



Central-I 总线型

直流驱控一体型

交流驱控一体型

运动控制器

Agito 控制器系列

Agito PCSuite 快速入门手册 v1.3



www.agito-akribis.com

Member of Akribis Systems group

版本记录

版本	描述	日期
1.0	首版发布	2023/8/11
1.1	调整部分内容排版	2023/9/20
1.2	修改 IDE 程序部分内容描述	2023/9/25
1.3	修改部分描述有误的内容，增加部分内容	2023/11/14

※本公司保留不定期更新的权利，根据产品硬件及软件的升级或更新迭代以及市场需求，本手册将会不定期进行内容上的更新调整，恕不另行告知，如需最新本档文档，请联系 Agito-Akribis 公司获取相应支持。

目录

1	介绍	5
1.1	关于手册	5
1.2	PCSuite简介	5
2	操作步骤	6
2.1	参数调试步骤	6
2.1.1	通讯连接	6
2.1.2	参数配置	7
2.2	换相及调参	14
2.2.1	自动换相	14
2.2.2	PIV 调参	17
2.3	简单运动控制	20
2.3.1	PTP 运动	20
2.3.2	Jog 运动	22
2.3.3	常用快捷指令	23
2.4	常用数字 I/O 功能配置	24
2.4.1	通用数字输入	25
2.4.2	通用数字输出	26
2.4.3	Hall 信号	27
2.4.4	限位信号	28
2.4.5	静态刹车	29
2.4.6	回零开关	31
2.4.7	编码器分频输出	31
2.4.8	PWM 输出	32
2.5	模拟 I/O 配置	33
2.5.1	模拟输入口配置	33
2.5.2	模拟输出口配置	35
2.6	更改控制器 IP 地址	36
3	PCSuite 常用工具介绍	37
3.1	参数导入/导出	37
3.2	示波器采集图像	41
3.2.1	常规/触发选项卡	42
3.2.2	图表设置选项卡	44
3.2.3	图形选项卡	46
3.2.4	分析选项卡	48
3.2.5	管理选项卡	49
3.2.6	数据记录控制	49
3.2.7	数据记录文件	50
3.3	悬浮窗工具	50
3.3.1	终端命令窗 (Terminal)	51
3.3.2	数据采集窗 (Data Recording)	51
3.3.3	数字输入悬浮窗 (Discret Inputs)	51
3.3.4	数字输出悬浮窗 (Discrete Output)	52
3.3.5	运动控制悬浮窗 (Motions)	52
3.3.6	运动状态显示窗 (Statuses)	53
3.3.7	图像采集窗 (Graphic Statuses)	53
3.3.8	参数监控窗 (Watch Statuses)	54
3.3.9	错误日志 (Error Log)	54
3.4	离线仿真	55

4	PCSuite IDE+编程环境	57
4.1	常用关键字	57
4.1.1	换相相关	57
4.1.2	回零相关	58
4.1.3	运动相关	59
4.1.4	电机参数相关	70
4.1.5	I/O 相关	72
4.1.6	系统相关	76
4.2	IDE 简介	79
4.3	常用 IDE Demo 程序	79
4.3.1	PTP 运动	80
4.3.2	Jog 运动	81
4.3.3	数字 I/O	81
4.3.4	飞拍 (Position Event)	82
4.3.5	误差补偿 (Err Mapping)	82
4.3.6	多线程 (Multi-Thread)	83
4.3.7	插补运动 (CNC)	83
5	常见故障代码及解决办法	84
6	附录	92

1 介绍

1.1 关于手册

感谢您选择 Agito 系列运动控制产品，我们将竭力为您提供追求速度与精度的极致运动控制方案，并提供全方位的技术支持；

本手册主要介绍如何快速使用 Agito PCSuite 软件配置电机参数，并控制电机正常运动的详细配置步骤，以及常用工具的使用说明；

关于其他功能的详细使用可参阅各功能应用文档。

1.2 PCSuite 简介

Agito PCSuite 是 Agito 控制器上位机软件，包含各类功能模块及工具，以及自带 IDE+编程环境，给用户提供一个功能齐全、使用方便快捷的调试工具。


2 操作步骤

- 在使用 PCSuite 配置参数前请确保外部电气线束已按照相应控制器使用手册连接好对应线束；
- 控制器已按照使用手册电压上电(注意控制器的上电电压请勿超过所允许电压范围)；
- 使用 PC Suite 设置参数时修改后需要点回车键保存或者点击“Apply and Next”保存；
- 断电前如果需要保存当前参数，点击菜单栏“Save to flash”保存参数到控制器 flash；
- 直线电机调试过程中，电机运动方向两侧请勿放置易损物品或人员，以防由于参数设置错误导致飞车造成危险；
- 在配置参数前，尽量使电机为空载状态。

2.1 参数调试步骤

2.1.1 通讯连接

Agito 控制器支持以太网、mini-USB、RS232、RS485、CANBus 等多种通讯方式，由于调试过程中数据交互量较大，因此推荐优先使用以太网通讯作为通讯方式，以下步骤以以太网通讯作为示例。

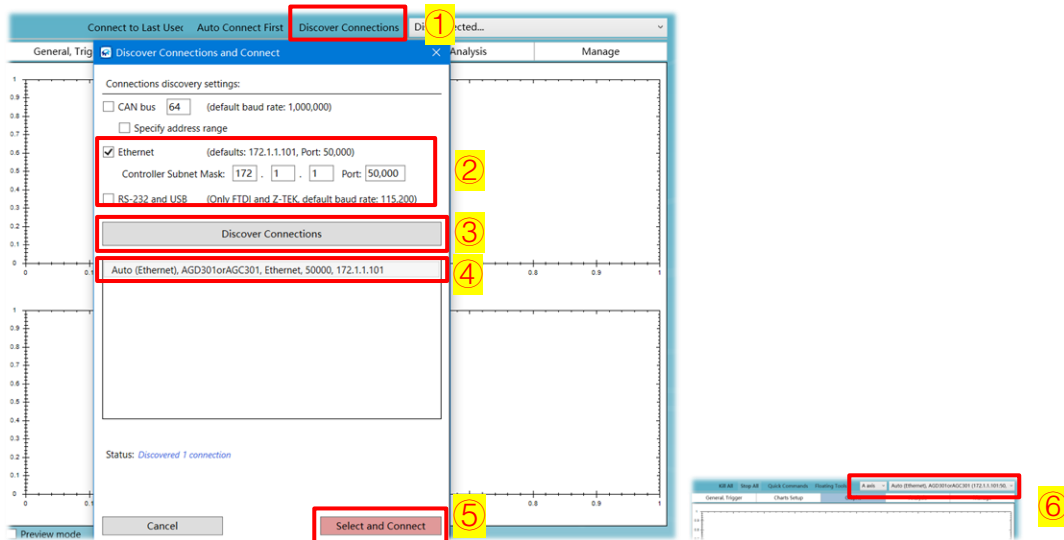
- **第 1 步：打开电脑网络连接：**  > 控制面板 > 网络和 Internet > 网络连接，并双击和控制器连接的网卡；



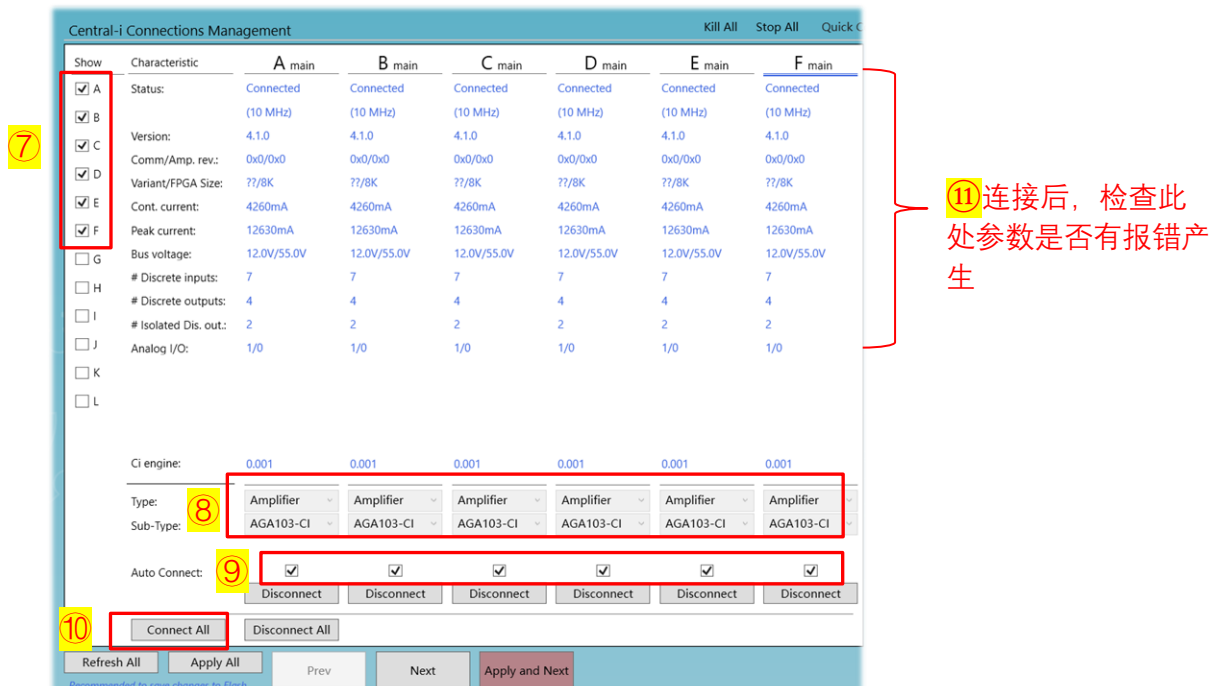
- **第 2 步：单击“属性”后，双击“TCP/IPv4”，将网卡设置为和驱动器同一网段，控制器默认 IP 为 172.1.1.101，可将网卡 IP 地址设为相同网段 IP，如 172.1.1.220，子网掩码设为 255.255.0.0；**



- 第 3 步：打开 PCSuite，依次点击①、②、③步骤，等待约 5s，弹出④表示读取到控制器，点击⑤连接控制器，图⑥表示连接成功。



- 第 4 步（如控制器为 AGM800）：如所连接的控制器为 AGM800 系列则需要点击底部功能栏“配置”→“Ci”，选择对应的放大器型号，连接放大器。



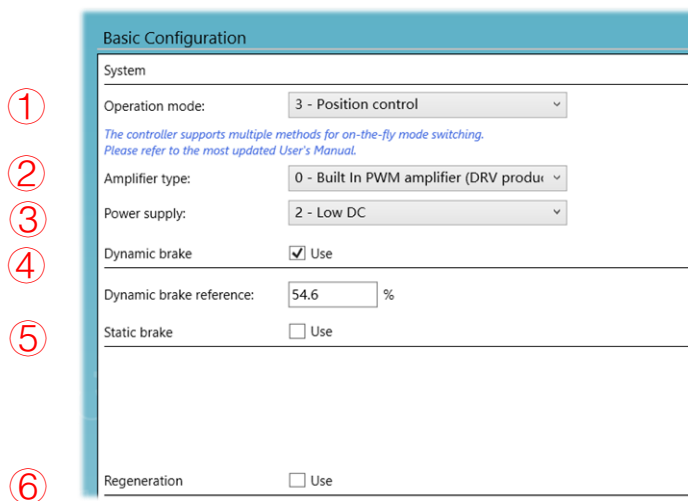
2.1.2 参数配置



- 第 1 步：选择底部菜单栏“配置”，子菜单“配置”选项，切换到基本配置

Configuration --> Basic Configuration

- ①**控制模式**：设置电机的控制模式，1~4 分别对应：电流(力矩)模式、速度模式、位置模式、力控模式。典型控制模式为“3-Position control”；
- ②**驱动方式**：可以选择 PWM 驱动、模拟量控制、脉冲/方向等。典型的伺服驱动为“0-Built in PWM amplifier”；Central-i 总线型 AGA 系列选择“0-Remote PWM amplifier”或“Remote P/D...”；Central-i 总线型 AGL 系列选择“2-Remote Analog(Current) command”。
- ③**供电电源**：1~3 分别为：单相交流、低压直流、三相交流，本例 AGD301 为直流控制器，选择“2-Low DC”，AGD155 为交流控制器选择“1-Single Phase”；
- ④**动态刹车**：电机运动过程中如果出现报警保护的时候，驱动器会将两相(或三相)短路进行紧急制动，推荐勾选；
- ⑤**静态刹车**：设置用户接外部抱闸(刹车)器相关参数，如用户电机有使用外部抱闸器，则需要勾选该选项。为方便调试建议在调试期间将刹车为持续上电松开状态，当参数调整完成后，再选择对应的打开抱闸方式（静态抱闸详细设置方法可参阅本文 2.4.4 章节）。
- ⑥**反电势吸收电阻**：开启之前先确保正确计算反电势再生电阻值，参数设置不合理将可能导致控制器损坏；
- ⑦**配置完成后**点击“Apply and Next”应用（后续步骤与此相同）。



The screenshot shows the 'Basic Configuration' window with the following settings:

- ① Operation mode: 3 - Position control
- ② Amplifier type: 0 - Built In PWM amplifier (DRV produ...
- ③ Power supply: 2 - Low DC
- ④ Dynamic brake: Use
- Dynamic brake reference: 54.6 %
- ⑤ Static brake: Use
- ⑥ Regeneration: Use

- 第 2 步：选择电机类型

Configuration --> Motor

根据对应轴所连接电机选择对应的电机类型；



The screenshot shows the 'Motor' configuration window with the following settings:

- ① Type: 4 - Rotary DC Brushless
- ② Number of pole pairs: 7

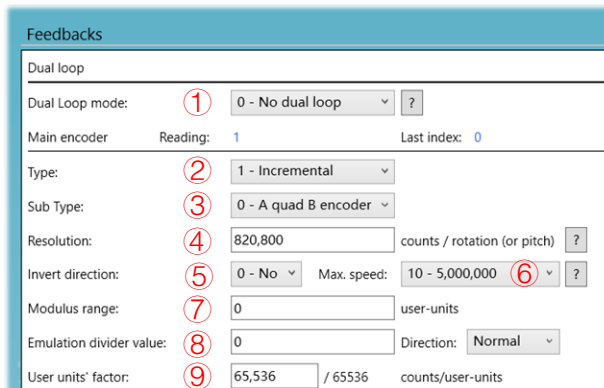
电机类型	极对数	备注信息
1-直流有刷电机	/	有刷电机无须设置
2-音圈	/	音圈电机无须设置

3-直线	1	定值，恒为 1
4-旋转伺服	根据电机手册填入	若手册是极数，则 $\text{极对数} = \text{极数} / 2$
5-仿真	离线仿真	参数来源于仿真文件
6-开环步进	典型 1.8° 两相步进极对数为 50	$\text{步距角} = 360^\circ / (2 * N * P)$ ，其中 N 为极对数，P 为电机相数
7-闭环步进	典型 1.8° 两相步进极对数为 50	$\text{步距角} = 360^\circ / (2 * N * P)$ ，其中 N 为极对数，P 为电机相数

■ **第 3 步：选择反馈类型**

Configuration --> Feedbacks

根据对应电机所使用的编码器类型选择对应的反馈类型；



- **①反馈模式：**选择是否启用全闭环控制，如使用全闭环功能请参阅《Agito 全闭环控制使用手册》，非全闭环控制选择“0-No dual loop”；
- **②反馈类型：**支持 ABI 增量式、EnDat 2.2 绝对式、BISS-C 绝对式、SinCos 模拟；
- **③子类别：**增量式编码器将会显示该选项，根据编码器手册选择，典型增量式编码器为“0-A quad B encoder”类型；
- **④分辨率：**根据编码器及电机参数手册设置，关于分辨率的详细设置方法见下一页，值得注意的是 Agito PCSuite 里默认是以 Counts 为单位；
- **⑤编码器方向：**设置编码器方向是否取反。以旋转电机为例，若“Invert direction”选择“0-No”时，CW 方向编码器反馈位置递增，此时若将“Invert direction”选择“1-Yes”后，编码器反馈位置将会改变为递减。注意电机在进行换相过程中，编码器必须朝正方向移动，模拟量编码器该选项不生效，请在模拟量编码器配置选项中设置“Invert direction”参数；
- **⑥最大速度：**此处设置控制器位置采样滤波频率，典型设置为默认“0-100,000,000”
- **⑦编码器计数模量范围：**值为 0 表示不启用该功能，当值设为 M 时表示编码器值将只会在 [0, M]Counts 之间范围内变化，如超过 M 时将从 0Counts 处重新计数，通常该值设为 0；
- **⑧编码器分频输出：**该值设置编码器分频系数，将主编码器读数按照设定系数比例通过 I/O 口分频输出，默认值为 0；
- **⑨用户单位：**用于设置用户自定义单位，Agito PCSuite 默认值为 65536 表示以 Counts 作为位置单位，如用户需要以其他单位作为位置信息从此处设置。

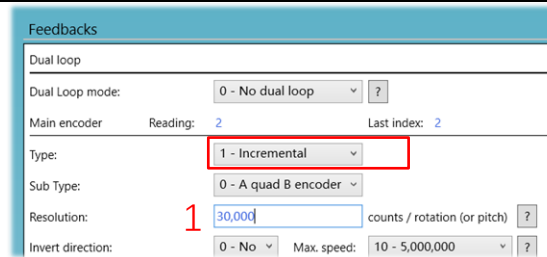
User-units 示例：

- ◆ 直线电机：若使用光栅尺分辨率为 1um，即 1um/Count，用户需要使用 mm 为单位，即 0.001mm/Count，则 count/user-unit 值设置为[1000*65536]=[65536000] Counts/user-unit，即表示所有的 user-unit 实际单位都为 mm；
- ◆ 旋转电机：若编码器分辨率为 5000Counts/rev，用户需要使用 Revolutions[转]为单位，则 counts/user-units 值设置为[5000*65536]=[327680000]，即表示所有的 user-unit 实际单位都是[转]；

✚ **编码器分辨率设置：** 以下将根据编码器类型详细介绍分辨率设置，

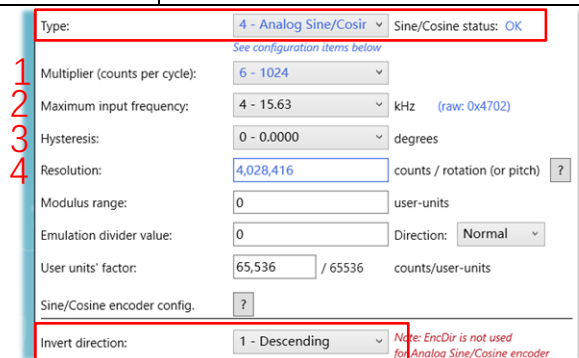
- **增量式编码器：** 仅支持 ABI 型 TTL 增量式编码器；

电机类型：	分辨率：
直线电机	Resolution=磁极距/光栅栅距； 例如本例中：Resolution=30mm / 1 μ m =30,000 Counts/pitch
旋转电机（以线数为单位）	Resolution=lines*4（常见为 4 倍频）
旋转电机（以脉冲为单位）	Counts 根据电机手册填写
音圈电机	Resolution≡1



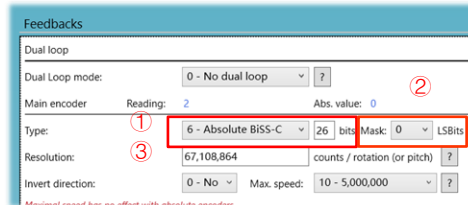
- **SinCos 模拟编码器：** 模拟量编码器分辨率有原始线数(line)和细分倍数得到；
 - ①**细分倍数：** 根据应用需求选择，与分辨率相关，典型设定值为“6-1024”；
 - ②**最大输入频率：** 用于编码器滤波，和细分倍数相关；
 - ③**磁滞角：** 仅在标定时使用，通常情况下设为默认“0-0.0000”即可；
 - ④**SinCos 分辨率：** 根据编码器线数和细分倍数设定。

电机类型：	分辨率：
直线电机	Resolution= $\frac{\text{电机磁极距}}{\text{光栅尺栅距}} * \text{细分倍数}$ ；
旋转电机	Resolution=lines*细分倍数
音圈电机	Resolution≡1



注意：模拟量编码器取反和其他不同，使用上图“Invert direction”；

- 绝对式编码器：支持 36-bits 及以下 BISS-C 和 EnDat2.2 编码器（单圈+多圈≤36-bits）；
 - ①绝对编码器位数：此处设置绝对编码器单圈位数，最大 32bit；
 - ②Mask：编码器屏蔽位，屏蔽低位；
 - ③绝对编码器分辨率：旋转电机根据编码器位数设定，直线电机根据电磁周期（磁极距）和光栅尺分辨率来设定（Resolution=磁极距/光栅栅距）；



- 旋转单圈绝对式编码器分辨率：Resolution=2^{AbsBit}，
以 26 位 BISS-C 单圈编码器为例：Resolution=2²⁶=67,108,864，
- 旋转多圈绝对式编码器分辨率：本例以 EnDat2.2 多圈旋转型为例，
EncAbsBits=SingleTurn Resolution+MultiTurn Resolution
Resolution=2^{SingleTurn Resolution}；

电机类型：	分辨率：
直线电机	Resolution= $\frac{\text{电机磁极距}}{\text{光栅尺栅距}}$
旋转电机	Resolution=2 ^{AbsBit} ；
音圈电机	Resolution=1

以单圈 13 位多圈 12 位 EnDat2.2 为例：

EncAbsBits=13+12=25（注：可分辨圈数 4096=2¹²）；

Resolution=2¹³=8192；

Encoder DataSheet:

每圈信号周期数	每圈位置数	可分辨圈数	接口
512	8192 (13 bit)	~4096	EnDat 2.2/01带~1Vpp

第 4 步：设定运动参数

Configuration --> MotionConfiguration

设定整定范围，即在设置的整定范围内(Setting tolerance)能维持所设定的整定时间(Setting time)，则控制器判断运动到位，

如以下示例当 PosErr 进入±10 范围内并在范围内维持至少 4ms，则控制器判断运动到位；



第 5 步：电机保护

Configuration --> Protections: Position and Velocity

本步骤用于设置电机相关保护参数避免某些意外情况导致设备损坏；

Protections: Position and Velocity	
Motion limitations	
① Position limits: Reverse: -160,000,000 Forward: 160,000,000 user-units	
Velocity/Acceleration limitations	
② Maximal allowed velocity: 40,000,000 user-units/sec	
③ Maximal allowed acceleration (future feature) 200,000,000 user-units/sec ²	
Maximal errors	
④ Maximal allowed position error: 15,000 user-units	
⑤ Maximal allowed velocity error: 20,000,000 user-units/sec	
Maximal errors in Open Loop modes (Open Loop, Injection and Identification)	
⑥ Maximal allowed open loop position error: 1,500,000 user-units	
⑦ Maximal allowed open loop velocity error: 20,000,000 user-units/sec	
Motor stuck	
⑧ Minimal current to activate protection: 1,500 mA	
⑨ Maximal velocity to define stuck: 1,000 user-units/sec	
⑩ Maximal time allowed in stuck: 1,000 msec	
Mask protections	
⑪ <input type="checkbox"/> Mask main encoder errors	⑫ <input checked="" type="checkbox"/> Mask auxiliary encoder errors (if exist)

- ①**软限位**：正负反向所允许到达的最大位置，超出该范围控制器将会报警（不下使能）；
- ②**最大运行速度**：运动过程中所允许的最大速度；
- ③**最大加速度**：该参数暂不支持用户更改；
- ④**最大跟随位置误差**：闭环模式运动过程中所允许的最大跟随位置误差；
- ⑤**最大跟随速度误差**：闭环模式运动过程中所允许的最大跟随速度误差；
- ⑥**最大开环位置误差**：开环模式运动过程中所允许的最大跟随位置误差（在 PIV 调参前建议将该参数尽量放大）；
- ⑦**最大开环速度误差**：开环模式运动过程中所允许的最大跟随速度误差；
- ⑧**堵转电流阈值**：判断电机堵转条件之一，若判断电机堵转(三条件同时满足)将会下使能；
- ⑨**堵转速度阈值**：判断电机堵转条件之一，若判断电机堵转(三条件同时满足)将会下使能；
- ⑩**堵转时间阈值**：判断电机堵转条件之一，若判断电机堵转(三条件同时满足)将会下使能；
- ⑪**屏蔽主编码器错误**：选择是否屏蔽主编码器报警，通常未接编码器线时勾选；
- ⑫**屏蔽辅编码器错误**：通常不使用全闭环(Dual-loop)功能时需要勾选；

第 6 步：硬件保护

Configuration --> Protections: Power and Current

本步骤用于设置电机电压电流等保护参数，防止电机损坏；

Protections: Power and Current

Current limitations and protections Current units ?

