令启动点位运动以后，控制器根据设定的运动参数自动生成相应的梯形曲线速度规划，并且在运动过程中可以随时修改目标位置和目标速度。

设定平滑时间能够得到平滑的速度曲线，从而使加减速过程更加平稳，如图 7-1 所示。

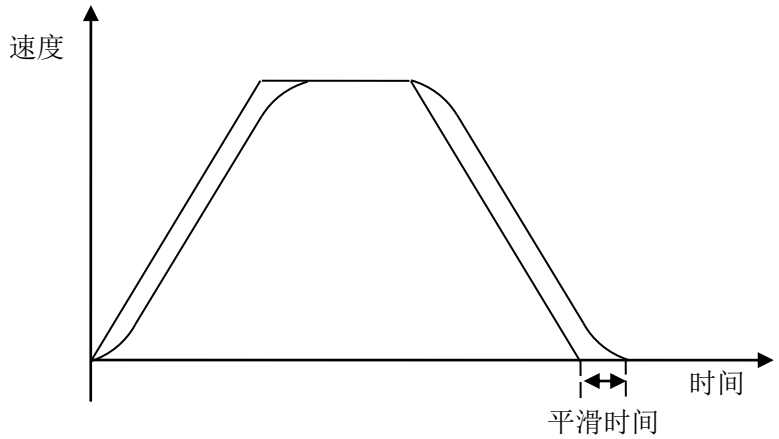

图 7-1 点位运动速度曲线

平滑时间是指加速度的变化时间，单位：ms，取值范围：[0, 50]。

7.2.3 例程

例程 7-1 点位运动

将第 1 轴设定为点位运动模式，并且以速度 50pulse/ms，加速度 0.25pulse/ms^2，减速度 0.125pulse/ms^2，平滑时间为 25ms 的运动参数正向运动 50000 个脉冲。

该例程生成一段梯形曲线速度规划，如图 7-2 所示：

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

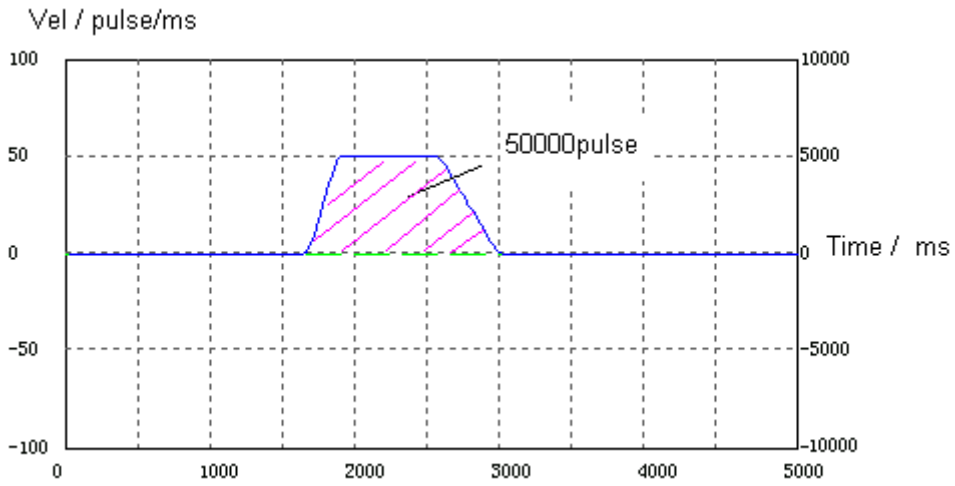

图 7-2 点位运动速度规划

```
PROGRAM PLC_PRG
VAR
    xInitDone:BOOL:= FALSE;
    rtn:INT;                    (*指令返回值变量*)
    Axis:INT:=1;                (*定义轴号: 1*)
    xStart: BOOL:= FALSE;
    trap:TTrapPrm;              (*点位运动参数*)
    lrPrfPos:LREAL;             (*规划位置*)
    dwAxisStatus:DWORD;         (*轴状态*)
END_VAR
(* @END_DECLARATION := '0' *)
IF NOT xInitDone THEN
    (*配置运动控制器*)
    (*注意：配置文件取消了各轴的报警和限位*)
    rtn:= GT_LoadConfig('test.cfg);
    commandhandler('GT_LoadConfig', rtn);
    (*清除各轴的报警和限位*)
    rtn:= GT_ClrSts(1,8);
    commandhandler('GT_ClrSts', rtn);
    (*伺服使能*)
    rtn:= GT_AxisOn(Axis);
    commandhandler('GT_AxisOn', rtn);
    xInitDone:= TRUE;
END_IF
IF xStart THEN
    (*位置清零*)
    rtn:= GT_ZeroPos(Axis,1);
    commandhandler('GT_ZeroPos', rtn);
    (*AXIS轴规划位置清零*)
    rtn:= GT_SetPrfPos(Axis, 0);
    commandhandler('GT_SetPrfPos', rtn);
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        (*将 AXIS 轴设为点位模式*)
        rtn:= GT_PrfTrap(Axis);
        commandhandler('GT_PrfTrap', rtn);
        (*读取点位运动参数(需要读取所有运动参数到上位机变量*)
        rtn:= GT_GetTrapPrm(Axis, ADR(trap));
        commandhandler('GT_GetTrapPrm', rtn);
        (*设置需要修改的运动参数*)
        trap.acc := 0.25;
        trap.dec := 0.125;
        trap.smoothTime := 25;
        (*设置点位运动参数*)
        rtn:= GT_SetTrapPrm(Axis, ADR(trap));
        commandhandler('GT_SetTrapPrm', rtn);
        (*设置 AXIS 轴的目标位置*)
        rtn:= GT_SetPos(Axis, 50000);
        commandhandler('GT_SetPos', rtn);
        (*设置AXIS轴的目标速度*)
        rtn:= GT_SetVel(Axis, 50);
        commandhandler('GT_SetVel', rtn);
        (*启动AXIS轴的运动*)
        rtn:= GT_Update(SHL(WORD#1,Axis-1));
        commandhandler('GT_Update', rtn);
        xStart:= FALSE;
    END_IF
    (*读取AXIS轴的状态*)
    rtn:= GT_GetSts(Axis, ADR(dwAxisStatus),1,0);
    (*读取AXIS轴的规划位置*)
    rtn:= GT_GetPrfPos(Axis, ADR(lrPrfPos),1,0);
    END_PROGRAM
```

7.3 Jog 运动模式

7.3.1 指令列表

表 7-3 Jog 运动模式指令列表

指令	说明	页码
GT_PrfJog	设置指定轴为 Jog 运动模式	132
GT_SetJogPrm	设置 Jog 运动模式下的运动参数	141
GT_GetJogPrm	读取 Jog 运动模式下的运动参数	121
GT_SetVel	设置目标速度	147
GT_GetVel	读取目标速度	128
GT_Update	启动 Jog 运动	149

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

7.3.2 重点说明

在 Jog 运动模式下，各轴可以独立设置目标速度、加速度、减速段、平滑系数等运动参数，能够独立运动或停止。

调用 GT_Update 指令启动 Jog 运动以后，按照设定的加速度加速到目标速度后保持匀速运动，在运动过程中可以随时修改目标速度，如图 7-3 所示：

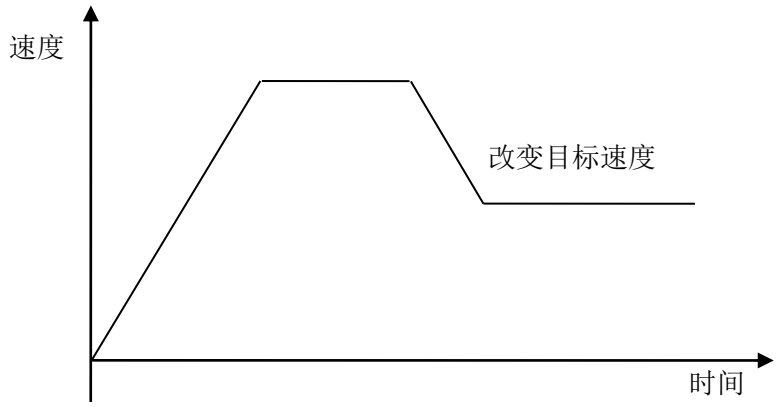

图 7-3 Jog 模式速度曲线

设定平滑系数能够得到平滑的速度曲线，从而使加减速过程更加平稳。平滑系数的取值范围是[0, 1)，越接近 1，加速度变化越平稳。

7.3.3 例程

例程 7-2 Jog 运动

轴 1 运动在 Jog 模式下，初始目标速度为 100pulse/ms。动态改变目标速度，当规划位置超过 100000pulse 时，修改目标速度为 50 pulse/ms。如图 7-4 所示：

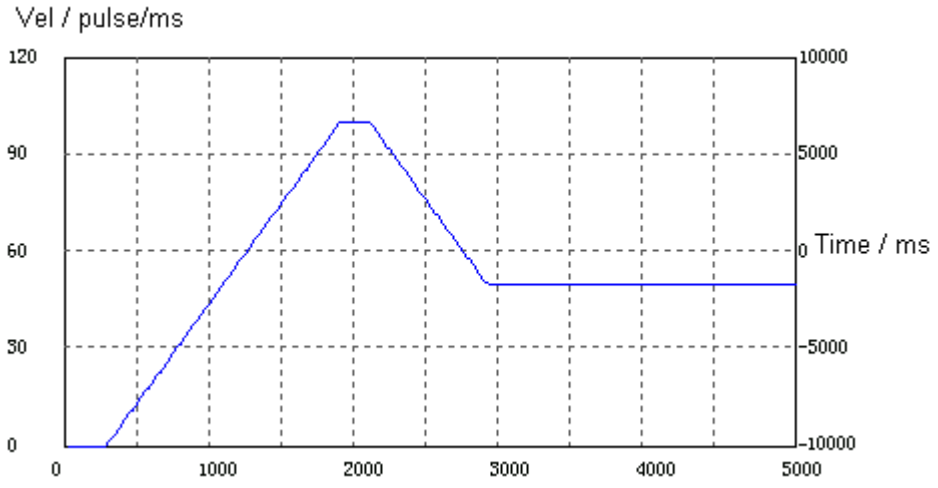

图 7-4 Jog 模式动态改变目标速度

```
PROGRAM PLC_PRG
VAR
    xInitDone:BOOL:= FALSE;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        rtn:INT;                    (*指令返回值变量*)
        Axis:INT:=1;                (*定义轴号: 1*)
        xStart: BOOL:= FALSE;
        xUpdate: BOOL:= FALSE;
        jog:TJogPrm;                (*Jog运动参数*)
        lrPrfPos:LREAL;             (*规划位置*)
        lrPrfVel:LREAL;             (*规划速度*)
        dwAxisStatus:DWORD;         (*轴状态*)
    END_VAR
    (* @END_DECLARATION := '0' *)
    IF NOT xInitDone THEN
        (*配置运动控制器*)
        (*注意：配置文件取消了各轴的报警和限位*)
        rtn:= GT_LoadConfig('test.cfg);
        commandhandler('GT_LoadConfig', rtn);
        (*清除各轴的报警和限位*)
        rtn:= GT_ClrSts(1,8);
        commandhandler('GT_ClrSts', rtn);
        (*伺服使能*)
        rtn:= GT_AxisOn(Axis);
        commandhandler('GT_AxisOn', rtn);
        xInitDone:= TRUE;
    END_IF
    IF xStart THEN
        (*位置清零*)
        rtn:= GT_ZeroPos(Axis,1);
        commandhandler('GT_ZeroPos', rtn);
        (*将 AXIS 轴设为 Jog 模式*)
        rtn:= GT_PrfJog(Axis);
        commandhandler('GT_PrfJog', rtn);
        (*读取 Jog 运动参数(需要读取全部运动参数到上位机变量)*)
        rtn:= GT_GetJogPrm(Axis, ADR(jog));
        commandhandler('GT_GetJogPrm', rtn);
        (*设置需要修改的运动参数*)
        jog.acc:= 0.0625;
        jog.dec:= 0.0625;
        (*设置 Jog 运动参数*)
        rtn:= GT_SetJogPrm(Axis, ADR(jog));
        commandhandler('GT_SetJogPrm', rtn);
        (*设置 AXIS 轴的目标速度*)
        rtn:= GT_SetVel(Axis, 100);
        commandhandler('GT_SetVel', rtn);
        (*启动AXIS轴的运动*)
        rtn:= GT_Update(SHL(WORD#1,Axis-1));
        commandhandler('GT_Update', rtn);
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        xStart:= FALSE;
    END_IF
    IF xUpdate THEN
        (*设置 AXIS 轴的目标速度*)
        rtn:= GT_SetVel(Axis, 50);
        commandhandler('GT_SetVel', rtn);
        (*启动AXIS轴的运动*)
        rtn:= GT_Update(SHL(WORD#1,Axis-1));
        commandhandler('GT_Update', rtn);
        xUpdate:= FALSE;
    END_IF

    (*读取AXIS轴的状态*)
    rtn:= GT_GetSts(Axis, ADR(dwAxisStatus),1,0);
    (*读取AXIS轴的规划位置*)
    rtn:= GT_GetPrfPos(Axis, ADR(lrPrfPos),1,0);
    (*读取AXIS轴的规划速度*)
    rtn:= GT_GetPrfVel(Axis, ADR(lrPrfVel),1,0);
    END_PROGRAM
```

7.4 PT 运动模式

7.4.1 指令列表

表 7-4 PT 运动模式指令列表

指令	说明	页码
GT_PrfPt	设置指定轴为 PT 运动模式	132
GT_PtSpace	查询 PT 运动模式指定 FIFO 的剩余空间	134
GT_PtData	向 PT 运动模式指定 FIFO 增加数据	133
GT_PtClear	清除 PT 运动模式指定 FIFO 中的数据 运动状态下该指令无效 动态模式下该指令无效	133
GT_SetPtLoop	设置 PT 运动模式循环执行的次数 动态模式下该指令无效	143
GT_GetPtLoop	查询 PT 运动模式循环执行的次数 动态模式下该指令无效	125
GT_PtStart	启动 PT 运动	134
GT_SetPtMemory	设置 PT 运动模式的缓存区大小	143
GT_GetPtMemory	读取 PT 运动模式的缓存区大小	125
GT_GetPtRemainder	读取 PT 运动模式的缓存区中待执行数据段数	125

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

7.4.2 重点说明

PT 模式非常灵活，能够实现任意速度规划。用户通过直接输入位置和时间参数描述运动规律。

PT 模式使用一系列“位置、时间”数据点描述速度规划，用户需要将速度曲线分割成若干段，如图 7-5 所示。

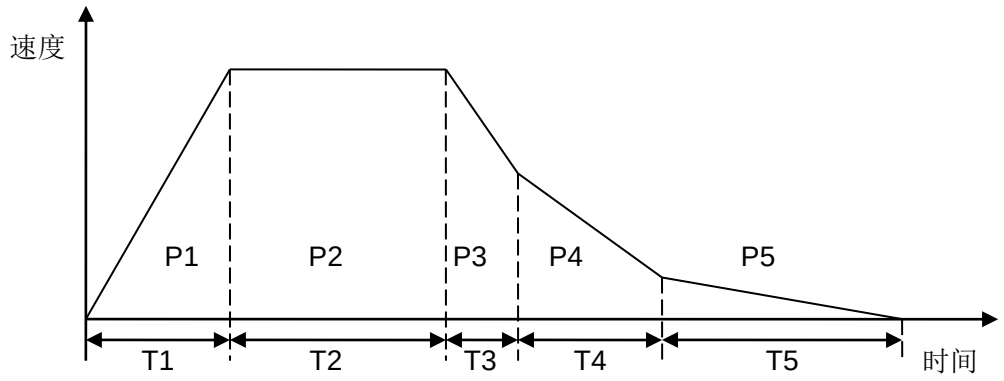

图 7-5 PT 运动速度曲线

整个速度曲线被分割成 5 段，第 1 段起点速度为 0，经过时间 T1 运动位移 P1，因此第 1 段的终点速度为 $v1=\frac{2P1}{T1}$；第 2 段起点速度为 v1，经过时间 T2 运动位移 P2，因此第 2 段的终点速度为 $v2=\frac{2P2}{T2}-v1$；第 3、4、5 段依此类推。PT 模式的数据段要求用户输入每段所需时间和位置点。

注意	在描述一次完整的 PT 运动时，第 1 段的起点位置和时间被假定为 0。压入控制器的数据为位置点，即相对于第 1 段的起点的绝对值，而不是每段位移长度。位置的单位是脉冲（pulse），时间单位是毫秒（ms）。

PT 模式在实现任意速度规划方面非常具有优势。用户将任意的速度规划分割为足够密的小段，用户只需要给出每段所需时间和位置点，运动控制器会计算段内各点的速度，生成一条连续的速度曲线。为了得到光滑的速度曲线，可以增加速度曲线的分割段数。

(1) 如何切换到 PT 模式？

用户必须要调用 GT_PrfPt，才能将指定轴设定为 PT 模式。

(2) 认识 PT 模式的数据段类型。如何向 PT 模式的 FIFO 中写入数据段？

- PT 模式的数据段有 3 种类型。
- PT_SEGMENT_NORMAL 表示普通段，FIFO 中第 1 段的起点速度为 0，从第 2 段起每段的起点速度等于上一段的终点速度。
- PT_SEGMENT_EVEN 表示匀速段，FIFO 中各段的段内速度保持不变，段内速度=段内位移/段内时间。如图 7-6 所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

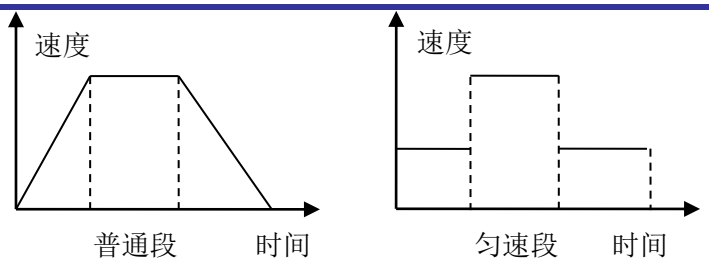

图 7-6 PT 模式匀速段类型

- PT_SEGMENT_STOP 表示停止段，该段的终点速度为 0，起点速度根据段内位移和段内时间计算得到，和上一段的终点速度无关。如图 7-7 所示。

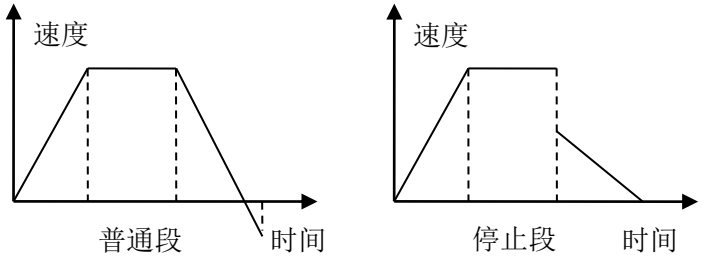

图 7-7 PT 模式停止段类型

- 调用 GT_PtData(short profile, double pos, long time, short type, short fifo)，将数据段写入指定 FIFO 中。profile 指轴号，pos 和 time 分别为一段曲线的位置点和时间，type 指数据段类型，fifo 是指定的 FIFO 编号。

(3) 如何设置 PT 循环次数？

如果轴的运动是周期性的，用户可以只写入一个周期的运动规划数据段到 FIFO 中，然后设定循环次数，即可实现周期性的 PT 运动。调用指令 GT_SetPtLoop 即可。

(4) PT 模型下，如何使用 FIFO？

- PT 模式下，有 2 个 FIFO 用来存放数据，分别为 FIFO1 和 FIFO2。PT 具有 2 种 FIFO 使用模式：静态模式和动态模式。
- 静态模式下，可以选择启动其中一个 FIFO，运动完成以后规划停止。控制器不会清除 FIFO 中的数据，用户可以重复使用 FIFO 中的数据。静止状态下调用 GT_PtClear 指令可以清空指定 FIFO。在运动状态下不能清空正在使用的 FIFO，但可以清除没有在使用的 FIFO。
- 动态模式下，不可以选择启动哪一个 FIFO，控制器会启用两个 FIFO（动态模式下，PT 指令中选择 FIFO 的参数都是无效的）。当一个 FIFO 中的数据用完以后会自动清空，同时切换到另一个 FIFO，此时可以向控制器发送新的 PT 数据。当 2 个 FIFO 中的数据都用完以后规划停止。为了避免异常停止，必须在 2 个 FIFO 中的数据都用完之前及时发送新的数据。调用 GT_PtSpace 指令可以查询剩余多少数据空间。
- 调用 GT_SetPtMemory 可以设置 FIFO 的大小，有 32 段和 1024 段两种选择。默认为 32 段。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

注意	在切换到 PT 模式时（调用指令 GT_PrfPt），应设置 FIFO 为“静态模式”或“动态模式”。运动时不能修改 FIFO 的使用模式。

7.4.3 例程

1. PT 静态 FIFO

例程 7-3 PT 静态 FIFO

该例程生成一段梯形曲线速度规划，一共三段数据段，如表 7-5 所示。

表 7-5 PT 静态 FIFO 例程数据段

	第一段	第二段	第三段
位置点(pos)	1024	2048	1024
时间(time)	1024	1024	1024
数据段类型	普通段	普通段	普通段

PT 模式梯形曲线速度规划如图 7-8 所示。

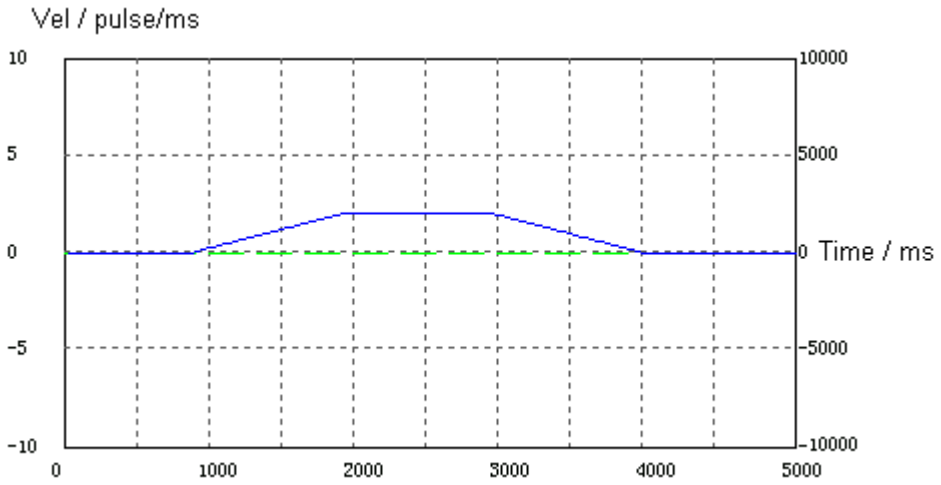

图 7-8 PT 模式梯形曲线速度规划

```
PROGRAM PLC_PRG
VAR
    xInitDone:BOOL:= FALSE;
    rtn:INT;                    (*指令返回值变量*)
    Axis:INT:=1;                (*定义轴号: 1*)
    xStart: BOOL:= FALSE;
    space:DINT;
    pos:LREAL;
    ttime:DINT;
    lrPrfPos:LREAL;             (*规划位置*)
    lrPrfVel:LREAL;             (*规划速度*)
    dwAxisStatus:DWORD;         (*轴状态*)
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
END_VAR
(* @END_DECLARATION := '0' *)
IF NOT xInitDone THEN
    (*配置运动控制器*)
    (*注意：配置文件取消了各轴的报警和限位*)
    rtn:= GT_LoadConfig('test.cfg);
    commandhandler('GT_LoadConfig', rtn);
    (*清除各轴的报警和限位*)
    rtn:= GT_ClrSts(1,8);
    commandhandler('GT_ClrSts', rtn);
    (*伺服使能*)
    rtn:= GT_AxisOn(Axis);
    commandhandler('GT_AxisOn', rtn);
    xInitDone:= TRUE;
END_IF
IF xStart THEN
    (*位置清零*)
    rtn:= GT_ZeroPos(Axis,1);
    commandhandler('GT_ZeroPos', rtn);
    (*将 AXIS 轴设为 PT 模式*)
    rtn:= GT_PrfPt(Axis, PT_MODE_STATIC);
    commandhandler('GT_PrfPt', rtn);
    (*清除AXIS轴的FIFO*)
    rtn:= GT_PtClear(Axis,0);
    commandhandler('GT_PtClear', rtn);
    (*查询PT模式FIFO的剩余空间*)
    rtn:= GT_PtSpace(Axis, ADR(space),0);
    commandhandler('GT_PtSpace', rtn);
    IF space > 0 THEN
        (*向FIFO中增加运动数据*)
        pos:= 1024;
        ttime:= 1024;
        rtn:= GT_PtData(Axis, pos, ttime, PT_SEGMENT_NORMAL,0);
        commandhandler('GT_PtData', rtn);
    END_IF
    (*查询PT模式FIFO的剩余空间*)
    rtn:= GT_PtSpace(Axis, ADR(space),0);
    commandhandler('GT_PtSpace', rtn);
    IF space > 0 THEN
        (*向FIFO中增加运动数据*)
        pos:= pos + 2048;
        ttime:= ttime + 1024;
        rtn:= GT_PtData(Axis, pos, ttime, PT_SEGMENT_NORMAL,0);
        commandhandler('GT_PtData', rtn);
    END_IF
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        (*查询PT模式FIFO的剩余空间*)
        rtn:= GT_PtSpace(Axis, ADR(space),0);
        commandhandler('GT_PtSpace', rtn);
        IF space > 0 THEN
            (*向FIFO中增加运动数据*)
            pos:= pos + 1024;
            ttime:= ttime + 1024;
            rtn:= GT_PtData(Axis, pos, ttime, PT_SEGMENT_NORMAL,0);
            commandhandler('GT_PtData', rtn);
        END_IF
        (*启动PT运动*)
        rtn:= GT_PtStart(SHL(WORD#1,Axis-1),0);
        commandhandler('GT_PtStart', rtn);
        xStart:= FALSE;
    END_IF

    (*读取AXIS轴的状态*)
    rtn:= GT_GetSts(Axis, ADR(dwAxisStatus),1,0);
    (*读取AXIS轴的规划位置*)
    rtn:= GT_GetPrfPos(Axis, ADR(lrPrfPos),1,0);
    (*读取AXIS轴的规划速度*)
    rtn:= GT_GetPrfVel(Axis, ADR(lrPrfVel),1,0);
    END_PROGRAM
```

2. PT 动态 FIFO

例程 7-4 PT 动态 FIFO

该例程生成一段正弦曲线速度规划，周期为 1s，循环次数为 2。将该正弦曲线分割成若干小线段，每段时间间隔为 0.016s。计算每个小线段的时间和位置点，传入 FIFO 中。由于数据段较多，使用 PT 的动态 FIFO 模式。在存入数据段的同时不断检查 FIFO 是否已满；当两个 FIFO 都满了时，启动运动；当其中一个 FIFO 中的数据遍历完了，控制器将清空这个 FIFO，此时查询到有空间，就继续存入后面的数据段；依次下来，直到整条描述正弦曲线的小线段全部遍历完。PT 模式正弦曲线速度规划如图 7-9 所示：

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

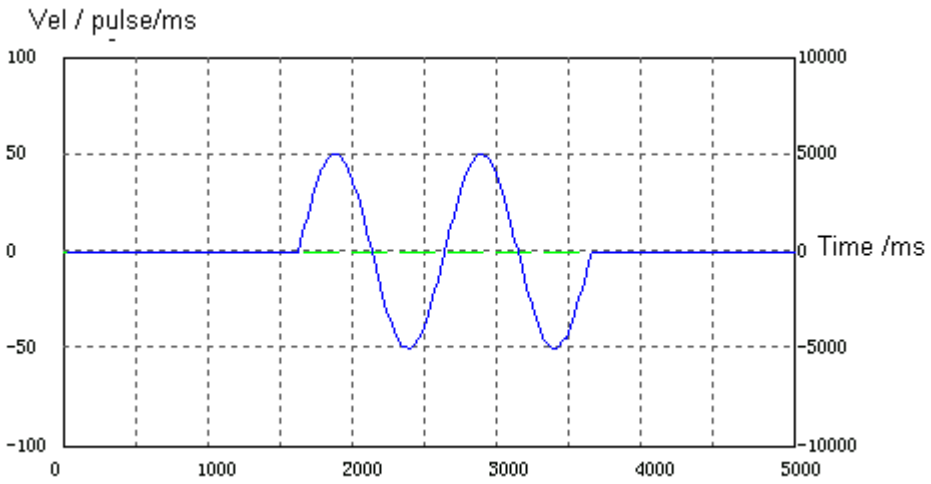

图 7-9 PT 模式正弦曲线速度规划

```
PROGRAM PLC_PRG
VAR
    xInitDone:BOOL:= FALSE;
    rtn:INT;                    (*指令返回值变量*)
    Axis:INT:=1;                (*定义轴号: 1*)
    xStart: BOOL:= FALSE;
    space:DINT;
    pos,vel,velPre:LREAL;
    ttime:DINT;
    loop:INT;
    lrPrfPos:LREAL;             (*规划位置*)
    lrPrfVel:LREAL;             (*规划速度*)
    dwAxisStatus:DWORD;         (*轴状态*)
END_VAR
VAR CONSTANT
    A:INT:= 50;                 (*幅值*)
    T:INT:= 1;                  (*周期*)
    DELTA:LREAL:= 0.016;        (*时间分段*)
    LOOP_TIMES:INT:= 2;         (*循环次数*)
    PI:LREAL:= 3.1415926;
END_VAR
(* @END_DECLARATION := '0' *)
IF NOT xInitDone THEN
    (*配置运动控制器*)
    (*注意：配置文件取消了各轴的报警和限位*)
    rtn:= GT_LoadConfig('test.cfg);
    commandhandler('GT_LoadConfig', rtn);
    (*清除各轴的报警和限位*)
    rtn:= GT_ClrSts(1,8);
    commandhandler('GT_ClrSts', rtn);
    (*伺服使能*)
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
            rtn:= GT_AxisOn(Axis);
            commandhandler('GT_AxisOn', rtn);
            xInitDone:= TRUE;
        END_IF
        IF xStart THEN
            (*位置清零*)
            rtn:= GT_ZeroPos(Axis,1);
            commandhandler('GT_ZeroPos', rtn);
            (*将 AXIS 轴设为 PT 模式*)
            rtn:= GT_PrfPt(Axis, PT_MODE_DYNAMIC);
            commandhandler('GT_PrfPt', rtn);
            (*清除AXIS轴的FIFO*)
            rtn:= GT_PtClear(Axis,0);
            commandhandler('GT_PtClear', rtn);
            pos = 0;
            vel = velPre = 0;
            ttime = 0;
            start = 0;
            loop = 1;
            xStarting:= TRUE;
            xStart:= FALSE;
            xStarted:= FALSE;
        END_IF
        IF xStarting THEN
            (*查询PT模式FIFO的剩余空间*)
            rtn:= GT_PtSpace(Axis, ADR(space),0);
            IF space>0 THEN
                ttime:= ttime+DELTA;
                (*计算段末速度*)
                vel:= A*SIN((2*PI)/T*ttime);
                (*计算段内位移*)
                pos:= pos+1000*(vel+velPre)*DELTA/2;
                velPre:= vel;
                IF ttime<loop*T THEN
                    (*发送新数据*)
                    rtn:= GT_PtData(Axis, pos, ttime*1000, PT_SEGMENT_NORMAL,0);
                ELSE
                    (*发送终点数据*)
                    rtn:= GT_PtData(Axis, 0, loop*T*1000, PT_SEGMENT_STOP,0);
                    pos:= 0;
                    ttime:= loop*T;
                    velPre:= 0;
                    loop:= loop+1;
                    IF loop>LOOP_TIMES THEN
                        IF NOT xStarted THEN
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
                    (*启动PT运动*)
                    rtn:= GT_PtStart(SHL(WORD#1,Axis-1),0);
                    commandhandler('GT_PtStart', rtn);
                    xStarted:= TRUE;
                END_IF
                xStarting:= FALSE;
            END_IF
        END_IF
    ELSIF NOT xStarted THEN
        (*启动PT运动*)
        rtn:= GT_PtStart(SHL(WORD#1,Axis-1),0);
        commandhandler('GT_PtStart', rtn);
        xStarted:= TRUE;
    END_IF
END_IF

(*读取AXIS轴的状态*)
rtn:= GT_GetSts(Axis, ADR(dwAxisStatus),1,0);
(*读取AXIS轴的规划位置*)
rtn:= GT_GetPrfPos(Axis, ADR(lrPrfPos),1,0);
(*读取AXIS轴的规划速度*)
rtn:= GT_GetPrfVel(Axis, ADR(lrPrfVel),1,0);
END_PROGRAM
```

7.5 电子齿轮（Gear）运动模式

7.5.1 指令列表

表 7-6 电子齿轮运动模式指令列表

指令	说明	页码
GT_PrfGear	设置指定轴为电子齿轮运动模式	131
GT_SetGearMaster	设置电子齿轮运动跟随主轴	140
GT_GetGearMaster	读取电子齿轮运动跟随主轴	120
GT_SetGearRatio	设置电子齿轮比	140
GT_GetGearRatio	读取电子齿轮比	121
GT_GearStart	启动电子齿轮运动	110

7.5.2 重点说明

电子齿轮模式能够将 2 轴或多轴联系起来，实现精确的同步运动，从而替代传统的机械齿轮连接。

我们把被跟随的轴叫主轴，把跟随的轴叫从轴。电子齿轮模式下，1 个主轴能够驱动多个从轴，从轴可以跟随主轴的规划位置、编码器位置。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

传动比：主轴速度与从轴速度的比例。电子齿轮模式能够灵活的设置传动比，节省机械系统的安装时间。当主轴速度变化时，从轴会根据设定好的传动比自动改变速度。电子齿轮模式也能够在运动过程中修改传动比。

离合区：当改变传动比时，可以设置离合区，实现平滑变速，如图所示，阴影区域为离合区。

电子齿轮模式速度曲线如图 7-10 所示。

注意	离合区位移是指从轴平滑变速过程中主轴运动的位移。不要计算成从轴变速时走过的位移

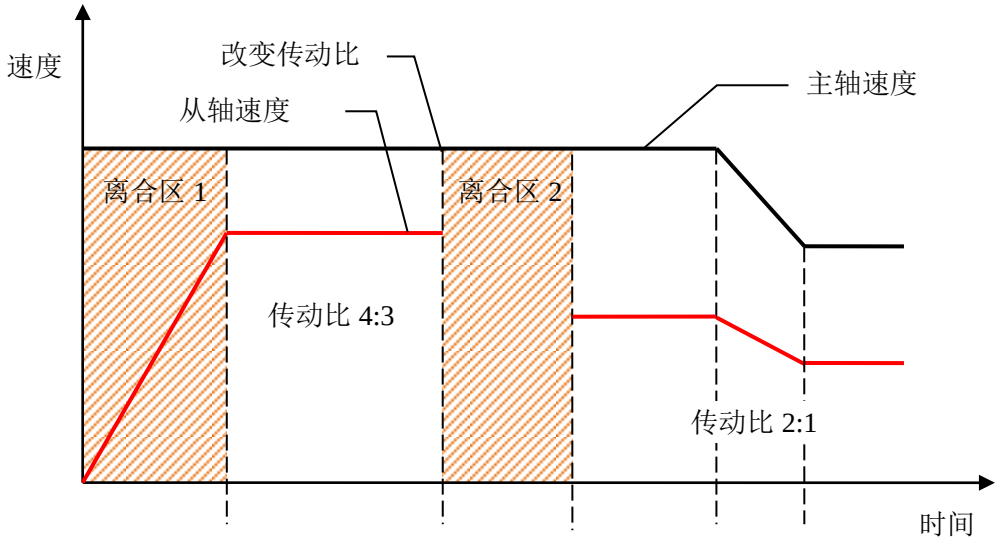

图 7-10 电子齿轮模式速度曲线

主轴匀速运动，从轴为电子齿轮模式。在离合区 1 从轴速度从 0 逐渐增大，直到到达传动比 4:3。当改变传动比至 2:1 时，在离合区 2 从轴速度逐渐变化直到满足新的传动比。离合区越大，从轴传动比的变化过程越平稳。当主轴速度变化时，从轴速度也随着变化，保持固定的传动比。

(1) 如何切换到电子齿轮模式？

用户必须要调用 GT_PrfGear(short profile, short dir)，才能将指定轴设定为 Gear 模式。应将从轴设定为 Gear 模式。

(2) 如何设置主轴？

调用 GT_SetGearMaster(short profile, short masterIndex, short masterType, short masterItem)。profile 为从轴轴号，mastIndex 为主轴轴号。

注意	为了减少跟随滞后，从轴的轴号应当大于主轴的轴号。

(3) 如何设定传动比和离合区？

1) 调用 GT_SetGearRatio(short profile, long masterEven, long slaveEven, long masterSlope) 指令来设置传动比和离合区。profile 是从轴轴号；masterEven 是主轴位移，slaveEven 是从轴位移，masterEven/slaveEven 等于传动比；slope 是主轴离合区位移，即主轴在离合区内走过的位移，用户应自行计算出来。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

2) 如果从轴轴号为 slave，当主轴位移 alpha 时从轴位移 beta，主轴运动 slope 位移后从轴到达设定传动比，应当调用以下指令：

GT_SetGearRatio(slave, alpha, beta, slope);

7.5.3 例程

例程 7-5 电子齿轮跟随

该例程主轴为 Jog 模式，从轴为电子齿轮模式，传动比为主轴速度：从轴速度=2：1，主轴运动离合区位移后（图中阴影部分的区域），从轴达到设定的传动比，如图 7-11 和图 7-12 所示。

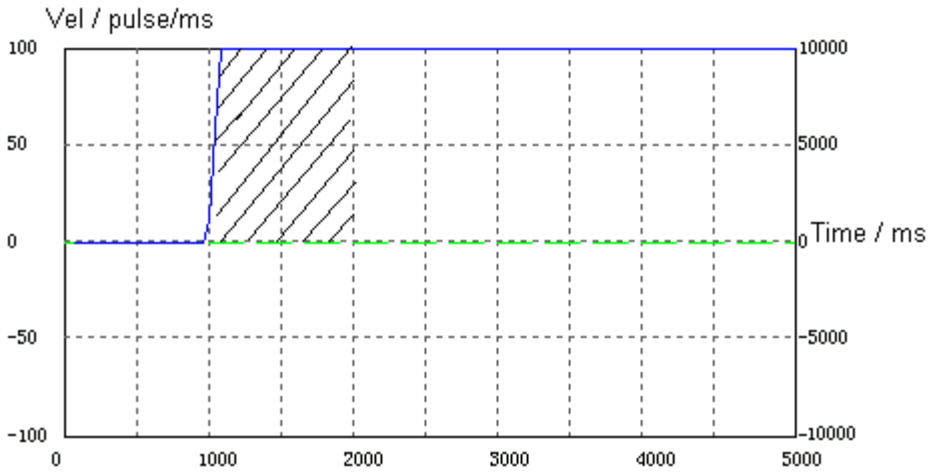

图 7-11 电子齿轮模式主轴速度规划

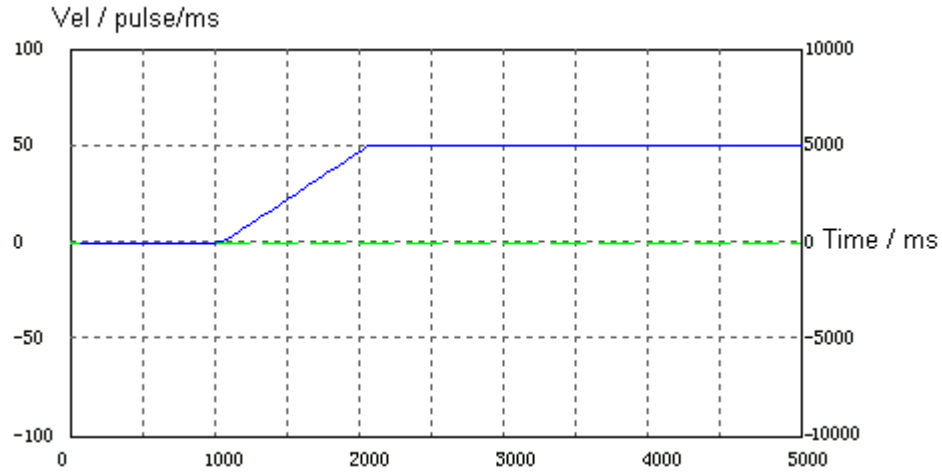

图 7-12 电子齿轮模式从轴速度规划

```
PROGRAM PLC_PRG
VAR
    xInitDone:BOOL:= FALSE;
    rtn:INT;                    (*指令返回值变量*)
    Master:INT:=1;              (*定义主轴轴号: 1*)
    Slave:INT:=2;               (*定义从轴轴号: 2*)
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        xStart: BOOL:= FALSE;
        jog:TJogPrm;                    (*Jog运动参数*)
        arrPrfVel:ARRAY [0..7] OF LREAL; (*规划速度*)
    END_VAR
    (* @END_DECLARATION := '0' *)
    IF NOT xInitDone THEN
        (*配置运动控制器*)
        (*注意：配置文件取消了各轴的报警和限位*)
        rtn:= GT_LoadConfig('test.cfg);
        commandhandler('GT_LoadConfig', rtn);
        (*清除各轴的报警和限位*)
        rtn:= GT_ClrSts(1,8);
        commandhandler('GT_ClrSts', rtn);
        (*伺服使能*)
        rtn:= GT_AxisOn(Master);
        commandhandler('GT_AxisOn', rtn);
        rtn:= GT_AxisOn(Slave);
        commandhandler('GT_AxisOn', rtn);
        xInitDone:= TRUE;
    END_IF
    IF xStart THEN
        (*位置清零*)
        rtn:= GT_ZeroPos(Master,1);
        commandhandler('GT_ZeroPos', rtn);
        rtn:= GT_ZeroPos(Slave,1);
        commandhandler('GT_ZeroPos', rtn);
        (*将主轴设为 Jog 模式*)
        rtn:= GT_PrfJog(Master);
        commandhandler('GT_PrfJog', rtn);
        (*设置主轴运动参数*)
        rtn:= GT_GetJogPrm(Master, ADR(jog));
        commandhandler('GT_GetJogPrm', rtn);
        jog.acc := 1;
        rtn:= GT_SetJogPrm(Master, ADR(jog));
        commandhandler('GT_SetJogPrm', rtn);
        rtn:= GT_SetVel(Master, 100);
        commandhandler('GT_SetVel', rtn);
        (*启动主轴*)
        rtn:= GT_Update(SHL(WORD#1,Master-1));
        commandhandler('GT_Update', rtn);
        (*将从轴设为 Gear 模式*)
        rtn:= GT_PrfGear(Slave,0);
        commandhandler('GT_PrfGear', rtn);
        (*设置主轴，跟随主轴规划位置*)
        rtn:= GT_SetGearMaster(Slave, Master, GEAR_MASTER_PROFILE,0);
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        commandhandler('GT_SetGearMaster', rtn);
        (*设置从轴的传动比和离合区*)
        rtn:= GT_SetGearRatio(Slave, 2, 1, 100000);
        commandhandler('GT_SetGearRatio', rtn);
        (*启动从轴*)
        rtn:= GT_GearStart(SHL(WORD#1,Slave-1));
        commandhandler('GT_GearStart',rtn);
        xStart:= FALSE;
    END_IF

    (*读取轴的规划速度*)
    rtn:= GT_GetPrfVel(1, ADR(arrPrfVel),8,0);
    END_PROGRAM
```

7.6 Follow 运动模式

7.6.1 指令列表

表 7-7 Follow 运动模式指令列表

指令	说明	页码
GT_PrfFollow	设置指定轴为 Follow 运动模式	131
GT_SetFollowMaster	设置 Follow 运动模式跟随主轴	139
GT_GetFollowMaster	读取 Follow 运动模式跟随主轴	119
GT_SetFollowLoop	设置 Follow 运动模式循环次数	138
GT_GetFollowLoop	读取 Follow 运动模式循环次数	119
GT_SetFollowEvent	设置 Follow 运动模式启动跟随条件	138
GT_GetFollowEvent	读取 Follow 运动模式启动跟随条件	118
GT_FollowSpace	查询 Follow 运动模式指定 FIFO 的剩余空间	109
GT_FollowData	向 Follow 运动模式指定 FIFO 增加数据	109
GT_FollowClear	清除 Follow 运动模式指定 FIFO 中的数据 运动状态下该指令无效	108
GT_FollowStart	启动 Follow 运动	109
GT_FollowSwitch	切换 Follow 运动模式所使用的 FIFO	110
GT_SetFollowMemory	设置 Follow 运动模式的缓存区大小	139
GT_GetFollowMemory	读取 Follow 运动模式的缓存区大小	120

7.6.2 重点说明

在很多应用中，两轴或多轴之间需要保证位置同步和速度同步。我们把被跟随的轴叫主轴，把跟随的轴叫从轴。一对典型的主轴和从轴的规划如图 7-13 所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

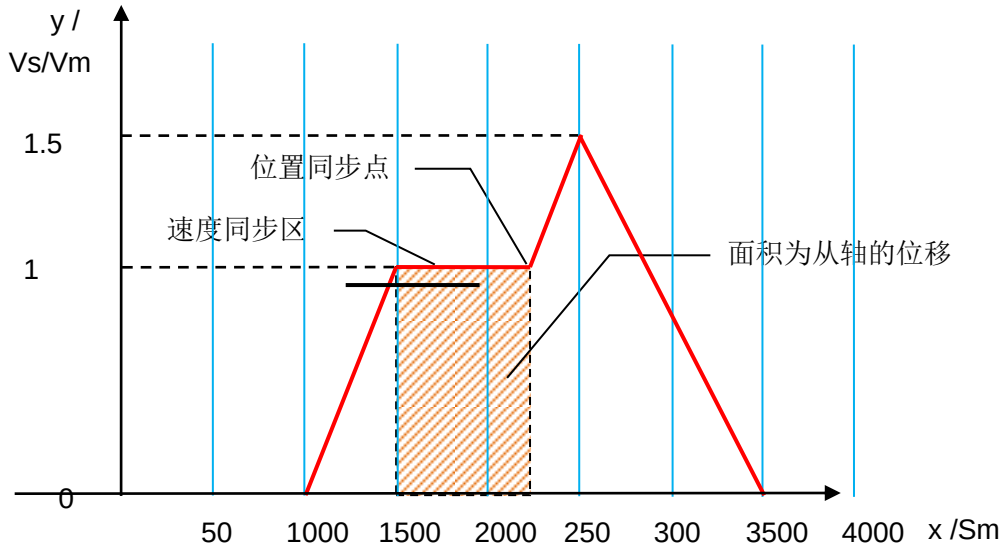

图 7-13 Follow 模式主从轴规划

图中画了一个周期的 Follow 运动，其中横坐标 x 表示主轴的位移，纵轴 y 表示从轴速度 Vs 与主轴速度 Vm 的比率。根据 Sm*(Vs/Vm)可得:Vs*(Sm/Vm)=Vs*t，即图 7-13 中的面积为从轴的位移。整个跟随模式如下：

(1) 在主轴运动到设定位置（1000pulse）时，从轴启动跟随。

(2) 在主轴运动到 1500pulse 时，从轴运动 250（(1500-1000)*1/2)pulse 到达速度同步区，即图中阴影部分所示的速度同步区。

(3) 在主轴运动到 2250pulse 时，从轴与主轴的分别同时到达各自的位置点，即图中标注的位置同步点。

位置同步点表示主轴和从轴必须同时到达各自指定位置。

速度同步区表示主轴和从轴之间必须保持准确的速度比。

Follow 模式就是针对这种应用，给用户提供了主轴和从轴的位置和速度规划方式。用户只需要学习如何设置主轴，从轴，从轴启动跟随的条件，如何设置 Follow 循环次数，如何利用 Follow 模式中的数据段类型实现应用中所需要的规划以及如何管理 FIFO，就可以轻松实现 Follow 运动。

(1) 如何切换到 Follow 模式？

用户必须要调用 GT_PrfFollow(short profile, short dir)，才能将指定轴设定为 Follow 模式。一般应将从轴设定为 Follow 模式。

(2) 如何设置主轴，从轴？

调用 GT_SetFollowMaster(short profile, short masterIndex, short masterType, short masterItem)。profile 为从轴轴号，masterIndex 为主轴轴号。

注意	为了减少跟随滞后，从轴的轴号应当大于主轴的轴号。

(3) 如何设定从轴启动跟随条件？

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

所谓从轴启动跟随条件，是描述什么情况下从轴开始启动运动。有两种情况：第一，调用指令 GT_FollowStart 以后从轴立即启动；第二，调用指令 GT_FollowStart 以后，从轴还要等待主轴穿越了设定位置以后才启动跟随运动。用户需调用指令 GT_SetFollowEvent 来选择使用哪种跟随条件。

(4) 如何设置 Follow 循环次数？

如果从轴的跟随运动是周期性的，用户可以只写入一个周期的运动规划到 FIFO 中，然后设定循环次数，即可实现周期性的 Follow 运动。调用指令 GT_SetFollowLoop 即可。

(5) 认识 Follow 模式的数据段类型。如何向 Follow 模式的 FIFO 中写入数据段？

调用指令 GT_FollowData(short profile, long masterSegment, double slaveSegment, short type= FOLLOW_SEGMENT_NORMAL, short fifo)，将数据段写入指定 FIFO 中。profile 指从轴轴号，masterSegment 和 slaveSegment 分别指主轴和从轴应同时走过的位移，type 指从轴的数据段类型，fifo 是指定的 FIFO 编号。

用户压入的数据段都是对位置点的描述，控制器会根据用户选择的类型来自行计算速度。

- Follow 模式的数据段有 4 种类型。注意，都是针对从轴的规划。
- FOLLOW_SEGMENT_NORMAL 表示普通段，FIFO 中第 1 段的起点速度比率为 0，从第 2 段起每段的起点速度比率等于上一段的终点速度比率。
- FOLLOW_SEGMENT_EVEN 表示恒速比率段，FIFO 中各段的段内速度比率保持不变。
- FOLLOW_SEGMENT_STOP 表示停止段，该段的终点速度比率为 0，起点速度比率根据段内位移和段内时间计算得到，和上一段的终点速度比率无关。
- FOLLOW_SEGMENT_CONTINUE 表示连续段，FIFO 中第一段的起点速度比率等于上个 FIFO 的终点速度比率，从第 2 段起每段的起点速度比率等于上一段的终点速度比率。

用户如何根据自己主轴和从轴的速度比率及位置规划，来设计相应的数据段类型？可以参考飞剪案例。

(6) 如何切换 FIFO？

Follow 模式下有 2 个独立的 FIFO 用来保存数据。2 个 FIFO 之间可以在运动状态下进行切换。下面描述一个切换 FIFO 的典型案例，如图 7-14 所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

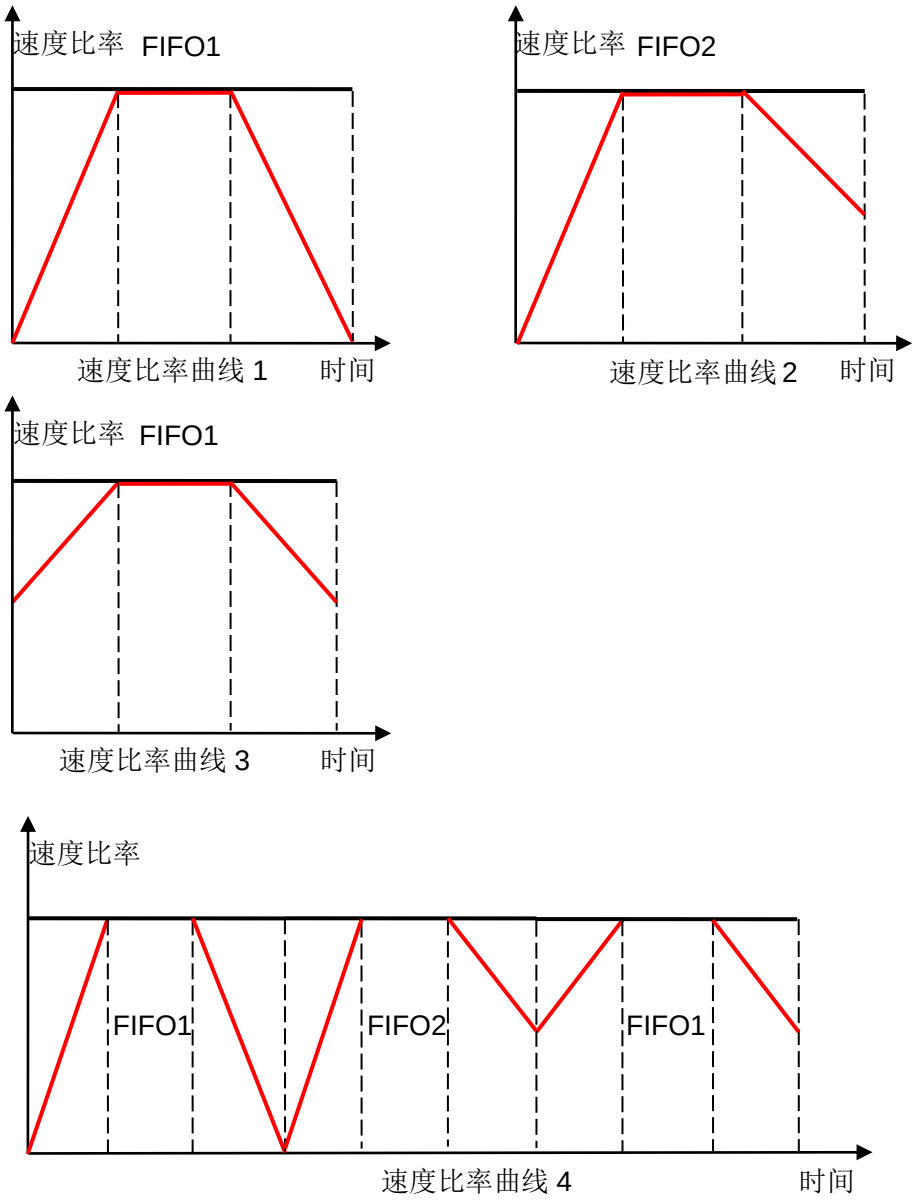

图 7-14 Follow 模式切换 FIFO

如图所示，黑色粗线为主轴规划，红色粗线为从轴规划。从轴的运动规律需要从“速度比率曲线 1”变化到“速度比率曲线 3”，为了实现从轴速度的平滑过渡，增加了一个“速度比率曲线 2”的过渡状态。“速度比率曲线 2”的起始速度比率和“速度比率曲线 1”相等，“速度比率曲线 2”的终点速度比率和“速度比率曲线 3”相等。“速度比率曲线 4”是“速度比率曲线 1”、“速度比率曲线 2”和“速度比率曲线 3”的合成。

具体的操作步骤是：

a) “速度比率曲线 1”放入 FIFO1 中，把“速度比率曲线 2”放入 FIFO2 中。假设当前正在运行的数据来自 FIFO1，调用指令 GT_FollowSwitch，控制器会在 FIFO1 中的数据全部运行完后，自动切换去运行 FIFO2 中的数据，并且将 FIFO1 全部清空。

注意	为了实现 2 个 FIFO 之间的速度连续，存入 FIFO2 的第一段数据的时候，调用 GT_FollowData 指令时应当将数据类型设置为 FOLLOW_SEGMENT_CONTINUE。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

b) 在控制器运行 FIFO2 的数据的时候，调用指令 GT_FollowSpace 查询 FIFO1 是否被清空。如果已被清空，就将“速度比率曲线 3”的数据存入 FIFO1 中。然后调用指令 GT_FollowSwitch，控制器会在 FIFO2 中的数据全部运行完后，自动切换去运行 FIFO1 中的数据，并且将 FIFO2 全部清空。

用户欲知道怎么用代码具体实现这个例子，可以查看“**例程 7-8 Follow 双 FIFO 切换**”。

7.6.3 例程

1. 飞剪案例

例程 7-6 飞剪中的 Follow 模式应用

飞剪应用背景介绍：

简化的飞剪设备结构是主传送带（主轴）匀速拉着待剪物品定向移动，同时，在传送带上方装有一带切刀的转子，当转子旋转一周（逆时针为正向）时，刚好和待剪物体接触，使之剪断，剪切长度为：10000pulse，转子旋转一周为：8000pulse。如图 7-15 所示。

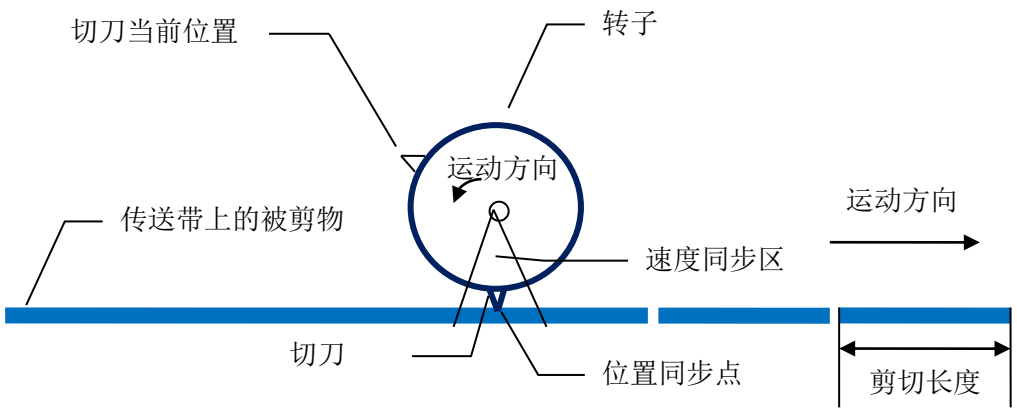

图 7-15 飞剪模型

下面我们分析如何为这样一个同步机构设计 Follow 模式下的速度规划。

首先要找出同步机构的**位置同步点**。位置同步点表示主轴和从轴必须同时到达各自指定位置（比如被剪物体每走完上图中的“剪切长度”，转轮就要刚好走完一圈，两者在各自最终位置点上必须同时到达）。该例中，待剪物被切断时主轴和从轴的位置即位置同步点。要求主轴走完 10000pulse 时，从轴必须走完 8000pulse。