①	Continuous limitation:	1,100	mA
②	Peak limitation:	3,500	mA
③	Peak maximum time:	500	msec
④	Maximal phase current:	3,600	mA
⑤	Maximal allowed motor current:	3,600	mA
⑥	Maximal allowed power unit temperature:	65	°C
⑦	Motor temperature sensor (PT100):	<input type="checkbox"/> Connected	
⑧	Maximal allowed motor temperature:	80	°C
Bus voltage protections			
⑨	Minimal allowed bus voltage:	16,000	mV
⑩	Maximal timed bus voltage:	80,000	mV
⑪	Maximal time for over voltage:	50	msec
⑫	Absolute maximal allowed bus voltage:	80,000	mV
PWM limitations			
⑬	PWM limitations:	95	%

Recommended to save changes to Flash

- ①**持续电流**：电机运行过程中的持续电流，根据电机手册设置，注意 PC Suite 中的电流都是峰值（Peak）电流，若电机手册上是有效值(Arms)，则该处设定值为 Arms 的 $\sqrt{2}$ 倍；
- ②**峰值电流**：运动过程中的瞬时峰值电流，根据电机手册设置，注意 PC Suite 中的电流都是峰值，若电机手册上是有效值(Arms)，则该处值设置为 Arms 的 $\sqrt{2}$ 倍；
- ③**峰值持续时间**：最大瞬时峰值电流所允许的持续时间，典型值为 500ms；
- ④**最大相电流**：电机每相所允许的最大电流，当该电流大于此设定值超过 0.25ms，电机将会下使能，典型值为 $\text{Peak limitation} * [101\% \sim 105\%]$ ；
- ⑤**电机允许最大电流**：连续 4 次采样值大于此值时，控制器将会报警，电机下使能，典型值为 $\text{Peak limitation} * [101\% \sim 105\%]$ ；
- ⑥**功率单元允许的最高温度**：典型值为 65°C；
- ⑦**电机温度传感器**：勾选将启用温度传感器功能，支持 PT100、PT1000 类型温度传感器；
- ⑧**电机允许最高温度**：根据电机的耐温常数填写；
- ⑨**最小母线电压**：根据控制器输入电压设定，典型低压直流控制器为 16,000mV，单相交流 220V 为 90,000mV；
- ⑩**最大母线电压**：根据控制器输入电压设定，典型低压直流控制器为 90,000mV，单相交流 220V 为 360,000mV；
- ⑪**最长允许过压时间**：超过最大电压允许最大持续时间，典型值为 200ms；
- ⑫**最大允许电压绝对值**：典型低压直流控制器为 90,000mV，单相交流 220V 为 360,000mV；
- ⑬**PWM 限制**：可设置 PWM 输出功率限制，推荐值为 85~95%；

完成以上参数设置后即可进行 2.2 章节调参步骤，I/O 等相关功能设置可待电机调参正常运动后进行设置。

2.2 换相及调参

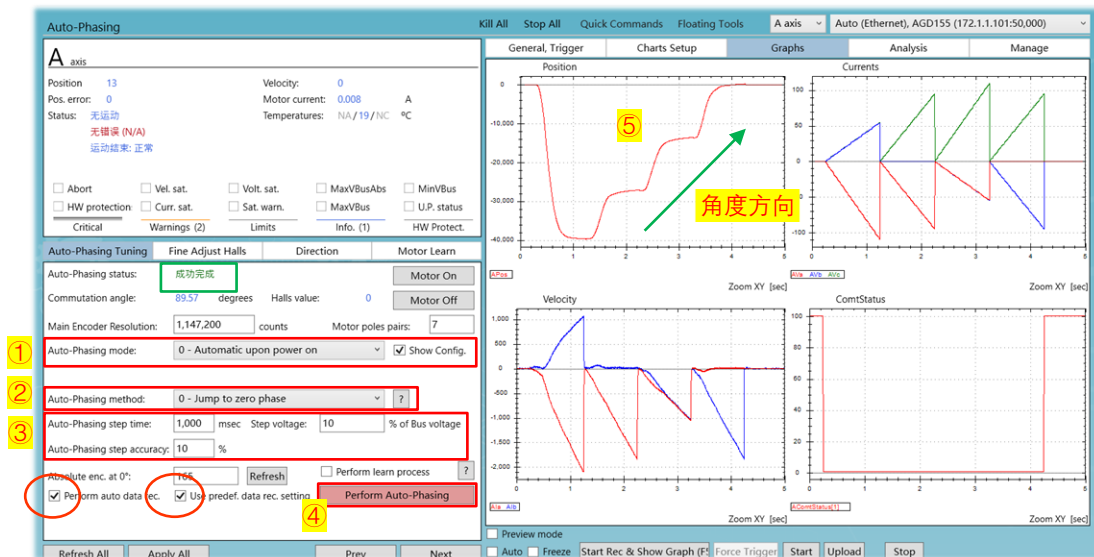
Agito 控制器支持多种换相方式，包括寻找电机电磁角度寻相和 Hall 换相，也支持手动调参和自动调参。本章节仅介绍典型换相及调参方式。

2.2.1 自动换相



(1) 跳转零位换相：除音圈、步进及有刷电机外，其他电机需要先进行换相，寻找电磁换相角。

TUNE ---> AutoPhasing



➤ ①换相模式：选择执行换相的触发方式，通常选择“0-Automatic upon power on”（上电自动换相），控制器在上电或重启时进行换相动作；

- | | |
|--|---------------|
| 0 - Automatic upon power on | 0: 上电自动换相 |
| 0 - Automatic upon power on | 1: 收到换相指令开始换相 |
| 1 - Only upon manual request | 2: 收到使能指令开始换相 |
| 2 - Automatic upon MotorOn (if needed) | |

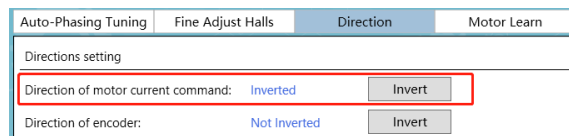
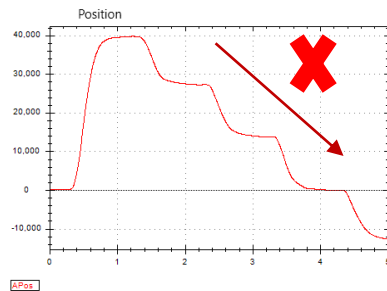
➤ ②寻相方式：选择寻找换相角的方式，典型方式为“0-Jump to zero phase”（跳转到相位零点）；

- | | |
|--|-----------------|
| 0 - Jump to zero phase | 0: 跳转到相位零点 |
| 0 - Jump to zero phase | 1: 手动换相 |
| 1 - Reserved | 2: 绝对编码器补偿 |
| 2 - Absolute encoder with predefined offset | 3: 带霍尔传感器的特殊编码器 |
| 3 - Special Encoder with Halls Encoder switching | 4: 带霍尔传感器编码器 |
| 4 - Encoder with Halls/Encoder switching | 5: 微动跳转搜寻 |
| 5 - Minimal jumps search | |

➤ ③步进延时、电压、精度：推荐值分别为时间 1000ms、电压 10[%]起（交流驱动器推荐从 2%开始）、精度 10[%]起；

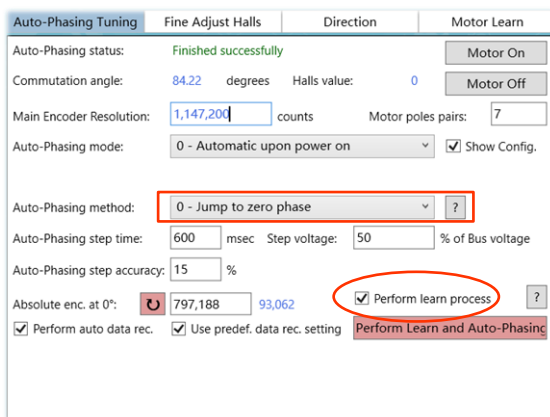
- ④执行换相：点击“Perform Auto-Phasing”开始进行自动换相；
- ⑤换相方向：换相跳转方向必须是正方向，反方向将会换相失败，可以将电流取反，此时运动方向将会改变为正向（如下图所示）。

（在换相成功的情况下，如用户需要将编码器方向取反，则此时电流方向也要同步与之前相反，或交换电机 U、W 动力线）；

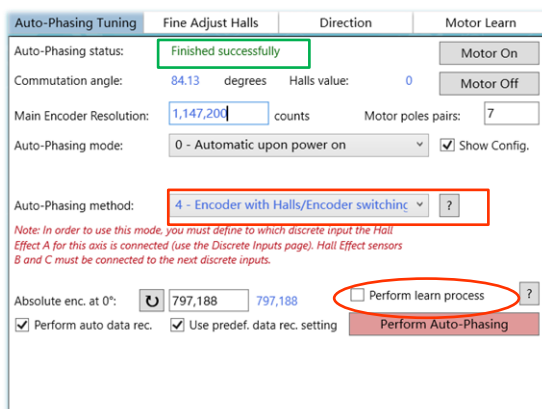


(2) Hall 换相：如运动系统要求在上电或重启时不允许进行任何运动，此时需要使用 Hall 换相。使用 Hall 换相前，请确保 Hall 信号线已正确连接并且在数字 I/O 中正确配置（请参考本文 2.4.3 章节内容）。值得注意的是在进行 Hall 学习前请先使用（1）中跳转到相位零点的方式进行换相，确保电机可以正常进行换相动作。

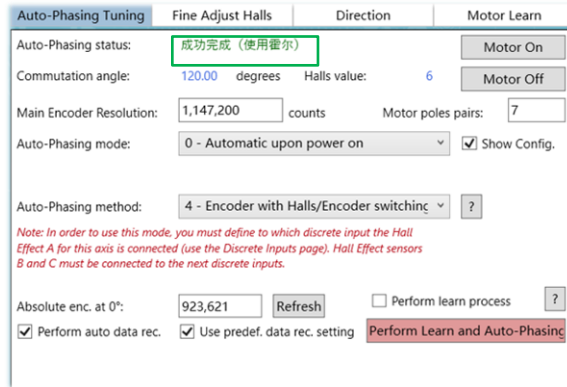
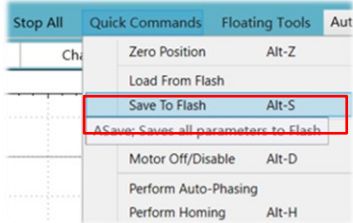
- 在使用“0-Jump to zero phase”换相成功后，先勾选图示“Perform learn process”，再点击“Perform Auto-Phasing”进行 Hall 换相学习；



- 换相学习成功后将 Auto-Phasing method 更改为“4-Encoder with Halls/Encoder switching”，并点击“Perform learn process”复选框将勾选去除；

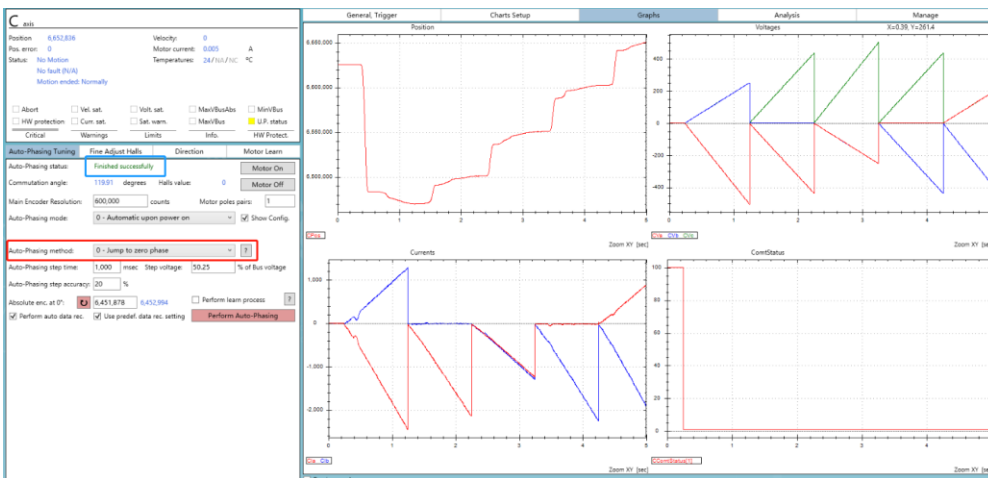


- 点击“Save to flash”保存参数到控制器，重新上电连接通讯后，可以看到提示使用霍尔换相成功。

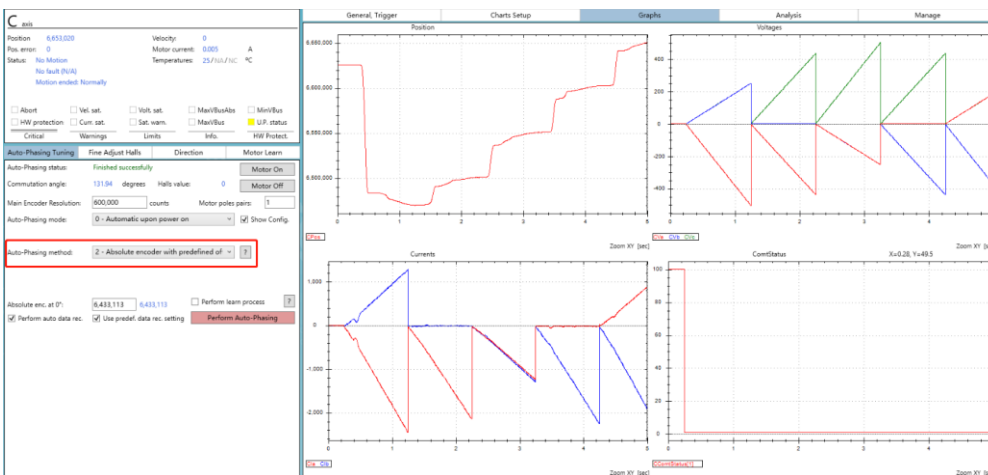


(3) 绝对值编码器换相:

- 绝对值编码器换相换相方式先按照(1)方式选择“0-Jump to zero phase”换相,



- 换相成功后，将换相方式切换为“2 - Absolute encoder ...”，并保存到控制器（Save to flash），下次重新上电就会自动使用已保存的换相角进行换相；



2.2.2 PIV 调参

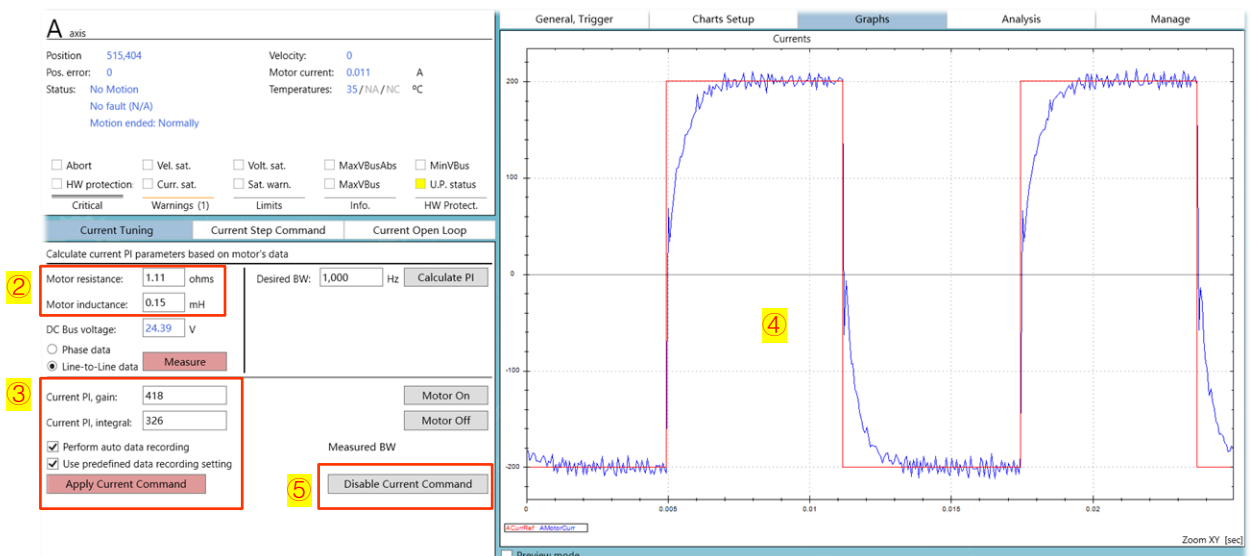
为了达到优良的控制效果，在进行控制电机运动之前，需要对系统的 PIV 参数进行调试。在调试 PIV 参数时，Agito PCSuite 上位软件提供多种函数发生器，包括方波、正弦、阶跃等多种激励源，在调参时可使用示波器功能实时自动抓取给定参数和反馈数据波形进行响应分析。如 PIV 参数设置不当，可能出现电机抖动或异常等不良情况产生。



第 1 步：电流环



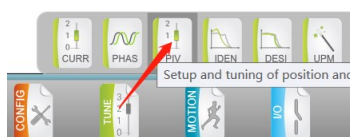
TUNE ---> Current Control

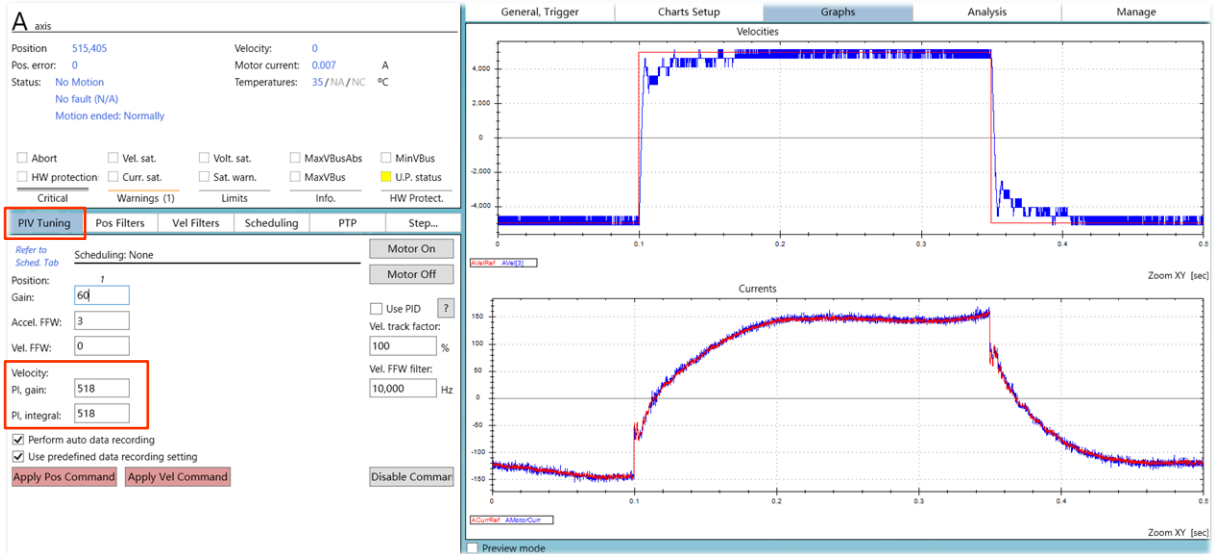


- ① 点击菜单栏切换到“Current Control”页面；
- ② 按照电机手册查询所连接电机相电阻、相电感，并分别填入上图②对应处；
- ③ 分别输入电流环 PI 参数比例增益和积分，Current PI gain 和 integral 推荐数值分别从 10、1 开始正向增大输入，以避免参数不合适电机产生啸叫声，点击“Apply Current Command”开始注入激励电流并响应。（注：通过滚动鼠标滚轮并同时点击键盘回车键可以实时更改 PI 参数并产生响应）。勾选“Perform auto data recording”表示连续自动抓取波形，勾选“Use predefined data recording setting”表示按照预定义参数进行抓图，通常情况下这两项均勾选；
- ④ 在调整参数的同时观察右侧采集到的电流指令（红：CurrRef）和电流反馈（蓝：MotorCurr）的实时波形，使两者最大程度重合（即使反馈电流跟随上指令电流）；
- ⑤ 点击“Disable Current Command”停止给入电流指令。