我们假设以切刀当前的位置来看，转轮还要正向运动 2500 个脉冲切刀才可达到位置同步点。

其次，查看该同步机构是否需要**速度同步区**。速度同步区表示在这段区域内主轴和从轴之间必须保持准确的速度比。该设备在待剪物被切断（位置同步点）前后一段距离内，需要有一速度同步区。在此速度同步区内，要求主轴和从轴速度相等。

因此，我们可以画出飞剪的主从轴速度曲线图，如图 7-16 所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

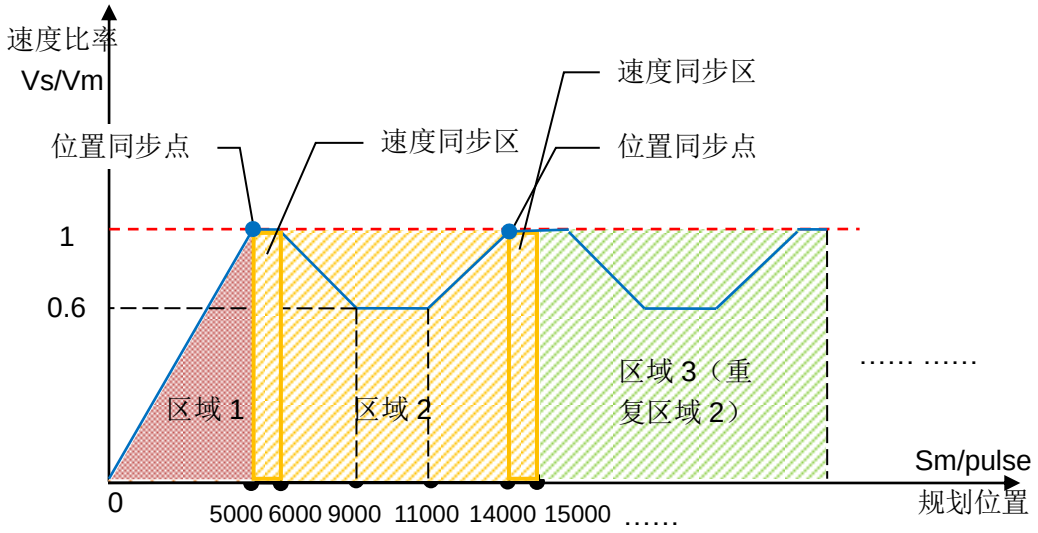

图 7-16 飞剪案例之 Follow 模式规划曲线

从上图来看，假设主轴（即传送带）是以 Jog 模式在运动，而从轴（即转轮）是 Follow 模式运动。区域 1（红色阴影部分）是从轴启动跟随，表示从轴追赶主轴到达位置同步点的位移。区域 2（黄色阴影部分）表示从轴旋转完整一周，回到起始点的位移，区域 3（绿色阴影部分）与区域 2 一样，表示从轴循环旋转以达到等长切断主轴传送带上的被剪物体。

蓝色实心点即位置同步点，橙色方框区域为速度同步区。区域 2 曲线表示要求从轴上的切刀在与主轴上的被剪物体接触后保持一定时间的同速运行，以便转轮上的切刀切断主轴传送带上的被剪物。之后以较低速度和主轴物体分离；当切刀再次运动到临接近被切物体时，又要与主轴的速度同步，以此类推，循环运行。

如何实现以上规划曲线，现以上图的具体数字说明。

我们注意到，区域 1 和区域 2 是功能完全不同的数据段。区域 1 的数据段只是过渡段，当速度和位置到达预定值后便不再执行了，区域 2 则是需要循环执行的段，因此需要将区域 1 的数据放在一个 FIFO，区域 2 的数据放在另外一个 FIFO。

区域 1：从轴追赶主轴的位移段，当主轴走完 5000pulse 时，从轴需要走 2500pulse，如表 7-8 所示。以主轴规划位置为参考，该数据段的起点为规划 0 位置。

表 7-8 飞剪案例区域 1 的数据段

	第一段(pulse)
主轴位置	5000
从轴位置	2500

区域 2：可以分成 5 个数据段：第一段为切刀从位置同步点离开的速度同步区段；第二段为切刀减速脱离速度同步区段；第三段为从轴恒速段；第四段为从轴往主轴速度变化的加速段；最后一段是切刀接近被剪物体的速度同步区段。计算可得如表 7-9 所示。以主轴规划位置为参考，该数据段的起点为规划位置 5000pulse。

表 7-9 飞剪案例区域 2 的数据段

	第一段	第二段	第三段	第四段	第五段
主轴位置	1000	4000	6000	9000	10000

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	第一段	第二段	第三段	第四段	第五段
从轴位置	1000	3400	4600	7000	8000
主轴位移长度	1000	3000	2000	3000	1000
从轴位移长度	1000	2400	1200	2400	1000

1. 压入控制器的数据为位置点（相对于数据段起点位置的位移），而不是位移长度。
2. 位置点的起点都是以启动 Follow 运动（调用指令 GT_FollowStart 之后）那一刻的位置点为零点（如设置的启动条件为 FOLLOW_EVENT_PASS，则以穿越点为零点基准）。
3. 若切换了 FIFO，位置点又是以换 FIFO 后的位置为零点。

2. Follow 单 FIFO

例程 7-7 Follow 单 FIFO 模式

该例程主轴为 Jog 模式，速度为 50pulse/ms。从轴为 Follow 模式，跟随主轴的规划位置。从轴启动的跟随条件是：主轴走过 50000pulse 后，从轴启动跟随。从轴的运动规律由 3 段组成，如表 7-10 所示，加速段跟随，匀速跟随，减速跟随，类似一个梯形曲线。并且无限次循环此数据段。主轴速度规划如图 7-17 所示，从轴速度规划如图 7-18 所示。

表 7-10 Follow 单 FIFO 数据段

	第一段	第二段	第三段
主轴位置	20000	20000	20000
从轴位置	10000	20000	10000

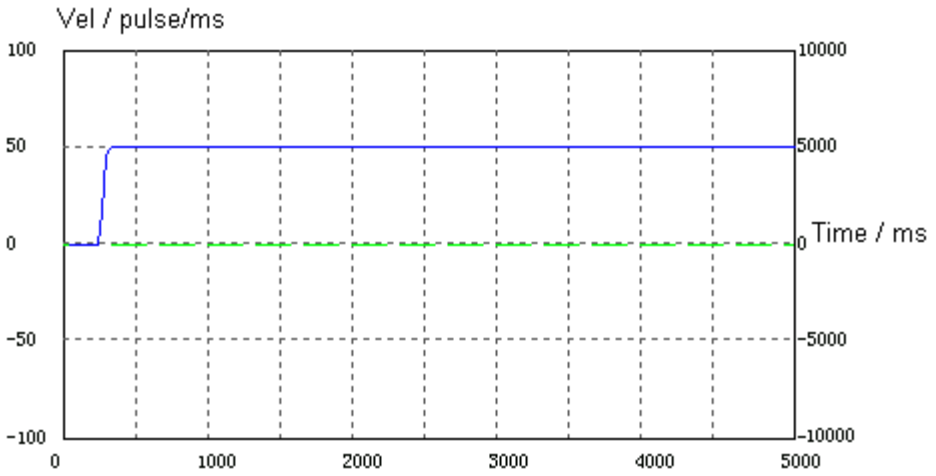

图 7-17 Follow 单 FIFO 模式主轴速度规划

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

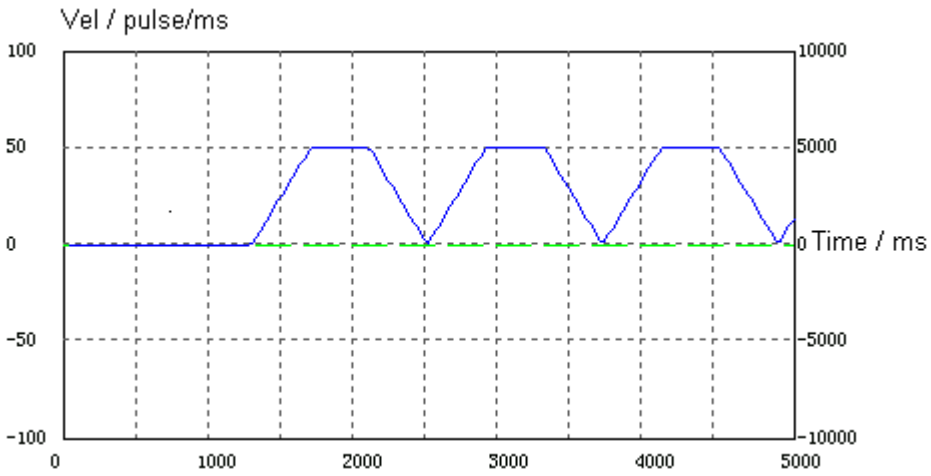

图 7-18 Follow 单 FIFO 模式从轴速度规划

```
PROGRAM PLC_PRG
VAR
    xInitDone:BOOL:= FALSE;
    rtn:INT;                    (*指令返回值变量*)
    Master:INT:=1;              (*定义主轴轴号: 1*)
    Slave:INT:=2;               (*定义从轴轴号: 2*)
    xStart: BOOL:= FALSE;
    jog:TJogPrm;                (*Jog运动参数*)
    arrPrfVel:ARRAY [0..7] OF LREAL; (*规划速度*)
    space:DINT;
    masterPos:DINT;
    slavePos:LREAL;
END_VAR
(* @END_DECLARATION := '0' *)
IF NOT xInitDone THEN
    (*配置运动控制器*)
    (*注意：配置文件取消了各轴的报警和限位*)
    rtn:= GT_LoadConfig('test.cfg);
    commandhandler('GT_LoadConfig', rtn);
    (*清除各轴的报警和限位*)
    rtn:= GT_ClrSts(1,8);
    commandhandler('GT_ClrSts', rtn);
    (*伺服使能*)
    rtn:= GT_AxisOn(Master);
    commandhandler('GT_AxisOn', rtn);
    rtn:= GT_AxisOn(Slave);
    commandhandler('GT_AxisOn', rtn);
    xInitDone:= TRUE;
END_IF
IF xStart THEN
    (*位置清零*)
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
rtn:= GT_ZeroPos(Master,1);
commandhandler('GT_ZeroPos', rtn);
rtn:= GT_ZeroPos(Slave,1);
commandhandler('GT_ZeroPos', rtn);
(*将主轴设为 Jog 模式*)
rtn:= GT_PrfJog(Master);
commandhandler('GT_PrfJog', rtn);
(*设置主轴运动参数*)
rtn:= GT_GetJogPrm(Master, ADR(jog));
commandhandler('GT_GetJogPrm', rtn);
jog.acc := 1;
rtn:= GT_SetJogPrm(Master, ADR(jog));
commandhandler('GT_SetJogPrm', rtn);
rtn:= GT_SetVel(Master, 50);
commandhandler('GT_SetVel', rtn);
(*启动主轴*)
rtn:= GT_Update(SHL(WORD#1,Master-1));
commandhandler('GT_Update', rtn);
(*将从轴设为 Follow 模式*)
rtn:= GT_PrfFollow(Slave,0);
commandhandler('GT_PrfFollow', rtn);
(*清空从轴 FIFO*)
rtn:= GT_FollowClear(Slave,0);
commandhandler('GT_FollowClear', rtn);
(*设置主轴，默认跟随主轴规划位置*)
rtn:= GT_SetFollowMaster(Slave,Master, FOLLOW_MASTER_PROFILE,1);
commandhandler('GT_SetFollowMaster', rtn);
(*查询 Follow 模式的剩余空间*)
rtn:= GT_FollowSpace(Slave,ADR(space),0);
(*向 FIFO 中增加运动数据*)
masterPos:= 20000;
slavePos:= 10000;
rtn:= GT_FollowData(Slave, masterPos, slavePos,
        FOLLOW_SEGMENT_NORMAL, 0);
commandhandler('GT_FollowData', rtn);
(*查询 Follow 模式的剩余空间*)
rtn:= GT_FollowSpace(Slave,ADR(space),0);
(*向 FIFO 中增加运动数据*)
masterPos:= masterPos+20000;
slavePos:= slavePos+20000;
rtn:= GT_FollowData(Slave, masterPos, slavePos,
        FOLLOW_SEGMENT_NORMAL,0);
commandhandler('GT_FollowData',rtn);
(*查询 Follow 模式的剩余空间*)
rtn:= GT_FollowSpace(Slave,ADR(space),0);
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        (*向 FIFO 中增加运动数据*)
        masterPos:= masterPos+20000;
        slavePos:= slavePos+10000;
        rtn:= GT_FollowData(Slave, masterPos, slavePos,
                FOLLOW_SEGMENT_NORMAL,0);
        commandhandler('GT_FollowData', rtn);
        (*设置循环次数为无限循环*)
        rtn:= GT_SetFollowLoop(Slave, 0);
        commandhandler('GT_SetFollowLoop', rtn);
        (*设置启动跟随条件*)
        rtn:= GT_SetFollowEvent(Slave, FOLLOW_EVENT_PASS, 1, 50000);
        commandhandler('GT_SetFollowEvent', rtn);
        (*启动从轴 Follow 运动*)
        rtn:= GT_FollowStart(SHL(WORD#1,Slave-1),0);
        commandhandler('GT_FollowStart', rtn);
        xStart:= FALSE;
    END_IF

    (*读取轴的规划速度*)
    rtn:= GT_GetPrfVel(1, ADR(arrPrfVel),8,0);
    END_PROGRAM
```

3. Follow 双 FIFO 切换

例程 7-8 Follow 双 FIFO 切换

该例程主轴为 Jog 模式，速度为 50pulse/ms。从轴为 Follow 模式，跟随主轴的规划位置。从轴启动的跟随条件是：从轴在调用指令 GT_FollowStart 后立即启动跟随。从轴在运动时更换跟随策略，其速度规划经过一个过渡的数据段，然后变成一个新的梯形曲线，并且无限次循环。如下面三个表所示。我们把表 7-11 中的数据写入 FIFO1 中，把表 7-12 中的数据写入 FIFO2 中，在运动过程中切换到 FIFO2，同时在检查并确认 FIFO1 被控制器自动清空之后，将表 7-13 中的数据写入 FIFO1 中，并切换运动 FIFO1 的数据。这样即可利用双 FIFO 切换完成跟随策略的更换。主轴速度规划如图 7-19 所示，从轴速度规划如图 7-20 所示。

表 7-11 Follow 双 FIFO 切换之原来的跟随策略

	第一段	第二段	第三段
主轴位置	20000	20000	20000
从轴位置	10000	20000	10000

表 7-12 Follow 双 FIFO 切换之更换跟随策略时的过渡段

	第一段	第二段	第三段
主轴位置	20000	20000	20000
从轴位置	10000	20000	16000

表 7-13 Follow 双 FIFO 切换之更换后的跟随策略

	第一段	第二段	第三段
主轴位置	20000	20000	20000

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	第一段	第二段	第三段
从轴位置	16000	20000	16000

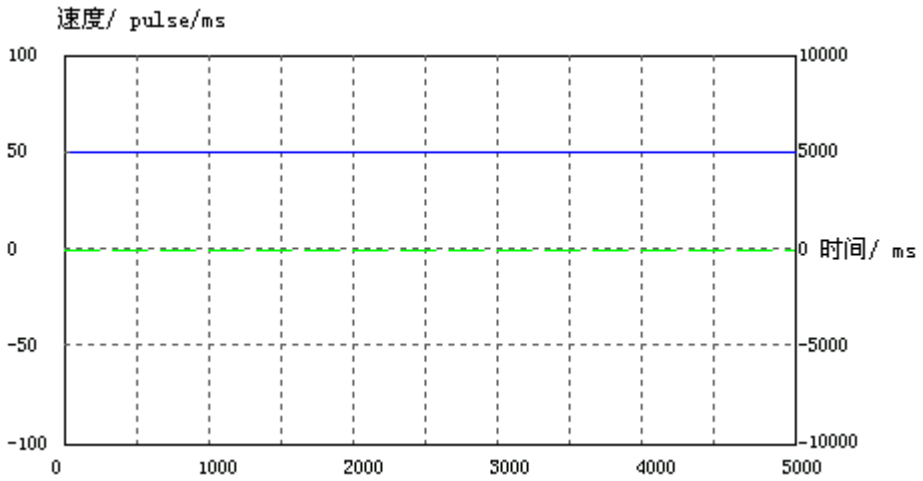

图 7-19 Follow 双 FIFO 切换主轴速度规划

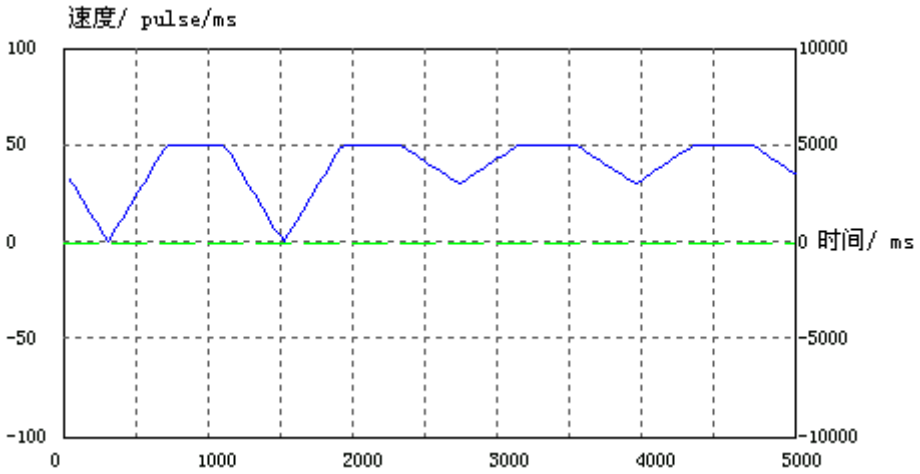

图 7-20 Follow 双 FIFO 切换从轴速度规划

```
PROGRAM PLC_PRG
VAR
    xInitDone:BOOL:= FALSE;
    rtn:INT;                    (*指令返回值变量*)
    Master:INT:=1;              (*定义主轴轴号: 1*)
    Slave:INT:=2;               (*定义从轴轴号: 2*)
    xStart: BOOL:= FALSE;
    jog:TJogPrm;                (*Jog运动参数*)
    arrPrfVel:ARRAY [0..7] OF LREAL;  (*规划速度*)
    space:DINT;
    masterPos:DINT;
    slavePos:LREAL;
    stage:INT:= STAGE_END;
END_VAR
VAR CONSTANT
    STAGE_FIFO1         :INT:= 1;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        STAGE_TO_FIFO2    :INT:= 2;
        STAGE_TO_FIFO1    :INT:= 3;
        STAGE_END         :INT:= 4;
    END_VAR
    (* @END_DECLARATION := '0' *)
    IF NOT xInitDone THEN
        (*配置运动控制器*)
        (*注意：配置文件取消了各轴的报警和限位*)
        rtn:= GT_LoadConfig('test.cfg);
        commandhandler('GT_LoadConfig', rtn);
        (*清除各轴的报警和限位*)
        rtn:= GT_ClrSts(1,8);
        commandhandler('GT_ClrSts', rtn);
        (*伺服使能*)
        rtn:= GT_AxisOn(Master);
        commandhandler('GT_AxisOn', rtn);
        rtn:= GT_AxisOn(Slave);
        commandhandler('GT_AxisOn', rtn);
        xInitDone:= TRUE;
    END_IF
    IF xStart THEN
        CASE stage OF
        STAGE_END:
            (*位置清零*)
            rtn:= GT_ZeroPos(Master,1);
            commandhandler('GT_ZeroPos', rtn);
            rtn:= GT_ZeroPos(Slave,1);
            commandhandler('GT_ZeroPos', rtn);
            (*将主轴设为 Jog 模式*)
            rtn:= GT_PrfJog(Master);
            commandhandler('GT_PrfJog', rtn);
            (*设置主轴运动参数*)
            rtn:= GT_GetJogPrm(Master, ADR(jog));
            commandhandler('GT_GetJogPrm', rtn);
            jog.acc := 1;
            rtn:= GT_SetJogPrm(Master, ADR(jog));
            commandhandler('GT_SetJogPrm', rtn);
            rtn:= GT_SetVel(Master, 50);
            commandhandler('GT_SetVel', rtn);
            (*启动主轴*)
            rtn:= GT_Update(SHL(WORD#1,Master-1));
            commandhandler('GT_Update', rtn);
            (*将从轴设为 Follow 模式*)
            rtn:= GT_PrfFollow(Slave,0);
            commandhandler('GT_PrfFollow', rtn);
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        (*清空从轴 FIFO*)
        rtn:= GT_FollowClear(Slave,0);
        commandhandler('GT_FollowClear', rtn);
        (*设置主轴，默认跟随主轴规划位置*)
        rtn:= GT_SetFollowMaster(Slave,Master, FOLLOW_MASTER_PROFILE,1);
        commandhandler('GT_SetFollowMaster', rtn);
        (*查询 Follow 模式的剩余空间*)
        rtn:= GT_FollowSpace(Slave,ADR(space),0);
        (*向 FIFO1 中增加运动数据*)
        masterPos:= 20000;
        slavePos:= 10000;
        rtn:= GT_FollowData(Slave, masterPos, slavePos,
                FOLLOW_SEGMENT_NORMAL,0);
        commandhandler('GT_FollowData', rtn);
        (*查询 Follow 模式 FIFO1 的剩余空间*)
        rtn:= GT_FollowSpace(Slave,ADR(space),0);
        (*向 FIFO1 中增加运动数据*)
        masterPos:= masterPos+20000;
        slavePos:= slavePos+20000;
        rtn:= GT_FollowData(Slave, masterPos, slavePos,
                FOLLOW_SEGMENT_NORMAL,0);
        commandhandler('GT_FollowData',rtn);
        (*查询 Follow 模式 FIFO1 的剩余空间*)
        rtn:= GT_FollowSpace(Slave,ADR(space),0);
        (*向 FIFO1 中增加运动数据*)
        masterPos:= masterPos+20000;
        slavePos:= slavePos+10000;
        rtn:= GT_FollowData(Slave, masterPos, slavePos,
                FOLLOW_SEGMENT_NORMAL,0);
        commandhandler('GT_FollowData', rtn);
        (*设置循环次数为无限循环*)
        rtn:= GT_SetFollowLoop(Slave, 0);
        commandhandler('GT_SetFollowLoop', rtn);
        (*设置启动跟随条件*)
        rtn:= GT_SetFollowEvent(Slave, FOLLOW_EVENT_START, 1, 0);
        commandhandler('GT_SetFollowEvent', rtn);
        (*启动从轴 Follow 运动*)
        rtn:= GT_FollowStart(SHL(WORD#1,Slave-1),0);
        commandhandler('GT_FollowStart', rtn);
        stage:= STAGE_FIFO1;

    STAGE_FIFO1:
        (*向FIFO2中发送过渡数据*)
        rtn:= GT_FollowClear(Slave, 1);
        masterPos:= 20000;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
                slavePos:= 10000;
                rtn:= GT_FollowData(Slave, masterPos, slavePos,
                        FOLLOW_SEGMENT_CONTINUE, 1);
                masterPos:= masterPos+20000;
                slavePos:= slavePos+20000;
                rtn:= GT_FollowData(Slave, masterPos, slavePos,
                        FOLLOW_SEGMENT_NORMAL, 1);
                masterPos:= masterPos+220000;
                slavePos:= slavePos+16000;
                rtn:= GT_FollowData(Slave, masterPos, slavePos,
                        FOLLOW_SEGMENT_NORMAL, 1);
                (*切换到FIFO2*)
                (*当前工作FIFO中的数据遍历完以后才会切换FIFO*)
                rtn:= GT_FollowSwitch(SHL(WORD#1,Slave-1));
                stage:= STAGE_TO_FIFO2;
            END_CASE
            xStart:= FALSE;
        END_IF
        (*检查 FIFO1 是否被清空，如果已被清空，则将新的速度规划传入 FIFO1 中，并且切换运行 FIFO1
    中数据*)
        CASE stage OF
        STAGE_TO_FIFO2:
            (*查询 FIFO1 的剩余空间*)
            rtn:= GT_FollowSpace(Slave,ADR(space),0);
            (*如果 FIFO1 被清空，说明已经切换到 FIFO2*)
            IF space = 16 THEN
                stage:= STAGE_TO_FIFO1;
                masterPos:= 20000;
                slavePos:= 16000;
                rtn:= GT_FollowData(Slave, masterPos, slavePos,
                        FOLLOW_SEGMENT_ CONTINUE,0);
                masterPos:= masterPos+20000;
                slavePos:= slavePos+20000;
                rtn:= GT_FollowData(Slave, masterPos, slavePos,
                        FOLLOW_SEGMENT_NORMAL,0);
                masterPos:= masterPos+20000;
                slavePos:= slavePos+16000;
                rtn:= GT_FollowData(Slave, masterPos, slavePos,
                        FOLLOW_SEGMENT_NORMAL,0);
                (*切换到 FIFO1*)
                (*当前工作 FIFO 遍历完以后才会切换 FIFO*)
                rtn:= GT_FollowSwitch(SHL(WORD#1,Slave-1));
            END_IF

        STAGE_TO_FIFO1:
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        (*查询 FIFO2 的剩余空间*)
        rtn:= GT_FollowSpace(Slave,ADR(space),1);
        (*如果 FIFO2 被清空，说明已经切换到 FIFO1*)
        IF space = 16 THEN
            stage:= STAGE_END;
        END_IF
    END_CASE

    (*读取轴的规划速度*)
    rtn:= GT_GetPrfVel(1, ADR(arrPrfVel),8,0);
    END_PROGRAM
```

以上几种运动模式的指令中有部分指令的调用比较消耗时间（约 1ms），请在编写程序时不要反复调用。他们是 GT_Update、GT_PtStart、GT_GearStart、GT_FollowStart。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第8章 访问硬件资源

8.1 本章简介

提示	本章表格中右侧的数字为“页码”，其中指令右侧的为“**第 13 章 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 GT_AlarmOff）均带有超级链接，点击可跳转至指令说明。

运动控制器包含的硬件资源分为如下几类：

表 8-1 运动控制器硬件资源

资源	说明
编码器	对外部编码器的脉冲输出进行计数

本章介绍如何在应用程序中使用这些硬件资源。

8.2 访问编码器

8.2.1 指令列表

表 8-2 访问编码器指令列表

指令	说明	页码
GT_GetEncPos	读取编码器位置	118
GT_GetEncVel	读取编码器速度	118
GT_SetEncPos	修改编码器位置	137

控制器内部为每个轴配置了脉冲计数装置。脉冲模块没有外部编码器反馈信号接口，所以调用 GT_GetEncPos 读取的将是运动控制器向驱动器发出的脉冲个数。如果读取的是 EtherCAT 总线轴的编码器位置，则是总线上反馈的实际编码器增量值。

调用 GT_SetEncPos 修改编码器位置的值。例如，设置轴 1 的编码器位置为 0，则接下来的编码器计数从 0 开始。若设置为 1000，则从 1000 开始。