第 2 步：速度环

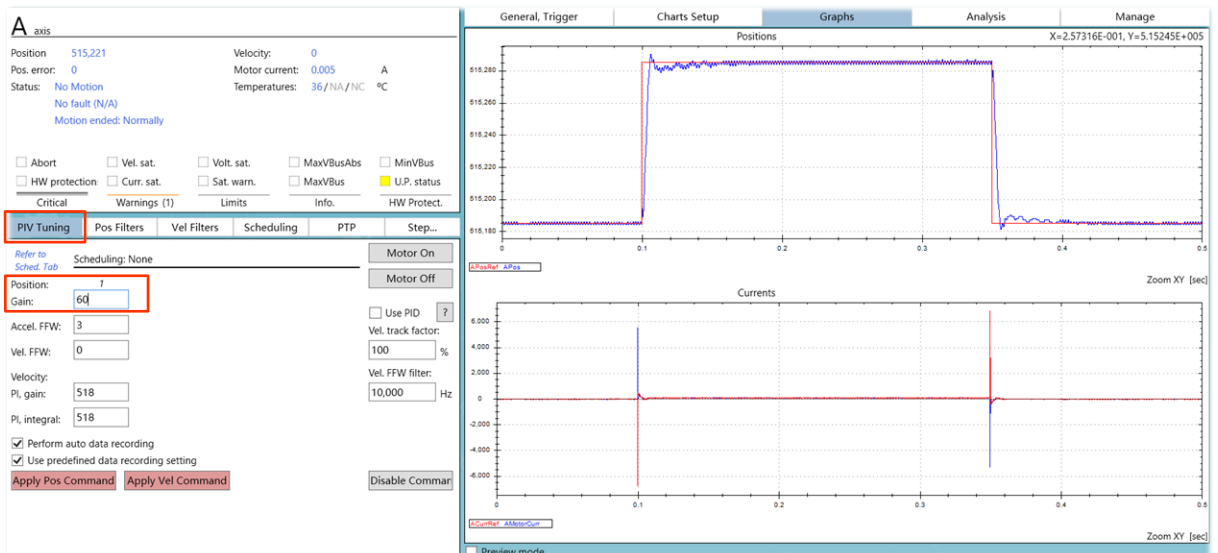
TUNE ---> PIV Tuning





- ① 点击菜单栏切换到“PIV Tuning”页面；
- ② 分别输入速度环 PI 参数比例增益和积分，Velocity PI gain 和 PI integral 推荐值分别从 10、1 开始正向增大输入，以避免参数不合适电机产生啸叫声，点击“Apply Vel Command”开始注入激励并响应；
- ③ 在调整参数的同时观察右侧采集到的速度指令红：VelRef）和速度反馈（蓝：Vel[3]）的实时波形，使两者最大程度重合（即使反馈速度跟随上指令速度）。
- ⑤ 点击“Disable Command”停止给入速度指令。

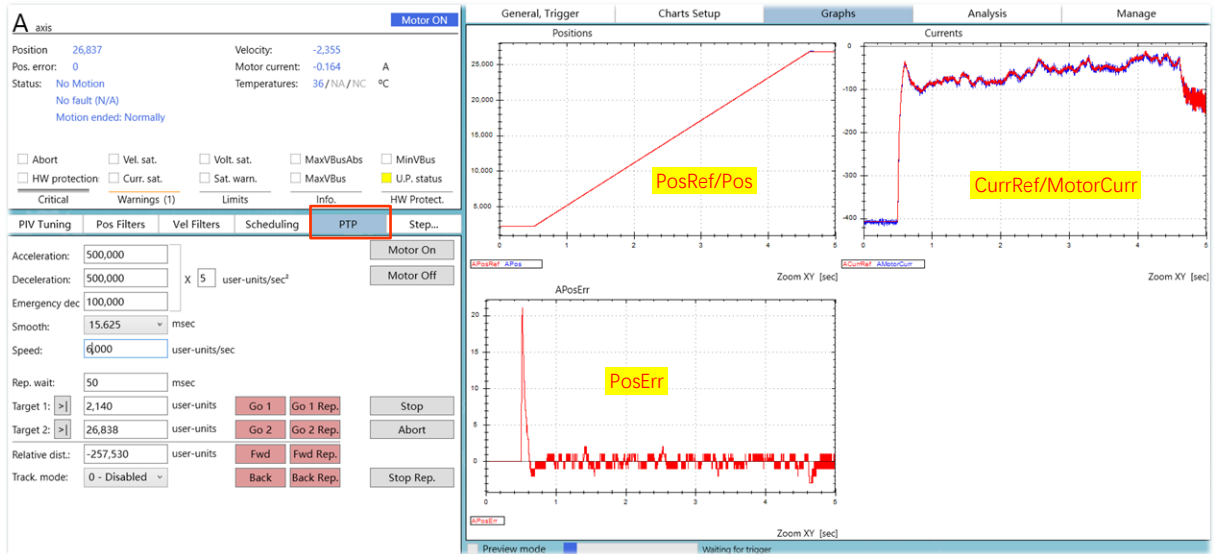
■ 第 3 步：位置环



TUNE ---> PIV Tuning

- ① 在位置环调试界面输入位置环增益值（Position Gain），推荐数值从 5 开始正向递增，以避免参数不合适电机产生啸叫声，点击“Apply Pos Command”开始注入激励并响应；
- ② 在调整参数的同时观察右侧采集到的位置指令红：PosRef）和位置反馈（蓝：Pos）的实时波形，使两者最大程度重合（即使反馈位置跟随上指令位置）。
- ③ 点击“Disable Command”给入位置指令。

第 4 步：试运行



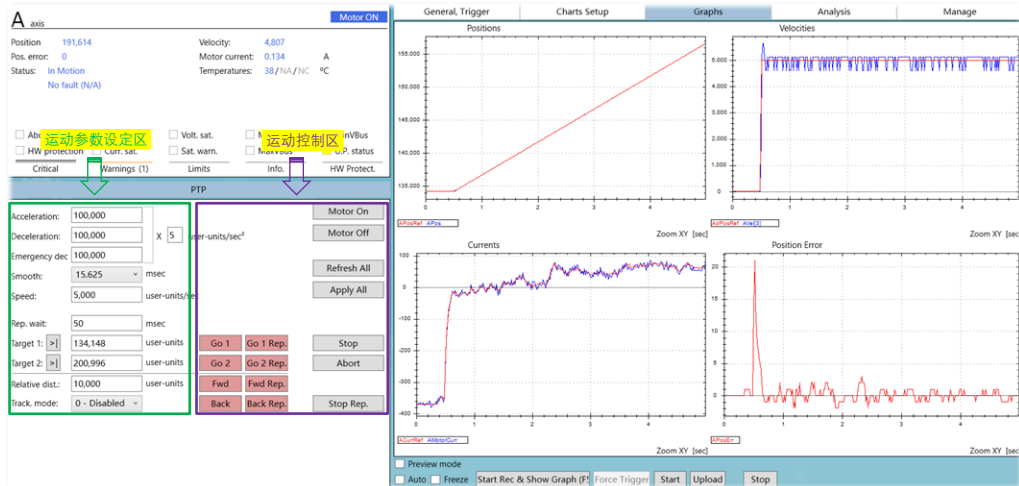
TUNE ---> PTP

- 从 PIV Tuning 界面切换到 PTP 运动界面，设置速度、加减速等运动相关参数，选取两个位置控制电机进行点到点运动，并使用示波器抓取运动过程中的 Pos 及 PosErr（位置跟随误差）等参数，确保电机的运动性能符合需求。若跟随误差过大可能需要调整 PIV 参数或者使用滤波器。

2.3 简单运动控制

本文档仅介绍使用 PC Suite 进行一些简单的运动操作，其他复杂功能（如插补、凸轮等）详见于各对应使用手册。

2.3.1 PTP 运动



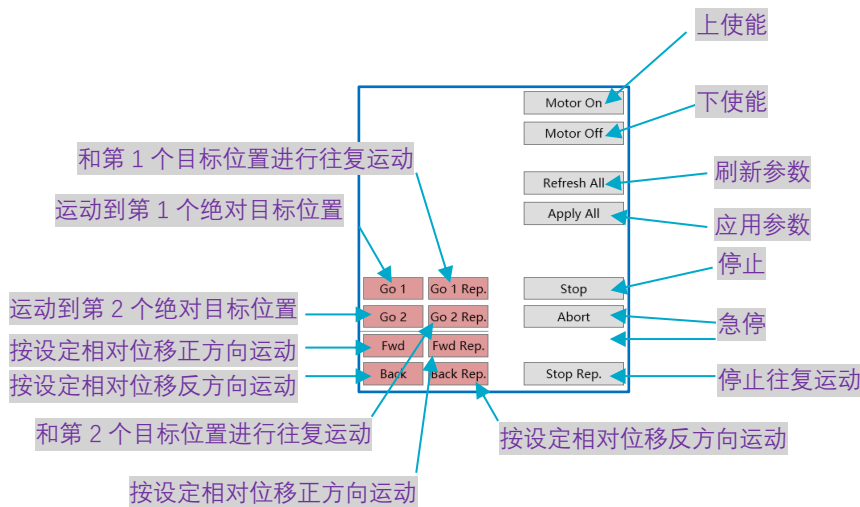
PTP 运动界面可以从多处调出，如 **TUNE--->PIV Control--->PTP**、**Motion--->Point to Point**、**TUNE--->Force Control--->Point to Point** 等界面均可以调出 PTP 运动界面，用户可根据使用习惯选择。

值得注意的是，运动过程中可以修改速度、加速度等参数并实时生效，但不能更改目标位置。

- **Acceleration:** 加速度；
- **Deceleration:** 减速度；
- **Emergency dec:** 急停减速度；
X□: 加速度、减速度、急停速度放大比例系数；
- **Smooth:** 加减速平滑；
- **Speed:** 速度；
- **Rep. wait:** 往复运动等待时间；
- **Target 1:** 第一个绝对目标位置，
点击右侧“>|”可以将当前位置设为 Target 1；
- **Target 2:** 第二个绝对目标位置；
点击右侧“>|”可以将当前位置设为 Target 2；
- **Relative dist:** 相对目标位移；

Acceleration:	100,000	
Deceleration:	100,000	X 5
Emergency dec:	100,000	
Smooth:	15.625	msec
Speed:	5,000	user-units/sec
Rep. wait:	50	msec
Target 1:	> 134,148	user-units
Target 2:	> 200,996	user-units
Relative dist.:	10,000	user-units
Track mode:	0 - Disabled	

- **Track. mode:** 跟随模式;



Go 1: 当鼠标点击“Go 1”时电机将运动到 Target 1 所设定位置，如下图所示;



Go 1 Rep: 当电机停止在某位置时（运动时禁止操作），点击“Go 1 Rep”将会在当前位置和 Target 1 之间按照设定的速度、等待时间进行往复运动，如下图所示;



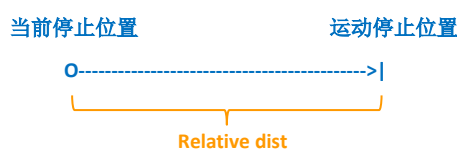
Go 2: 当鼠标点击“Go 2”时电机将运动到 Target 2 所设定位置，如下图所示;



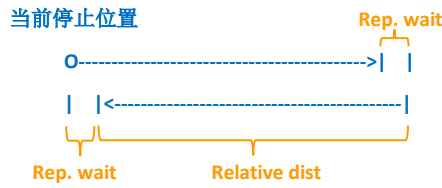
Go 2 Rep: 当电机停止在某位置时（运动时禁止操作），点击“Go 2 Rep”将会在当前位置和 Target 2 之间按照设定的速度、等待时间进行往复运动，如下图所示;



Fwd: 当鼠标点击“Fwd”时电机将会以相对当前位置按照设定 Relative dist 距离正方向执行单次 PTP 运动，如下图所示;



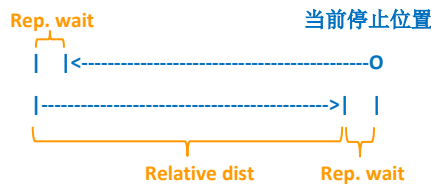
Fwd Rep.: 当鼠标点击“Fwd”时电机将会以相对当前位置按照设定 Relative dist 距离**正方向连续**执行等距 PTP 运动，如下图所示；



Back: 当鼠标点击“Fwd”时电机将会以相对当前位置按照设定 Relative dist 距离**正方向**执行单次 PTP 运动，如下图所示；



Back Rep.: 当鼠标点击“Back”时电机将会以相对当前位置按照设定 Relative dist 距离**反方向连续**执行等距 PTP 运动，如下图所示；



Motor On/Off: 电机上/下使能，上使能电机将保持在当前位置；

Refresh All: 刷新当前页面参数，当修改的参数**未保存**（蓝色字体）时，点击“Refresh All”参数将刷新为修改之前的参数；

Apply All: 应用当前修改的参数；

Stop: 按照设定的减速度停止运动；

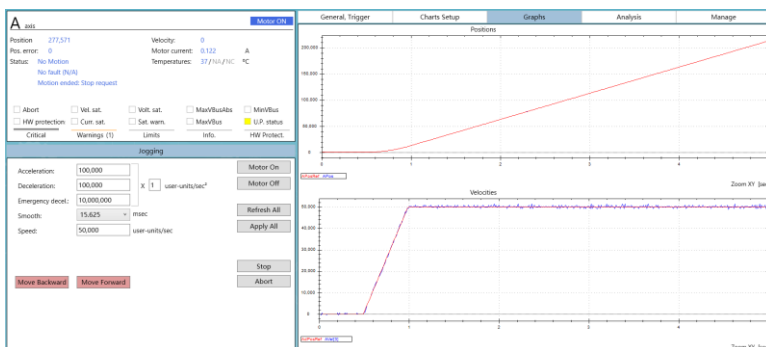
Abort: 将速度置 0 立即停止运动（电机保持使能状态）；

Stop Rep.: 停止往复运动。

2.3.2 Jog 运动



Motion ---> Jog



Move Backward: 从当前停止位置按照设定的速度向反方向连续运动;



当前停止位置

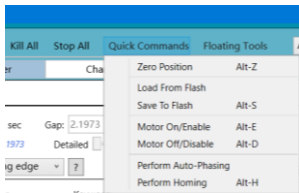
Move Forward: 从当前停止位置按照设定的速度向正方向连续运动;



当前停止位置

其他参数设定同 PTP 运动。

2.3.3 常用快捷指令



Kill All: 将所有轴下使能;

Stop All: 停止所有轴运动 (不会下使能);

Quick Commands → **Zero Position:** 将当前位置置 0;

Save To Flash: 保存参数到控制器;

Motor On/Enable: 当前轴上使能;

Motor Off/Disable: 当前轴下使能;

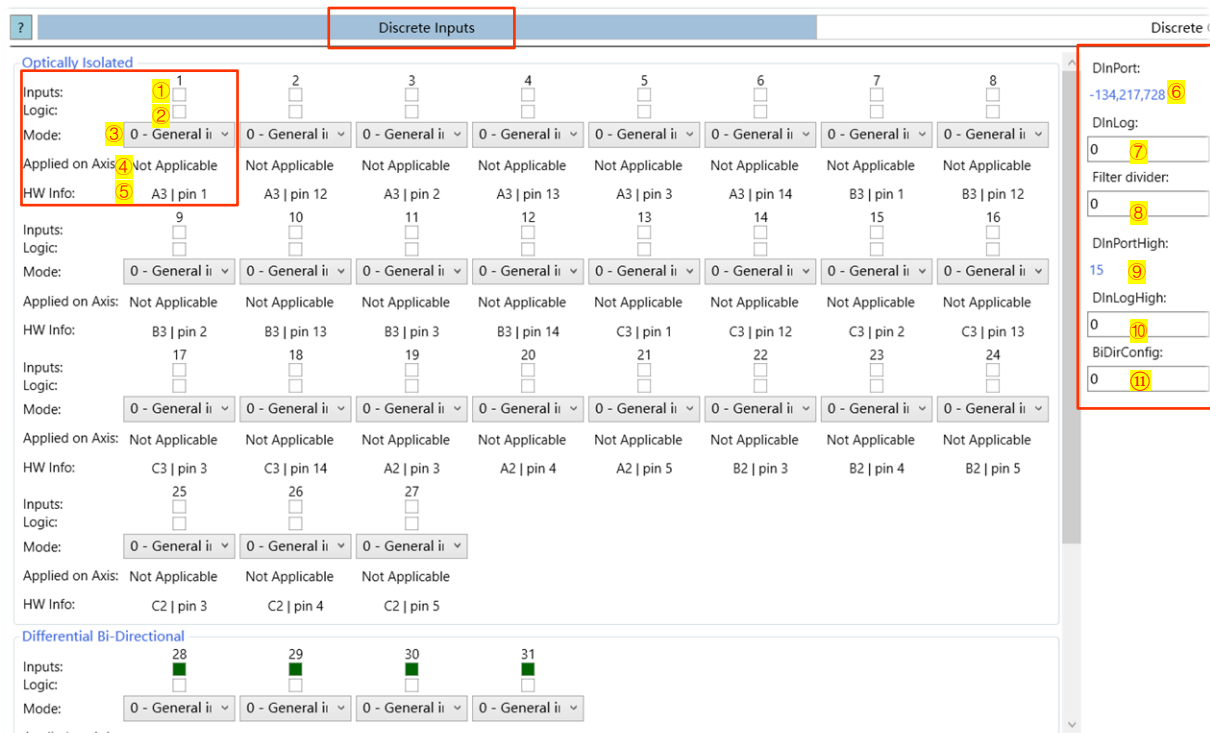
Perform Homing: 执行当前轴回零。

2.4 常用数字 I/O 功能配置



I/O ---> Discrete I/Os

- 数字 I/O 配置窗口可用于设定数字 I/O 功能和读取 I/O 状态；
- 该工具有两个选项卡：数字输入（Discrete Inputs）和数字输出（Discrete Outputs）；
- PCSuite 可从已连接的控制器中读取 I/O 数据，并反映控制器 I/O 的实际数量和状态；
- 本章节主要介绍几种常见的 I/O 功能应用，在使用 I/O 功能前先参阅各控制器对应的的用户手册（《Agito-AG□xxx-Product-Manual-Rev.xx》），了解各 I/O 口的电气特性，根据实际应用接线。
- 不同型号的控制器 I/O 数量及相关的电气特性存在差异，使用时请务必参阅对应产品使用手册，以下示例主要以 AGD301 为主，其他控制参数设置类似。
- 以下为数字 I/O 窗口简介（以 AGD301 为例，其他控制器可类比）：



- ①Inputs: 显示当前输入通道状态；
- ②Logic: 显示当前逻辑状态，点击 Logic 复选方框可以逻辑取反，对应 DinLog；
- ③Mode: 选择功能模式，[Axis_Index]DInMode[DInPort_Index]，bits 0~15；
- ④Applied on Axis: 选择该功能关联的轴号，[Axis_Index]DInMode[DInPort_Index]，bits 16~32；
- ⑤HW Infor: 指示该接口对应的硬件接口；
- ⑥DinPort: 显示当前所有数字输入通道状态，对应 Inputs；
- ⑦DinLog: 显示当前所有数字输入逻辑状态，对应 Logics；
- ⑧Filter divider: 数字输入滤波；

- ⑨DInPortHigh: 显示当前高速差分数字输入通道输入状态;
- ⑩DinLogHigh: 显示当前高速差分数字输入逻辑状态;
- ⑪BiDirConfig: 显示所有双向逻辑输入口配置状态 (配置为输入或输出)。