8.2.2 例程

例程 8-1 访问编码器

```
PROGRAM PLC_PRG
VAR
    rtn:INT;  (*指令返回值*)
    i:INT;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        enc:ARRAY [0..7] OF LREAL;
        vel:ARRAY [0..7] OF LREAL;
    END_VAR
    (* @END_DECLARATION := '0' *)
    (*读取8个编码器的位置值*)
    rtn:= GT_GetEncPos(1, ADR(enc), 8,0);
    (*读取8个编码器的速度值*)
    rtn:= GT_GetEncVel(1, ADR(vel), 8,0);
    END_PROGRAM
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第9章 安全机制

9.1 本章简介

本章介绍运动控制器提供给用户的所有可用安全机制，包括：限位、报警、平滑停止、紧急停止。

提示	本章表格中右侧的数字为“页码”，其中指令右侧的为“**第 13 章 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 GT_AlarmOff）均带有超级链接，点击可跳转至指令说明。

9.2 限位

运动控制器能够通过安装限位开关或者设置软限位来限制各轴的运动范围，如图 9-1 所示：

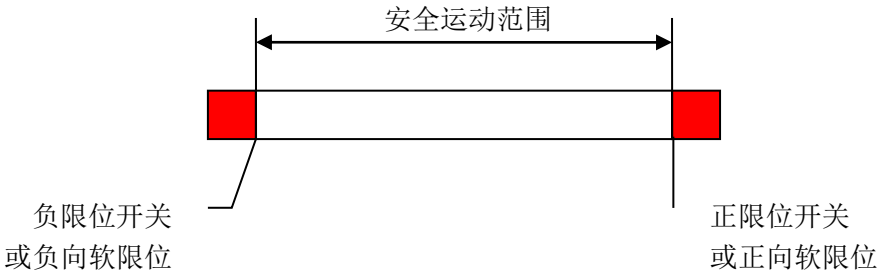

图 9-1 轴运动范围

工作台碰到限位开关或者规划位置超越软限位时，运动控制器紧急停止工作台的运动。限位触发以后，运动控制器禁止触发限位方向上运动，同时该轴的限位触发状态置 1。离开限位回到安全运动范围以后，需要调用指令 GT_ClrSts 清除限位触发状态，才能使控制轴回到正常运动状态。

控制器复位后，默认软限位是无效的，没有触发的。

注意	限位标志为模态标志，一旦置起，需在离开限位回到安全运动范围以后，调用指令 GT_ClrSts 清除限位触发状态

9.2.1 指令列表

表 9-1 软限位指令列表

指令	说明	页码
GT_SetSoftLimit	设置轴正向软限位和负向软限位	144
GT_GetSoftLimit	读取轴正向软限位和负向软限位	125

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

9.2.2 重点说明

应当在回原点以后再设置软限位。正向软限位必须大于负向软限位。软限位和限位开关可以同时使用，当软限位触发时也会置起限位触发标志。

限位触发以后使用急停加速度紧急停止。默认急停加速度为 1000 脉冲/毫秒 ^{2}，如何设置急停加速度请参见“4.3.3 配置 profile”。

9.2.3 例程

例程 9-1 软限位使用

该例程设置了第一轴的软限位，正向在 20000 处，负向在-20000 处。启动第一轴的点位运动后，通过下图 9-2 可以看到，当运动超过正向软限位时，限位信号触发，运动停止。

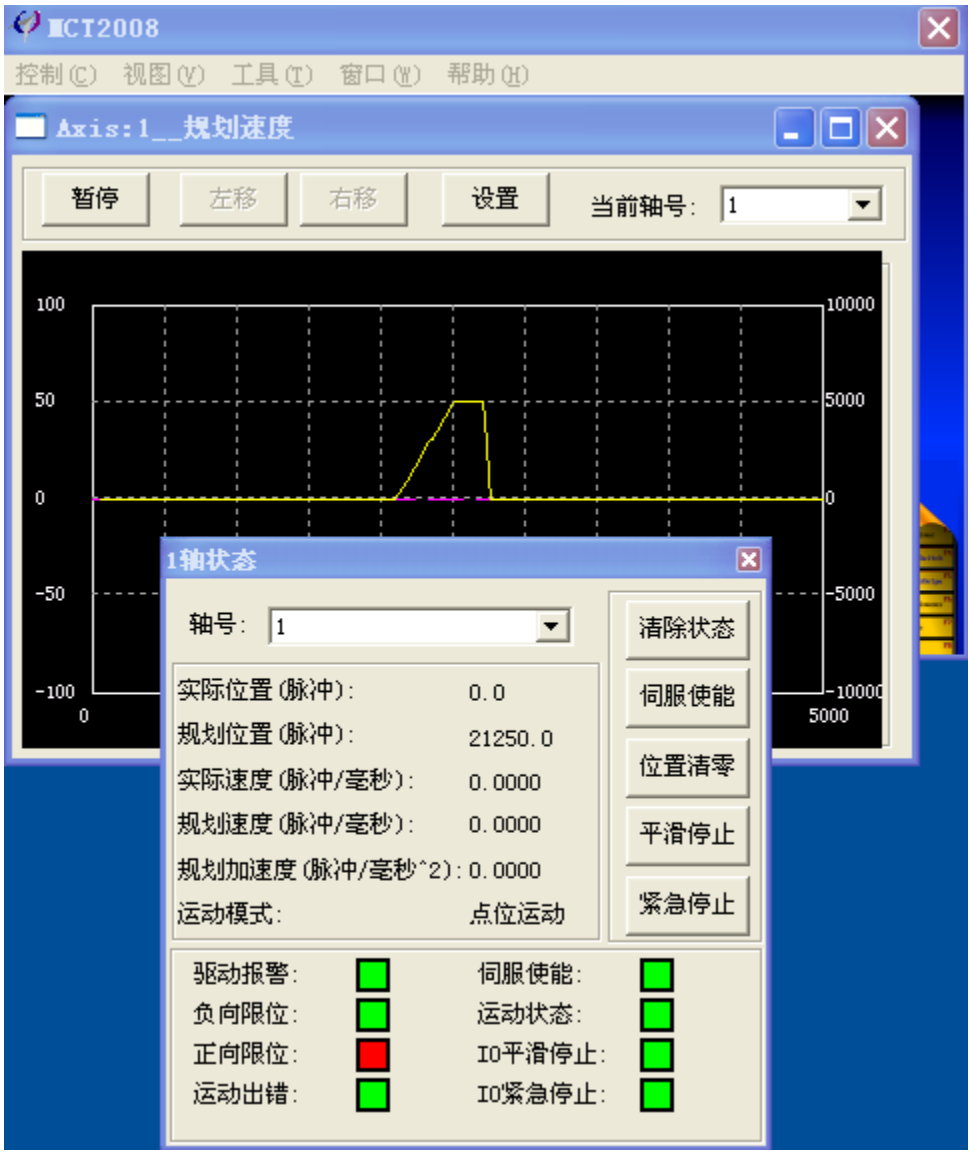

图 9-2 软限位触发

```
PROGRAM PLC_PRG
VAR
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        xInitDone:BOOL:= FALSE;
        rtn:INT;                    (*指令返回值变量*)
        Axis:INT:=1;                (*定义轴号: 1*)
        xStart: BOOL:= FALSE;
        trap:TTrapPrm;              (*点位运动参数*)
        lrPrfPos:LREAL;             (*规划位置*)
        dwAxisStatus:DWORD;         (*轴状态*)
    END_VAR
    (* @END_DECLARATION := '0' *)
    IF NOT xInitDone THEN
        (*配置运动控制器*)
        rtn:= GT_LoadConfig('test.cfg);
        (*清除轴状态*)
        rtn:= GT_ClrSts(1,8);
        (*设置软限位*)
        rtn:= GT_SetSoftLimit(Axis, 20000, -20000);
        (*伺服使能*)
        rtn:= GT_AxisOn(Axis);
        xInitDone:= TRUE;
    END_IF
    IF xStart THEN
        (*将AXIS轴设为点位模式*)
        rtn:= GT_PrfTrap(Axis);
        (*读取点位运动参数(需要读取所有运动参数到上位机变量*)
        rtn:= GT_GetTrapPrm(Axis, ADR(trap));
        (*设置需要修改的运动参数*)
        trap.acc := 0.125;
        trap.dec := 0.125;
        trap.smoothTime := 25;
        (*设置点位运动参数*)
        rtn:= GT_SetTrapPrm(Axis, ADR(trap));
        (*设置 AXIS 轴的目标位置*)
        rtn:= GT_SetPos(Axis, 1000000);
        (*设置AXIS轴的目标速度*)
        rtn:= GT_SetVel(Axis, 50);
        (*启动AXIS轴的运动*)
        rtn:= GT_Update(SHL(WORD#1,Axis-1));
        xStart:= FALSE;
    END_IF
    (*读取AXIS轴的状态*)
    rtn:= GT_GetSts(Axis, ADR(dwAxisStatus),1,0);
    (*读取AXIS轴的规划位置*)
    rtn:= GT_GetPrfPos(Axis, ADR(lrPrfPos),1,0);
    END_PROGRAM
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

9.3 报警

运动控制器提供专用的驱动报警信号输入接口。当检测到驱动器报警信号以后，运动控制器将关闭该轴的伺服使能，急停运动规划，同时该轴报警触发标志置 1。

驱动器报警信号产生以后，应当执行以下操作：

1. 确定引起驱动器报警的原因，并加以改正
2. 复位驱动器
3. 调用GT_ClrSts清除报警，重新回机床原点

9.4 平滑停止和急停

运动控制器的每个轴都可以定义平滑停止 IO 和急停 IO。

当平滑停止 IO 输入为触发电平时（触发电平可以设置），运动控制器自动平滑停止所关联的控制轴，并将轴状态字（bit7）置 1。

当急停 IO 输入为触发电平时（触发电平可以设置），运动控制器自动紧急停止所关联的控制轴，并将轴状态字（bit8）置 1。

IO 平滑停止或者 IO 急停完成以后，必须调用 GT_ClrSts 指令清除停止标志位（bit7 和 bit8），才能继续运动。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第10章 运动程序

10.1 本章简介

本章将介绍下载在控制器中运行的程序——运动程序。用户想要使用运动程序，需要了解相应的语法，以及下载步骤。本章将一一介绍。

提示	本章表格中右侧的数字为“页码”，其中指令右侧的为“**第 13 章 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 GT_AlarmOff）均带有超级链接，点击可跳转至指令说明。

10.2 运动程序概述

为了表述方便，直接在 PC 机上调用动态链接库发送指令访问控制器的程序称为'应用程序'，下载到运动控制器上执行的程序称为'运动程序'。运动程序与应用程序的关系如图 10-1 所示。

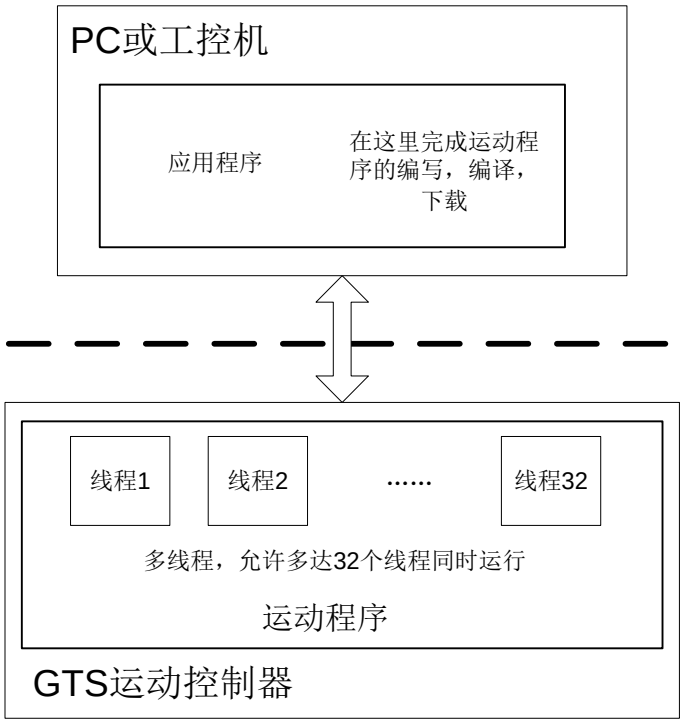

图 10-1 运动程序与应用程序的关系

运动程序的三个特点：独立性，实时性，并行性。

独立性：运动程序能够脱离主机在运动控制器上独立执行，主机能够将CPU资源分配给其它任务，从而将主机从繁琐的运动逻辑管理中解放出来。当然，如果需要，主机仍然可以在任何时候向控制器发送指令，即使运动控制器上的运动程序正在执行。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

注意	当主机指令和运动控制器上的运动程序控制相同的轴时，需仔细设计运动逻辑，以免造成混乱。

实时性：运动控制器上执行运动程序由于不需要通过总线和主机进行频繁的数据交换，因此具有更高的实时性。和在主机上执行的应用程序不同之处在于，运动程序对GT指令调用不必再通过PC总线，因此具有更高的执行效率。平均执行速度约为100指令/毫秒，是PC机执行API指令速度的5倍。

并行性：支持多任务，允许多达32个运动程序在运动控制器上同时执行。

注意	在多线程环境下，一个线程中连续的 2 条指令在执行时有可能被插入其它线程的指令。当启动多个线程并行执行时，应当仔细考虑线程之间是否会相互影响。

10.3 运动程序的使用

10.3.1 编写运动程序

运动程序可以使用C语言编写。但是一些编写规则和C语言略有不同。用户编写运动程序时应遵照”10.4.1 语言元素”、”10.4.2 运算指令”中的说明，否则有可能编译不通过。

运动程序可以和应用程序一样调用GT指令。用户可查阅”10.5 可在运动程序中使用的指令”知道哪些指令可以在运动程序中调用。

注意	运动程序中，调用 GT 指令必须完整描述函数的每一个参数。 例如：short GT_GetClock(unsigned long *pClock, unsigned long *pLoop=NULL)。 在应用程序中调用，可以写成如下形式，使用 VC 可以编译通过： long lClock; GT_GetClock(&lClock); 在运动程序中调用，必须写成如下形式，否则编译不通过： long lClock, lLoop; GT_GetClock(&lClock, &lLoop);

10.3.2 编译

为了让运动控制器能够执行用户用C语言编写的运动程序，必须对运动程序进行编译。使用MCT2008编译运动程序，生成目标程序文件（*.bin）和符号文件（*.ini）。目标文件用来下载到运动控制器。符号文件用来保存运动程序编译信息。应用程序必须使用这2个文件才能正确下载和启动运动程序，访问运动程序的变量。

关于如何使用MCT2008编译运动程序，请参考’MCT2008使用帮助’。具体操作：启动MCT2008，点击主界面菜单’帮助’-->‘MCT2008使用帮助’，将会弹出MCT2008使用帮助对话框。在对话框左侧’目录’一页点击’工具’-->‘运动程序编译器.mht’，可以查看详细的如何编译运动程序的说明。如图 10-2所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

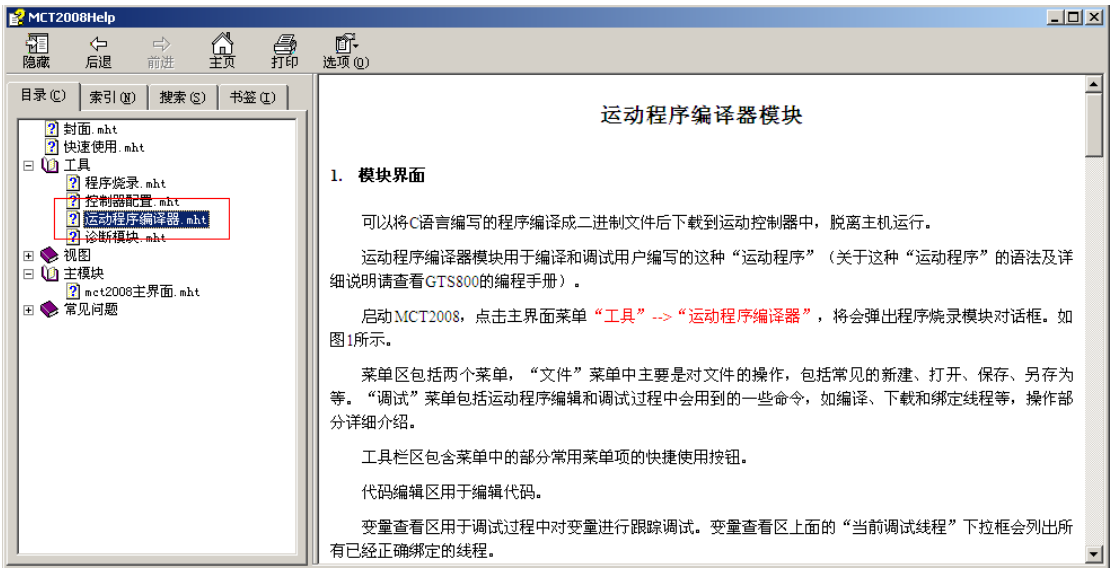

图 10-2 MCT2008 运动程序编译说明界面

编译成功后，目标程序文件（*.bin）和符号文件（*.ini）会出现在运动程序文件（*.c）的同一目录下。

注意	如果编译不成功，则无法进行下一步操作。用户应按照编译器的提示仔细检查错误原因。

10.3.3 指令列表

后面的操作需要用户在应用程序中调用相关的运动程序指令来完成。

表 10-1 运动程序指令列表

指令	说明	页码
GT_Compile	编译运动程序	105
GT_Download	下载运动程序到运动控制器	105
GT_GetFunId	读取运动程序中函数的标识	120
GT_GetVarId	读取运动程序中变量的标识	128
GT_Bind	绑定线程、函数、数据页	104
GT_RunThread	启动线程	135
GT_StopThread	停止正在运行的线程	149
GT_PauseThread	暂停正在运行的线程	131
GT_GetThreadSts	读取线程的状态	126
GT_SetVarValue	设置运动程序中变量的值	146
GT_GetVarValue	读取运动程序中变量的值	128

10.3.4 下载

编译成功后，用户需要在应用程序中调用 GT_Download 指令将目标文件（*.bin）下载到运动控制器的 SDRAM 中。用户应保证目标文件和符号文件与应用程序在同一目录下。

当下载新的运动程序时会覆盖原有的运动程序。运动控制器每次上电以后需要重新下载运动程序。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

10.3.5 绑定线程、函数和数据页

运动程序下载到运动控制器后，还不能立即执行。运动控制器会自动为运动程序中的每个函数，全局变量和局部变量定义好 ID，调用 GT_GetFunId 读取函数 ID，调用 GT_GetVarId 读取变量 ID。

为了执行运动程序，必须调用 GT_Bind 指令绑定线程、函数和数据页。

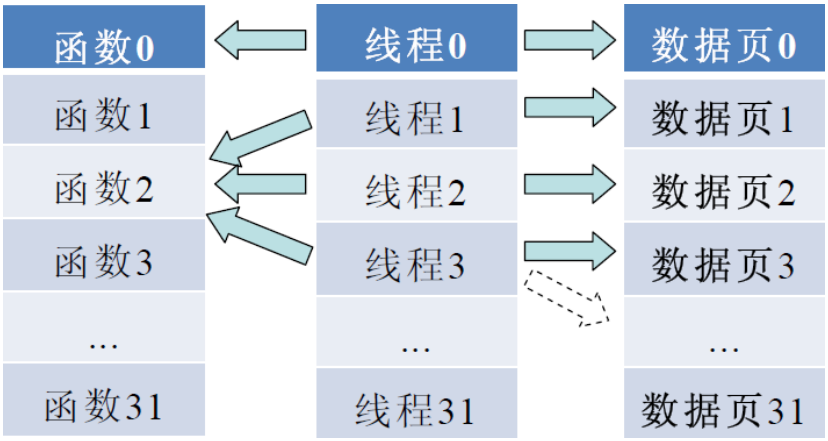

图 10-3 线程、函数和数据页的关系

运动控制器支持 32 个线程同时运行，一个线程只能分配一个函数，但是一个函数可以分配给多个线程同时执行，例如多轴回零时，可以让多个线程绑定同一个回零函数，然后同时启动这些线程就可以实现多轴同时回零。在线程执行过程中不允许绑定新的函数，除非线程执行完毕。

各函数的局部变量放在相互独立的数据页中。运动控制器提供 32 个数据页。在绑定线程和函数时，必须指明所使用的数据页。一个数据页只能分配给一个线程，但是一个线程可以使用多个数据页。线程在执行过程中可以切换数据页。其关系示意如图 10-3 所示。

10.3.6 启动、停止、暂停线程

将线程、函数和数据页绑定好后，调用 GT_RunThread 指令就可以启动某一个线程。调用 GT_StopThread

停止某个正在运行的线程，调用 GT_PauseThread

暂停线程。

10.3.7 查询线程状态

应用程序可以随时调用 GT_GetThreadSts 指令查询线程的执行状态。包括该线程是否正在运行，线程绑定的函数的返回值，当前正在执行的指令所在行数，当前正在执行的指令的返回值。

应用程序可以随时调用 GT_SetVarValue 指令更新运动程序中所有变量的值。

应用程序可以随时调用 GT_GetVarValue 指令查询运动程序中所有变量的值。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

10.3.8 例程

1. 单线程累加求和

例程 10-1 运动程序单线程累加求和

运动程序完成累加求和任务。定义了全局变量 sum 用于保存累加和，局部变量 begin 用于保存累加起点，局部变量 end 用于保存累加终点。累加完成以后程序结束。

```
/*----------------------------------------------
      累加求和
      begin  累加起点
      end     累加终点
----------------------------------------------*/
intsum;

intadd(intbegin, intend)
{
      inti;
      intcc;

      i=begin;
lbl_loop:
      cc = i>end;
      if(cc) gotolbl_end;
      sum = sum + i;
      i = i + 1;
      gotolbl_loop;
lbl_end:
      returnsum;
}
```

应用程序负责编译、下载、初始化、启动运动程序。

```
PROGRAM PLC_PRG
VAR
      xInitDone:BOOL:= FALSE;
      rtn:INT;
      sDownloadFileName:STRING:= 'sum.bin';
      sFunname:STRING:= ‘add’;
      sVarname:STRING:= ‘sum’;
      funId:INT;
      sum, begin, end:TVarInfo;
      value:LREAL;
      thread:TThreadSts;
END_VAR
(* @END_DECLARATION := '0' *)
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
IF NOT xInitDone THEN
    (*下载运动程序sum.bin，必须保证sum.bin文件位于工程文件夹中*)
    rtn:= GT_Download(ADR(sDownloadFileName));
    (*获取函数ID*)
    rtn:= GT_GetFunId(ADR(sFunname), ADR(funId));
    (*获取全局变量sum的ID*)
    sVarName:= 'sum';
    rtn:= GT_GetVarId(0, ADR(sVarname), ADR(sum));
    (*获取局部变量begin的ID*)
    sVarName:= 'begin';
    rtn:= GT_GetVarId(ADR(sFunname), ADR(sVarname), ADR(begin));
    (*获取局部变量end的ID*)
    sVarName:= 'end';
    rtn:= GT_GetVarId(ADR(sFunname), ADR(sVarname), ADR(end));
    (*绑定线程，函数，数据页*)
    rtn:= GT_Bind(0, funId, 0);
    value = 0;
    (*初始化运动程序的全局变量sum*)
    rtn:= GT_SetVarValue(-1, ADR(sum), ADR(value),1);
    value = 1;
    (*初始化运动程序的局部变量begin*)
    rtn:= GT_SetVarValue(0, ADR(begin), ADR(value),1);
    value = 100;
    (*初始化运动程序的局部变量end*)
    rtn:= GT_SetVarValue(0, ADR(end), ADR(value),1);
    (*启动线程*)
    rtn:= GT_RunThread(0);
    xInitDone:= TRUE;
END_IF
(*查询线程状态*)
rtn:= GT_GetThreadSts(0, ADR(thread));
(*查询全局变量sum的值*)
rtn:= GT_GetVarValue(-1, ADR(sum), ADR(value),1);
END_PROGRAM
```

2. 多线程累加求和

例程 10-2 运动程序多线程累加求和

运动程序代码和例程 10-1 相同。

应用程序负责编译、下载、初始化、启动运动程序。和例程 10-1 不同之处在于启动 2 个线程完成累加运算任务。

```
PROGRAM PLC_PRG
VAR
    xInitDone:BOOL:= FALSE;
    rtn:INT;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        sDownloadFileName:STRING:= 'sum.bin';
        sFunname:STRING:= ‘add’;
        sVarname:STRING:= ‘sum’;
        funId:INT;
        sum, begin, end:TVarInfo;
        value:LREAL;
        thread:TThreadSts;
    END_VAR
    (* @END_DECLARATION := '0' *)
    IF NOT xInitDone THEN
        (*下载运动程序sum.bin，必须保证sum.bin文件位于工程文件夹中*)
        rtn:= GT_Download(ADR(sDownloadFileName));
        (*获取函数ID*)
        rtn:= GT_GetFunId(ADR(sFunname), ADR(funId));
        (*获取全局变量sum的ID*)
        sVarName:= ‘sum’;
        rtn:= GT_GetVarId(0, ADR(sVarname), ADR(sum));
        (*获取局部变量begin的ID*)
        sVarName:= ‘begin’;
        rtn:= GT_GetVarId(ADR(sFunname), ADR(sVarname), ADR(begin));
        (*获取局部变量end的ID*)
        sVarName:= ‘end’;
        rtn:= GT_GetVarId(ADR(sFunname), ADR(sVarname), ADR(end));
        (*绑定线程，函数，数据页*)
        rtn:= GT_Bind(0, funId, 0);
        rtn:= GT_Bind(1, funId, 1);
        value = 0;
        (*初始化运动程序的全局变量sum*)
        rtn:= GT_SetVarValue(-1, ADR(sum), ADR(value),1);
        value = 1;
        (*初始化运动程序的局部变量begin*)
        rtn:= GT_SetVarValue(0, ADR(begin), ADR(value),1);
        value = 50;
        (*初始化运动程序的局部变量end*)
        rtn:= GT_SetVarValue(0, ADR(end), ADR(value),1);
        value = 51;
        (*初始化运动程序的局部变量begin*)
        rtn:= GT_SetVarValue(1, ADR(begin), ADR(value),1);
        value = 100;
        (*初始化运动程序的局部变量end*)
        rtn:= GT_SetVarValue(1, ADR(end), ADR(value),1);
        (*启动线程*)
        rtn:= GT_RunThread(0);
        rtn:= GT_RunThread(1);
        xInitDone:= TRUE;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
    END_IF
    (*查询线程状态*)
    rtn:= GT_GetThreadSts(0, ADR(thread));
    rtn:= GT_GetThreadSts(1, ADR(thread));
    (*查询全局变量sum的值*)
    rtn:= GT_GetVarValue(-1, ADR(sum), ADR(value),1);
    END_PROGRAM
```

10.4 如何编写运动程序

10.4.1 语言元素

1. 数据类型

支持整型和浮点型 2 种数据类型。

- 整型 32 位，取值范围是-2, 147, 483, 648 ~ 2, 147, 483, 647。
- 浮点型采用定点格式，32 位整数，16 位小数。所能表示的最小精度为(1/2)^16=0.0000152587890625。

2. 常量

可以在程序中直接使用立即数和宏。立即数可以是 10 进制整数、16 进制整数和浮点数。

3. 变量

可以声明局部变量和全局变量。每个函数最多可声明 1024 个局部变量。全局变量最多可声明 1024 个。整型类型说明符为 int。浮点型类型说明符为 double。

4. 数组

支持一维数组，支持常量下标索引和变量下标索引。

不支持多维数组，不支持用数组元素进行下标索引。

5. 函数

函数可以定义返回值类型和输入形参类型。

不支持在函数中调用自定义函数，但是可以调用 GT 运动控制指令。

6. 数据类型转换

支持强制数据类型转换。强制数据类型转换符有 int, double。

- 数据类型转换符必须加括号，如 a=(int)b
- 数据类型转换不会改变变量本身的数据类型定义

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

10.4.2 运算指令

支持算术运算、逻辑运算、关系运算、位运算，语法规则和 C 语言相同，但是不支持复杂表达式，只能使用 2 个操作数进行运算，而且这 2 个操作数的数据类型必须相同。

注意：由于运动程序中的浮点数据类型只有 16 位小数精度，请不要在运动程序中进行高精度浮点运算。

1. 算术运算

用于各类数值运算。包括加(+)、减(-)、乘(*)、除(/)、求余(或称模运算，%)共五种。

2. 逻辑运算

逻辑运算包括与(&&)、或(||)、非(!)三种。参数可以是整型变量、或者整型常数。

3. 关系运算

关系运算符用于比较运算。包括大于(>)、小于(<)、等于(==)、大于等于(>=)、小于等于(<=)和不等于(!=)六种。参与比较的参数类型必须一致。

4. 位运算

参与运算的量，按二进制位进行运算。包括位与(&)、位或(|)、位非(~)、位异或(^)、左移(<<)、右移(>>)六种。

5. 流程控制

支持条件跳转、条件返回，语法规则和 C 语言相同。

1) 条件跳转如下所示：

```
if(var) goto label;
```

当条件变量 var 非 0 时，跳转到标记为 label 的指令。

不支持表达式作为判断条件。

2) 条件返回如下所示：

```
if(var) return value;
```

当条件变量 var 非 0 时，程序返回，返回值为 value。

不支持表达式作为判断条件。

10.4.3 流程控制与标准 C 语言的流程控制对比

(1) While 语句

运动程序实现：

```
inti,sum,cc;
i = 1;
sum = 0;
```

标准 C 实现：

```
inti,sum;
i = 1;
sum = 0;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
lbl_loop:
    cc = i> 10;
    if(cc) gotolbl_end;
    sum = sum + i;
    i = i + 1;
    gotolbl_loop;
lbl_end:
……
```

```
while(i<= 10)
{
    sum = sum + i;
    i = i + 1;
}
```

以上程序等同于：

运动程序实现：

```
inti,sum,cc;
i = 1;
sum = 0;
lbl_loop:
        sum = sum + i;
        i = i + 1;

    cc = i<= 10;
    if(cc)gotolbl_loop;
lbl_end:
……
```