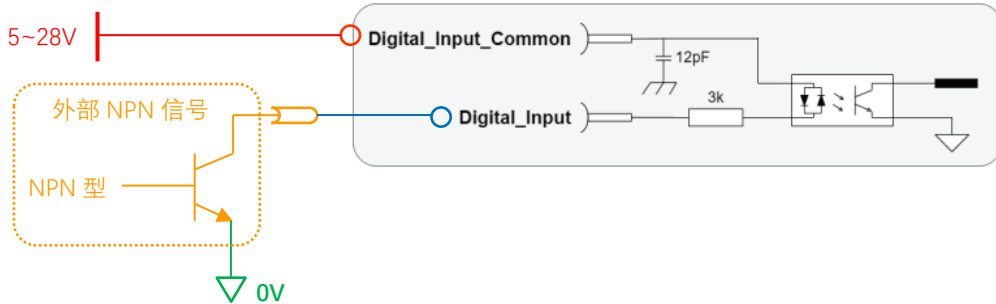
2.4.1 通用数字输入

以下以将 AGD301 Digital_Input_1 和 Bi-Dir_Diff_IO_1 设为通用数字输入口为例。

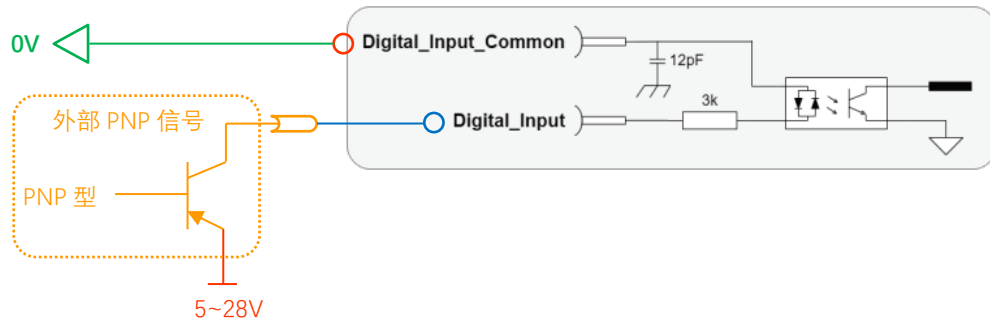
(1) 单端数字输入

▪ 第 1 步: 电气接线

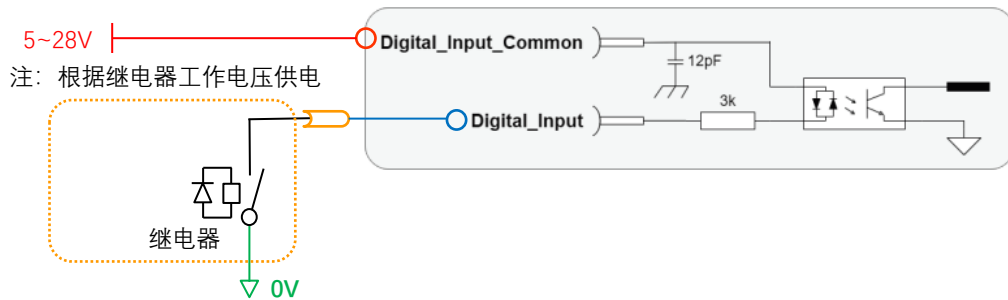
①外部 NPN 型 (外部 Sink 型):



②外部 PNP 型 (外部 Source 型):

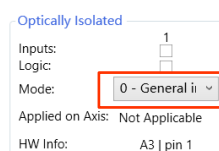


③外部继电器输出型:



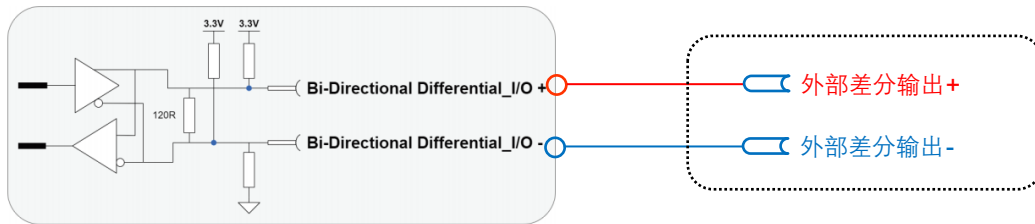
▪ 第 2 步: 设置对应数字输入口功能

切换到 Discrete I/Os 界面, 然后设置 Mode 为 “0-General input”;



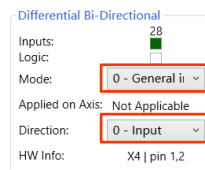
(2) 差分数字输入

第 1 步：电气接线



第 2 步：设置对应差分输入口功能

如是双向 I/O 先将 Direction 设为“0-Input”，然后设置 Mode 为“0-General input”。



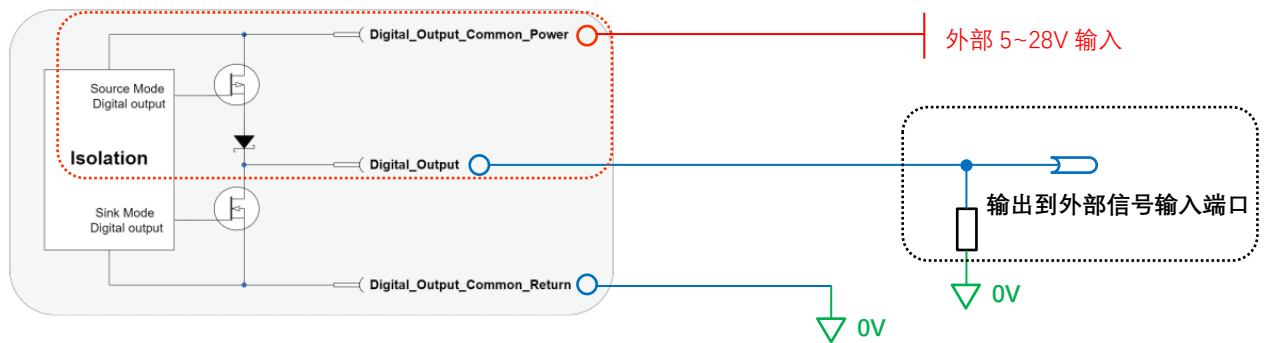
2.4.2 通用数字输出

以下以将 AGD301 Digital_Input_1 和 Bi-Dir_Diff_IO_1 设为通用数字输入口为例。

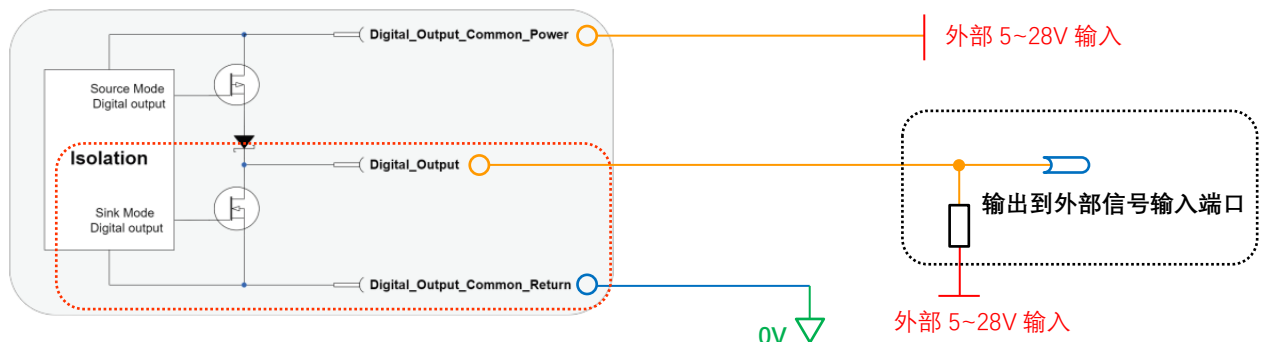
(1) 单端数字输出

第 1 步：电气接线

① Source 输出型（高电平有效）

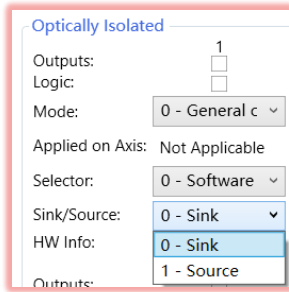


② Sink 输出型（低电平有效）



▪ **第 2 步：设置对应单端数字输出口功能**

切换到 Discrete I/Os ---> Discrete Output 界面，选择 Mode 为 “0-General output”，根据对应接线方式选择 “0-Sink” 或 “1-Source” 输出模式；



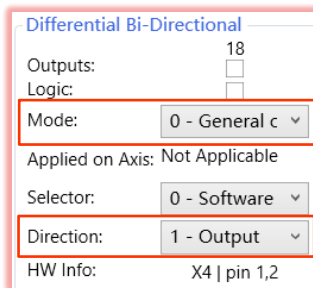
(2) 差分数字输出

▪ **第 1 步：电气接线**



▪ **第 2 步：设置对应差分输入口功能**

如是双向 I/O 先将 Direction 设为 “0-Output”，然后设置 Mode 为 “0-General output”。

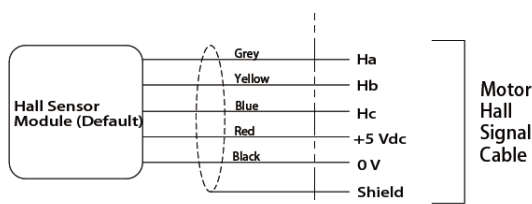


2.4.3 Hall 信号

部分型号控制器有预定义 Hall 接口（如 AGD301），有些控制器无预定义 Hall 接口，但是都支持 Hall 信号，即使是有预定义 Hall 接口普通数字输入信号也需要在数字 I/O 界面设置为 Hall 功能或将其他数字输入通道设为 Hall 信号接口。

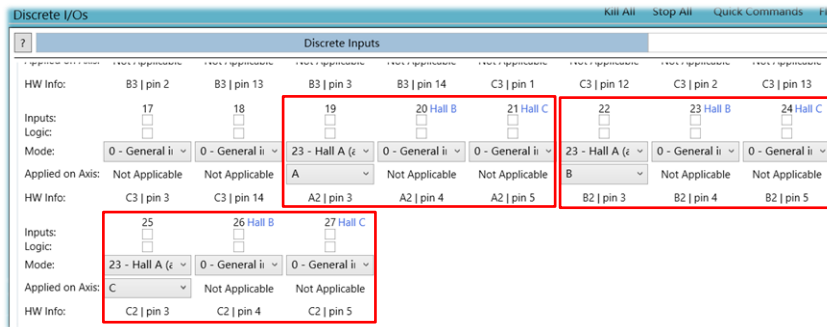
▪ **第 1 步：电气接线**

将 Hall 信号线按 Hall A、Hall B、Hall C 按顺序依次接到连续 3 个同一类型的单端数字输入口，5V、0V 分别接到控制器 I/O 口对应的 5V、GND 引脚上；



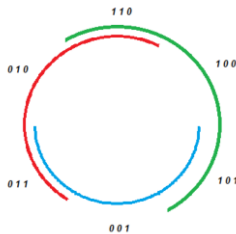
第 2 步：配置 I/O

将第 1 步 Hall 所接的数字输入 Mode 设置为“23-Hall A”，后两路将依次自动设为 Hall B、Hall C，并将关联轴（Applied on Axis）设置为对应的运动轴；



第 3 步：测试 Hall 信号

转动电机轴可以观察到 Hall A、Hall B、Hall C 信号依次连续变化（Hall 信号不存在同时为 000 和 111 的情况）。



第 4 步：进行 Hall 换相

换相详细步骤请参阅本文 2.2.1-(2) 章节，此处不再累述。

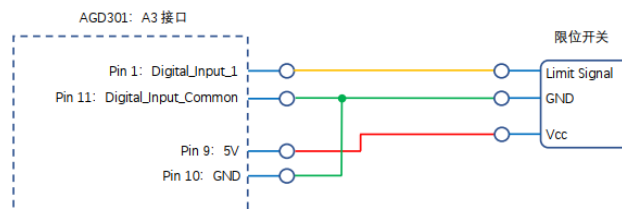
2.4.4 限位信号

限位信号是用于限制电机运动范围的一种物理信号，常用于直线运动系统和转角运动系统。当控制器检测到限位信号时电机将会停止运动（保持使能）。

以下以 AGD301 使用 Digital_Input_1 和 Digital_Input_2 分别设置负限位和正限位为示例。

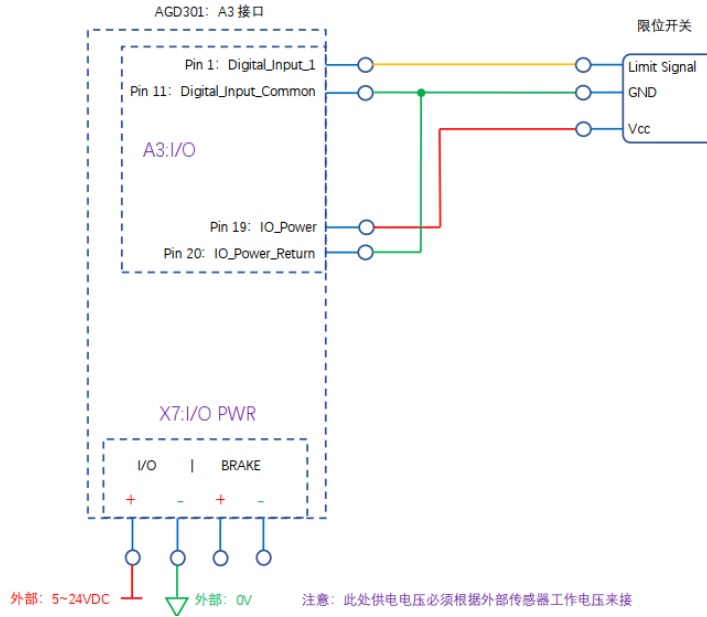
第 1 步：电气接线

(1) 方法一：使用驱动器内部 5V 输出供电；



(2) 方法二：使用驱动器外部逻辑电源供电（如开关电源）；

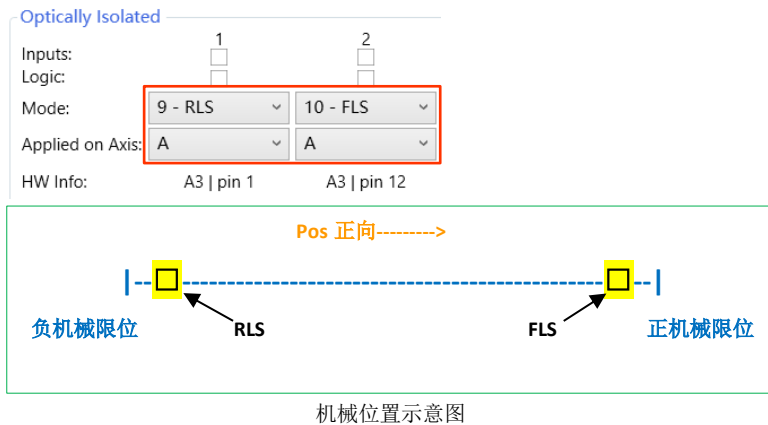
注意：供电电压必须根据外部传感器工作电压来接，X7 口 I/O 电源和其他数字信号接口 I/O PWR 内部连通。



第 2 步：配置 I/O

将对应数字输入口设置为限位信号，

负限位设为“9-RLS”，正限位设为“10-FLS”，值得注意的是正负限位需要和位置方向一致，并且应用轴必须和运动轴对应。



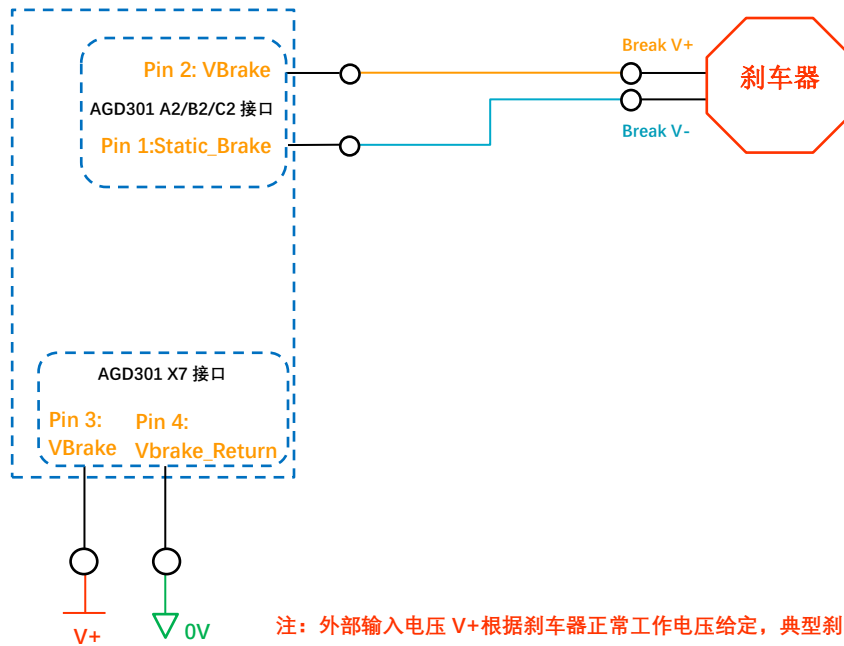
2.4.5 静态刹车

当用户使用外部抱闸时，需要使用数字输入口作为控制信号，部分型号控制器有专用静态刹车接口，当然也支持用户自定义数字输入口作为刹车控制信号。

以下以 AGD301 为例。

(1) 方式一：使用专用静态刹车接口作为静态刹车控制信号

第 1 步：电气接线



■ **第 2 步：配置信号及开启方式**

Config ---> Basic Configuration

- 切换菜单栏到 Basic Configuration 界面，
- 勾选 “Stactic brake” ；
- 选择开启刹车模式，典型的选择 “3-Automaic-By MotorOn State” ，即上使能自动松开抱闸，下使能自动锁紧；
- 输入松开/抱紧抱闸延时，典型值为 100ms。



(2) 方式二：绑定数字输入信号作为静态刹车控制信号

以下以 AGD301 绑定 Digital_Input_3 作为 A 轴抱闸输入信号为例，使用其他数字输入通道可类比；

■ **第 1 步：电气接线**

刹车及刹车电源部分接线同方法一完全相同，然后将 Digital_Input_3 按照普通 I/O sink 或 source 形式接入对应 AGD301 口；

■ **第 2 步：配置信号及开启方式**

Config ---> Basic Configuration

- 切换菜单栏到 Basic Configuration 界面；
- 勾选 “Stactic brake” ；
- 选择开启刹车模式为 “4-Automaic-By Discrete Input” ，即绑定隔离数字输入信号；

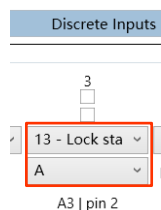
- 输入松开/抱紧抱闸延时，典型值为 100ms；



第 3 步：设置数字通道功能

Config ---> Discrete I/Os ---> Discrete Inputs

- 切换菜单栏到 Discrete Inputs 界面；
- 将 Digital_Input_3 模式设为 “13-Lock Static Brake”，将应用轴选为对应运动轴，比如 A 轴；



第 4 步：完成以上后即可通过 I/O 信号控制抱闸状态

2.4.6 回零开关

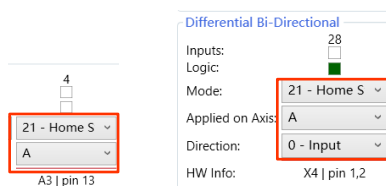
回零开关可用于回零点时的位置参考信号，和回零时的 index 信号类似（可参阅《Agito 回零使用手册》）。

第 1 步：电气接线

电气接线和普通数字输入信号（差分或单端）接线相同；

第 2 步：设置数字通道功能

将对应输入通道模式设为 “21-Home Switch”，将应用轴选为对应运动轴，比如 A 轴；



2.4.7 编码器分频输出

当用户需要使用将主编码器信号等效输出时，可以将差分数字输出设置为等效编码器信号输出通道。以下以 AGD301 控制器为例，其他型号可类比。

第 1 步：设置差分数字通道功能

- 切换菜单栏到 Discrete Outputs 下滑到 Differential Bi-Directional 界面；

Config ---> Discrete I/Os ---> Discrete Outputs

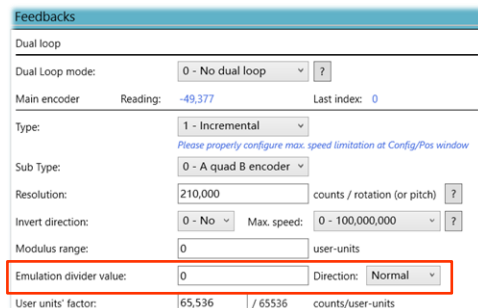
- 将 Direction 设为 “1-Output 模式”
- 按照下图设置对应轴的 Selector 为 “13-□ Enc.emul”；

Differential Bi-Directional										
Outputs:	<input type="checkbox"/> 18 HW	<input type="checkbox"/> 19 HW	<input type="checkbox"/> 20 HW	<input type="checkbox"/> 21 HW	<input type="checkbox"/> 22 HW	<input type="checkbox"/> 23 HW	<input type="checkbox"/> 24 HW	<input type="checkbox"/> 25 HW		
Logic:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Mode:	0 - General c	0 - General c	0 - General c	0 - General c	0 - General c	0 - General c	0 - General c	0 - General c		
Applied on Axis:	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable	Not Applicable		
Selector:	13 - A Enc. e	13 - A Enc. e	13 - B Enc. e	13 - B Enc. e	13 - C Enc. e	13 - C Enc. e	13 - A Enc. e	13 - B Enc. e		
Direction:	1 - Output	1 - Output	1 - Output	1 - Output	1 - Output	1 - Output	1 - Output	1 - Output		
HW Info:	X4 pin 1,2	X4 pin 19,20	X4 pin 3,4	X4 pin 21,22	X4 pin 25,26	X4 pin 9,10	X4 pin 27,28	X4 pin 29,30		
	A 轴编码器 A/B 信号		B 轴编码器 A/B 信号		C 轴编码器 A/B 信号		A 轴编码器 Z 信号	B 轴编码器 Z 信号		

第 2 步：设置分频输出比例因子

当 EmulRat=0 时，为 1:1 比例等效输出；

当 EmulRat > 2 时，按右式比例等效输出：

$$\text{emulation encoder output pulse} = \frac{\text{main encoder input pulse}}{\text{EmulRat}}$$


- 第 3 步：完成以上参数设置后，当主编码器口有脉冲信号输入时，差分输出口按上述设置等效输出脉冲信号；

2.4.8 PWM 输出

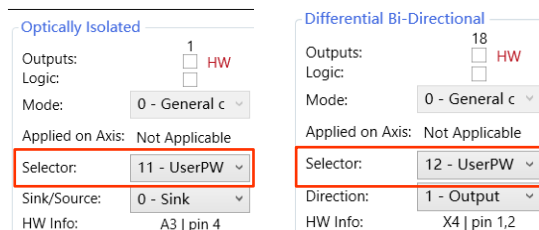
Agito 控制器数字输出通道可以按照一定频率和脉宽输出 PWM 脉冲，可用于控制一些加热器、灯光亮度等的应用。