标准 C 实现：

```
inti,sum;
i = 1;
sum = 0;
while(i<= 10)
{
    sum = sum + i;
    i = i + 1;
}
```

(2) for 语句

运动程序实现：

```
inti,sum,cc;
i = 1;
sum = 0;
lbl_loop:
    cc = i> 10;
    if(cc) gotolbl_end;
    sum = sum + i;
    i = i + 1;
    gotolbl_loop;
lbl_end:
……
```

标准 C 实现：

```
inti,sum;
sum = 0;
for(i=1;i<= 10;i++)
{
    sum = sum + i;
}
```

(3) if… else 语句

运动程序实现：

```
inta,b;
intcc;
a = 5;
b = 10;
cc = a<=b;
    if(cc) gotolbl_Jump;
```

标准 C 实现：

```
inta,b;
a = 5;
b = 10;
if(a>b)
{
    a = a - b;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
a = a - b;
gotolbl_end;
lbl_Jump:
    a = b - a;
lbl_end:
 ……
```

```
}
else
{
    a = b - a;
}
```

(4) switch…case

运动程序实现：

```
#define MC_GPO 12   //注意：该宏定义应放在函数体外
inta;
intcc;
a = 5;
cc = a==1;
    if(cc) gotolbl_1;
cc = a==2;
    if(cc) gotolbl_2;
cc = a==5;
    if(cc) gotolbl_3;
gotolbl_end;
lbl_1:
    GT_SetDo (MC_GPO,0x01);
    gotolbl_end;
lbl_2:
    GT_SetDo (MC_GPO,0x02);
    gotolbl_end;
lbl_3:
    GT_SetDo(MC_GPO,0x04);
    gotolbl_end;
lbl_end:
……
```

标准 C 实现：

```
inta;

a = 5;
switch(a)
{
case 1:
    GT_SetDo(MC_GPO,0x01);
    break;
  case 2:
    GT_SetDo (MC_GPO,0x02);
    break;
  case 5:
    GT_SetDo(MC_GPO,0x04);
break;
  default:
    break;
}
```

注意

编写运动程序时：

1. 对于 GT 指令，其指令的参数个数必须写全；
2. GT 指令中的参数宏定义，在运动程序当中不支持，需要重新定义。

10.5 可在运动程序中使用的指令

表 10-2 可在运动程序中使用的指令

编号	指令	指令	指令
1	GT_Delay	GT_PrfTrap	GT_GetExtAdValue
4	GT_DelayHighPrecision	GT_SetTrapPrm	GT_GetExtAdVoltage
7	GT_SetDataPage	GT_GetTrapPrm	GT_SetExtDaValue

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

编号	指令	指令	指令
10	GT_SetDo	GT_PrfJog	GT_SetExtDaVoltage
13	GT_SetDoBit	GT_SetJogPrm	GT_GetStsExtMdl
16	GT_GetDo	GT_GetJogPrm	GT_AlarmOff
19	GT_GetDi	GT_PrfGear	GT_AlarmOn
22	GT_GetDiReverseCount	GT_SetGearMaster	GT_LmtsOn
25	GT_SetDiReverseCount	GT_GetGearMaster	GT_LmtsOff
28	GT_SetDac	GT_SetGearRatio	GT_ProfileScale
31	GT_GetDac	GT_GetGearRatio	GT_EncScale
34	GT_GetDacValue	GT_GearStart	GT_StepDir
37	GT_GetEncPos	GT_ZeroPos	GT_StepPulse
40	GT_SetTrigger	GT_SetExtIoValue	GT_SetMtrBias
43	GT_GetTriggerStatus	GT_GetExtIoValue	GT_GetMtrBias
46	GT_LinkCaptureOffset	GT_SetExtIoBit	GT_SetMtrLmt
49	GT_GetCaptureOffset	GT_GetExtIoBit	GT_GetMtrLmt
52	GT_GetClock	GT_SetPid	GT_EncSns
55	GT_GetSts	GT_GetPid	GT_EncOn
58	GT_AxisOn	GT_Update	GT_EncOff
61	GT_Stop	GT_SetPos	GT_SetPosErr
64	GT_SynchAxisPos	GT_GetPos	GT_GetPosErr
67	GT_GetSoftLimit	GT_SetVel	GT_SetStopDec
70	GT_GetAxisBand	GT_GetVel	GT_GetStopDec
73	GT_GetPrfVel	GT_ZeroPos	GT_LmtSns
76	GT_GetPrfMode	GT_SetExtIoValue	GT_CtrlMode
79	GT_GetAxisPrfVel	GT_GetExtIoValue	GT_CrdSpace
82	GT_GetAxisEncPos	GT_SetExtIoBit	GT_CrdStart
85	GT_GetAxisEncAcc	GT_GetExtIoBit	GT_SetOverride
88	GT_SetControlFilter	GT_CrdStatus	GT_GetCrdVel
91	GT_GetControlFilter		

注意	在编写运动程序过程中必须使用 GT 指令，GT 指令请参照 GTS 相关的编程手册

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第11章 其它指令

11.1 本章简介

本章介绍运动控制器为用户提供的其他指令。包括：打开/关闭运动控制器、读取固件版本号、读取系统时钟、打开/关闭电机使能信号、维护位置值、电机到位检测等。

本章表格中右侧的数字为“页码”，其中指令右侧的为“**第 13 章 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 GT_AlarmOff）均带有超级链接，点击可跳转至指令说明。

11.2 复位运动控制器

表 11-1 复位运动控制器指令列表

指令	说明	页码
GT_Reset	复位运动控制器	135

调用 GT_Reset 指令将使运动控制器的所有寄存器恢复到默认状态，一般在打开运动控制器之后调用该指令。

11.3 读取固件版本号

表 11-2 读取固件版本号指令列表

指令	说明	页码
GT_GetVersion	读取运动控制器固件的版本号	129

为了方便用户核对运动控制器固件版本，提供 GT_GetVersion 指令来读取运动控制器固件版本号，版本号是一个含有 18 个字符的字符串：aaa bbbbbb ccc dddddd。具体的定义如表 11-3 所示。

表 11-3 固件版本号的定义格式

aaa	固件 1 的版本号，如 100，即表示版本号为：1.00
bbbbbb	固件 1 的版本号的生成时间，如 090908，即表示该版本生成于：2009 年 9 月 8 日
ccc	固件 2 的版本号
dddddd	固件 2 的版本号的生成时间

例程 11-1 读取控制器固件版本号

```
PROGRAM PLC_PRG
VAR
    rtn:INT;
    pVersion: POINTER TO STRING;          (*定义指向版本号字符串的指针*)
    firmwareVersion:STRING;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
END_VAR
(* @END_DECLARATION := '0' *)
rtn:= GT_GetVersion(ADR(pVersion));          (*读取版本号*)
firmwareVersion:=pVersion^;
END_PROGRAM
```

11.4 读取系统时钟

表 11-4 读取系统时钟指令列表

指令	说明	页码
GT_GetClock	读取运动控制器系统时钟	113

运动控制器上电初始化之后，内部计数时钟从 0 开始计数，每 1 毫秒增加 1，通过 GT_GetClock 指令可以读取该计数时钟的值。调用 GT_Reset 指令将会使计数时钟值清零。

11.5 打开/关闭电机使能信号

表 11-5 打开/关闭电机使能信号指令列表

指令	说明	页码
GT_AxisOn	打开驱动器使能	104
GT_AxisOff	关闭驱动器使能	103

调用 GT_AxisOn 指令将打开指定控制轴所连电机的伺服使能信号，使指定控制轴进入控制状态。

11.6 维护位置值

表 11-6 维护位置值指令列表

指令	说明	页码
GT_SetPrfPos	修改指定轴的规划位置	142
GT_SynchAxisPos	axis 合成规划位置和所关联的 profile 同步 axis 合成编码器位置和所关联的 encoder 同步	149
GT_ZeroPos	清零规划位置和实际位置，并进行零漂补偿	150

第 4 章 里提到，axis 含有 profile 和 encoder 的当量变换的功能，如果调用了 GT_SetPrfPos 指令或者 GT_SetEncPos 指令之后，profile 的输出值或者 encoder 的输出发生了变化，如果需要将 axis 当量变换之后的值与 profile 或者 encoder 的值同步，需要调用 GT_SynchAxisPos 指令。

11.7 电机到位检测

表 11-7 电机到位检测指令列表

指令	说明	页码
GT_SetAxisBand	设置轴到位误差带 规划器静止，规划位置和实际位置的误差小于设定误差带，并且在误差带内保持设定时间后，置起到位标志	135

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

GT_GetAxisBand	读取轴到位误差带	111

用户使用伺服电机时，由于伺服电机在运动的过程中可能会存在运动滞后，会出现规划停止，而实际位置并没有到位的情况。用户可以使用运动控制器的运动到位检测功能来判断电机是否实际到位。运动控制器默认该功能是无效的，当调用 GT_SetAxisBand 指令设置了相应的误差带和保持时间之后，该功能生效。

该功能生效后，当规划器处于静止状态，即轴状态寄存器 bit10 为 0(参见 6.3.1)，并且规划位置和编码器位置的误差在设定的误差带内保持了设定时间，轴状态寄存器 bit11 将被置 1(参见 6.3.1)。当规划器运动，或者规划位置和编码器位置的误差超出误差带时立即清 0。检测电机到位标志可以保证系统的定位精度，应当根据机械系统的实际情况设置合适的到位误差带和误差带保持时间。如果到位误差带设置的太小，或者误差带保持时间太长，都会使到位时间增长，影响加工效率。

如图 11-1 所示。当轴 1 启动电机到位检测功能，设置误差带为 100pulse，误差带保持时间为 500μs，当电机在 1000pulse 位置处静止时，会有震荡。控制器判断其震荡位于误差带内 500μs 后，就将轴状态的 bit11 电机到位标志置 1。蓝色阴影区域表示电机到位标志还未置 1，橙色阴影区域表示已置 1。可以看出，误差带保持时间越短，误差带越大，控制器会在越短时间内将电机到位标志置 1，但是并不是越短越好。电机是否连接机械本体，也会影响控制器判断电机到位所需的时间。

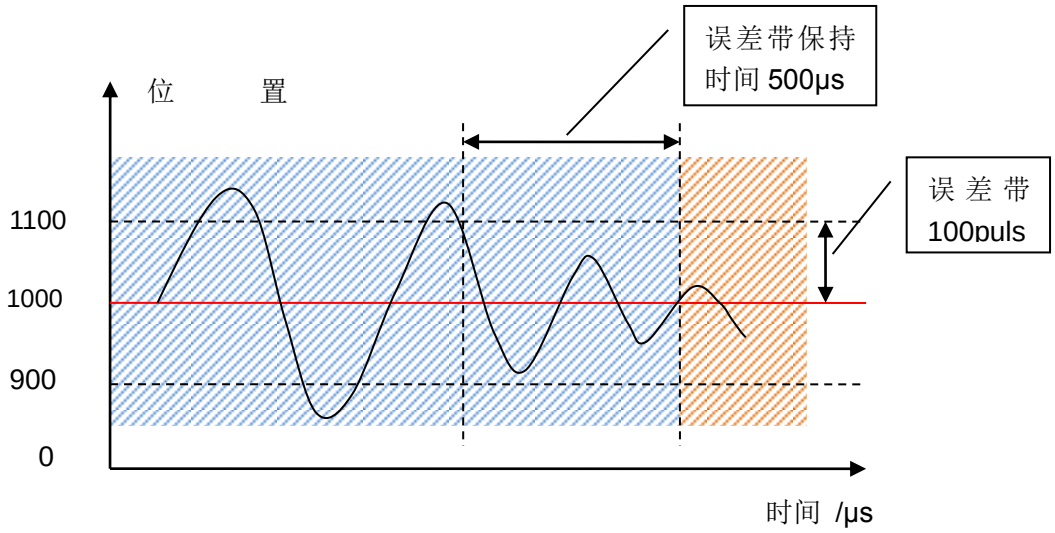

图 11-1 电机到位检测功能

注意	使用电机到位检测功能必须注意以下几点： 1. axis 正确关联编码器，并且编码器方向和规划运动方向必须一致。 2. 正确设置到位误差带，默认情况下到位误差带无效 3. 调用 GT_ZeroPos 进行对位置进行清零，同时进行自动零漂补偿。

例程 11-2 电机到位检测功能

下面这个例程示例电机到位检测的使用方法。一个轴运动到位以后，启动另一个轴的运动。电机到位的运动状态检测如图 11-2 所示。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

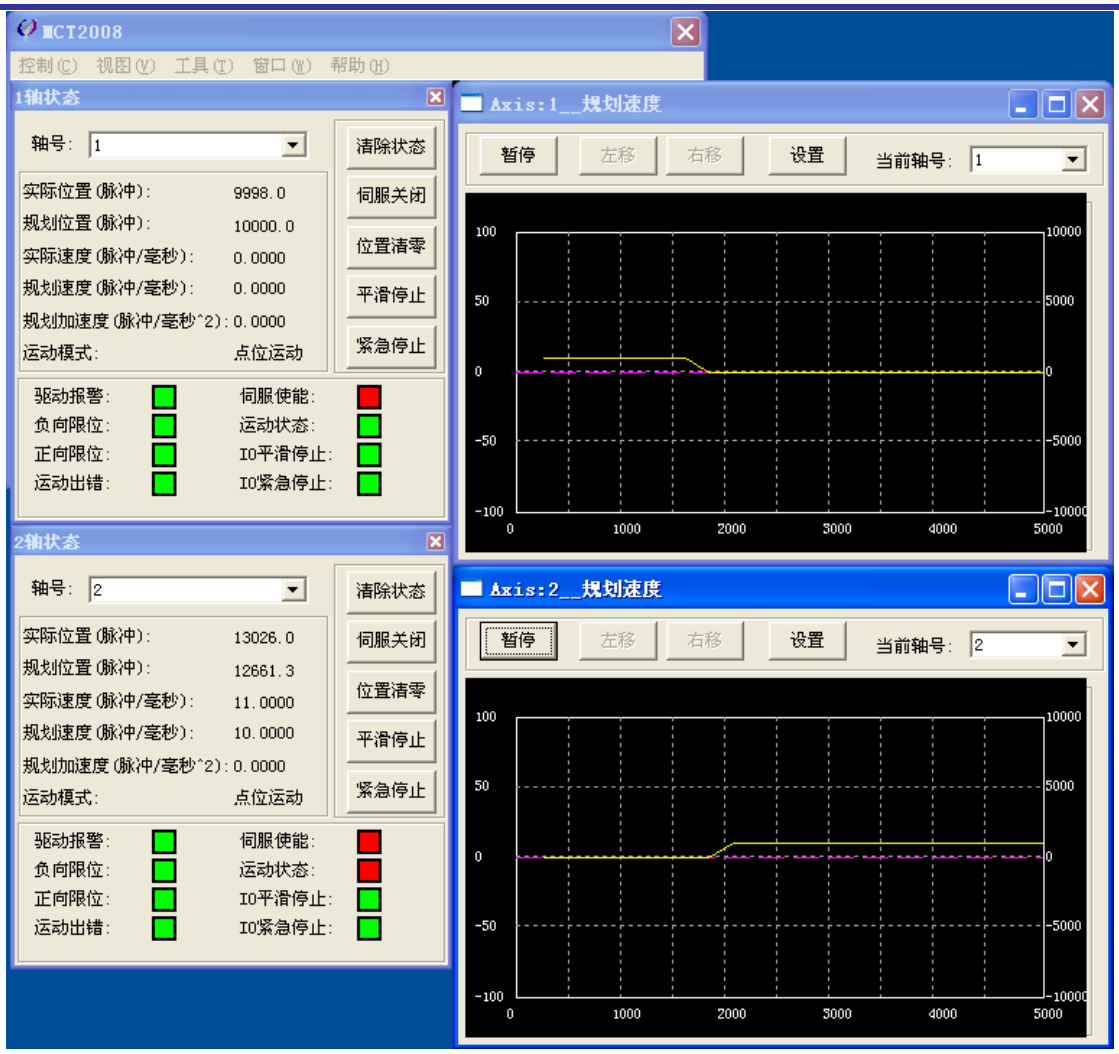

图 11-2 电机到位的运动状态检测

```
PROGRAM PLC_PRG
VAR
    xInitDone:BOOL:= FALSE;
    rtn:INT;
    xStart: BOOL:= FALSE;
    xStarted_X: BOOL:= FALSE;
    xStarted_Y: BOOL:= FALSE;
    pid:TPid;
    trap:TTrapPrm;
    posX:DINT:= 10000;
    PosY:DINT:= 20000;
    lrPrfPosX,lrPrfVelX:LREAL;
    lrPrfPosY,lrPrfVelY:LREAL;
    stsX,stsY:DWORD;
END_VAR
VAR CONSTANT
    AXIS_X:INT:= 1;
    AXIS_Y:INT:= 2;
END_VAR
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
(* @END_DECLARATION := '0' *)
IF NOT xInitDone THEN
    (*配置运动控制器为伺服模式*)
    rtn:= GT_LoadConfig('servo.cfg');
    (*清除各轴的报警和限位*)
    rtn:= GT_ClrSts(1,8);
    (*伺服使能*)
    rtn:= GT_AxisOn(AXIS_X);
    rtn:= GT_AxisOn(AXIS_Y);
    (*位置清零，并进行自动零漂补偿*)
    rtn:= GT_ZeroPos(AXIS_X,1);
    rtn:= GT_ZeroPos(AXIS_Y,1);
    (*设置X轴到位误差带*)
    rtn:= GT_SetAxisBand(AXIS_X, 20, 5);
    (*设置Y轴到位误差带*)
    rtn:= GT_SetAxisBand(AXIS_Y, 20, 5);
    xInitDone:= TRUE;
    xStarted_X:= FALSE;
    xStarted_Y:= FALSE;
END_IF
IF xStart THEN
    (*将X轴设为点位模式*)
    rtn:= GT_PrfTrap(AXIS_X);
    rtn:= GT_GetTrapPrm(AXIS_X, ADR(trap));
    (*设置 X 轴点位运动参数*)
    trap.acc := 1;
    trap.dec := 0.5;
    rtn:= GT_SetTrapPrm(AXIS_X, ADR(trap));
    (*设置X轴的目标速度*)
    rtn:= GT_SetVel(AXIS_X, 10);

    (*将Y轴设为点位模式*)
    rtn:= GT_PrfTrap(AXIS_Y);
    rtn:= GT_GetTrapPrm(AXIS_Y, ADR(trap));
    (*设置 Y 轴点位运动参数*)
    trap.acc := 1;
    trap.dec := 0.5;
    rtn:= GT_SetTrapPrm(AXIS_Y, ADR(trap));
    (*设置Y轴的目标速度*)
    rtn:= GT_SetVel(AXIS_Y, 10);

    (*设置 X 轴的目标位置*)
    rtn:= GT_SetPos(AXIS_X, posX);
    (*启动X轴的运动*)
    rtn:= GT_Update(SHL(WORD#1, AXIS_X -1));
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
        xStarted_X:= TRUE;
        xStarted_Y:= FALSE;
        xStart:= FALSE;
    END_IF

    (*等待X轴进入误差带*)
    IF xStarted_X AND stsX.11 AND NOT xStarted_Y THEN
        (*设置 Y 轴的目标位置*)
        rtn:= GT_SetPos(AXIS_Y, posY);
        (*启动Y轴的运动*)
        rtn:= GT_Update(SHL(WORD#1, AXIS_Y -1));
        xStarted_Y:= TRUE;
    END_IF

    (*读取X轴的状态*)
    rtn:= GT_GetSts(AXIS_X, ADR(stsX),1,0);
    (*读取X轴的规划位置*)
    rtn:= GT_GetPrfPos(AXIS_X, ADR(lrPrfPosX),1,0);
    (*读取X轴的规划速度*)
    rtn:= GT_GetPrfVel(AXIS_X, ADR(lrPrfVelX),1,0);
    (*读取Y轴的状态*)
    rtn:= GT_GetSts(AXIS_Y, ADR(stsY),1,0);
    (*读取Y轴的规划位置*)
    rtn:= GT_GetPrfPos(AXIS_Y, ADR(lrPrfPosY),1,0);
    (*读取Y轴的规划速度*)
    rtn:= GT_GetPrfVel(AXIS_Y, ADR(lrPrfVelY),1,0);
    END_PROGRAM
```

11.8 铁电功能

表 11-8 铁电功能指令列表

指令	说明	页码
GT_SetMRAMData	传输数据到控制器	142
GT_GetMRAMData	从控制器读取数据	122
GT_StartAccessMRAM	启动存储器读/写操作	147
GT_StopAccessMRAM	停止存储器读/写操作	148
GT_GetMRAMStatus	获取存储器读/写完成状态	122

iDEABOX 3 智能控制器包含铁电存储器芯片，将 ROM 的非易失性数据存储特性和 RAM 的无限次读写、高速读写以及低功耗等优势结合在一起，可用于实时的数据采集与记录。尤其是发生电源失效的情况下，及时记录系统状态和相关参数，便于下次上电时恢复系统状态或者确认一个系统错误。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

11.8.1 写数据至存储器

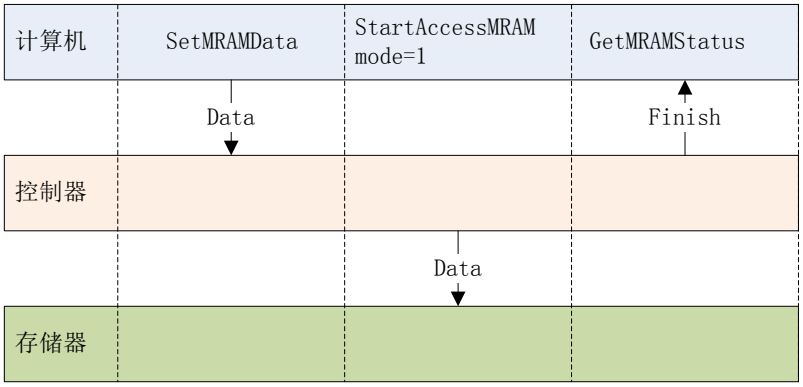

图 11-3 写数据流程示意图

写数据时，需首先调用 GT_SetMRAMData 指令将数据发送到控制器数据缓冲区；可以分段传递，参数 address 用于表示存放至数据缓冲区的起始序号，但每次最长传递 24words；然后调用 GT_StartAccessMRAM 指令，控制器将缓冲区中数据写入存储器，可以选择这次从数据缓冲区写入到存储器数据的起始序号与写入长度；可以调用 GT_GetMRAMStatus 指令获取控制器是否完成数据存储。

11.8.2 从存储器读数据

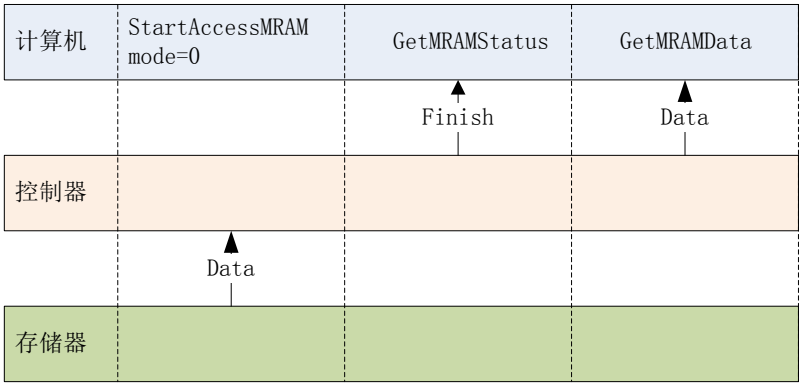

图 11-4 读数据流程示意图

读数据时，需首先调用 GT_StartAccessMRAM 指令，控制器将从存储器中读取数据至缓冲区，可以选择这次从存储器读取数据的起始序号与读取长度；然后调用 GT_GetMRAMStatus 指令获取控制器是否完成数据读取；数据读取完成后，可调用 GT_GetMRAMData 指令将缓冲区中数据传回计算机；可以分段传递，参数 address 用于表示缓冲区中数据的起始序号，但每次最长传递 24words。

11.8.3 例程

例程 11-3 写数据至存储器

```
PROGRAM PLC_PRG
VAR
    rtn: INT;
    xWriteMRAM: BOOL;
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

```
    WriteMRAMR: R_TRIG;
    WriteMramBuf: ARRAY [0..3, 0..15] OF INT := 12345;
    MramSts: INT;
END_VAR
(* @END_DECLARATION := '0' *)
WriteMRAMR(CLK:= xWriteMRAM);
IF xWriteMRAM THEN
    IF WriteMRAMR.Q THEN
        FOR i:= 0 TO 3 DO
            rtn:= GT_SetMRAMData(i*16,ADR(WriteMramBuf[i,0]),16);
        END_FOR
        rtn:= GT_StartAccessMRAM(0,64,1);     (* 一次性写入64个words的数据 *)
    END_IF
    rtn:= GT_GetMRAMStatus(ADR(MramSts));
    IF MramSts = 1 THEN       (* 写数据完成标志 *)
        xWriteMRAM:= FALSE;
    END_IF
END_IF
END_PROGRAM
```

例程 11-4 从存储器读数据

```
PROGRAM PLC_PRG
VAR
    rtn: INT;
    xReadMRAM: BOOL;
    ReadMRAMR: R_TRIG;
    ReadMramBuf: ARRAY [0..3, 0..15] OF INT;
    MramSts: INT;
END_VAR
(* @END_DECLARATION := '0' *)
ReadMRAMR(CLK:= xReadMRAM);
IF xReadMRAM THEN
    IF ReadMRAMR.Q THEN
        rtn:= GT_StartAccessMRAM(0,64,0);     (* 一次性读取64个words的数据 *)
    END_IF
    rtn:= GT_GetMRAMStatus(ADR(MramSts));
    IF MramSts = 1 THEN       (* 读数据完成标志 *)
        FOR i:= 0 TO 3 DO
            rtn:= GT_GetMRAMData(i*16,ADR(ReadMramBuf[i,0]),16);
        END_FOR
        xReadMRAM:= FALSE;
    END_IF
END_IF
END_PROGRAM
```


上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第12章 加密机制

1. 目前支持的两类加密形式

(1) 软件加密：即应用程序开发人员在 OtoStudio 环境下开发应用程序过程中，设置密码保护程序。具体分为两种：

1) 保护程序代码：在 OtoStudio->选项->密码：设置密码和保护密码。

2) 保护程序运行：例如，在 OtoStudio 程序开发中，在程序中给出一个密码比较语句，作为是否运行程序的条件。

(2) 硬件加密：固高欧辰可以提供两种硬件加密方式：

1) 在运行 GRT.exe 时候，自动检查硬件版本号，如果版本号不正确，则不能正常启动。（版本号由固高欧辰提供，同一系列的固高欧辰控制器，版本号是相同的）

2) 提供绑定网卡地址的函数 GetMacAddress()，应用程序开发人员可通过读取网卡地址来加密应用程序。（每套硬件平台都有唯一的网卡地址，且不可更改）

2. 关于回款加密

固高欧辰可以提供参考方案如下：

在应用程序中设置定时器，读取 CPU 的时钟，例如，希望客户在 3 个月内付款，否则应用程序无法工作。则可通过此种方式，先等待三个月的系统时钟，然后验证软件加密[2]。

表 12-1 加密函数指令列表

指令	说明	页码
GetMacAddress	读取网卡的物理地址	103

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第13章 指令详细说明

提示	以下表格中的“章节页码”即为此指令在章节中的位置。可以使用“超级链接”进行索引。“指令示例”即为与此指令相关的例程，可以使用“超级链接”进行索引。

指令 1 GetMacAddress

指令原型	short GetMacAddress(unsigned long ulAdapterNumber, unsigned char *pMacAddress, unsigned long ulAddressSize)		
指令说明	控制相应轴驱动报警信号无效。		
指令类型	立即指令，调用后立即生效。	章节页码	102
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
ulAdapterNumber	当前控制器网卡端口号，默认为 0。		
pMacAddress	输出网卡地址，为数组首地址。		
ulAddressSize	网卡数组长度，单位为 Byte。		
指令返回值	返回值为 0 表示成功，其他值表示失败。		
相关指令	无		
指令示例	无		

指令 2 GT_AlarmOff

指令原型	short GT_AlarmOff(short axis)		
指令说明	控制相应轴驱动报警信号无效。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
axis	控制轴号。		
指令返回值	若返回值为 1：请检查相应轴在配置文件中是否已经配置了报警无效。 其他返回值：请参照指令返回值列表。		
相关指令	GT_AlarmOn		
指令示例	无		

指令 3 GT_AlarmOn

指令原型	short GT_AlarmOn(short axis)		
指令说明	控制轴驱动报警信号有效。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
axis	控制轴号。		
指令返回值	若返回值为 1：请检查相应轴在配置文件中是否已经配置了报警无效。 其他返回值：请参照指令返回值列表。		
相关指令	GT_AlarmOff		
指令示例	无		

指令 4 GT_AxisOff

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令原型	short GT_AxisOff(short axis)		
指令说明	关闭驱动器使能。		
指令类型	立即指令，调用后立即生效。	章节页码	95
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
axis	关闭伺服使能的轴的编号。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_AxisOn		
指令示例	无		

指令 5 GT_AxisOn

指令原型	short GT_AxisOn(short axis)		
指令说明	打开驱动器使能。		
指令类型	立即指令，调用后立即生效。	章节页码	95
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
axis	打开伺服使能的轴的编号。		
指令返回值	若返回值为 1： (1) 若在配置文件中报警有效，请检查驱动器是否有报警。 (2) 若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 其他返回值：请参照指令返回值列表。		
相关指令	GT_AxisOff		
指令示例	**例程 7-1 点位运动**		

指令 6 GT_Bind

指令原型	short GT_Bind(short thread, short funId, short page)		
指令说明	绑定线程、函数、数据页。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
thread	线程编号，取值范围：[0, 31]。		
funId	函数标识，可以调用 GT_GetFunId 查询。		
page	数据页编号，取值范围：[0, 31]。		
指令返回值	若返回值为 1：请检查绑定的线程号是否已经有线程绑定并正在运行。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 10-1 运动程序单线程累加求和**		

指令 7 GT_Close

指令原型	short GT_Close()		
指令说明	关闭运动控制器。		
指令类型	立即指令，调用后立即生效。	章节页码	
指令参数	无		
指令返回值	请参照指令返回值列表。		
相关指令	GT_Open		
指令示例	无		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令 8 GT_ClrSts

指令原型	short GT_ClrSts(short axis, short count=1)		
指令说明	清除驱动器报警标志、跟随误差越限标志、限位触发标志。 1. 只有当驱动器没有报警时才能清除轴状态字的报警标志； 2. 只有当跟随误差正常以后，才能清除跟随误差越限标志； 3. 只有当离开限位开关，或者规划位置在软限位行程以内时才能清除轴状态字的限位触发标志。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
axis	起始轴号，正整数。		
count	读取的轴数，默认为 1，正整数。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_GetSts		
指令示例	**例程 7-1 点位运动**		

指令 9 GT_Compile

指令原型	short GT_Compile(char *pFileName, TCompileInfo *pWrongInfo)		
指令说明	编译运动程序。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
pFileName	要编译的运动程序文件名。		
pWrongInfo	typedef struct CompileInfo { char *pFileName; 编译文件名 short *pLineNo; 错误行号 char *pMessage; 错误信息 } TCompileInfo;		
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 10 GT_Download

指令原型	short GT_Download(char *pFileName)		
指令说明	下载运动程序到运动控制器。注：文件包含的线程数不能大于 32。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
pFileName	下载到运动控制器的运动程序文件名。		
指令返回值	若返回值为 2007： (1) 请检查文件名是否太长。 (2) 请检查需要下载的文件是否存在。 若返回值为 2008：表示打开文件失败。 (1) 请检查文件是否损坏，编译是否成功。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	(2) 请检查 ini 和 bin 文件是否在同一根目录。 若返回值为 7：请检查线程数量是否大于 32。 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	**例程 10-1 运动程序单线程累加求和**

指令 11 GT_EcatIOReadInput

指令原型	short GT_EcatIOReadInput(unsigned short slave, unsigned short offset, unsigned short nSize, unsigned char *pValue)		
指令说明	读取EtherCAT IO模块数字量输入。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
slave	从站号，包括伺服模块的排序。		
offset	IO 的偏移位置，指从站内的 IO 地址偏移，越界将返回 7 参数错误。		
nSize	读取的字节数，不能超越最大的数据长度，越界将返回 7 参数错误。		
pValue	读取的数据指针。		
指令返回值	若返回值为 1：请检查相应从站是否已正确连接。 其他返回值：请参照指令返回值列表。		
相关指令	GT_EcatIOWriteOutput		
指令示例	**例程 5-4 EtherCAT IO 的**		

指令 12 GT_EcatIOWriteOutput

指令原型	short GT_EcatIOWriteOutput(unsigned short slave, unsigned short offset, unsigned short nSize, unsigned char *pValue)		
指令说明	写入EtherCAT IO模块数字量输入。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
slave	从站号，包括伺服模块的排序。		
offset	IO 的偏移位置，指从站内的 IO 地址偏移，越界将返回 7 参数错误。		
nSize	写入的字节数，不能超越最大的数据长度，越界将返回 7 参数错误。		
pValue	写入的数据指针。		
指令返回值	若返回值为 1：请检查相应从站是否已正确连接。 其他返回值：请参照指令返回值列表。		
相关指令	GT_EcatIOReadInput		
指令示例	**例程 5-4 EtherCAT IO 的**		

指令 13 GT_EcatSDODownload

指令原型	short GT_EcatSDODownload(unsigned short slave_position, unsigned short index, unsigned char subindex, unsigned char *pData, unsigned long data_size, unsigned long *pAbort_code)		
指令说明	通用 SDO 下载（Service Data Object，参考 IEC 61800）。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 6 个参数，参数的详细信息如下。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

slave_position	从站号。
index	SDO 的 Index (参考 IEC 61800)。
subindex	SDO 的 Subindex (参考 IEC 61800)。
pData	下载的数据指针。
data_size	下载的数据量，按 Byte 计算。
pAbort_code	异常退出代码。
指令返回值	请参照指令返回值列表。
相关指令	GT_EcatSDOUpLoad
指令示例	无

指令 14 GT_EcatSDOUpLoad

指令原型	short GT_EcatSDOUpload(unsigned short slave_position, unsigned short index, unsigned char subindex, unsigned char *pTarget, unsigned long target_size, unsigned long *pResult_size, unsigned long *pAbort_code)		
指令说明	通用 SDO 上传。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 7 个参数，参数的详细信息如下。		
slave_position	从站号。		
index	SDO 的 Index (参考 IEC 61800)。		
subindex	SDO 的 Subindex (参考 IEC 61800)。		
pTarget	上传的数据指针。		
target_size	目标上传的数据量，按 Byte 计算。		
pResult_size	实际上传的数据量。		
pAbort_code	异常退出代码。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_EcatSDODownload		
指令示例	无		

指令 15 GT_EncOff

指令原型	short GT_EncOff(short encoder)		
指令说明	设置为“脉冲计数器”计数方式。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
encoder	编码器通道号。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_EncOn		
指令示例	无		

指令 16 GT_EncOn

指令原型	short GT_EncOn(short encoder)		
指令说明	设置为“外部编码器”计数方式。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 1 个参数，参数的详细信息如下。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

encoder	编码器通道号。
指令返回值	请参照指令返回值列表。
相关指令	GT_EncOff
指令示例	无

指令 17 GT_EncScale

指令原型	short GT_EncScale(short axis, short alpha, short beta)		
指令说明	设置控制轴的编码器当量变换值。		
25	立即指令，调用后立即生效。	**章节页码**	25
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
axis	控制轴号。		
alpha	规划器当量的alpha值，取值范围：(-32767, 0)和(0, 32767)。		
beta	规划器当量的beta值，取值范围：(-32767, 0)和(0, 32767)。		
指令返回值	若返回值为 1：若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 其他返回值：请参照指令返回值列表。		
相关指令	GT_ProfileScale		
指令示例	无		

指令 18 GT_EncSns

指令原型	short GT_EncSns(unsigned short sense)		
指令说明	设置编码器的计数方向。		
指令类型	立即指令，调用后立即生效。	**章节页码**	25
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
sense	按位标识编码器的计数方向。 bit0~bit11依次对应编码器1~12。 0：该编码器计数方向不取反。 1：该编码器计数方向取反。 系统复位后，默认为编码器计数方向取反。		
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 4-1 设置编码器计数方向**		

指令 19 GT_FollowClear

指令原型	short GT_FollowClear(short profile, short fifo=0)		
指令说明	清除 Follow 运动模式指定 FIFO 中的数据。 运动状态下该指令无效。		
指令类型	立即指令，调用后立即生效。	**章节页码**	61
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
fifo	指定需要清除的 FIFO，取值范围：0、1 两个值。默认为 0。		
指令返回值	若返回值为 1： (1) 请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	(2) 请检查要清除的 FIFO 是否正在使用，运动是否结束。 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	**例程 7-7 Follow 单 FIFO**

指令 20 GT_FollowData

指令原型	short GT_FollowData(short profile, long masterSegment, double slaveSegment, short type= FOLLOW_SEGMENT_NORMAL, short fifo=0)		
指令说明	向 Follow 运动模式指定 FIFO 增加数据。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 5 个参数，参数的详细信息如下。		
profile	规划轴号。		
masterSegment	主轴位移，单位：pulse。		
slaveSegment	从轴位移，单位：pulse。		
type	数据段类型： FOLLOW_SEGMENT_NORMAL（该宏定义为 0）普通段。默认为该类型。 FOLLOW_SEGMENT_EVEN（该宏定义为 1）匀速段。 FOLLOW_SEGMENT_STOP（该宏定义为 2）减速到 0 段。 FOLLOW_SEGMENT_CONTINUE（该宏定义为 3）保持 FIFO 之间速度连续。		
fifo	指定存放数据的 FIFO，取值范围：0、1 两个值。默认为 0。		
指令返回值	若返回值为 1： (1) 请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。 (2) 请检查是否有足够的空间放新的数据。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 7-7 Follow 单 FIFO**		

指令 21 GT_FollowSpace

指令原型	short GT_FollowSpace(short profile, short *pSpace, short fifo=0)		
指令说明	查询 Follow 运动模式指定 FIFO 的剩余空间。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
profile	规划轴号。		
pSpace	读取 FIFO 的剩余空间。 说明此空间的含义。		
fifo	指定所要查询的 FIFO，取值范围：0、1 两个值。默认为 0。		
指令返回值	若返回值为 1：请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	无		

指令 22 GT_FollowStart

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令原型	short GT_FollowStart(long mask, long option)
指令说明	启动 Follow 运动。
指令类型	立即指令，调用后立即生效。 章节页码 61
指令参数	该指令共有 2 个参数，参数的详细信息如下。
mask	按位指示需要启动 Follow 运动的轴号。当 bit 位为 1 时表示启动对应的轴。

Bit	11	10	9	8	7	6	5	4	3	2	1	0
对应轴	12 轴	11 轴	10 轴	9 轴	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴

option：按位指示所使用的 FIFO，默认为 0。当 bit 位为 0 时表示对应的轴使用 FIFO1。当 bit 位为 1 时表示对应的轴使用 FIFO2。

Bit	11	10	9	8	7	6	5	4	3	2	1	0
对应轴	12 轴	11 轴	10 轴	9 轴	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴

指令返回值：

若返回值为 1：

(1) 请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。

(2) 检查运动是否结束，运动进行时，指令调用会失败；

(3) 检查相应轴是否设置了跟随主轴；

(4) 检查 FIFO 是否有数据；

(5) 检查 mask 参数是否设置了启动相应的轴。

其他返回值：请参照指令返回值列表。

相关指令	无
指令示例	**例程 7-7 Follow 单 FIFO**

指令 23 GT_FollowSwitch

指令原型	short GT_FollowSwitch(long mask)
指令说明	切换 Follow 运动模式所使用的 FIFO。
指令类型	立即指令，调用后立即生效。 章节页码 61
指令参数	该指令共有 1 个参数，参数的详细信息如下。
mask	按位指示需要切换 Follow 工作 FIFO 的轴号。当 bit 位为 1 时表示切换对应的轴的 FIFO。

Bit	11	10	9	8	7	6	5	4	3	2	1	0
对应轴	12 轴	11 轴	10 轴	9 轴	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴

指令返回值：

若返回值为 1：

(1) 请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。

(2) 检查运动是否进行，只有运动中才能切换。

(3) 检查目标 FIFO 是否为空。

(4) 检查 mask 参数是否设置了启动相应的轴。

其他返回值：请参照指令返回值列表。

相关指令	无
指令示例	**例程 7-8 Follow 双 FIFO 切换**

指令 24 GT_GearStart

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令原型	short GT_GearStart(long mask)
指令说明	启动电子齿轮运动。
指令类型	立即指令，调用后立即生效。 章节页码 57
指令参数	该指令共有 1 个参数，参数的详细信息如下。
mask	按位指示需要启动 Gear 运动的轴号。当 bit 位为 1 时表示启动对应的轴。

Bit	11	10	9	8	7	6	5	4	3	2	1	0
对应轴	12 轴	11 轴	10 轴	9 轴	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴

指令返回值	若返回值为 1： (1) 请检查当前轴是否为电子齿轮模式，若不是，请先调用 GT_PrfGear 将当前轴设置为电子齿轮模式。 (2) 请检查主轴是否已设置。 (3) 请检查传动比是否已设置。 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	**例程 7-5 电子齿轮跟随**

指令 25 GT_GetAxisBand

指令原型	short GT_GetAxisBand(short axis, long *pBand, long *pTime)
指令说明	读取轴到位误差带。
指令类型	立即指令，调用后立即生效。 章节页码 95
指令参数	该指令共有 3 个参数，参数的详细信息如下。
axis	轴号。
pBand	读取误差带大小。
pTime	读取误差带保持时间。
指令返回值	请参照指令返回值列表。
相关指令	GT_SetAxisBand
指令示例	无

指令 26 GT_GetAxisEncAcc

指令原型	short GT_GetAxisEncAcc(short axis, double *pValue, short count=1, unsigned long *pClock=NULL)
指令说明	读取 encoder 输出值经过当量变换之后的编码器加速度值。
指令类型	立即指令，调用后立即生效。 章节页码 38
指令参数	该指令共有 4 个参数，参数的详细信息如下。
axis	起始轴号，正整数。
pValue	轴的编码器加速度。单位：pulse/ms^2。
count	读取的轴数，默认为 1，正整数。
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。
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 27 GT_GetAxisEncPos

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令原型	short GT_GetAxisEncPos(short axis, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取 encoder 输出值经过当量变换之后的编码器位置值。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
axis	起始轴号，正整数。		
pValue	轴的编码器位置。单位：pulse。		
count	读取的轴数，默认为 1，正整数。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
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 28 GT_GetAxisEncVel

指令原型	short GT_GetAxisEncVel(short axis, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取 encoder 输出值经过当量变换之后的编码器速度值。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
axis	起始轴号，正整数。		
pValue	轴的编码器速度。单位：pulse/ms。		
count	读取的轴数，默认为 1，正整数。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
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 29 GT_GetAxisError

指令原型	short GT_GetAxisError(short axis, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取 profile 经过当量变换之后的规划位置与 encoder 经过当量变换之后的编码器位置的差值。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
axis	起始轴号，正整数。		
pValue	轴的规划位置与编码器位置的差值。单位：pulse。		
count	读取的轴数，默认为 1，正整数。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
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 30 GT_GetAxisPrfAcc

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令原型	short GT_GetAxisPrfAcc(short axis, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取 profile 输出值经过当量变换之后的规划加速度。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
axis	起始轴号，正整数。		
pValue	轴的规划加速度。单位：pulse/ms^2。		
count	读取的轴数，默认为 1，正整数。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
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 31 GT_GetAxisPrfPos

指令原型	short GT_GetAxisPrfPos(short axis, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取 profile 输出值经过当量变换之后的规划位置。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
axis	起始轴号，正整数。		
pValue	轴的规划位置。单位：pulse。		
count	读取的轴数，默认为 1，正整数。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
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 32 GT_GetAxisPrfVel

指令原型	short GT_GetAxisPrfVel(short axis, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取 profile 输出值经过当量变换之后的规划速度。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
axis	起始轴号，正整数。		
pValue	轴的规划速度。单位：pulse/ms。		
count	读取的轴数，默认为 1，正整数。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
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 33 GT_GetClock

指令原型	short GT_GetClock(unsigned long *pClock, unsigned long *pLoop=NULL)

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令说明	读取运动控制器系统时钟。		
指令类型	立即指令，调用后立即生效。	章节页码	95
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
pClock	读取的运动控制器的时钟，单位：ms。		
pLoop	内部使用，默认值为：NULL，即不读取该值。		
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 34 GT_GetEcatAxisAI

指令原型	short GT_GetEcatAxisAI(short axis, short channel, short *pValue)		
指令说明	读取EtherCAT轴模拟量输入。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
axis	轴号，取值范围：[1, 12]。		
channel	模拟量通道号，取值范围：[1, 2]。		
pValue	模拟量输入数值。		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，20F2h、20F9h）配置为 PDO。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	无		

指令 35 GT_GetEcatAxisAtlCurrent

指令原型	short GT_GetEcatAxisAtlCurrent(short axis, short * pCur);		
指令说明	读取 EtherCAT 轴的电流值。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
pCur	伺服电机电流值。		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，6078h）配置为 PDO。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	无		

指令 36 GT_GetEcatAxisAtlTorque

指令原型	short GT_GetEcatAxisAtlTorque(short axis, short * pTorque);		
指令说明	读取 EtherCAT 轴的力矩值。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
pTorque	伺服电机力矩值。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，6077h）配置为 PDO。 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	无

指令 37 GT_GetEcatAxisDI

指令原型	short GT_GetEcatAxisDI(short axis,unsigned long *pDi)		
指令说明	读取EtherCAT轴的数字量输入值。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
pDi	数字 IO 输入状态，按位指示 IO 输入电平。		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，60FDh）配置为 PDO。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	无		

指令 38 GT_GetEcatAxisDO

指令原型	short GT_GetEcatAxisDO(short axis,unsigned long *pDo)		
指令说明	读取EtherCAT轴的数字量输出值。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
pDo	数字 IO 输出状态，按位指示 IO 输出电平。		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，60FEh）配置为 PDO。 其他返回值：请参照指令返回值列表。		
相关指令	GT_SetEcatAxisDO		
指令示例	无		

指令 39 GT_GetEcatAxisMode

指令原型	short GT_GetEcatAxisMode(short axis, short *pMode);		
指令说明	读取 EtheCAT 轴的操作模式		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
pMode	伺服操作模式（以 GTHD 为例）： 0：No mode assigned 1：Profile position 3：Profile velocity 4：Profile torque		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	6：Homing 7：Interpolated position 8：Cyclic sync position 9：Cyclic sync velocity 10：Cyclic sync torque
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，6061h）配置为 PDO。 其他返回值：请参照指令返回值列表。
相关指令	GT_SetEcatAxisMode
指令示例	无

指令 40 GT_GetEcatAxisPE

指令原型	short GT_GetEcatAxisPE(short axis,long *pPosErr)		
指令说明	读取EtherCAT轴的位置误差。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
pPosErr	规划和编码器的位置差值。		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，60F4h）配置为 PDO。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	无		

指令 41 GT_GetEcatEncPos

指令原型	short GT_GetEcatEncPos(short axis, long *pEncPos)		
指令说明	读取 EtherCAT 轴的编码器位置。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
pEncPos	轴编码器实际位置。		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，6064h）配置为 PDO。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	无		

指令 42 GT_GetEcatEncVel

指令原型	short GT_GetEcatEncVel(short axis, long *pEncVel);		
指令说明	读取 EtherCAT 轴的编码器速度。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

pEncVel	编码器实际速度。
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，606Ch）配置为 PDO。 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	无

指令 43 GT_GetEcatHomingStatus

指令原型	short GT_GetEcatHomingStatus(short axis, short *pHomingStatus)		
指令说明	查询EtherCAT轴的回零状态。		
指令类型	立即指令，调用后立即生效。	**章节页码**	29
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，取值范围：[1, 12]。		
pHomeStatus	回零过程的状态值：		

BIT	状态
0	0：正在回零 1：回零完成
1	0：无意义 1：回零成功完成
2	0：无意义 1：回零过程出错

指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，6041h）配置为 PDO。 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	**例程 5-2 采用 3 号回零方式**

指令 44 GT_GetEcatRawData

指令原型	short GT_GetEcatRawData(unsigned short offset, unsigned short size, unsigned char *pValue)		
指令说明	获取控制器 PDO 数据。		
指令类型	立即指令，调用后立即生效。	**章节页码**	29
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
offset	读取 PDO 数据偏移值，取值范围：0 ～ 当前 PDO 长度，单位：字节。		
size	读取数据长度，单位：字节。		
pValue	读取数据缓冲区地址		
指令返回值	若返回值为 1：请检查 EtherCAT 通讯是否正常。 其它返回值：请参照指令返回值列表。		
相关指令	GT_SetEcatRawData		
指令示例	无		

指令 45 GT_GetEcatSlaves

指令原型	short GT_GetEcatSlaves(short *pSlaveCnt)		
指令说明	读取EtherCAT总线的从站数目。		
指令类型	立即指令，调用后立即生效。	**章节页码**	29
指令参数	该指令共有 1 个参数，参数的详细信息如下。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

pSlaveCnt	在线的 EtherCAT 从站数目。
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 46 GT_GetEncPos

指令原型	short GT_GetEncPos(short encoder, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取编码器位置。		
指令类型	立即指令，调用后立即生效。	章节页码	76
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
encoder	编码器起始轴号，正整数，取值范围：[1, 12]。		
pValue	编码器位置。		
count	读取的轴数。默认为 1。 1 次最多可以读取 8 个编码器轴。		
pClock	读取控制器时钟。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_SetEncPos		
指令示例	**例程 8-1 访问编码器**		

指令 47 GT_GetEncVel

指令原型	short GT_GetEncVel(short encoder, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取编码器速度。		
指令类型	立即指令，调用后立即生效。	章节页码	76
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
encoder	编码器起始轴号，正整数，取值范围：[1, 12]。		
pValue	编码器速度。		
count	读取的轴数。默认为 1。 1 次最多可以读取 8 个编码器轴。		
pClock	读取控制器时钟。		
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 8-1 访问编码器**		

指令 48 GT_GetFollowEvent

指令原型	short GT_GetFollowEvent(short profile, short *pEvent, short *pMasterDir, long *pPos)		
指令说明	读取 Follow 运动模式启动跟随条件。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	规划轴号。		
pEvent	读取启动跟随条件： FOLLOW_EVENT_START（该宏定义为 1）表示调用 GT_FollowStart 以后立即启动。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	FOLLOW_EVENT_PASS（该宏定义为 2）表示主轴穿越设定位置以后启动跟随。
pMasterDir	读取主轴运动方向： 1 主轴正向运动，-1 主轴负向运动。
pPos	读取穿越位置，单位：pulse。 当 event 为 FOLLOW_EVENT_PASS 时有效。
指令返回值	若返回值为 1：请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。 其他返回值：请参照指令返回值列表。
相关指令	GT_SetFollowEvent
指令示例	无

指令 49 GT_GetFollowLoop

指令原型	short GT_GetFollowLoop(short profile, long *pLoop)		
指令说明	读取 Follow 运动模式循环已经执行完成的次数。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
pLoop	读取 Follow 模式循环已经执行完成的次数。		
指令返回值	若返回值为 1：请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_SetFollowLoop		
指令示例	无		

指令 50 GT_GetFollowMaster

指令原型	short GT_GetFollowMaster(short profile, short *pMasterIndex, short *pMasterType, short *pMasterItem)		
指令说明	读取 Follow 运动模式跟随主轴。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	规划轴号。		
pMasterIndex	读取主轴索引。 主轴索引不能与规划轴号相同，最好主轴索引号小于规划轴号，如主轴索引为 1 轴，规划轴号为 2 轴。		
pMasterType	读取主轴类型。 FOLLOW_MASTER_PROFILE（该宏定义为 2）表示跟随规划轴(profile)的输出值，默认为此类型。 FOLLOW_MASTER_ENCODER（该宏定义为 1）表示跟随编码器(encoder)的输出值。 FOLLOW_MASTER_AXIS（该宏定义为 3）表示跟随轴(axis)的输出值。		
pMasterItem	合成轴类型，当 masterType= FOLLOW_MASTER_AXIS 时起作用。 0 表示 axis 的规划位置输出值，默认为该值。 1 表示 axis 的编码器位置输出值。		
指令返回值	若返回值为 1：请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	当前轴设置为 Follow 模式。 其他返回值：请参照指令返回值列表。