以下以 AGD301 单端及差分输出 PWM 为例。

- 第 1 步：设置输出功能

Config ---> Discrete I/Os ---> Discrete Outputs

- 切换菜单栏到 Discrete Outputs，如是双向差分，先选择模式为“1-Output”模式；
- 选择 Selector 为“11-UserPWM 1”（第 1 路输出）或“12-UserPWM 2”（第 2 路输出）；



值得注意的是单端通道输出的为 0/5V 脉冲信号，差分通道输出 ±2.5V 脉冲信号，支持多通道同时输出 PWM 1 或 PWM 2 脉冲。

- 第 2 步：设置 PWM 参数

在数字输出界面右侧菜单栏设置 PWM 输出相关参数；

- PWM frequency: 选择 PWM 输出频率（不可自定义）；
- User PWM 1、User PWM 2: 设置 PWM 占空比；

Discrete Outputs

Refresh Outputs

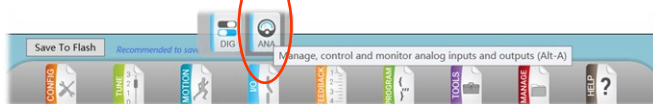
PWM frequency:
3 - 762.9 Hz

User PWM 1:
50 %

User PWM 2:
60 %

- 完成以上参数设置后，对应数字输出通道将开始输出 PWM 脉冲，用户可实时改变 PWM 占空比来改变输出功率。

2.5 模拟 I/O 配置



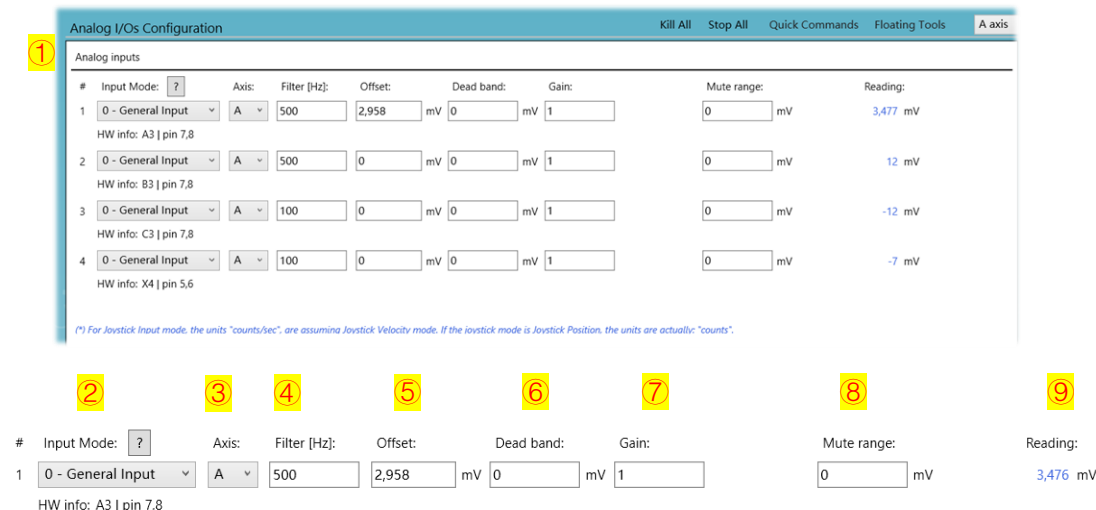
I/O ---> Analog I/Os

模拟 I/O 工具可用于定义模拟 I/O 属性和读取 I/O 状态，该工具有两个选项卡：模拟输入（Analog Inputs）和模拟输出（Analog Outputs），PCSuite 可从已连接的控制器中读取 I/O 数据，并反映控制器 I/O 的实际数量和状态。

模拟输入输出的数量因具体产品型号而异，在使用模拟 I/O 功能前先参阅各控制器对应的的用户手册（《Agito-AG□xxx-Product-Manual-Rev.xx》），充分了解各 I/O 口的电气特性，根据实际应用接线。

PCSuite 可检测数量并显示所连接控制器中模拟输入的实际数量。模拟输入属性（滤波器、偏移量、死区和增益）可以编辑。模拟输入的读数会像状态一样连续显示。

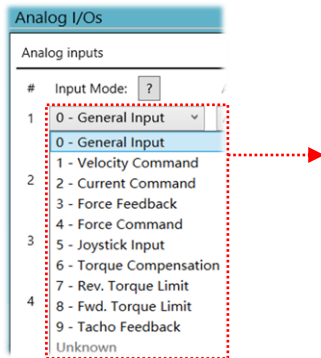
2.5.1 模拟输入口配置



#	Input Mode	Axis	Filter [Hz]	Offset	Dead band	Gain	Mute range	Reading
1	0 - General Input	A	500	2,958 mV	0 mV	1	0 mV	3,477 mV
HW info: A3 pin 7,8								
2	0 - General Input	A	500	0 mV	0 mV	1	0 mV	12 mV
HW info: B3 pin 7,8								
3	0 - General Input	A	100	0 mV	0 mV	1	0 mV	-12 mV
HW info: C3 pin 7,8								
4	0 - General Input	A	100	0 mV	0 mV	1	0 mV	-7 mV
HW info: X4 pin 5,6								

(*) For Joystick Input mode, the units "counts/sec", are assuming Joystick Velocity mode. If the joystick mode is Joystick Position, the units are actually "counts".

- ①选项栏：模拟量输入配置区，不同型号控制器模拟输入通道数量有差异；
- ②输入模式 (AInMode[]), bits 0~15：选择模拟输入信号控制功能模式；



模式编号	功能名称	常见应用
0	通用输入	读取模拟输入信号，±10V；
1	速度指令	速度模式控制信号
2	电流指令	电流(力矩)模式控制信号
3	力反馈	力传感器信号反馈，力控模式
4	力矩指令	力矩模式控制信号
5	摇杆输入	摇杆输入控制信号
6	力矩补偿	扭矩补偿，仅用于位置和速度模式
7	反向扭矩限制	反向扭矩反馈
8	正向扭矩限制	正向扭矩反馈
9	转速计反馈	转速计信号

- ③关联轴：选择模拟量控制的关联轴；
- ④滤波 (AInFilt[]) :模拟输入口软件滤波频率，典型值为 500Hz；
- ⑤偏置 (AInOffset[]) :模拟输入值偏移量，

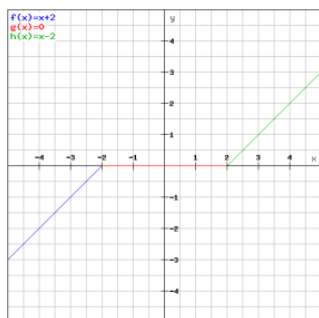
$$\text{AInPort Reading Value} = \text{AinPort Actual Value} + \text{AInOffset};$$

- ⑥死区 (AInDB[]) :模拟输入在 $[-\text{AInDB}, +\text{AInDB}]$ 范围内时读取值为 0；

$$\text{AInPort Reading Value} = \begin{cases} \text{AinPort Actual Value} - \text{AInDB}, & \text{当 } \text{AinPort Actual Value} \in (\text{AInDB}, +12000] \text{ 时;} \\ 0, & \text{当 } \text{AinPort Actual Value} \in [-\text{AInDB}, +\text{AInDB}] \text{ 时;} \\ \text{AinPort Actual Value} + \text{AInDB}, & \text{当 } \text{AinPort Actual Value} \in [-12000, -\text{AInDB}] \text{ 时;} \end{cases}$$

例如当 $\text{AInDB}[\text{AIn_Index}] = 2$ 时，实际读数如下图所示；

(图见下页)



- ⑦增益 (AInGain[]) :模拟输入信号增益，见⑨公式；
- ⑧蠕变范围 (AInMuteRange[]) :模拟输入信号变化量在此范围内将保持不变；
- ⑨模拟输入量 (AInPort[]) :

$$\text{AInPort Reading Value} = \text{Gain} * (\text{AIn Actual Value} + \text{AInOffset}),$$

同时该值也受 AInFilt, AInDB, AInOffset, AInMuteRange 同时影响，±10V, 12bit 或 16bit；

2.5.2 模拟输出口配置

Analog outputs (When an analog output is used to connect to an external analog amplifier, the settings here have no effect.)

Output number: HW info: A3 | pin 17 Analog Offset: mV

Direct command:

Or use for monitoring: Scaling:

Axis: Category: Keyword:

Source:

Refresh All Apply All Prev Next Apply and Next

- 模式（一）：直接电压指令输出模式

Analog outputs (When an analog output is used to connect to an external analog amp)

Output number: HW info: A3 | pin 17 Analog Offset: mV

Direct command: Out value: mV

Or use for monitoring:

- **Output number:** 选择模拟输出通道；
- **Direct command:** 选择输出模式为直接电压指令输出模式；
- **Analog Offset:** 设置输出偏移量；
- **Out Value:** 设置模拟电压输出值。

- 模式（二）：命令跟随模式

Analog outputs (When an analog output is used to connect to an external analog amp)

Output number: HW info: A3 | pin 17 Analog Offset: mV

Direct command:

Or use for monitoring: Scaling:

Axis: Category: Keyword:

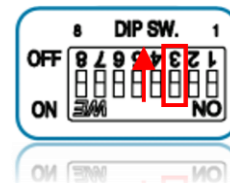
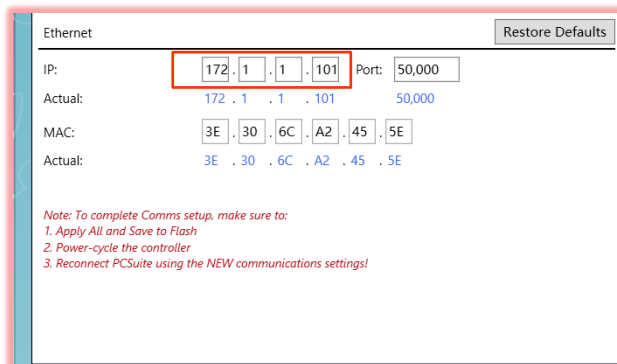
Source:

- **Output number:** 选择模拟输出通道；
- **Or use for monitoring:** 选择输出模式为命令跟随模式；
- **Analog Offset:** 设置输出偏移量；
- **Scaling:** 设置[跟随对象值] \propto [AOutPort]的比例因子，
比例系数=2^(Scaling Value)，即实际输出电压值=源指令值*比例系数=源指令值*2^(Scaling Value)；
- **Source:** 选择被跟随对象，即源指令。

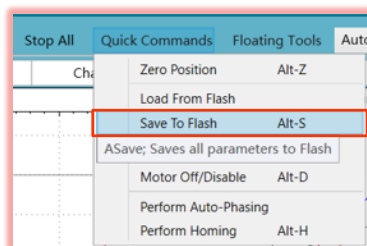
2.6 更改控制器 IP 地址



- 第 1 步：切换到菜单到 **Configuration ---> Communication**;
- 第 2 步：修改 IP 为所需的 IP 地址（值得注意的是修改 IP 地址前需要确认侧面 DIP 开关的 SW 3 是否是拨到 OFF 状态，如未拨到 OFF 需要在修改 IP 前将拨码拨到 OFF 状态，并重新上电重启控制器）；



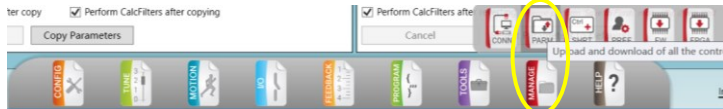
- 第 3 步：点击“Save to flash”保存参数到控制器，重新上电连接通讯后，可以看到 Communication 界面的 IP 变为更改后的 IP 地址；



3 PCSuite 常用工具介绍

Agito PCSuite 上位机软件提供丰富多彩的工具窗口供用户调试及分析问题使用，以下介绍几种常用的工具方便用户快速入门。

3.1 参数导入/导出

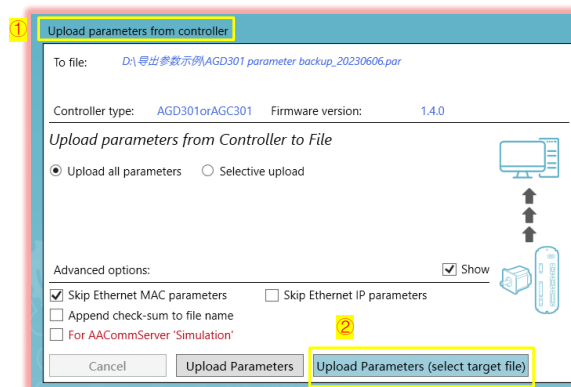


用户在使用过程中经常需要上传或者下载控制器参数，或者不同轴之间参数相互交换等操作，下文将介绍几种常用的参数导出/导出方式。

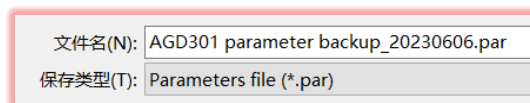
MANAGE--->Parameters

类型 1：导出所有参数到本地文件

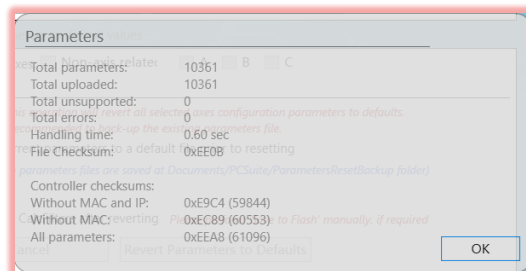
- 第 1 步：选择“Upload parameters from controller”窗口，点击“Upload parameters(select target file)”；



- 第 2 步：选择文件保存路径，输入文件名称，点击“保存”后数据将开始导出参数到本地所选择目录，文件为.par 格式；



- 第 3 步：保存成功后，将会弹出以下提示框，点击“OK”；

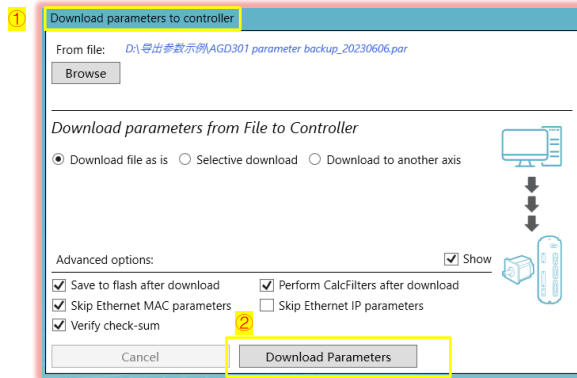


※值得注意的是导出参数前请点击“Save to flash”，确保在导出参数前设定参数均已保存到控制器内存中。

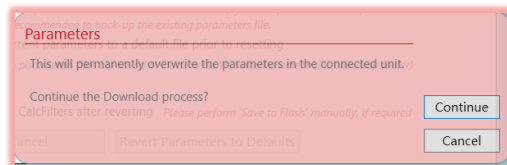
■ **类型 2：将本地文件所有参数导入到控制器中**

主要应用于参数备份和恢复使用，或多台相同配置控制器参数克隆

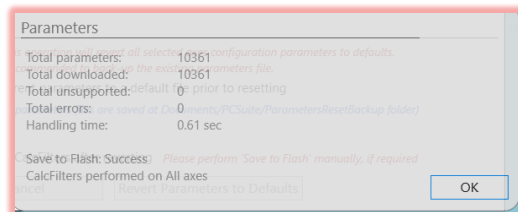
- 第 1 步：选择“Download parameters to controller”窗口，点击“Download parameters”；



- 第 2 步：将弹出如下对话框，提示控制器原本参数将会被覆盖（无法撤销回原来参数），点击“Continue”；（注：如不希望将下载的参数保存到控制器中，可去掉“Save to flash after download”勾选）；



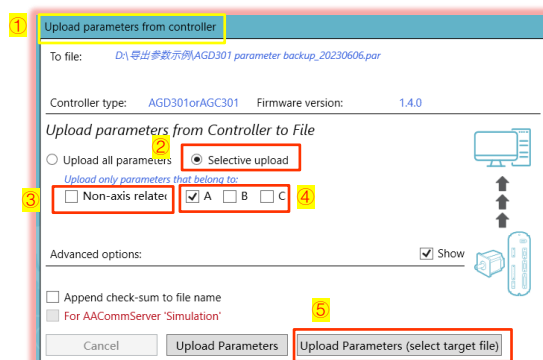
- 第 3 步：参数导入成功后，将弹出如下对话框，点击“OK”，参数导入完成。



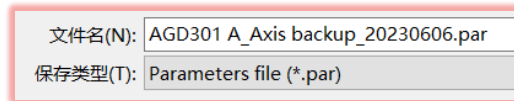
■ **类型 3：导出部分轴参数（应用于多轴控制器），如仅导出 A 轴**

主要应用于交换不同轴参数，如交换 A、B 两轴电机参数等操作

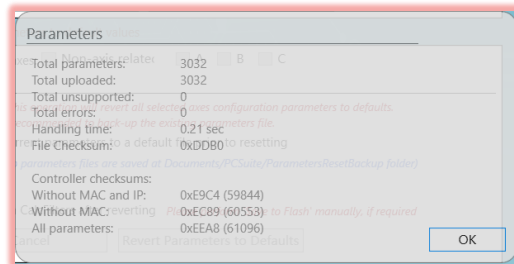
- 第 1 步：选择“Upload parameters from controller”窗口，①选择导出参数方式为“Selective upload”，②选择是否包含轴无关参数（如 DInPort[]等），③选择运动轴；



- 第 2 步：④点击“Upload Parameters(select target file)”，选择文件保存路径，输入文件名称，点击“保存”后数据将开始导出参数到本地所选择目录，文件后缀为.par 格式；

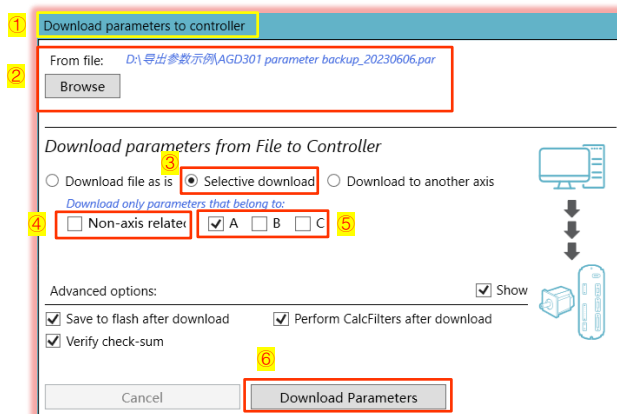


- 第 3 步：参数导入成功后，将弹出如下对话框，点击“OK”，参数导入完成。

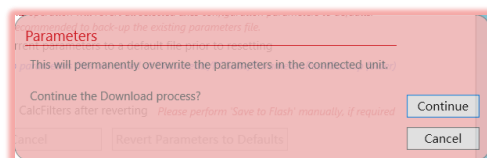


■ 类型 4：从本地备份参数文件（文件需包含该导入轴）中导入部分轴参数到控制器

- 第 1 步：①选择“Download parameters to controller”窗口，②点击“Browse”选择本地备份参数文件，③选择导出参数方式为“Selective download”，④选择是否包含轴无关参数（如 DIInPort[]等），⑤选择需要导入参数的运动轴，如 A 轴；

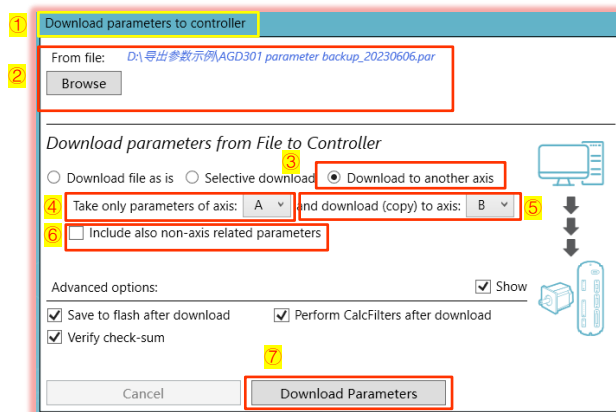


- 第 2 步：⑥点击“Download Parameters”，将会弹出以下对话框，点击“Continue”将开始下载参数，本地备份参数文件中的 A 轴参数将会下载到控制器中并保存。

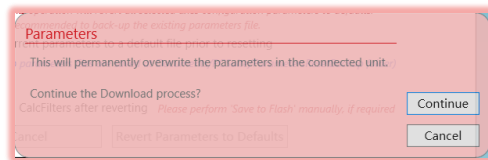


■ 类型 5：将本地备份参数文件中其中一轴参数导入到控制器另一轴中，如将备份参数文件中的 A 轴参数导入下载到控制器 B 轴中

- 第 1 步：①选择“Download parameters to controller”窗口，②点击“Browse”选择本地备份参数文件，③选择导出参数方式为“Download to another axis”，④选择文件中对应轴参数，如 A 轴，⑤选择需要导入参数的运动轴，如 B 轴，⑥选择是否包含轴无关参数（如 DIInPort[]等）；

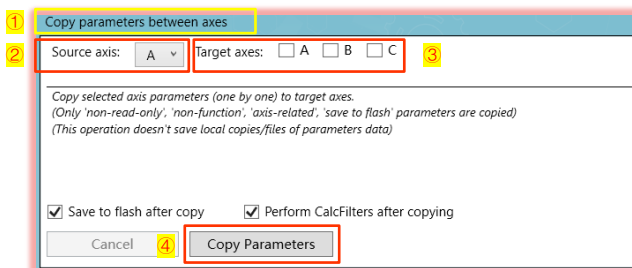


- 第 2 步：⑦点击“Download Parameters”，将会弹出以下对话框，点击“Continue”将开始下载参数，本地备份参数文件中的 A 轴参数将会下载控制器 B 轴中并保存。

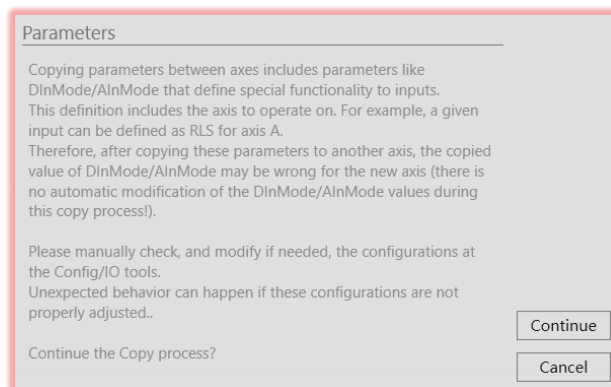


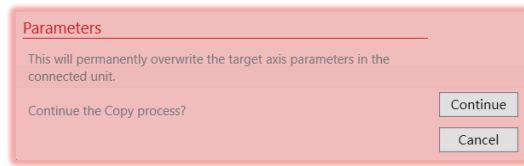
类型 6：同一控制器不同轴之间参数复制（克隆）

- 第 1 步：①选择“Copy parameters between axes”窗口，②选择被复制参数轴号，③选择目标轴轴号；



- 第 2 步：④点击“Copy Parameters”将会弹出以下对话框，依次点击“Continue”将开始下载参数；





3.2 示波器采集图像

示波器采集图像工具是一个嵌入在 PCSuite 大多数工具页面中都包含的图形查看器窗格，如 Auto-Phasing、PIV Control、Point to Point 等界面会自动调出示波器图像采集界面，也可以从悬浮工具（Floating Tools）中“Data Recording”选项调出。

每个图形查看器窗格都可根据当前使用的工具显示数据。

图形查看器窗格有五个选项卡：常规/触发、图表设置、图形、分析、管理。



强烈建议您掌握示波器采集图像工具的使用方法，该功能可显著提高您使用 PCSuite 进行调试和分析问题的效率。

示波器采集数据一般工作流程如下：

- 第 1 步：设置数据采集持续时间、采样间隔和触发条件、触发起始位置等；
- 第 2 步：设置采集参数（最大同时 8 通道）、图表（最大 6 张图表）、颜色等；
- 第 3 步：执行数据记录（自动触发或手动触发）；
- 第 4 步：分析结果（可对采集到的图像进行放大、运算、对比等操作）。



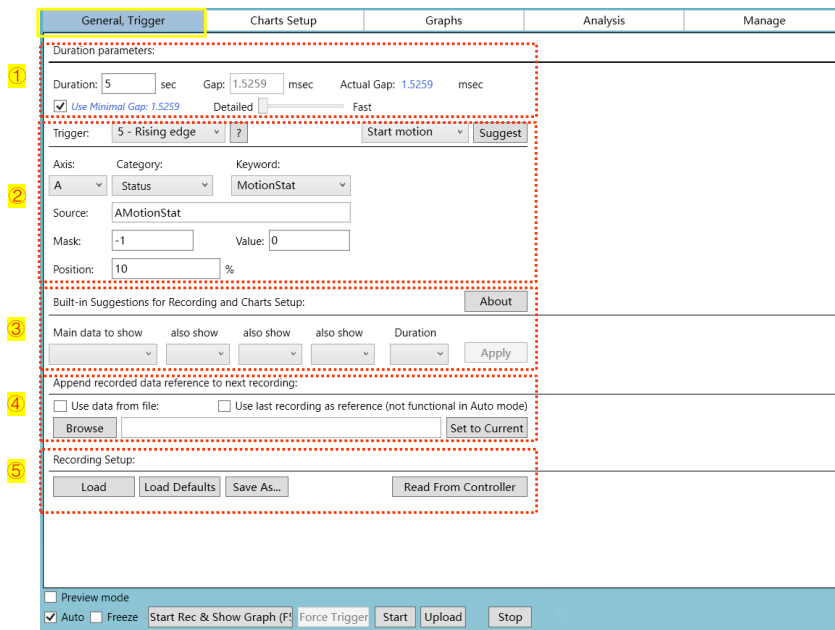
使用过程中需注意以下规则：

- 1. 最小采样间隔：0.061msec，最大记录点数：16500，设置超过该采样点数将会弹出相应错误信息；
- 2. 最大记录通道数：8 通道，即同时最多可以记录 8 个参数；
- 3. 数组参数在使用时需要输入数组元素索引；
- 4. 当多参数放置于同一表格中时注意两者值数量级，如相差太大在图表中不宜直接观测，建议加入比例因子 factor 进行缩放；
- 5. 默认 X 轴为时间；
- 6. v13.6.0-4.0 及以上版本 PCSuite 支持以用户自定义单位显示。

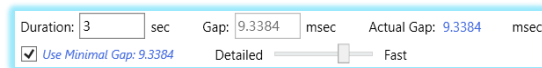
3.2.1 常规/触发选项卡

常规/触发选项卡可设置数据采集持续时间、采样间隔和触发条件等参数。该选项卡也可以加载并显示之前记录的数据，从而覆盖当前显示的数据，或者和当前数据做比对。

此步骤为关键步骤，触发条件参数设置不合理可能会导致无法自动触发数据采集。



①周期参数（Duration Parameters）

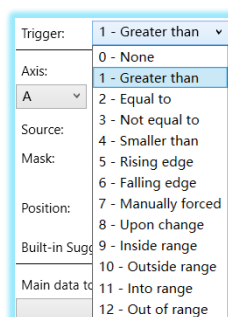


- **持续时间（Duration）**：数据采集总记录时长，单位：s；
- **采样间隔（Gap）**：两个连续的记录点之间的间隔，单位：ms，最小为 0.061ms；当勾选 Use Minimal Gap 时，可通过拖拽条设置 Gap；

②触发参数

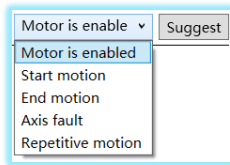
- **触发条件（Trigger）**：设置开始记录数据时的触发器类型；

注：对于下表的作用域，X 指作用于触发器的参数值（关键字：RecTrigSrc）



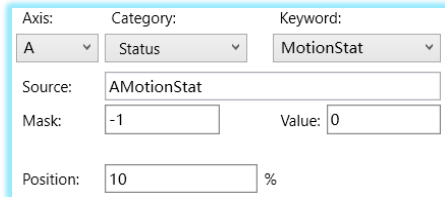
编号	触发器类型	描述
0	无	无需等待触发条件；
1	>	当 $X > \text{RecTrigVal}$ 时；
2	=	当 $X = \text{RecTrigVal}$ 时；
3	≠	当 $X \neq \text{RecTrigVal}$ 时；
4	<	当 $X < \text{RecTrigVal}$ 时；
5	↑	当 $X \leq \text{RecTrigVal}$ 变为 $> \text{RecTrigVal}$ 时；
6	↓	当 $X \geq \text{RecTrigVal}$ 变为 $< \text{RecTrigVal}$ 时；
7	手动	手动点击“Start Rec.”时；
8	变化	当 X 开始变化时；
9	范围内	当 $X \in [\text{RecTrigVal}, \text{RecTrigValMax}]$ ，即在设定范围内时；
10	范围外	当 $X \notin [\text{RecTrigVal}, \text{RecTrigValMax}]$ ，即在设定范围外时；
11	进入范围	当 X 进入 $[\text{RecTrigVal}, \text{RecTrigValMax}]$ 范围内时；
12	超出范围	当 X 超出 $[\text{RecTrigVal}, \text{RecTrigValMax}]$ 范围外时；

- **常用推荐触发方式（Suggest）**：此处将会预定义一个触发条件，选择所需触发方式后，点击“Suggest”将会按照预定义参数自动填入与之相关触发条件；



预定义触发方式	描述
Motor is enabled	电机使能时;
Start motion	电机开始运动时;
End motion	电机停止运动时;
Axis fault	轴报错时;
Repetitive motion	电机往复运动时;

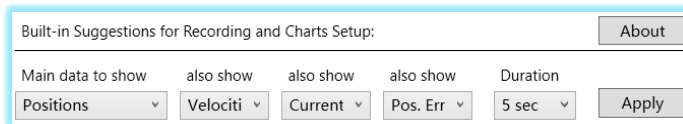
- **Axis、Category、Keyword、Value:** 设置作用于触发器的参数值及关联轴，可选择所有变量；



- **掩码 (Mask):** 在判断触发条件前，与触发器的参数值进行按位“&”运算。这是一种对特定位置进行触发的（不常用）方式，默认值为-1，表示不使用掩码；
- **位置 (Position):** 定义触发事件在整个记录时间内的位置（以百分比表示），以便记录和查看触发之前的数据。

③内置常用推荐触发条件 (Built-in Suggestions ...)

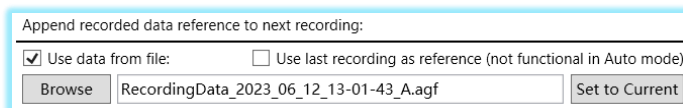
帮助您自动设置参数记录表，最多可选择 4 张图表，依次选择对应关键字、持续时间等记录参数后，点击“Apply”自动将按照预定义参数填入表格设置选项卡 (Charts Setup) 中；



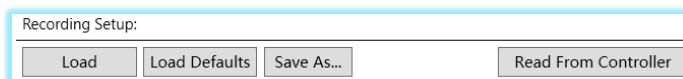
④将之前记录的某数据引用到下一个记录的数据图表中作参考

Browse: 选择本地文件路径；

Set to Current: 将本地文件中的参数设置设为当前；

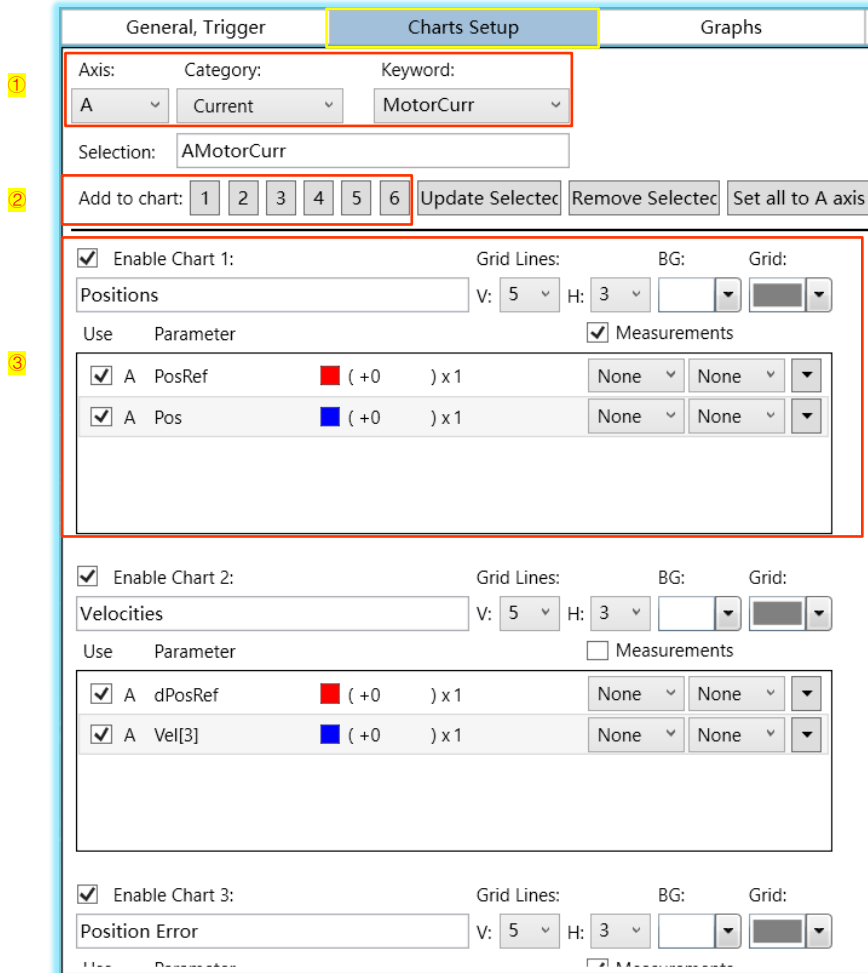


⑤导入/导出触发参数设置

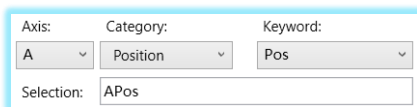


3.2.2 图表设置选项卡

图表设置选项卡可配置待记录的控制器参数、图表数量以及各种图表和显示选项。选择完需要需要监控和参数后即可点击底部“Start Rec & Show Graph”开始采集数据。

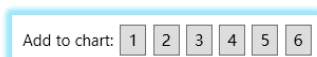


①选择待记录的参数关键字



- **Axis:** 关联轴;
- **Category、Keyword:** 常见关键字可从关键字所属类里快速选择, 非常用可从“ALL”中选择;

②选择图表

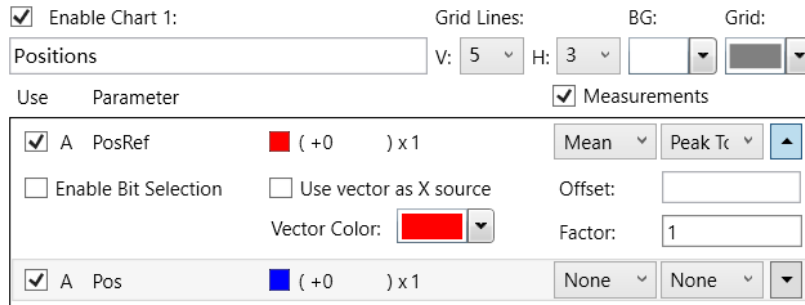


选择待记录参数的放置图表, 同一图表可放置多个参数, 数字表示第几张图表, 值得注意的是尽管同一图表中的参数放置最多可以放置 8 个, 但是受限于总的通道数 ≤ 8 , 当使用多个图表时, 所有图表的参数总数和必须 ≤ 8 ;

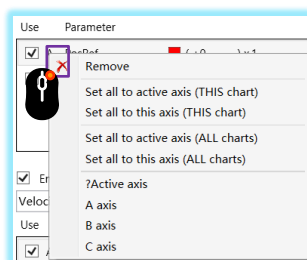
点击对应表格编号, ①中选择的变量将会被放置于对应的图表设置窗中;

③设置图表参数

从②中选择的待记录参数，在③窗口可以设置详细的图表参数，以下以 A 轴 PosRef 为示例，其他参数类比：

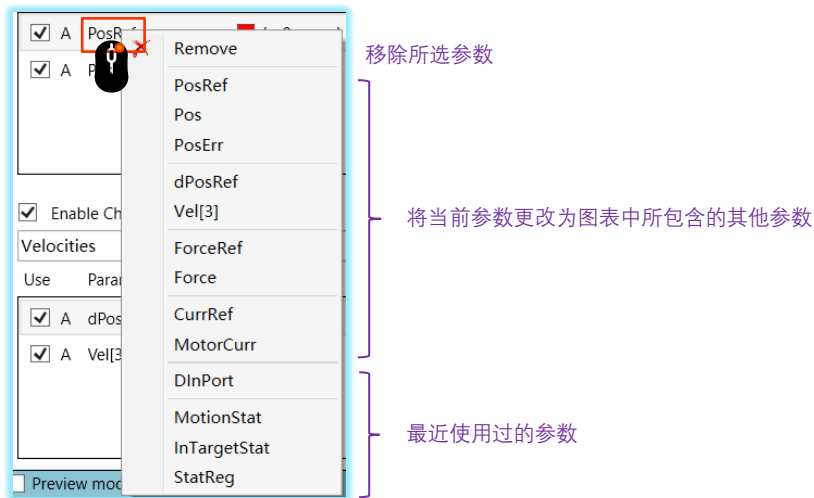


- **Enable Chart X**: 表示启用当前图表记录，下文本框中可自定义表名称；
- **Grid Lines**: **V**: 纵轴，**H**: 横轴；
- BG**: 图表背景色；
- Grid**: 网格颜色；
- 点击右侧 “ **A PosRef**: 勾选表示启用当前参数记录；
- Measurements**: 勾选表示启用测量工具，每张图表最多可选择 2 项，如示例图中选择 “Mean”（平均值）、“Peak to Peak”（峰-峰值），默认不启用；
- Enable Bit Selection**: 勾选将展开具体的位 ，常用于 I/O 变化的记录，默认不启用；
- Use vector as X source**: 将该参数作为 X 轴坐标，默认不勾选 X 轴为时间；
- Vector Color**: 设置图像颜色；
- Offset**: 设置偏移量，默认值为 0；
- Factor**: 设置实际值和绘图比例尺系数，默认值为 1，即 实测：图像=1:1；
- ;
- Update Selected**: 左键点击图表中的某参数，在①处选择需要更改的参数关键字，然后再点击该选项将会更新参数为重新选择的参数；
- Remove Selected**: 左键点击图表中的某参数，点击该选项将会从图表中移除该参数；
- Set all to X axis**: 此处 X（轴号）自动与当前控制轴相关联，点击该选项图表中所有参数将会被关联到当前控制轴；
- 右键单击轴的名称，可显示一个右键菜单



选项	含义
Remove	移除该参数
Set...active...(THIS...)	将本图表中所有参数关联为当前控制轴
Set...this...(THIS..)	将本图表中所有参数关联为当前参数轴
Set...active...(ALL...)	将所有图表中所有参数关联为当前控制轴
Set...this...(ALL..)	将所有图表中所有参数关联为当前参数轴
A axis	将关联轴切换为 A 轴（不同型号轴数将会有差异），其他可类比；

- 右键单击参数名称，可显示一个右键菜单



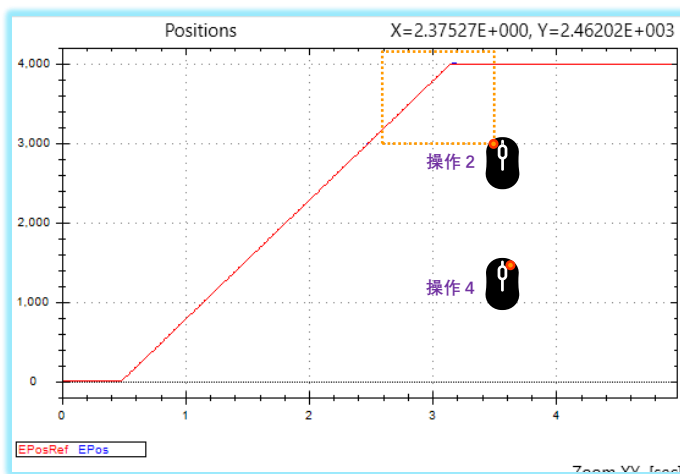
3.2.3 图形选项卡



使用“图形”选项卡可查看和管理记录的数据和图表属性。

图表选项卡中有许多功能，可使用鼠标按键和滚轮进行缩放、平移，并右键点击右键菜单。

图表属性可以通过右键菜单访问。

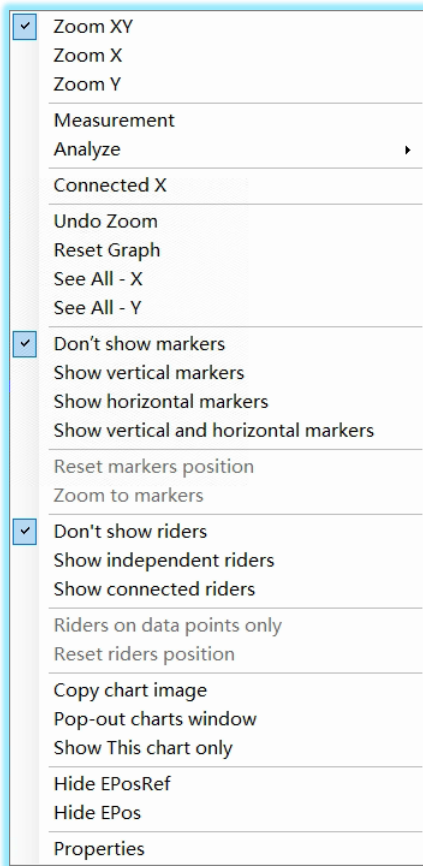


操作 1：滚动鼠标滚轮可缩放图表；

操作 2：按住鼠标左键拖选图像区域后松开可放大拖选区域；

操作 3：移动鼠标光标，可实时捕捉光标坐标；

操作 4：鼠标右键点击图表区域可调出右键菜单；



选项名称	功能
Zoom XY	缩放图像时同时缩放 X 轴和 Y 轴；
Zoom X	缩放图像时仅缩放 X 轴；
Zoom Y	缩放图像时仅缩放 Y 轴；
Measurement	启用测量工具；
Analyze	选择启用分析工具，如频率、微分、积分等；
Connected X	关联所有图表 X 轴；
Undo Zoom	撤销上一步的缩放；
Reset Graph	将图像恢复到原始状态；
See All - X	显示图表 X 轴全部数据；
See All - Y	显示图表 Y 轴全部数据；
Don't show markers	不显示标尺；
Show vertical markers	显示 X 轴(垂直)游标；
Show horizontal markers	显示 Y 轴(水平)游标；
Show vertical and horizontal markers	同时显示 X、Y 轴游标；
Reset markers position	重置游标到初始位置；
Zoom to markers	放大游标之间区域；
Don't show riders	不显示游码；
Show independent riders	仅当前图表显示轨迹坐标点
Show connected riders	全部关联图表显示轨迹坐标点；
Riders on data points only	
Reset riders position	重置轨迹坐标点到初始位置；
Copy chart image	复制图表区域为图像到剪贴板；
Pop-out charts window	将图表弹出为悬浮窗；
Show This chart only	仅显示当前图表（点击“Back to all charts”退出）；
Hide EPosRef	隐藏当前图表中第 1 个参数；
Hide EPos	隐藏当前图表中第 2 个参数；
Properties	显示当前图表属性；