相关指令	GT_SetFollowMaster
指令示例	无

指令 51 GT_GetFollowMemory

指令原型	short GT_GetFollowMemory(short profile, short *pMemory)		
指令说明	读取 Follow 运动模式的缓存区大小。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号，正整数。取值范围：[1, 12]。		
pMemory	读取 Follow 运动缓存区大小标志。 0：每个 Follow 运动缓存区有 16 段空间。 1：每个 Follow 运动缓存区有 512 段空间。		
指令返回值	若返回值为 1：请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_SetFollowMemory		
指令示例	无		

指令 52 GT_GetFunId

指令原型	short GT_GetFunId(char *pFunName, short *pFunId)		
指令说明	读取运动程序中函数的标识。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
pFunName	运动程序函数名称。		
pFunId	根据运动程序函数名称查询函数标识。		
指令返回值	若返回值为 1，2007 或者 2008： 请检查重新检查 GT_Download 是否调用成功。 若失败，请根据 GT_Download 返回值提示操作，直至成功。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 10-1 运动程序单线程累加求和**		

指令 53 GT_GetGearMaster

指令原型	short GT_GetGearMaster(short profile, short *pMasterIndex, short *pMasterType, short *pMasterItem)		
指令说明	读取电子齿轮运动跟随主轴。		
指令类型	立即指令，调用后立即生效。	章节页码	57
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	规划轴号。		
pMasterIndex	读取主轴索引。		
pMasterType	读取主轴类型：		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	GEAR_MASTER_PROFILE（该宏定义为）表示跟随规划轴(profile)的输出值。 GEAR_MASTER_ENCODER（该宏定义为）表示跟随编码器(encoder)的输出值。 GEAR_MASTER_AXIS（该宏定义为）表示跟随轴(axis)的输出值。
pMasterItem	读取输出位置类型。当 masterType=GEAR_MASTER_AXIS 时起作用。 0 表示 axis 的规划位置输出值，默认为该值。 1 表示 axis 的编码器位置输出值。
指令返回值	若返回值为 1：请检查当前轴是否为电子齿轮模式，若不是，请先调用 GT_PrfGear 将当前轴设置为电子齿轮模式。 其他返回值：请参照指令返回值列表。
相关指令	GT_SetGearMaster
指令示例	无

指令 54 GT_GetGearRatio

指令原型	short GT_GetGearRatio(short profile, long *pMasterEven, long *pSlaveEven, long *pMasterSlope)		
指令说明	读取电子齿轮比。		
指令类型	立即指令，调用后立即生效。	**章节页码**	57
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	规划轴号。		
pMasterEven	读取传动比系数，主轴位移。单位：pulse。		
pSlaveEven	读取传动比系数，从轴位移。单位：pulse。		
pMasterSlope	读取主轴离合区位移。单位：pulse。		
指令返回值	若返回值为 1：请检查当前轴是否为电子齿轮模式，若不是，请先调用 GT_PrfGear 将当前轴设置为电子齿轮模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_SetGearRatio		
指令示例	无		

指令 55 GT_GetJogPrm

指令原型	short GT_GetJogPrm(short profile, TJogPrm *pPrm)		
指令说明	读取 Jog 运动模式下的运动参数。		
指令类型	立即指令，调用后立即生效。	**章节页码**	46
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
pPrm	设置 Jog 模式运动参数。该参数为一个**结构体**，包含三个参数，详细的参数定义及说明请参照 GT_SetJogPrm 指令说明。		
指令返回值	若返回值为 1： (1) 若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 (2) 请检查当前轴是否为 Jog 模式，若不是，请先调用 GT_PrfJog 将当前轴设置为 Jog 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_SetJogPrm		
指令示例	**例程 7-2 Jog 运动**		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令 56 GT_GetMcEcatAxisNum

指令原型	short GT_GetMcEcatAxisNum(short *pNum)		
指令说明	读取 EtherCAT 伺服轴数。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
pPrm	EtherCAT 伺服轴数。		
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 57 GT_GetMRAMData

指令原型	short GT_GetMRAMData(short address, short *pData, short count)		
指令说明	从控制器读取数据。		
指令类型	立即指令，调用后立即生效。	章节页码	99
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
address	控制器缓冲区数据起始地址，取值范围：[0, 4096)。		
pData	待读取数据缓冲区指针。		
count	数据长度，取值范围：[1, 24]。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_SetMRAMData		
指令示例	**例程 11-4 从存储器读数据**		

指令 58 GT_GetMRAMStatus

指令原型	short GT_GetMRAMStatus(short *pSts)		
指令说明	获取存储器读/写完成状态。		
指令类型	立即指令，调用后立即生效。	章节页码	99
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
pSts	状态数据指针，0 未完成；1 完成。		
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 11-3 写数据至存储器、例程 11-4 从存储器读数据**		

指令 59 GT_GetPos

指令原型	short GT_GetPos(short profile, long *pPos)		
指令说明	读取目标位置。		
指令类型	立即指令，调用后立即生效。	章节页码	43
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
pos	读取目标位置，单位：pulse。		
指令返回值	若返回值为 1： 请检查当前轴是否为 Trap 模式，若不是，请先调用 GT_PrfTrap 将当前轴设置为 Trap 模式。 其他返回值：请参照指令返回值列表。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

相关指令	GT_SetPos
指令示例	**例程 7-1 点位运动**

指令 60 GT_GetPosErr

指令原型	short GT_GetPosErr(short control, long *pError)		
指令说明	读取跟随误差极限值。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
control	伺服控制器编号。		
pError	读取的跟随误差极限值。单位：pulse。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_SetPosErr		
指令示例	无		

指令 61 GT_GetPosScale

指令原型	short GT_GetPosScale(short axis, unsigned short *pScale);		
指令说明	读取编码器倍率。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
pScale	编码器倍率，2 的幂运算的指数值。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_SetPosScale		
指令示例	无		

指令 62 GT_GetPrfAcc

指令原型	short GT_GetPrfAcc(short profile, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取规划加速度。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	起始规划轴号。		
pValue	规划加速度。单位：pulse/ms^2。		
count	读取的轴数，默认为 1。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
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 6-1 获取轴 1 的轴状态、运动模式、位置、速度和加速度**		

指令 63 GT_GetPrfMode

指令原型	short GT_GetPrfMode(short profile, long *pValue, short count=1, unsigned long *pClock=NULL)
指令说明	读取轴运动模式。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	起始规划轴号。		
pValue	轴运动模式。 0：点位运动，控制器上电后默认为该模式； 1：Jog 模式； 2：PT 模式； 3：电子齿轮模式； 4：Follow 模式； 5：插补模式； 6：Pvt 模式。		
count	读取的轴数，默认为 1。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
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 6-1 获取轴 1 的轴状态、运动模式、位置、速度和加速度**		

指令 64 GT_GetPrfPos

指令原型	short GT_GetPrfPos(short profile, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取规划位置。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	起始规划轴号。		
pValue	规划位置。单位：pulse。		
count	读取的轴数，默认为 1。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_SetPrfPos		
指令示例	**例程 6-1 获取轴 1 的轴状态、运动模式、位置、速度和加速度**		

指令 65 GT_GetPrfVel

指令原型	short GT_GetPrfVel(short profile, double *pValue, short count=1, unsigned long *pClock=NULL)		
指令说明	读取规划速度。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	起始规划轴号。		
pValue	规划速度。单位：pulse/ms。		
count	读取的轴数，默认为 1。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
指令返回值	请参照指令返回值列表。		
相关指令	无		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令示例	例程 6-1 获取轴 1 的轴状态、运动模式、位置、速度和加速度

指令 66 GT_GetPtLoop

指令原型	short GT_GetPtLoop(short profile, long *pLoop)		
指令说明	查询 PT 运动模式循环执行的次数。动态模式下该指令无效。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
pLloop	查询 PT 模式循环已经执行完成的次数。动态模式下该参数无效。		
指令返回值	若返回值为 1：请检查当前轴是否为 PT 模式，若不是，请先调用 GT_PrfPt 将当前轴设置为 PT 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_SetPtLoop		
指令示例	无		

指令 67 GT_GetPtMemory

指令原型	short GT_GetPtMemory(short profile, short *pMemory)		
指令说明	读取 PT 运动模式的缓存区大小。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
pMemory	读取 PT 运动缓存区大小标志。		
指令返回值	若返回值为 1：请检查当前轴是否为 PT 模式，若不是，请先调用 GT_PrfPt 将当前轴设置为 PT 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_SetPtMemory		
指令示例	无		

指令 68 GT_GetPtRemainder

指令原型	short GT_GetPtRemainder(short profile, long *pRemainder)		
指令说明	读取 PT 运动模式的缓存区中待执行数据段数。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
pRemainder	读取 PT 运动缓存区内待执行数据段数。		
指令返回值	若返回值为 1：请检查当前轴是否为 PT 模式，若不是，请先调用 GT_PrfPt 将当前轴设置为 PT 模式。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	无		

指令 69 GT_GetSoftLimit

指令原型	short GT_GetSoftLimit(short axis, long *pPositive, long *pNegative)

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令说明	读取轴正向软限位和负向软限位。		
指令类型	立即指令，调用后立即生效。	章节页码	78
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
axis	轴号。		
pPositive	读取正向软限位。		
pNegative	读取负向软限位。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_SetSoftLimit		
指令示例	无		

指令 70 GT_GetStopDec

指令原型	short GT_GetStopDec(short profile, double *pDecSmoothStop, double *pDecAbruptStop)		
指令说明	读取平滑停止减速度和急停减速度。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划器的编号。		
pDecSmoothStop	读取的平滑停止减速度。单位：pulse/ms^2。		
pDecAbruptStop	读取的急停减速度。单位：pulse/ms^2。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_SetStopDec		
指令示例	无		

指令 71 GT_GetSts

指令原型	short GT_GetSts(short axis, long *pSts, short count=1, unsigned long *pClock=NULL)		
指令说明	读取轴状态。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
axis	起始轴号。		
pSts	32 位轴状态字，详细定义参见”6.3.1”。		
count	读取的轴数，默认为 1。		
pClock	读取控制器时钟，默认值为：NULL，即不用读取控制器时钟。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_ClrSts		
指令示例	**例程 7-1 点位运动**		

指令 72 GT_GetThreadSts

指令原型	short GT_GetThreadSts(short thread, TThreadSts *pThreadSts)		
指令说明	读取线程的状态。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
thread	线程编号。取值范围：[0, 31]。		
pThreadSts	读取的线程状态。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	typedef struct ThreadSts
	{
	short run; // 运行状态
	short error; // 当前执行的指令返回值
	double result; // 函数返回值
	short line; // 当前执行的指令行号
	} TThreadSts;

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 10-1 运动程序单线程累加求和**

指令 73 GT_GetTouchProbeStatus

指令原型	short GT_GetTouchProbeStatus(short axis, unsigned short *probeStatus, long *probe1RiseValue, long *probe1FallValue, long *probe2RiseValue, long *probe2FallValue)		
指令说明	查询EtherCAT轴的探针状态和捕获值。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 6 个参数，参数的详细信息如下。		
axis	轴号，取值范围：[1, 12]。		
probeStatus	探针状态：		

BIT	描述
0	Probe1 开关: 0 OFF; 1 Enable
1	Probe1 上升沿: 0 未捕获; 1 捕获
2	Probe1 下降沿: 0 未捕获; 1 捕获
3-7	保留
8	Probe2 开关: 0 OFF; 1 Enable
9	Probe2 上升沿: 0 未捕获; 1 捕获
10	Probe2 下降沿: 0 未捕获; 1 捕获
11-15	保留

probe1RiseValue	探针 1 的上升沿捕获值。
probe1FallValue	探针 1 的下降沿捕获值。
probe2RiseValue	探针 2 的上升沿捕获值。
probe2FallValue	探针 2 的下降沿捕获值。
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，60B9h、60BAh、60BBh）配置为 PDO。 其他返回值：请参照指令返回值列表。
相关指令	GT_SetTouchProbeFunction、GT_SetTouchProbeFunctionEx
指令示例	**例程 5-3 探针 1 上升沿连续捕获（以 GTHD 为例）**

指令 74 GT_GetTrapPrm

指令原型	short GT_GetTrapPrm(short profile, TTrapPrm *pPrm)		
指令说明	读取点位运动模式下的运动参数。		
指令类型	立即指令，调用后立即生效。	章节页码	43

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令参数	该指令共有 2 个参数，参数的详细信息如下。
profile	规划轴号。
pPrm	读取点位运动模式运动参数，该参数为一个**结构体**，包含四个参数，详细的参数定义及说明请参照 GT_SetTrapPrm 指令说明。
指令返回值	若返回值为 1： 请检查当前轴是否为 Trap 模式，若不是，请先调用 GT_PrfTrap 将当前轴设置为 Trap 模式。 其他返回值：请参照指令返回值列表。
相关指令	GT_SetTrapPrm
指令示例	**例程 7-1 点位运动**

指令 75 GT_GetVarId

指令原型	short GT_GetVarId(char *pFunName, char *pVarName, TVarInfo *pVarInfo)		
指令说明	读取运动程序中变量的标识。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
pFunName	全局变量输入 0。 局部变量所在函数的名称。		
pVarName	运动程序变量名称。		
pVarInfo	根据运动程序函数名称和变量名称查询变量标识。		
指令返回值	若返回值为 1，2007 或者 2008： 请检查重新检查 GT_Download 是否调用成功。 若失败，请根据 GT_Download 返回值提示操作，直至成功。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 10-1 运动程序单线程累加求和**		

指令 76 GT_GetVarValue

指令原型	short GT_GetVarValue(short page, TVarInfo *pVarInfo, double *pValue, short count=1)		
指令说明	读取运动程序中变量的值。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
page	数据页编号。 全局变量为-1；局部变量取值范围：[0, 31]。		
pVarInfo	需要访问的变量标识。		
pValue	需要读取的变量值。		
count	需要读取的变量值的数量，取值范围：[1, 6]。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_SetVarValue		
指令示例	**例程 10-1 运动程序单线程累加求和**		

指令 77 GT_GetVel

指令原型	short GT_GetVel(short profile, double *pVel)
指令说明	读取目标速度。

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令类型	立即指令，调用后立即生效。	章节页码	43
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
pVel	读取目标速度。单位：pulse/ms。		
指令返回值	若返回值为 1： 请检查当前轴是否为 Trap 模式，若不是，请先调用 GT_PrfTrap 将当前轴设置为 Trap 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_SetVel		
指令示例	无		

指令 78 GT_GetVersion

指令原型	short GT_GetVersion(char *pVersion)		
指令说明	读取运动控制器固件的版本号。		
指令类型	立即指令，调用后立即生效。	章节页码	94
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
pVersion	读取的运动控制器的固件版本号字符串。		
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 11-1 读取控制器固件版本号**		

指令 79 GT_InitEcatComm

指令原型	short GT_InitEcatComm()		
指令说明	EtherCAT初始化。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	无		
指令返回值	0 初始化成功；-8 配置错误，请检查 eni 配置文件和实际连接从站是否匹配；-9 未找到 eni 配置文件，请将 eni 文件放到可执行程序的相同目录；-10 未找到从站，请检查从站接线是否稳固		
相关指令	无		
指令示例	**例程 5-5 C 或 C++环境 EtherCAT**		

指令 80 GT_IsEcatReady

指令原型	short GT_IsEcatReady(short *pStatus)		
指令说明	查询EtherCAT通讯状态。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
pStatus	通讯状态： 0：通讯未完全建立；1：通讯完全建立。		
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 5-1 读取总线通讯状态**		

指令 81 GT_LmtsOff

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令原型	short GT_LmtsOff(short axis, short limitType=-1)		
指令说明	控制轴限位信号无效。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
axis	控制轴号。		
limitType	需要有效的限位类型： MC_LIMIT_POSITIVE(该宏定义为0)：将该轴的正限位无效。 MC_LIMIT_NEGATIVE(该宏定义为1)：将该轴的负限位无效。 -1：将该轴的正限位和负限位都无效，默认为该值。		
指令返回值	若返回值为 1：请检查相应轴限位报警，配置文件是否已经配置了限位无效。 其他返回值：请参照指令返回值列表。		
相关指令	GT_LmtsOn		
指令示例	无		

指令 82 GT_LmtsOn

指令原型	short GT_LmtsOn(short axis, short limitType=-1)		
指令说明	控制轴限位信号有效。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
axis	控制轴号。		
limitType	需要有效的限位类型： MC_LIMIT_POSITIVE(该宏定义为0)：将该轴的正限位有效。 MC_LIMIT_NEGATIVE(该宏定义为1)：将该轴的负限位有效。 -1：将该轴的正限位和负限位都有效，默认为该值。		
指令返回值	若返回值为 1：请检查相应轴限位报警，配置文件是否已经配置了限位无效。 其他返回值：请参照指令返回值列表。		
相关指令	GT_LmtsOff		
指令示例	无		

指令 83 GT_LoadConfig

指令原型	short GT_LoadConfig(char *pFile)		
指令说明	下载配置信息到运动控制器，调用该指令后需再调用GT_ClrSts才能使该指令生效。		
指令类型	立即指令，调用后立即生效。	章节页码	24
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
pFile	配置文件的文件名。文件名格式：*.cfg 或*.CFG。用户可根据自己的需求，使用运动控制器管理软件 MCT2008 生成此配置文件。		
指令返回值	若返回值为 1：请停止各轴规划运动后再设置。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 7-1 点位运动**		

指令 84 GT_Open

指令原型	short GT_Open(short channel=0, short param=1)

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令说明	打开运动控制器。		
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。		
channel	打开运动控制器的方式。		
param	保留。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_Close		
指令示例	无		

指令 85 GT_PauseThread

指令原型	short GT_PauseThread(short thread)		
指令说明	暂停正在运行的线程。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
thread	线程编号。取值范围：[0, 31]。		
指令返回值	若返回值为 1： (1) 请检查线程号是否已经绑定。 (2) 请检查相应的数据页中是否超出范围。 其他返回值：请参照指令返回值列表。		
相关指令	GT_RunThread; GT_StopThread		
指令示例	无		

指令 86 GT_PrfFollow

指令原型	short GT_PrfFollow(short profile, short dir)		
指令说明	设置指定轴为 Follow 运动模式。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
dir	设置跟随方式： 0 表示双向跟随；1 表示正向跟随；-1 表示负向跟随。		
指令返回值	若返回值为 1： (1) 若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 (2) 当前已经是 Follow 模式，但再次设置的 dir 与当前的 dir 不一致。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 7-7 Follow 单 FIFO 模式**		

指令 87 GT_PrfGear

指令原型	short GT_PrfGear(short profile, short dir)		
指令说明	设置指定轴为电子齿轮运动模式。		
指令类型	立即指令，调用后立即生效。	章节页码	57
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

dir	设置跟随方式： 0 表示双向跟随；1 表示正向跟随；-1 表示负向跟随。
指令返回值	若返回值为 1： (1)若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 (2)当前已经是电子齿轮模式，但再次设置的 dir 与当前的 dir 不一致。 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	**例程 7-5 电子齿轮跟随**

指令 88 GT_PrfJog

指令原型	short GT_PrfJog(short profile)		
指令说明	设置指定轴为 Jog 运动模式。		
指令类型	立即指令，调用后立即生效。	章节页码	46
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
profile	规划轴号。		
指令返回值	若返回值为 1：若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 7-2 Jog 运动**		

指令 89 GT_PrfPt

指令原型	short GT_PrfPt(short profile, short mode=PT_MODE_STATIC)		
指令说明	设置指定轴为 PT 运动模式。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
mode	指定 FIFO 使用模式： PT_MODE_STATIC（该宏定义为 0）静态模式。默认为该模式。 PT_MODE_DYNAMIC（该宏定义为 1）动态模式。		
指令返回值	若返回值为 1：若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 7-3 PT 静态 FIFO**		

指令 90 GT_PrfTrap

指令原型	short GT_PrfTrap(short profile)		
指令说明	设置指定轴为点位运动模式。		
指令类型	立即指令，调用后立即生效。	章节页码	43
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
profile	规划轴号。		
指令返回值	若返回值为 1：若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 其他返回值：请参照指令返回值列表。		
相关指令	无		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令示例	例程 7-1 点位运动

指令 91 GT_ProfileScale

指令原型	short GT_ProfileScale(short axis, short alpha, short beta)		
指令说明	设置控制轴的规划器当量变换值。注意：alpha的绝对值要大于beta的绝对值。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
axis	控制轴号。		
alpha	规划器当量的alpha值，取值范围：(-32767, 0)和(0, 32767)。		
beta	规划器当量的beta值，取值范围：(-32767, 0)和(0, 32767)。		
指令返回值	若返回值为 1：若当前轴再规划运动，请调用 GT_Stop 停止运动在调用该指令。 其他返回值：请参照指令返回值列表。		
相关指令	GT_EncScale		
指令示例	无		

指令 92 GT_PtClear

指令原型	short GT_PtClear(short profile, short fifo)		
指令说明	清除 PT 运动模式指定 FIFO 中的数据。 运动状态下该指令无效。 动态模式下该指令无效。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
fifo	指定所要查询的 FIFO，取值范围：0、1 两个值。默认为 0。 动态模式下该参数无效。		
指令返回值	若返回值为 1： (1) 静态模式下，检查要清除的 FIFO 是否正在使用，在运动； (2) 动态模式下，不能在运动时清 FIFO； (3) 请检查当前轴是否为 PT 模式，若不是，请先调用 GT_PrfPt 将当前轴设置为 PT 模式。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 7-3 PT 静态 FIFO**		

指令 93 GT_PtData

指令原型	short GT_PtData(short profile, double pos, long time, short type, short fifo=0)		
指令说明	向 PT 运动模式指定 FIFO 增加数据。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 5 个参数，参数的详细信息如下。		
profile	规划轴号。		
pos	段末位置，单位：pulse。		
time	段末时间，单位：ms。		
type	数据段类型：		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	PT_SEGMENT_NORMAL（该宏定义为 0）普通段。默认为该类型。 PT_SEGMENT_EVEN（该宏定义为 1）匀速段。 PT_SEGMENT_STOP（该宏定义为 2）减速到 0 段。
fifo	指定所要查询的 FIFO，取值范围：0、1 两个值。默认为 0。 动态模式下该参数无效。
指令返回值	若返回值为 1： (1) 请检查 Space 是否小于 0，若是，则等待 Space 大于 0； (2) 请检查当前轴是否为 PT 模式，若不是，请先调用 GT_PrfPt 将当前轴设置为 PT 模式。 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	**例程 7-3 PT 静态 FIFO**

指令 94 GT_PtSpace

指令原型	short GT_PtSpace(short profile, short *pSpace, short fifo=0)		
指令说明	查询 PT 运动模式指定 FIFO 的剩余空间。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
profile	规划轴号。		
pSpace	读取 PT 指定 FIFO 的剩余空间。		
fifo	指定所要查询的 FIFO，取值范围：0、1 两个值。默认为 0。 动态模式下该参数无效。		
指令返回值	若返回值为 1：请检查当前轴是否为 PT 模式，若不是，请先调用 GT_PrfPt 将当前轴设置为 PT 模式。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 7-3 PT 静态 FIFO**		

指令 95 GT_PtStart

指令原型	short GT_PtStart(long mask, long option)		
指令说明	启动 PT 运动。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
mask	按位指示需要启动 PT 运动的轴号。当 bit 位为 1 时表示启动对应的轴。		
option	按位指示所使用的 FIFO，默认为 0。 当 bit 位为 0 时表示对应的轴使用 FIFO1（即 fifo=0） 当 bit 位为 1 时表示对应的轴使用 FIFO2（即 fifo=1）		

Bit	11	10	9	8	7	6	5	4	3	2	1	0
对应轴	12 轴	11 轴	10 轴	9 轴	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令返回值	动态模式下该参数无效。 若返回值为 1：请检查相应轴的 FIFO 是否有数据，若没有，请先压入数据； 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	**例程 7-3 PT 静态 FIFO**

指令 96 GT_Reset

指令原型	short GT_Reset()		
指令说明	复位运动控制器。		
指令类型	立即指令，调用后立即生效。	章节页码	94
指令参数	无		
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 4-3 初始化配置控制器**		

指令 97 GT_RunThread

指令原型	short GT_RunThread(short thread)		
指令说明	启动线程。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
thread	线程编号，取值范围：[0, 31]。		
指令返回值	若返回值为 1： (1) 请检查线程号是否已经绑定。 (2) 请检查相应的数据页中是否超出范围。 其他返回值：请参照指令返回值列表。		
相关指令	GT_PauseThread; GT_StopThread		
指令示例	**例程 10-1 运动程序单线程累加求和**		

指令 98 GT_SetAxisBand

指令原型	short GT_SetAxisBand(short axis, long band, long time)		
指令说明	设置轴到位误差带。 规划器静止，规划位置和实际位置的误差小于设定误差带，并且在误差带内保持设定时间后，置起到位标志。		
指令类型	立即指令，调用后立即生效。	章节页码	95
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
axis	轴号。		
band	误差带大小，单位：脉冲。		
time	误差带保持时间，单位：250 微秒。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_GetAxisBand		
指令示例	**例程 11-2 电机到位检测功能**		

指令 99 GT_SetEcatAxisDO

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令原型	short GT_SetEcatAxisDO(short axis, unsigned long DoValue)		
指令说明	设置EtherCAT轴的数字量输出。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
DoValue	数字量输出值。		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，60FEh）配置为 PDO。 其他返回值：请参照指令返回值列表。		
相关指令	GT_GetEcatAxisDO		
指令示例	无		

指令 100 GT_SetEcatAxisMode

指令原型	short GT_SetEcatAxisMode(short axis, short mode);		
指令说明	设置 EtheCAT 轴的操作模式		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
mode	伺服操作模式（以 GTHD 为例）： 0：No mode assigned 1：Profile position 3：Profile velocity 4：Profile torque 6：Homing 8：Cyclic sync position 9：Cyclic sync velocity 10：Cyclic sync torque		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，6060h）配置为 PDO。 其他返回值：请参照指令返回值列表。		
相关指令	GT_GetEcatAxisMode		
指令示例	无		

指令 101 GT_SetEcatAxisPT

指令原型	short GT_SetEcatAxisPT(short axis, short torque);		
指令说明	设置 EtherCAT 轴的力矩。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
torque	力矩值		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，6071h）配置为 PDO。 其他返回值：请参照指令返回值列表。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

相关指令	无
指令示例	无

指令 102 GT_SetEcatAxisPV

指令原型	short GT_SetEcatAxisPV(short axis, long velocity);		
指令说明	设置 EtherCAT 轴的速度		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
velocity	速度值		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，60FFh）配置为 PDO。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	无		

指令 103 GT_SetEcatHomingPrm

指令原型	short GT_SetEcatHomingPrm(short axis, short method, double speed1, double speed2, double acc, long offset, unsigned short probeFunction);		
指令说明	设置EtherCAT轴的回零参数。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 7 个参数，参数的详细信息如下。		
axis	轴号，取值范围：[1, 12]。		
method	回零方式。		
speed1	搜索开关速度，单位：驱动器设置的用户速度单位		
speed2	搜索 index 标识速度，单位：驱动器设置的用户速度单位		
acc	搜索加速度，单位：驱动器设置的用户加速度单位		
offset	原点偏移量，单位：驱动器设置的用户位置单位		
probeFunction	探针功能。		
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 5-2 采用 3 号回零方式**		

指令 104 GT_SetEcatRawData

指令原型	short GT_SetEcatRawData(unsigned short offset, unsigned short size, unsigned char *pValue)		
指令说明	设置控制器 PDO 数据。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
offset	设置 PDO 数据偏移值，取值范围：0 ～ 当前 PDO 长度，单位：字节。		
size	设置数据长度，单位：字节。		
pValue	设置数据缓冲区地址。		
指令返回值	若返回值为 1：请检查 EtherCAT 通讯是否正常。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	其它返回值：请参照指令返回值列表。
相关指令	GT_GetEcatRawData
指令示例	无

指令 105 GT_SetEncPos

指令原型	short GT_SetEncPos(short encoder, long encPos)		
指令说明	修改编码器位置。		
指令类型	立即指令，调用后立即生效。	章节页码	76
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
encoder	编码器起始轴号，取值范围：[1, 12]。		
encPos	编码器位置。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_GetEncPos		
指令示例	无		

指令 106 GT_SetFollowEvent

指令原型	short GT_SetFollowEvent(short profile, short event, short masterDir, long pos)		