分析工具（Analyze）：

分析工具中包含多种数学计算工具，用户可通过这些工具来分析问题及优化系统运动性能；

（1）频率分析：常用于分析系统噪声

Analyze	Frequency	Perform FFT (Full range)	傅里叶变换（图表全部区域）
	Derivative	Perform FFT (zoomed region)	傅里叶变换（图表放大区域）
	Integral	Perform PSD (Full range)	功率谱密度（图表全部区域）
		Perform PSD (zoomed region)	功率谱密度（图表放大区域）
		Perform CPS (Full range)	累积功率谱（图表全部区域）
		Perform CPS (zoomed region)	累积功率谱（图表放大区域）

（2）微积分：

Analyze	Frequency		
	Derivative	First order	一阶微分
	Integral	Second order	二阶微分
Analyze	Frequency		
	Derivative		
	Integral	First order	一次积分
		Second order	二次积分

图表属性（Properties）：可对图表属性进行相关更改处理，如颜色、X 轴、显示位等，点击“应用并显示图形（Apply and Show Graphs）”，可保存更改并返回到图表的图形视图；

General, Trigger	Charts Setup	Graphs																																
Chart title: <input type="text" value="Positions"/>		Chart setup:																																
Available vectors Recording Time	Chart vectors APosRef APos New vector	Chart background color <input type="text" value="White"/>																																
Line setup: APosRef Color: <input type="text" value="Red"/> Symbol: <input type="text" value="None"/> X axis vector: <input type="text" value="Recording Time"/> <input type="checkbox"/> Logarithmic		Markers <input checked="" type="radio"/> Don't show markers <input type="radio"/> Show vertical markers <input type="radio"/> Show horizontal markers																																
<input type="checkbox"/> Display data masked by 'logical bits' <table border="1"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr> <tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr> </table>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Riders <input checked="" type="radio"/> Don't show riders <input type="radio"/> Show independent riders <input type="radio"/> Show connected riders
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																			
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																			
<input type="checkbox"/> Display measurements on Chart Measurement #1: <input type="text" value="None"/> Measurement #2: <input type="text" value="None"/>		Grid lines setup <input checked="" type="checkbox"/> Show grid Color: <input type="text" value="Gray"/> # vertical lines: <input type="text" value="5"/> # horizontal lines: <input type="text" value="3"/>																																
<input checked="" type="checkbox"/> Apply Offset/Factor to recorded data Offset: <input type="text" value="0"/> Factor: <input type="text" value="0.0001"/>		<input type="button" value="Apply and Show Graphs"/>																																

3.2.4 分析选项卡

分析（Analysis）选项卡可让您能够对图形文件中的数据进行分析计算。

General, Trigger	Charts Setup	Graphs	Analysis	Manage
Select analysis type: <input type="text" value="Calculus"/>		<ul style="list-style-type: none"> Math operations Bit manipulation Conditional operations Calculus Frequency Performance 		
New vector name: <input type="text"/>		<input checked="" type="radio"/> Add to existing chart: <input type="text" value="Chart #1: Positions"/>		
New vector color: <input type="text"/>		Operation: <input type="text" value="Derivative"/>		
New vector chart: <input type="text"/>		Operation order: <input type="text" value="Second"/>		
Operation: <input type="text"/>		Vector to use: $d(\text{APos})/d(\text{RecTime})$		
Operation order: <input type="text"/>		<input checked="" type="radio"/> No filter <input type="radio"/> Filtered		
Vector to use: <input type="text"/>		<input type="button" value="Calculate and Add Result Vector"/>		

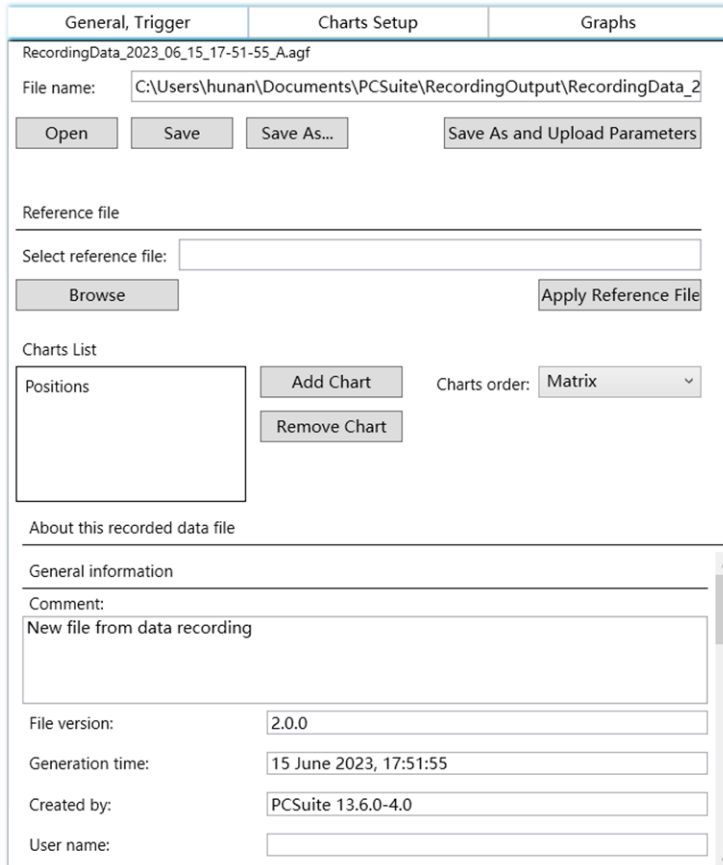
PCSuite 提供了不同类型的分析：

- 在图形文件中创建一个新数据向量的分析；例如，将两个现有向量相加，将一个现有向量乘以一个常数；
- 分析现有向量并提供数值结果的分析。例如，估计带宽、平均值、最小值、有效值；

如果创建了一个新向量，则需要对其命名。一旦创建，新向量就可以像其他数据向量一样，用于显示和更多分析；

3.2.5 管理选项卡

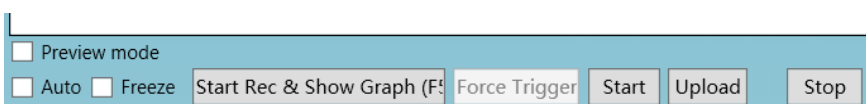
使用“管理”选项卡可打开并保存 *.agf（Agito 图形文件）文件、应用参考文件、修改图表、查看已记录数据的信息等。



The screenshot shows the 'Management' tab with the following sections:

- General, Trigger**: File name: C:\Users\hunan\Documents\PCSuite\RecordingOutput\RecordingData_2. Buttons: Open, Save, Save As..., Save As and Upload Parameters.
- Charts Setup**: Reference file: Select reference file: [text box]. Buttons: Browse, Apply Reference File.
- Graphs**: Charts List: Positions [text box]. Buttons: Add Chart, Remove Chart. Charts order: Matrix (dropdown).
- About this recorded data file**: General information: Comment: New file from data recording. Fields: File version: 2.0.0, Generation time: 15 June 2023, 17:51:55, Created by: PCSuite 13.6.0-4.0, User name: [text box].

3.2.6 数据记录控制



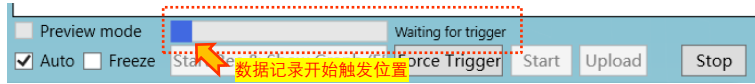
The screenshot shows the data recording control panel with the following elements:

- Preview mode
- Auto Freeze
- Start Rec & Show Graph (F5)
- Force Trigger
- Start
- Upload
- Stop

- **Start Rec & Show Graph:** 点击将开始记录和显示图表（Start Rec & Show Graph (F5)）。为“常规/触发”选项卡执行全部应用动作，并开始记录。在记录过程中，RecStat（记录状态）参数将被监测，并通过进度条显示。记录完成后，将上传并显示数据，所有记录的数据都将按照所设定的属性显示在对应的图表中。如果选择了一个模板文件，则模板文件中的图形属性将被应用到该图表中；
- **Force Trigger:** 点击将强制触发数据记录。当触发条件不便设定或监测对象非瞬时的情况下，该触发方式将会非常实用；
- **Start:** 执行全部应用并开始记录数据（不会等待记录停止，也不上传显示图象）；
- **Upload:** 上传最近记录的数据并显示图表；
- **Stop:** 在记录过程中停止；
- **Auto:** 选中该选项，示波器会在前一个数据采集完后才能后自动开启新的数据记录并显示图象；

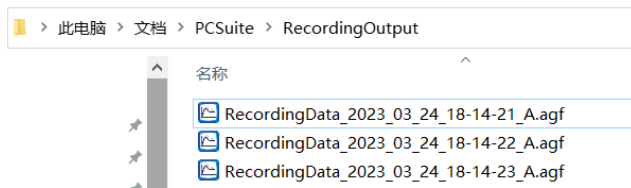
- **Freeze:** 选中该选项，示波器将不会重新绘制图表的坐标轴，将会在当前图表中显示新记录的数据；
- **Preview mode:** 可以快速查看 RecodingOutput 目录下的所有记录。

值得注意的是使用自动触发时，开始触发的时间点需要发生在开始数据记录触发点之后。



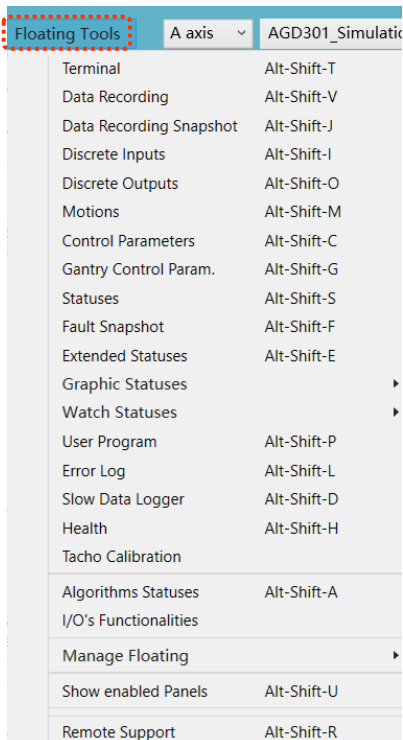
3.2.7 数据记录文件

每次执行数据记录的数据成功上传到电脑上时，都会保存到电脑本地文件夹中，用户可以访问以下浏览该路径查看已保存图像，如下所示：



3.3 悬浮窗工具

菜单栏 → Floating Tools



- **Terminal:** 打开终端命令输入窗口；
- **Data Recording:** 打开示波器悬浮窗；
- **Discrete Inputs:** 打开数字输入悬浮窗；
- **Discrete Outputs:** 打开数字输出悬浮窗；
- **Motions:** 打开运动控制悬浮窗；
- **Statuses:** 打开轴状态窗；
- **Graphic Statuses:** 实时图像采集窗（实时自动采集）；
- **Watch Statuses:** 实时变量监控窗；
- **Error Log:** 错误日志。

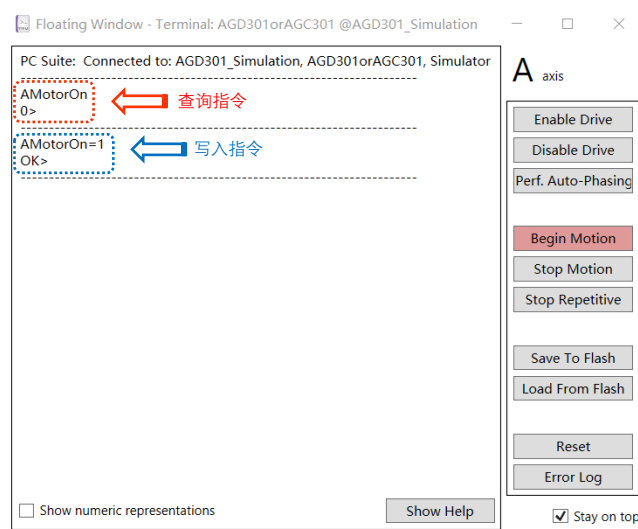
以下对上述常用悬浮窗工具使用进行简要说明：

3.3.1 终端命令窗 (Terminal)

Terminal 悬浮窗主要用于给控制器发送或查询指令（左侧），同时也包含部分常用指令按键（右侧），点击“Show help”将会弹出指令格式说明，“>”左侧值表示返回值，返回“OK”表示写入成功，格式错误将会返回错误提示；

指令主要遵循以下规则：

- 终端指令不区分大小写，例如 AMotorOn 和 AMOTORON 是同一指令；
- UserProgram 别名区分大小写，例如 AprogRun[1],MyTaskName 中的 MyTaskName 必须和 IDE 程序中所定义的 Task 名称一致，包括大小写；
- 查询单个参数值：Axis_index+Keyword，例如查询 A 轴使能状态发送“AMotorOn”；
- 查询所有轴该参数值：*+Keyword，例如查询所有轴使能状态发送“*MotorOn”；
- 查询数组某一元素值：Axis_index+Keyword+[Element_index]，例如查询全局变量数组 GenData 第 500 个元素存储的值发送“AGenData[500]”；
- 查询数组中所有元素：Axis_index+Keyword+[*]，例如查询所有模拟输入端口输入电压值发送“AAInPort[*]”；
- 写入指令格式：Axis_index+Keyword=Value，例如将 A 轴使能发送“AMotorOn=1”；
- 向数组元素写入：Axis_index+Keyword+[Element_index]=Value，例如将模拟输入 1 通道偏置设为 10mV 发送“AAInOffset[1]=10”；
- 输入指令，按回车（Enter）发送指令。

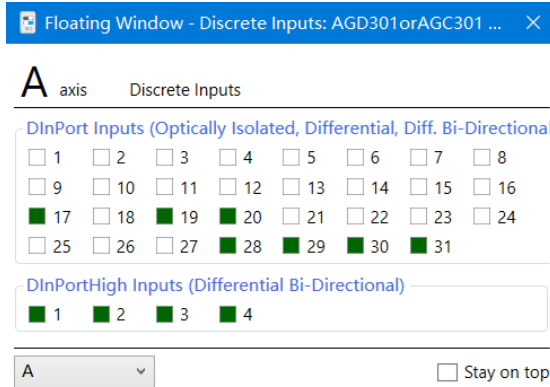


3.3.2 数据采集窗 (Data Recording)

Data Recording 悬浮窗用于调出示波器窗口，详细使用见本文 3.2 章节详细介绍。

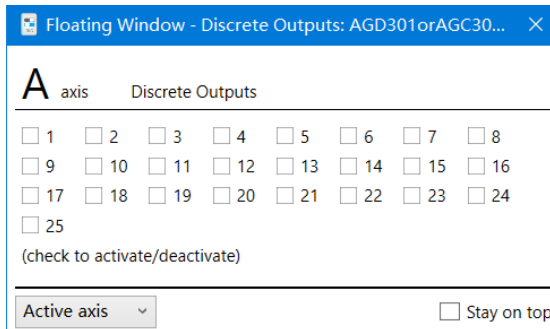
3.3.3 数字输入悬浮窗 (Discret Inputs)

Discrete Inputs 悬浮窗以缩略图显示当前控制器输入逻辑状态。



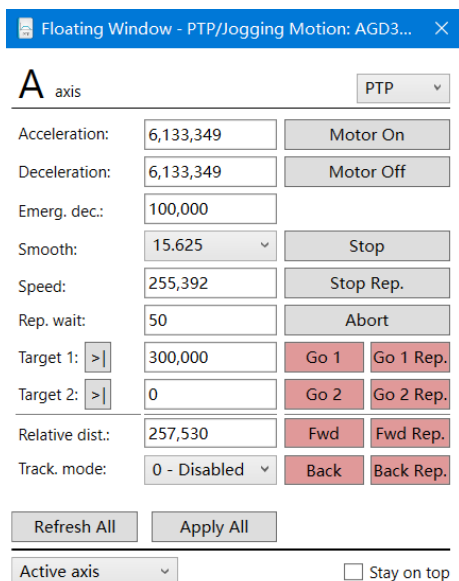
3.3.4 数字输出悬浮窗（Discrete Output）

Discrete Output 悬浮窗可观测及设置数字输出逻辑状态，点击对应输出端口编号左侧“□”可以改变该输出口逻辑状态，注意此处点击为实时生效，无需按回车键（和 I/O 界面设置有差异）；



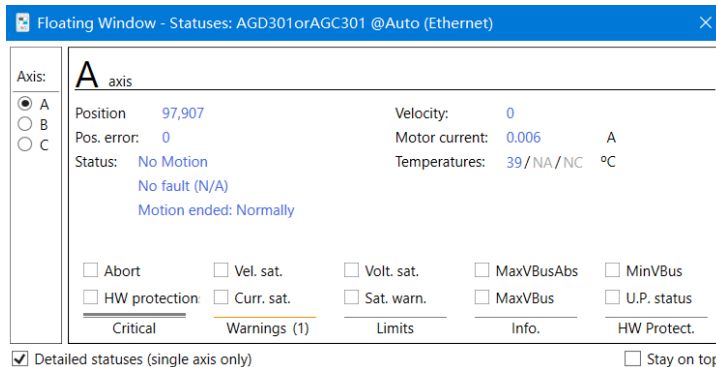
3.3.5 运动控制悬浮窗（Motions）

此处以悬浮窗的方式进行简单运动控制，如 PTP、Jog 等。



3.3.6 运动状态显示窗 (Statuses)

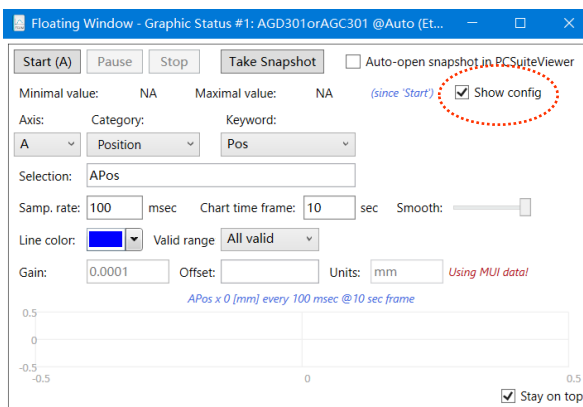
Statuses 悬浮窗显示当前轴运动状态、报警信息等；



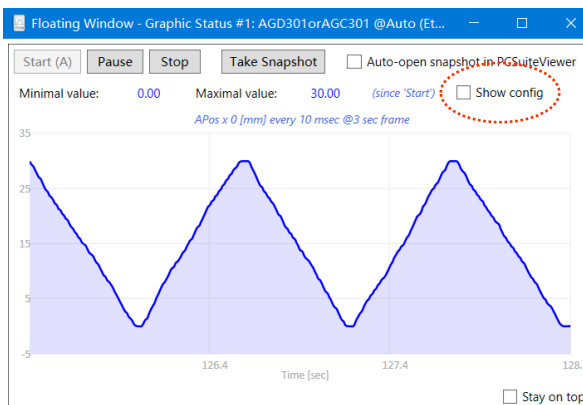
3.3.7 图像采集窗 (Graphic Statuses)

此窗口可实时自动抓取所定义变量数据到图像中显示 (v13.6.0-4.0 及以上版本 PCSuite 支持以用户自定义单位显示)，最多可同时打开 3 个监控窗口。

- 第 1 步：设定待记录参数：其中 Samp.rate (采样率) 最短 10msec，尽管 Chart time frame 值范围为 1~100sec，但受制于最大采样点数，其可实际可设定值与采样率正相关 (即 Samp.rate 值越大，Chart time frame 值越大)，超出范围将会弹出错误提示；

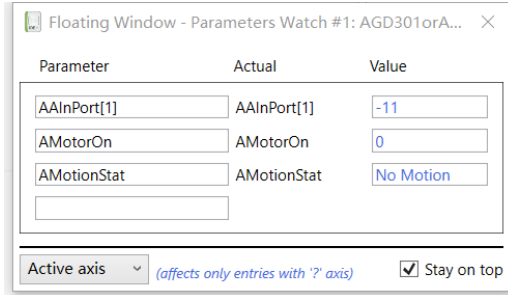


- 第 2 步：设置完参数后，点击 “Start” 即开始记录数据，再次点击 “Show config” 复选框隐藏参数设置，自动放大图表内容。



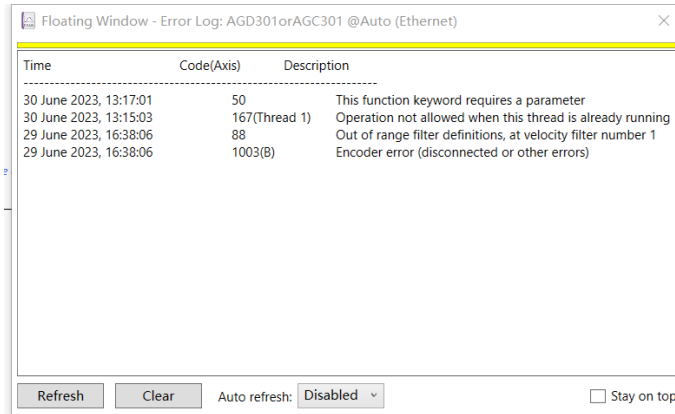
3.3.8 参数监控窗 (Watch Statuses)

Watch Statuses 窗口可实时显示所定义变量值，左侧设置变量，设置完变量后点回车键 (Enter) 确认，变量设置错误将会在 Value 处提示。



3.3.9 错误日志 (Error Log)