指令说明	设置 Follow 运动模式启动跟随条件。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	规划轴号。		
event	启动跟随条件： FOLLOW_EVENT_START（该宏定义为 1）表示调用 GT_FollowStart 以后立即启动。 FOLLOW_EVENT_PASS（该宏定义为 2）表示主轴穿越设定位置以后启动跟随。		
masterDir	穿越启动时，主轴的运动方向。 1 主轴正向运动，-1 主轴负向运动。		
pos	穿越位置，单位：pulse。 当 event 为 FOLLOW_EVENT_PASS 时有效。		
指令返回值	若返回值为 1：请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_GetFollowEvent		
指令示例	**例程 7-7 Follow 单 FIFO**		

指令 107 GT_SetFollowLoop

指令原型	short GT_SetFollowLoop(short profile, short loop)		
指令说明	设置 Follow 运动模式下的循环次数。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
loop	指定 Follow 模式循环执行的次数。 取值范围：[-32768, 32767]。注：loop 小于 1 表示无限次循环。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令返回值	若返回值为 1：请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。 其他返回值：请参照指令返回值列表。
相关指令	GT_GetFollowLoop
指令示例	**例程 7-7 Follow 单 FIFO**

指令 108 GT_SetFollowMaster

指令原型	short GT_SetFollowMaster(short profile, short masterIndex, short masterType = FOLLOW_MASTER_PROFILE, short masterItem)		
指令说明	设置 Follow 运动模式下的跟随主轴。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	规划轴号。		
masterIndex	主轴索引。 主轴索引不能与规划轴号相同，最好主轴索引号小于规划轴号，如主轴索引为 1 轴，规划轴号为 2 轴。		
masterType	主轴类型： FOLLOW_MASTER_PROFILE（该宏定义为 2）表示跟随规划轴(profile)的输出值。默认为该类型。 FOLLOW_MASTER_ENCODER（该宏定义为 1）表示跟随编码器(encoder)的输出值。 FOLLOW_MASTER_AXIS（该宏定义为 3）表示跟随轴(axis)的输出值。		
masterItem	合成轴类型，当 masterType= FOLLOW_MASTER_AXIS 时起作用。 0 表示 axis 的规划位置输出值，默认为该值。 1 表示 axis 的编码器位置输出值。		
指令返回值	若返回值为 1： (1) 若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 (2) 请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置为 Follow 模式。 其他返回值：请参照指令返回值列表		
相关指令	GT_GetFollowMaster		
指令示例	**例程 7-7 Follow 单 FIFO**		

指令 109 GT_SetFollowMemory

指令原型	short GT_SetFollowMemory(short profile, short memory)		
指令说明	设置 Follow 运动模式的缓存区大小。		
指令类型	立即指令，调用后立即生效。	章节页码	61
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
memory	Follow 运动缓存区大小标志。 0：每个 Follow 运动缓存区有 16 段空间。 1：每个 Follow 运动缓存区有 512 段空间。		
指令返回值	若返回值为 1： (1) 若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 (2) 请检查当前轴是否为 Follow 模式，若不是，请先调用 GT_PrfFollow 将当前轴设置		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	为 Follow 模式。 其他返回值：请参照指令返回值列表。
相关指令	GT_GetFollowMemory
指令示例	无

指令 110 GT_SetGearMaster

指令原型	short GT_SetGearMaster(short profile, short masterIndex, short masterType, short masterItem)		
指令说明	设置电子齿轮运动跟随主轴。		
指令类型	立即指令，调用后立即生效。	章节页码	57
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	规划轴号。		
masterIndex	主轴索引。 主轴索引不能与规划轴号相同，最好主轴索引号小于规划轴号，如主轴索引为 1 轴，规划轴号为 2 轴。		
masterType	主轴类型。 GEAR_MASTER_PROFILE（该宏定义为 2）表示跟随规划轴(profile)的输出值。默认为该类型。 GEAR_MASTER_ENCODER（该宏定义为 1）表示跟随编码器(encoder)的输出值。 GEAR_MASTER_AXIS（该宏定义为 3）表示跟随轴(axis)的输出值。		
masterItem	轴类型，当 masterType=GEAR_MASTER_AXIS 时起作用。 0 表示 axis 的规划位置输出值。默认为该值。 1 表示 axis 的编码器位置输出值。		
指令返回值	若返回值为 1： (1) 若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 (2) 请检查当前轴是否为电子齿轮模式，若不是，请先调用 GT_PrfGear 将当前轴设置为电子齿轮模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_GetGearMaster		
指令示例	**例程 7-5 电子齿轮跟随**		

指令 111 GT_SetGearRatio

指令原型	short GT_SetGearRatio(short profile, long masterEven, long slaveEven, long masterSlope)		
指令说明	设置电子齿轮比。		
指令类型	立即指令，调用后立即生效。	章节页码	57
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	规划轴号。		
masterEven	传动比系数，主轴位移。 正整数，单位：pulse。		
slaveEven	传动比系数，从轴位移。 单位：pulse。		
masterSlope	主轴离合区位移。 单位：pulse。取值范围：不能小于 0 或者等于 1。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令返回值	若返回值为 1：请检查当前轴是否为电子齿轮模式，若不是，请先调用 GT_PrfGear 将当前轴设置为电子齿轮模式。 其他返回值：请参照指令返回值列表。
相关指令	GT_GetGearRatio
指令示例	**例程 7-5 电子齿轮跟随**

指令 112 GT_SetHomingMode

指令原型	short GT_SetHomingMode(short axis, short mode)		
指令说明	切换EtherCAT轴的回零模式。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
mode	模式选择。		
指令返回值	若返回值为 1：若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 5-2 采用 3 号回零方式**		

指令 113 GT_SetJogPrm

指令原型	short GT_SetJogPrm(short profile, TJogPrm *pPrm)
指令说明	设置 Jog 运动模式下的运动参数。
指令类型	立即指令，调用后立即生效。 章节页码 46
指令参数	该指令共有 2 个参数，参数的详细信息如下。
profile	规划轴号。
pPrm	设置 Jog 模式运动参数。该参数为一个**结构体**，包含三个参数，详细的参数定义及说明如下： typedef struct JogPrm { double acc; double dec; double smooth; } TJogPrm; **acc:** 点位运动的加速度。正数，单位：pulse/ms^2。 **dec:** 点位运动的减速度。正数，单位：pulse/ms^2。未设置减速度时，默认减速度和加速度相同。 **smooth:** 平滑系数。取值范围：[0, 1)。平滑系数的数值越大，加减速过程越平稳。
指令返回值	若返回值为 1： (1) 若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 (2) 请检查当前轴是否为 Jog 模式，若不是，请先调用 GT_PrfJog 将当前轴设置为 Jog 模式。 其他返回值：请参照指令返回值列表。
相关指令	GT_GetJogPrm
指令示例	**例程 7-2 Jog 运动**

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令 114 GT_SetMRAMData

指令原型	short GT_SetMRAMData(short address, short *pData, short count)		
指令说明	传输数据到控制器。		
指令类型	立即指令，调用后立即生效。	章节页码	99
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
address	控制器缓冲区数据起始地址，取值范围：[0, 4096)。		
pData	待传输数据缓冲区指针。		
count	数据长度，取值范围：[1, 24]。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_GetMRAMData		
指令示例	**例程 11-3 写数据至存储器**		

指令 115 GT_SetPos

指令原型	short GT_SetPos(short profile, long pos)		
指令说明	设置目标位置。		
指令类型	立即指令，调用后立即生效。	章节页码	43
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
pos	设置目标位置，单位：pulse。取值范围：[-1073741823, 1073741823]。		
指令返回值	若返回值为 1： 请检查当前轴是否为 Trap 模式，若不是，请先调用 GT_PrfTrap 将当前轴设置为 Trap 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_GetPos		
指令示例	**例程 7-1 点位运动**		

指令 116 GT_SetPosErr

指令原型	short GT_SetPosErr(short control, long error)		
指令说明	设置跟随误差极限值。		
指令类型	立即指令，调用后立即生效。	章节页码	25
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
control	伺服控制器编号。		
error	跟随误差极限值，取值范围：(0, 2147483648]。单位：pulse。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_GetPosErr		
指令示例	无		

指令 117 GT_SetPosScale

指令原型	short GT_SetPosScale(short axis, unsigned short scale);		
指令说明	设置编码器倍率，主要用于编码器分辨率较高的时候。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

scale	编码器倍率，2 的幂运算的指数值。 例如：编码器是 23 位，那么此值如果是 6，则运控换算变成 17 位。
指令返回值	若返回值为 1：若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 其他返回值：请参照指令返回值列表。
相关指令	GT_GetPosScale
指令示例	无

指令 118 GT_SetPrfPos

指令原型	short GT_SetPrfPos(short profile, long prfPos)		
指令说明	修改指定轴的规划位置。禁止在运动状态下修改规划位置。		
指令类型	立即指令，调用后立即生效。	章节页码	95
指令参数	该指令共有 2 个参数，参数的详细信息如下：		
profile	规划轴编号。		
prfPos	设置的规划位置的值。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_GetPrfPos		
指令示例	无		

指令 119 GT_SetPtLoop

指令原型	short GT_SetPtLoop(short profile, long loop)		
指令说明	设置 PT 运动模式循环执行的次数。动态模式下该指令无效。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴编号。		
loop	指定 PT 模式循环执行的次数。 取值范围：非负整数。如果需要无限循环，设置为 0。 动态模式下该参数无效。		
指令返回值	若返回值为 1：请检查当前轴是否为 PT 模式，若不是，请先调用 GT_PrfPt 将当前轴设置为 PT 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_GetPtLoop		
指令示例	无		

指令 120 GT_SetPtMemory

指令原型	short GT_SetPtMemory(short profile, short memory)		
指令说明	设置 PT 运动模式的缓存区(FIFO)大小。		
指令类型	立即指令，调用后立即生效。	章节页码	49
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴编号。		
memory	PT 运动缓存区大小标志： 0：每个 PT 运动缓存区有 32 段空间。 1：每个 PT 运动缓存区有 1024 段空间。		
指令返回值	若返回值为 1：		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	(1) 若当前轴在规划运动，请调用 GT_Stop()停止运动再调用该指令。 (2) 请检查当前轴是否为 PT 模式，若不是，请先调用 GT_PrfPt 将当前轴设置为 PT 模式。 其他返回值：请参照指令返回值列表。
相关指令	GT_GetPtMemory
指令示例	无

指令 121 GT_SetSoftLimit

指令原型	short GT_SetSoftLimit(short axis, long positive, long negative)		
指令说明	设置轴正向软限位和负向软限位。		
指令类型	立即指令，调用后立即生效。	章节页码	78
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
axis	轴号。		
positive	正向软限位，当规划位置大于该值时，正限位触发。 默认值为：0x7fffffff，表示正向软限位无效。		
negative	负向软限位，当规划位置小于该值时，负限位触发。 默认值为：0x80000000，表示负向软限位无效。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_GetSoftLimit		
指令示例	**例程 9-1 软限位使用**		

指令 122 GT_SetStopDec

指令原型	short GT_SetStopDec(short profile, double decSmoothStop, double decAbruptStop)		
指令说明	设置平滑停止减速度和急停减速度。		
指令类型	立即指令，调用后立即生效。	章节页码	25
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。		
profile	规划器的编号。		
decSmoothStop	平滑停止减速度，取值范围：(0, 32767]。单位：pulse/ms^2。		
decAbruptStop	急停减速度，取值范围：(0, 32767]。单位：pulse/ms^2。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_GetStopDec		
指令示例	无		

指令 123 GT_SetTouchProbeFunction

指令原型	short GT_SetTouchProbeFunction(short axis, short ProbePrm)		
指令说明	设置EtherCAT轴的探针参数（参数方式）。		
指令类型	立即指令，调用后立即生效。	章节页码	29
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，取值范围：[1, 12]。		
ProbePrm	按位设置探针参数。		

BIT	描述
0	Probe1 开关　0:OFF　1:Enable

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	1	Probe1 触发类型 0：单次 1：连续
	2-3	保留
	4	上升沿触发 0：无效 1：有效
	5	下降沿触发 0：无效 1：有效
	6-7	保留
	8	Probe2 开关 0:OFF 1:Enable
	9	Probe2 触发类型 0：单次 1：连续
	10-11	保留
	12	上升沿触发 0：无效 1：有效
	13	下降沿触发 0：无效 1：有效
	14-15	保留
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，60B8h）配置为 PDO。 其他返回值：请参照指令返回值列表。	
相关指令	无	
指令示例	**例程 5-3 探针 1 上升沿连续捕获（以 GTHD 为例）**	

指令 124 GT_SetTouchProbeFunctionEx

指令原型	short GT_SetTouchProbeFunctionEx(short axis, short Probe1Enable, short Probe1TriggerType, short Probe1TriggerLevel, short Probe2Enable, short Probe2TriggerType, short Probe2TriggerLevel)		
指令说明	设置EtherCAT轴的探针参数（功能描述方式）。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 7 个参数，参数的详细信息如下。		
axis	轴号，取值范围：[1, 12]。		
Probe1Enable	探针通道 1 使能状态 0：不使能 1：使能。		
Probe1TriggerType	探针 1 的触发类型： ECAT_PROBE_TRIGGER_TYPE_CONTINUES 连续触发； ECAT_PROBE_TRIGGER_TYPE_FIRST_EVENT 单次触发。		
Probe1TriggerLevel	探针 1 的触发电平： ECAT_PROBE_TRIGGER_LEVEL_POS 上升沿触发； ECAT_PROBE_TRIGGER_LEVEL_NEG 下降沿触发； ECAT_PROBE_TRIGGER_LEVEL_DUL 双沿触发。		
Probe2Enable	探针通道 2 使能状态 0： 不使能 1：使能。		
Probe2TriggerType	探针 2 的触发类型： ECAT_PROBE_TRIGGER_TYPE_CONTINUES 连续触发； ECAT_PROBE_TRIGGER_TYPE_FIRST_EVENT 单次触发。		
Probe2TriggerLevel	探针 2 的触发电平： ECAT_PROBE_TRIGGER_LEVEL_POS 上升沿触发； ECAT_PROBE_TRIGGER_LEVEL_NEG 下降沿触发； ECAT_PROBE_TRIGGER_LEVEL_DUL 双沿触发。		
指令返回值	若返回值为 1：请检查相应轴在 Gecat 配置文件中是否已经将相关对象（以 GTHD 为例，60B8h）配置为 PDO。 其他返回值：请参照指令返回值列表。		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

相关指令	无
指令示例	无

指令 125 GT_SetTrapPrm

指令原型	short GT_SetTrapPrm(short profile, TTrapPrm *pPrm)		
指令说明	设置点位模式运动下的运动参数。		
指令类型	立即指令，调用后立即生效。	章节页码	43
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
profile	规划轴号。		
pPrm	设置点位运动模式运动参数，该参数为一个**结构体**，包含四个参数，详细的参数定义及说明如下： typedef struct TrapPrm { double acc; double dec; double velStart; short smoothTime; } TTrapPrm; **acc:** 点位运动的加速度。正数，单位：pulse/ms^2。 **dec:** 点位运动的减速度。正数，单位：pulse/ms^2。未设置减速度时，默认减速度和加速度相同。 **velStart:** 起跳速度。正数，单位：pulse/ms。默认值为 0。 **smoothTime:** 平滑时间。正整数，取值范围：[0, 50]，单位 ms。平滑时间的数值越大，加减速过程越平稳。		
指令返回值	若返回值为 1： (1) 若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 (2) 请检查当前轴是否为 Trap 模式，若不是，请先调用 GT_PrfTrap 将当前轴设置为 Trap 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_GetTrapPrm		
指令示例	**例程 7-1 点位运动**		

指令 126 GT_SetVarValue

指令原型	short GT_SetVarValue(short page, TVarInfo *pVarInfo, double *pValue, short count=1)		
指令说明	设置运动程序中变量的值。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令参数	该指令共有 4 个参数，参数的详细信息如下。		
page	数据页编号。 全局变量为-1；局部变量取值范围：[0, 31]。		
pVarInfo	需要访问的变量标识。		
pValue	需要写入的变量值。		
count	需要写入的变量值的数量，取值范围：[1, 6]。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_GetVarValue		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令示例	例程 10-1 运动程序单线程累加求和

指令 127 GT_SetVel

指令原型	short GT_SetVel(short profile, double vel)		
指令说明	设置目标速度。		
指令类型	立即指令，调用后立即生效。	章节页码	43
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
profile	规划轴号。		
vel	设置目标速度，单位：pulse/ms。		
指令返回值	若返回值为 1： 请检查当前轴是否为 Trap 模式，若不是，请先调用 GT_PrfTrap 将当前轴设置为 Trap 模式。 其他返回值：请参照指令返回值列表。		
相关指令	GT_GetVel		
指令示例	例程 7-1 点位运动		

指令 128 GT_StartAccessMRAM

指令原型	short GT_StartAccessMRAM(short address, short count, short mode)		
指令说明	启动存储器读/写操作。		
指令类型	立即指令，调用后立即生效。	章节页码	99
指令参数	该指令共有 3 个参数，参数的详细信息如下。		
address	控制器缓冲区数据起始地址，取值范围：[0, 4096)。		
count	数据长度，取值范围：(0, 4096]。		
mode	操作模式，0 读存储器；1 写存储器。		
指令返回值	请参照指令返回值列表。		
相关指令	GT_StopAccessMRAM		
指令示例	例程 11-3 写数据至存储器、例程 11-4 从存储器读数据		

指令 129 GT_StartEcatComm

指令原型	short GT_StartEcatComm()		
指令说明	启动DSP总线运动控制。成功调用这条指令后，才可以调用其他运动指令。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	无		
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 5-5 C 或 C++环境 EtherCAT		

指令 130 GT_StartEcatHoming

指令原型	short GT_StartEcatHoming(short axis)		
指令说明	启动EtherCAT轴回零。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
axis	轴号，取值范围：[1, 12]。		
指令返回值	若返回值为 1：请检查相应轴是否满足以下条件：		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

	1. 目标驱动器处于伺服使能状态； 2. 目标驱动器的操作模式是回零模式； 3. 已成功设置回零参数。 其他返回值：请参照指令返回值列表。
相关指令	无
指令示例	**例程 5-2 采用 3 号回零方式**

指令 131 GT_Stop

指令原型	short GT_Stop(long mask, long option)		
指令说明	停止一个或多个轴的规划运动，停止坐标系运动。		
指令类型	立即指令，调用后立即生效。	章节页码	38
指令参数	该指令共有 2 个参数，参数的详细信息如下。		

mask

按位指示需要停止运动的轴号或者坐标系号。当 bit 位为 1 时表示停止对应的轴或者坐标系。

Bit	13	12	11	10	9	8	7	6	5	4	3	2	1	0
对应轴或坐标系	坐标系 2	坐标系 1	12 轴	11 轴	10 轴	9 轴	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴

option

按位指示停止方式。当 bit 位为 0 时表示平滑停止对应的轴或坐标系，当 bit 位为 1 时表示急停对应的轴或坐标系。

Bit	13	12	11	10	9	8	7	6	5	4	3	2	1	0
对应轴或坐标系	坐标系 2	坐标系 1	12 轴	11 轴	10 轴	9 轴	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴

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 132 GT_StopAccessMRAM

指令原型	short GT_StopAccessMRAM()		
指令说明	停止存储器读/写操作。		
指令类型	立即指令，调用后立即生效。	章节页码	99
指令参数	无		
指令返回值	请参照指令返回值列表。		
相关指令	GT_StartAccessMRAM		
指令示例	无		

指令 133 GT_StopEcatHoming

指令原型	short GT_StopEcatHoming(short axis)		
指令说明	停止EtherCAT轴回零。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
axis	轴号，取值范围：[1, 12]。		
指令返回值	请参照指令返回值列表。		
相关指令	无		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令示例	无

指令 134 GT_StopThread

指令原型	short GT_StopThread(short thread)		
指令说明	停止正在运行的线程。		
指令类型	立即指令，调用后立即生效。	章节页码	84
指令类型	立即指令，调用后立即生效。		
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
thread	线程编号，取值范围：[0, 31]。		
指令返回值	若返回值为 1： (1) 请检查线程号是否已经绑定。 (2) 请检查相应的数据页中是否超出范围。 其他返回值：请参照指令返回值列表。		
相关指令	GT_RunThread; GT_PauseThread		
指令示例	**例程 10-1 运动程序单线程累加求和**		

指令 135 GT_SynchAxisPos

指令原型	short GT_SynchAxisPos(long mask)		
指令说明	axis 合成规划位置和所关联的 profile 同步。 axis 合成编码器位置和所关联的 encoder 同步。		
指令类型	立即指令，调用后立即生效。	章节页码	95
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
mask	按位标识需要进行位置同步的轴号。0：表示不需要进行位置同步，1：需要进行位置同步。		
指令返回值	若返回值为 1：请检查参数 mask 是否设置为 0。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	无		

Bit	13	12	11	10	9	8	7	6	5	4	3	2	1	0
对应轴或坐标系	坐标系 2	坐标系 1	12 轴	11 轴	10 轴	9 轴	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴

指令 136 GT_TerminateEcatComm

指令原型	short GT_TerminateEcatComm()		
指令说明	结束EtherCAT通讯。		
指令类型	立即指令，调用后立即生效。	章节页码	29
指令参数	无		
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 137 GT_Update

指令原型	short GT_Update(long mask)

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令说明	启动点位运动或 Jog 运动。		
指令类型	立即指令，调用后立即生效。	章节页码	43
指令参数	该指令共有 1 个参数，参数的详细信息如下。		
mask	按位指示需要启动点位运动或 Jog 运动的轴号。当 bit 位为 1 时表示启动对应的轴。		

Bit	11	10	9	8	7	6	5	4	3	2	1	0
对应轴	12 轴	11 轴	10 轴	9 轴	8 轴	7 轴	6 轴	5 轴	4 轴	3 轴	2 轴	1 轴

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 7-1 点位运动**

指令 138 GT_ZeroPos

指令原型	short GT_ZeroPos(short axis, short count)		
指令说明	清零规划位置和实际位置，并进行零漂补偿。		
指令类型	立即指令，调用后立即生效。	章节页码	95
指令参数	该指令共有 2 个参数，参数的详细信息如下。		
axis	需要位置清零的起始轴号。		
count	需要位置清零的轴数。		
指令返回值	若返回值为 1：若当前轴在规划运动，请调用 GT_Stop 停止运动再调用该指令。 其他返回值：请参照指令返回值列表。		
相关指令	无		
指令示例	**例程 7-1 点位运动**		

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

第14章 索引

14.1 指令索引

指令 1 GetMacAddress 103
指令 2 GT_AlarmOff 103
指令 3 GT_AlarmOn 103
指令 4 GT_AxisOff 103
指令 5 GT_AxisOn 104
指令 6 GT_Bind 104
指令 7 GT_Close 104
指令 8 GT_ClrSts 105
指令 9 GT_Compile 105
指令 10 GT_Download 105
指令 11 GT_EcatIOReadInput 106
指令 12 GT_EcatIOWriteOutput 106
指令 13 GT_EcatSDODownload 106
指令 14 GT_EcatSDOUpLoad 107
指令 15 GT_EncOff 107
指令 16 GT_EncOn 107
指令 17 GT_EncScale 108
指令 18 GT_EncSns 108
指令 19 GT_FollowClear 108
指令 20 GT_FollowData 109
指令 21 GT_FollowSpace 109
指令 22 GT_FollowStart 109
指令 23 GT_FollowSwitch 110
指令 24 GT_GearStart 110
指令 25 GT_GetAxisBand 111
指令 26 GT_GetAxisEncAcc 111
指令 27 GT_GetAxisEncPos 111
指令 28 GT_GetAxisEncVel 112
指令 29 GT_GetAxisError 112
指令 30 GT_GetAxisPrfAcc 112
指令 31 GT_GetAxisPrfPos 113
指令 32 GT_GetAxisPrfVel 113
指令 33 GT_GetClock 113
指令 34 GT_GetEcatAxisAI 114
指令 35 GT_GetEcatAxisAtlCurrent 114
指令 36 GT_GetEcatAxisAtlTorque 114
指令 37 GT_GetEcatAxisDI 115
指令 38 GT_GetEcatAxisDO 115
指令 39 GT_GetEcatAxisMode 115

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令 40 GT_GetEcatAxisPE 116
指令 41 GT_GetEcatEncPos 116
指令 42 GT_GetEcatEncVel 116
指令 43 GT_GetEcatHomingStatus 117
指令 44 GT_GetEcatRawData 117
指令 45 GT_GetEcatSlaves 117
指令 46 GT_GetEncPos 118
指令 47 GT_GetEncVel 118
指令 48 GT_GetFollowEvent 118
指令 49 GT_GetFollowLoop 119
指令 50 GT_GetFollowMaster 119
指令 51 GT_GetFollowMemory 120
指令 52 GT_GetFunId 120
指令 53 GT_GetGearMaster 120
指令 54 GT_GetGearRatio 121
指令 55 GT_GetJogPrm 121
指令 56 GT_GetMcEcatAxisNum 122
指令 57 GT_GetMRAMData 122
指令 58 GT_GetMRAMStatus 122
指令 59 GT_GetPos 122
指令 60 GT_GetPosErr 123
指令 61 GT_GetPosScale 123
指令 62 GT_GetPrfAcc 123
指令 63 GT_GetPrfMode 123
指令 64 GT_GetPrfPos 124
指令 65 GT_GetPrfVel 124
指令 66 GT_GetPtLoop 125
指令 67 GT_GetPtMemory 125
指令 68 GT_GetPtRemainder 125
指令 69 GT_GetSoftLimit 125
指令 70 GT_GetStopDec 126
指令 71 GT_GetSts 126
指令 72 GT_GetThreadSts 126
指令 73 GT_GetTouchProbeStatus 127
指令 74 GT_GetTrapPrm 127
指令 75 GT_GetVarId 128
指令 76 GT_GetVarValue 128
指令 77 GT_GetVel 128
指令 78 GT_GetVersion 129
指令 79 GT_InitEcatComm 129
指令 80 GT_IsEcatReady 129
指令 81 GT_LmtsOff 129
指令 82 GT_LmtsOn 130
指令 83 GT_LoadConfig 130
指令 84 GT_Open 130

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令 85 GT_PauseThread 131
指令 86 GT_PrfFollow 131
指令 87 GT_PrfGear 131
指令 88 GT_PrfJog 132
指令 89 GT_PrfPt 132
指令 90 GT_PrfTrap 132
指令 91 GT_ProfileScale 133
指令 92 GT_PtClear 133
指令 93 GT_PtData 133
指令 94 GT_PtSpace 134
指令 95 GT_PtStart 134
指令 96 GT_Reset 135
指令 97 GT_RunThread 135
指令 98 GT_SetAxisBand 135
指令 99 GT_SetEcatAxisDO 135
指令 100 GT_SetEcatAxisMode 136
指令 101 GT_SetEcatAxisPT 136
指令 102 GT_SetEcatAxisPV 137
指令 103 GT_SetEcatHomingPrm 137
指令 104 GT_SetEcatRawData 137
指令 105 GT_SetEncPos 138
指令 106 GT_SetFollowEvent 138
指令 107 GT_SetFollowLoop 138
指令 108 GT_SetFollowMaster 139
指令 109 GT_SetFollowMemory 139
指令 110 GT_SetGearMaster 140
指令 111 GT_SetGearRatio 140
指令 112 GT_SetHomingMode 141
指令 113 GT_SetJogPrm 141
指令 114 GT_SetMRAMData 142
指令 115 GT_SetPos 142
指令 116 GT_SetPosErr 142
指令 117 GT_SetPosScale 142
指令 118 GT_SetPrfPos 143
指令 119 GT_SetPtLoop 143
指令 120 GT_SetPtMemory 143
指令 121 GT_SetSoftLimit 144
指令 122 GT_SetStopDec 144
指令 123 GT_SetTouchProbeFunction 144
指令 124 GT_SetTouchProbeFunctionEx 145
指令 125 GT_SetTrapPrm 146
指令 126 GT_SetVarValue 146
指令 127 GT_SetVel 147
指令 128 GT_StartAccessMRAM 147
指令 129 GT_StartEcatComm 147

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

指令 130 GT_StartEcatHoming 147
指令 131 GT_Stop 148
指令 132 GT_StopAccessMRAM 148
指令 133 GT_StopEcatHoming 148
指令 134 GT_StopThread 149
指令 135 GT_SynchAxisPos 149
指令 136 GT_TerminateEcatComm 149
指令 137 GT_Update 149
指令 138 GT_ZeroPos 150

14.2 例程索引

例程 3-1 检测 GTN 指令是否正常执行 13
例程 4-1 设置编码器计数方向 26
例程 4-2 设置限位触发报警无效 26
例程 4-3 初始化配置控制器 27
例程 5-1 读取总线通讯状态 34
例程 5-2 采用 3 号回零方式 34
例程 5-3 探针 1 上升沿连续捕获（以 GTHD 为例） 35
例程 5-4 EtherCAT IO 的使用 36
例程 5-5 C 或 C++环境 EtherCAT 初始化 37
例程 6-1 获取轴 1 的轴状态、运动模式、位置、速度和加速度 40
例程 7-1 点位运动 44
例程 7-2 Jog 运动 47
例程 7-3 PT 静态 FIFO 52
例程 7-4 PT 动态 FIFO 54
例程 7-5 电子齿轮跟随 59
例程 7-6 飞剪中的 Follow 模式应用 65
例程 7-7 Follow 单 FIFO 模式 67
例程 7-8 Follow 双 FIFO 切换 70
例程 8-1 访问编码器 76
例程 9-1 软限位使用 79
例程 10-1 运动程序单线程累加求和 86
例程 10-2 运动程序多线程累加求和 87
例程 11-1 读取控制器固件版本号 94
例程 11-2 电机到位检测功能 96
例程 11-3 写数据至存储器 100
例程 11-4 从存储器读数据 101

14.3 表格索引

表 1-1 指令列表 7

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

表 3-1 运动控制器指令返回值定义 13
表 4-1 下载配置文件指令 24
表 4-2 配置信息指令列表 25
表 4-3 编码器计数方向设置 26
表 4-4 控制器配置初始化状态 28
表 5-1 EtherCAT 库指令列表 29
表 6-1 运动状态检测指令列表 38
表 6-2 轴状态定义 39
表 7-1 设置运动模式指令列表 43
表 7-2 点位运动模式指令列表 43
表 7-3 Jog 运动模式指令列表 46
表 7-4 PT 运动模式指令列表 49
表 7-5 PT 静态 FIFO 例程数据段 52
表 7-6 电子齿轮运动模式指令列表 57
表 7-7 Follow 运动模式指令列表 61
表 7-8 飞剪案例区域 1 的数据段 66
表 7-9 飞剪案例区域 2 的数据段 66
表 7-10 Follow 单 FIFO 数据段 67
表 7-11 Follow 双 FIFO 切换之原来的跟随策略 70
表 7-12 Follow 双 FIFO 切换之更换跟随策略时的过渡段 70
表 7-13 Follow 双 FIFO 切换之更换后的跟随策略 70
表 8-1 运动控制器硬件资源 76
表 8-2 访问编码器指令列表 76
表 9-1 软限位指令列表 78
表 10-1 运动程序指令列表 84
表 10-2 可在运动程序中使用的指令 92
表 11-1 复位运动控制器指令列表 94
表 11-2 读取固件版本号指令列表 94
表 11-3 固件版本号的定义格式 94
表 11-4 读取系统时钟指令列表 95
表 11-5 打开/关闭电机使能信号指令列表 95
表 11-6 维护位置值指令列表 95
表 11-7 电机到位检测指令列表 95
表 11-8 铁电功能指令列表 99
表 12-1 加密函数指令列表 102

14.4 图片索引

图 4-1 开环运动控制系统的配置 16
图 4-2 闭环运动控制系统的配置 17
图 4-3 MCT2008 运动控制器管理软件界面 18
图 4-4 打开控制器配置 18
图 4-5 axis 配置界面 19
图 4-6 axis 配置对控制系统的影响 19

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层

图 4-7 encoder 配置界面 22
图 4-8 encoder 配置对控制系统的影响 22
图 4-9 encoder 输入脉冲反转项的影响 23
图 4-10 profile 配置界面 24
图 4-11 生成配置文件界面 25
图 5-1 PDO 配置界面 31
图 5-2 GTHD 默认配置之 PDO 信息 32
图 5-3 EtherCAT IO 默认配置之 PDO 信息 32
图 5-4 EtherCAT 耦合器和 IO 的 PDO 配置信息 33
图 5-5 EtherCAT 耦合器和 IO 的 IO Mapping 信息 34
图 7-1 点位运动速度曲线 44
图 7-2 点位运动速度规划 45
图 7-3 Jog 模式速度曲线 47
图 7-4 Jog 模式动态改变目标速度 47
图 7-5 PT 运动速度曲线 50
图 7-6 PT 模式匀速段类型 51
图 7-7 PT 模式停止段类型 51
图 7-8 PT 模式梯形曲线速度规划 52
图 7-9 PT 模式正弦曲线速度规划 55
图 7-10 电子齿轮模式速度曲线 58
图 7-11 电子齿轮模式主轴速度规划 59
图 7-12 电子齿轮模式从轴速度规划 59
图 7-13 Follow 模式主从轴规划 62
图 7-14 Follow 模式切换 FIFO 64
图 7-15 飞剪模型 65
图 7-16 飞剪案例之 Follow 模式规划曲线 66
图 7-17 Follow 单 FIFO 模式主轴速度规划 67
图 7-18 Follow 单 FIFO 模式从轴速度规划 68
图 7-19 Follow 双 FIFO 切换主轴速度规划 71
图 7-20 Follow 双 FIFO 切换从轴速度规划 71
图 9-1 轴运动范围 78
图 9-2 软限位触发 79
图 10-1 运动程序与应用程序的关系 82
图 10-2 MCT2008 运动程序编译说明界面 84
图 10-3 线程、函数和数据页的关系 85
图 11-1 电机到位检测功能 96
图 11-2 电机到位的运动状态检测 97
图 11-3 写数据流程示意图 100
图 11-4 读数据流程示意图 100

上海固高欧辰智能科技有限公司 www.softlinkcloud.cn 021-54708786 021-54708386 上海市闵行区东川路555号4号楼一层