Error Log: 此窗口用于查询控制器报警历史。



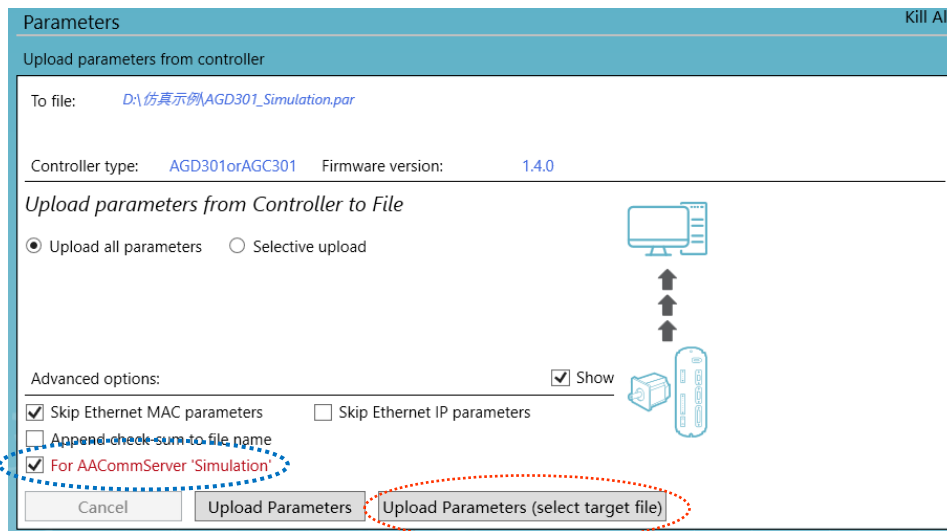
3.4 离线仿真

当用户需要在离线状态下查看或更改控制器参数，可以先创建一个离线仿真控制器文件进行相关操作，值得注意的是离线仿真需要本地已建立控制器信息，即在仿真前本地连接过真实的控制器，PCSuite 会自动在本地创建一个包含控制器信息的文件。

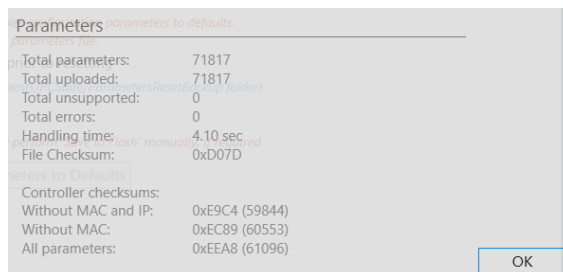
仿真使用步骤如下：


- 步骤 1：先连接真实控制器，并将控制器参数导出为仿真文件；

切换菜单到 **MANAGE**→**Parameters**，勾选“**For AACommServer ‘Simulation’**”，点击“**Upload ...**”，选择保存目录保存文件，

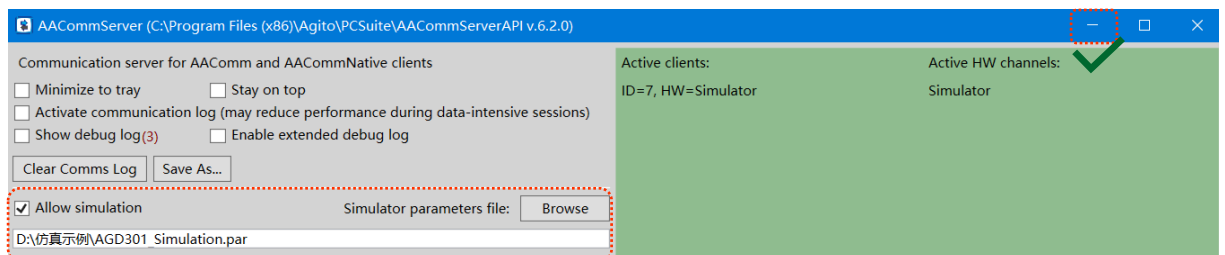


- 步骤 2：保存成功后弹出如下对话框；



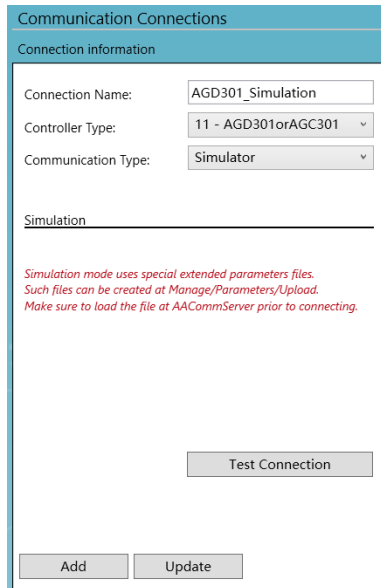
- 步骤 3：打开隐藏任务栏“”程序 AACommServer；

勾选“**Allow simulation**”，点击“**Browse**”选择第 1 步中保存的文件，然后**最小化** AACommServer 窗口，**注意：请勿点“X”关闭 AACommServer**



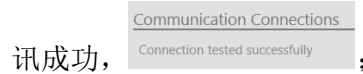
- 步骤 4：建立仿真通讯；

切换菜单到 **MANAGE**→**Communication Connections**，



分别填入/选择以下信息：

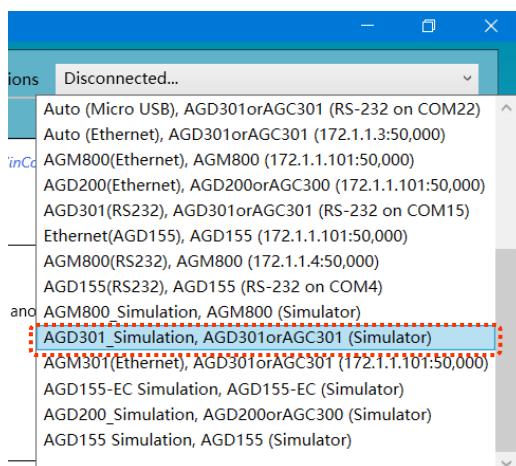
- **Connection Name:** 输入仿真控制器名称；
- **Controller Type:** 选择对应控制器类型；
- **Communication Type:** 选择“Simulator”；
- 点击“Test Connection”可以在右下角看到弹出通



- 讯成功，
- 点击“Add”，可以在右侧 List of Connections 看到

AGD301_Simulation, AGD301orAGC301, Simulator

- **步骤 5:** 退出通讯配置页面（切换至其他任一菜单均可退出），在右上角控制器连接处选择上一步加入的虚拟控制器即可连接上虚拟控制器



- **步骤 6:** 连接成功后可导入备份控制器参数（.Par）进行查看、修改参数等操作。

值得注意的是：步骤 3 中的仿真文件必须和步骤 4（或步骤 5 选择）控制器类型一致，不一致将会弹出错误提示。

4 PCSuite IDE+编程环境



IDE+为 Agito PCSuite 自带编程环境，用户可以在该环境下开发程序，并下载到控制器，使机器能够按照用户预定义的事件独立运行。使用 IDE+编程环境前需要对 Agito 控制器常用关键字有一定了解。

4.1 常用关键字

以下仅对部分常用关键字作介绍，其他详细关键字使用文档请参阅《Akribis-Agito Controller Keywords Reference》文档。

语法格式中 Axis_index 表示轴号，如 A 轴是 A、B 轴是 B，依次类推。

4.1.1 换相相关

ComtMode

ComtMode[]是一个用于控制换相过程的数组。

关键字	ComtMode
类型	参数
语法格式	读: Axis_index+ComtMode[Array_index]; 写: Axis_index+ComtMode[Array_index]=Value;
值类型	int
访问	读/写
是否允许运动时写入	否
是否允许使能时写入	否
是否保存到 flash	是
数组范围	1:24
是否轴相关	是
常用指令	ComtMode[5]=1282, 重新开始进行换相操作

ComtStatus

ComtStatus[]用于返回换相状态。

关键字	ComtStatus
类型	参数
语法格式	读: Axis_index+ComtStatus[Array_index];
值类型	int
访问	只读
是否允许运动时写入	是
是否允许使能时写入	是
是否保存到 flash	是
数组范围	1:2
是否轴相关	是
值含义	Axis_index ComtStatus[1]=100, 表示换相成功。

4.1.2 回零相关

回零相关参数主要由以下关键字定义：

▪ HomingOn

上电时或复位后 HomingOn 值将会被置 0，当用户将其置 1 时，控制器将会按照所定义的回零顺序执行回零操作。在回零结束后（无论成功还是报错失败），HomingOn 值都将重新被置 0。

关键字	HomingOn
类型	参数
语法格式	读：Axis_index+HomingOn; 写：Axis_index+HomingOn=Value (Value=0、1)
值范围	Int: 0~1
访问	读/写
是否允许运动时写入	否
是否允许使能时写入	是
是否保存到 flash	否
是否轴相关	是
值含义	Axis_index HomingOn=1，开始进行回零程序。

▪ HomingDef

HomingDef 用于定义回零过程参数，包括步骤和类型等参数，一般该参数通过 Pcsuite 回零工具导出（详细内容请参阅《Agito 回零使用手册》）。

关键字	HomingDef
类型	参数
语法格式	读：Axis_index+HomingDef[Array_index]; 写：Axis_index+HomingDef[Array_index]=Value
数组范围	[1:150]
访问	读/写
是否允许运动时写入	是
是否允许使能时写入	是
是否保存到 flash	是
是否轴相关	是
值含义	HomingDef[1]定义步骤类型，HomingDef[2-10]定义第 1 步参数；类似的 HomingDef[21-30]定义第 2 步，依次类推；

▪ HomingStat

HomingStat 用于描述回零状态。

关键字	HomingStat
类型	参数
语法格式	读：Axis_index+HomingStat;
访问	只读
是否允许运动时写入	是
是否允许使能时写入	是
是否保存到 flash	否
是否轴相关	是

值含义	值	含义
	0	上电或重启后未执行回零。
	正值 (除 100 以外)	回零进行中，HomingStat 值表示当前回零进行的步骤。
	-1	由于 HomingDef 参数错误导致的回零失败，在回零开始前每个步骤都会检查每步相关参数。
	-2	由于超时导致的回零失败，每个步骤都定义了执行延时时间，若该时间内未完成该步骤将会报错。
	-3	由于回零过程中电机意外下使能导致回零失败。
	-4	由于运动错误导致的回零失败，如未定义运动结束方式。
	-5	由于定义了不可识别的步骤类型导致的回零失败，如 HomingDef[1]=50。
	-6	由于检测到在步骤切换时电机运动状态为运动中导致的回零失败。
	-7	有错误的步骤导致的回零失败，意味着回零程序的最后一步不是“0-End Homing”，比如最后一步 HomingDef[61]=1 是错误的。
	-8	由于未知的步骤类型导致的回零失败。
	100	回零完成并成功。

4.1.3 运动相关

▪ Abort

将速度置 0 立即停止运动（电机保持使能状态）。

关键字	Abort
类型	命令
语法格式	Axis_index+Abort;
访问	只读
是否允许运动时写入	是
是否允许使能时写入	是
是否保存到 flash	否
是否轴相关	是

▪ AbsTrgt

AbsTrgt 用于设置 PTP(点到点运动)和往复运动的绝对目标位置。该目标位置和当前位置无关，AbsTrgt 定义运动停止的位置。对于 PTP 运动，该位置是电机运动停止的位置；对于往复 PTP

运动，AbsTrgt 位置将会是 PTP 运动的其中一目标点，而把当前位置作为往复运动另一目标点。如果 RelTrgt(相对目标位置)不为 0，则忽略 AbsTrgt，并使用 RelTrgt 来实际执行下一个目标动作，AbsTrgt 可以在运动过程中被修改并且立即执行。

关键字	AbsTrgt
类型	变量
语法格式	读: Axis_index+AbsTrgt;
值范围	-2,147,483,648~2,147,483,647
访问	读/写
是否允许运动时写入	是
是否允许使能时写入	是
是否保存到 flash	否
是否轴相关	是

示例:

如当前位置 Pos=1000,AbsTrgt=5000,RelTrgt=0,则下一个 PTP 运动将会停止在 Pos=5000 位置处。

如当前位置 Pos=1000, AbsTrgt=5000,RelTrgt=7000,则下一个 PTP 运动将会停止在 Pos=8000 位置处(目标位置=Pos(当前)+RelTrgt)。

如当前位置 Pos=1000, AbsTrgt = 5000 and RelTrgt = 0,则下一个往复 PTP 运动将会先运动到 Pos=5000 位置处，然后再运动到 Pos=1000 处，重复进行上述运动。

另请参阅: [RelTrgt](#)

▪ Accel

Accel 在所有运动中使用 user units/sec² 来表示加速度，Accel 的值可以在运动过程中被改变，并且会被立即执行，如当前正在做加速运动，加速度会被当前指令改变，如果此时已达目标速度，则新的加速度将会在下一次加速运动中执行。

关键字	Accel
类型	变量
语法格式	读: Axis_index+Accel;
值范围	100~2,000,000,000
访问	读/写
是否允许运动时写入	是
是否允许使能时写入	是
是否保存到 flash	是
是否轴相关	是

另请参阅: [Vel](#), [Speed](#), [Decel](#), [MaxAcc](#).

▪ Begin

Begin 命令用于在设置完所有运动参数（如运动模式、加减速度、速度等）后开始运动，此命令必须在电机使能之后才能生效，否则将会报错。将开始的运动类型由 MotionMode 定义，发送“Begin”命令之后，可以实时更改运动参数（速度、加减速等）并立即生效。

关键字	Begin
类型	命令

语法格式	Axis_index+Begin;
访问	只读
是否允许运动时写入	否
是否允许使能时写入	是
是否保存到 flash	否
是否轴相关	是

▪ BeginOnToPos

BeginOnToPos 命令用于在运动过程中将控制模式更改为位置模式，主要应用于力控。

关键字	BeginOnToPos
类型	参数
语法格式	读: Axis_index+BeginOnToPos; 写: Axis_index+BeginOnToPos=Value, Value=0,1;
值范围	0,1
默认值	0
访问	读/写
是否允许运动时写入	是
是否允许使能时写入	是
是否保存到 flash	否
是否轴相关	是

▪ Decel

Decel 用于定义所有运动的减速度，单位为 user units/sec²。Decel 可以在运动过程中进行更改，更新后的值将会立即生效。如当前运动在减速过程中，则减速率将会改变。

Stop 指令以 Decel 定义值减速停止，急停使用 EmrgDec 定义减速度停止。

关键字	Decel
类型	参数
语法格式	读: Axis_index+Decel; 写: Axis_index+Decel=Value;
值范围	100~2,000,000,000
默认值	100,000
访问	读/写
是否允许运动时写入	是
是否允许使能时写入	是
是否保存到 flash	否
是否轴相关	是

▪ AuxPos

AuxPos 反馈辅编码器读数。

关键字	AuxPos
类型	参数
语法格式	读: Axis_index+AuxPos; 写: Axis_index+AuxPos=Value;
值 (Value) 范围	-2,147,483,648~2,147,483,647

访问	读/写
是否允许运动时写入	否
是否允许使能时写入	否
是否保存到 flash	否
是否轴相关	是

▪ CurrRef

CurrRef 返回控制器回路电流（指令电流），单位 mA。

关键字	CurrRef
类型	参数
语法格式	读: Axis_index+CurrRef;
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ IndexPos

IndexPos 保持检测到的最后一个 index 信号位置（主编码器）。

关键字	IndexPos
类型	参数
语法格式	读: Axis_index+IndexPos;
值范围	-2,147,483,648~2,147,483,647
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ InTargetStat

InTargetStat 指示运动的进程，直到到达目标为止。当以 user-units 为单位表示的位置误差在到位时间（InTargetTime）内小于最大允许到位差（InTargetTol）时，则 InTargetStat 将会返回到到达目标位置。

关键字	InTargetStat	
类型	参数	
语法格式	读: Axis_index+InTargetStat;	
默认值	0	
访问	只读	
是否允许运动时读取	是	
是否允许使能时读取	是	
是否保存到 flash	否	
是否轴相关	是	
值含义	值	含义

	0	下使能
	1	上使能
	2	运动中
	3	等待 InTargetTime 结束
	4	到达目标位置

■ InTargetTime

InTargetTime 指示电机应在所允许的误差范围内到达目标位置的时间（以 ms 为单位）。如位置误差在 InTargetTime 时间内小于 InTargetTol，则 InTargetStat 返回目标到达。

关键字	InTargetTime
类型	参数
语法格式	读: Axis_index+InTargetTime;
值范围	0~1,000
默认值	3
访问	读/写
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

■ Jerk

Jerk 定义了基于轨迹规划的运动中建立所需加减速度所需的时间，

建立加减速度的时间计算如下：

$Jerk\ Time = SAMPLE_TIME * 2^{Jerk}$ ，SAMPLE_TIME 为采样时间，Agito 控制器采样时间为 $1/16384=61\mu s$ 。

在 PCSuite 界面以 JerkTime 显示，

注意 Jerk 不允许在运动中更改，显然 Jerk 值越大，运动时间越长，但是运动将会更平稳，通常过冲也会减小，应根据具体应用调整以获得最佳性能。

关键字	Jerk
类型	参数
语法格式	读: Axis_index+Jerk;
值范围	0~9
默认值	0
访问	读/写
是否允许运动时写入	否
是否允许使能时读写	是
是否保存到 flash	否
是否轴相关	是

▪ ModRev

ModRev 用于定义控制器反馈的模量，当该值为 0 时表示关闭该功能，当值设为 M 时表示编码器值将只在 [0, M)Counts 之间范围内变化，如超过 M 时将从 0Counts 处重新计数。

这允许旋转轴在同一方向无限运动（[0, M]在[RevPLim, FwdPLim]范围内）。

注意，不要将 ModeRev 和 Shaping 功能同时使用。

如果使用 SetPosition 手动将位置设置为超出 ModRev 范围外的值，则可能意外发生。

无限运动（ModRev 不为 0）可以和平滑（Jerk 不为 0）同时使用，但用户必须要正确设置 ModRev 和 Jerk 值，即在最大速度下，移动 ModRev 距离的时间应该大于 JerkTime，

例如：如果最大速度是 10,000,000 [counts/sec]，Jerk = 9 (即 $2^9 = 512$ samples)，控制器的采样率为 16,384 samples/sec，则 ModRev 必须大于 $10,000,000 / 16,384 * 512 = 312,500$ [counts]。

与 $Jerk(2^{Jerk[samples]})$ 和 ModRev ([counts]) 采样数的乘积必须小于 $(2^{31}-1)$ ，

例如 Jerk=9，则 ModRev 必须小于 $\frac{2^{31}-1}{2^9} \approx 4,194,304$ [counts]

关键字	ModRev
类型	参数
语法格式	读：Axis_index+ModRev; 写：Axis_index+ModRev=Value;
值范围	0~2,000,000,000
默认值	0
访问	读/写
是否允许运动时写入	否
是否允许使能时写入	否
是否保存到 flash	是
是否轴相关	是

▪ MotionMode

MotionMode 用于定义下一次“Begin”命令之后将执行的运动类型，注意在当前运动结束前，无法更改 MotionMode。

关键字	MotionMode	
类型	参数	
语法格式	读：Axis_index+MotionMode; 写：Axis_index+MotionMode=Value;	
值范围	0~18	
默认值	0	
访问	读/写	
是否允许运动时写入	否	
是否允许使能时写入	是	
是否保存到 flash	是	
是否轴相关	是	
值含义	值	含义
	0	Jog（点动）
	1	PTP（点到点）

	2	Point to point repetitive (往复 PTP)
	3	Pulse Direction direct mode (直接脉冲方向)
	4	Pulse Direction indirect mode (非直接脉冲方向)
	5	Gear direct motion (直接齿轮模式)
	6	Gear indirect motion (间接齿轮模式)
	7	ECAM direct motion
	8	ECAM indirect motion
	9	FIFO motion
	10	Slave position reference
	11	CNC group A motion
	12	Joystick direct motion, input analog in signal representing position reference
	13	Joystick indirect motion, input analog in signal representing position reference, and this profiler can be setting by user
	14	Joystick direct motion, input analog in signal representing velocity reference
	15	Joystick indirect motion, input analog in signal representing velocity reference ,and this profiler can be setting by user
	16	Vector motion
	17	CNC group B motion
	18	Spline buffer motion

▪ MotionReason

MotionReason 返回值用于指示导致最后一次运动结束的原因；

“Begin” 指令会将 MotionReason 值重置为 0.

关键字	MotionReason	
类型	参数	
语法格式	读: Axis_index+MotionReason;	
值范围	0~11	
默认值	0	
访问	只读	
是否允许运动时读取	是	
是否允许使能时读取	是	
是否保存到 flash	否	
是否轴相关	是	
值含义	值	含义
	0	当前运动还未结束或运动正常
	1	Stop 命令停止

	2	Abort 命令停止
	3	StopRep 命令停止
	4	检测到负限位开关停止
	5	检测到正限位开关停止
	6	负软限位停止
	7	正软限位停止
	8	电机下使能停止
	9	StopECAM 命令停止
	10	StopFIFO 命令停止
	11	检测到 index 停止（仅 Jogging 模式）

▪ MotionStat

MotionStat 返回当前运动状态，MotionStat 中的每个位表示一种运动状态，在某些情况下，可能会处于多个运动状态，MotionStat=0（即每位都为 0）为运动停止。MotionStat 值为十进制，可以转换成二进制数或者按位与（&）来根据每一个 bit 指示当前运动状态。

Bit	9	8	7	6	5	4	3	2	1	0
-----	---	---	---	---	---	---	---	---	---	---

关键字	MotionStat	
类型	参数	
语法格式	读：Axis_index+MotionStat;	
默认值	0	
访问	只读	
是否允许运动时读取	是	
是否允许使能时读取	是	
是否保存到 flash	否	
是否轴相关	是	
值含义	Bit	含义
	0	当前运动还未结束或运动正常
	1	Stop 命令停止
	2	Abort 命令停止
	3	StopRep 命令停止
	4	检测到负限位开关停止
	5	检测到正限位开关停止
	6	负软限位停止
	7	正软限位停止
	8	电机下使能停止
	9	StopECAM 命令停止

▪ MotorOn

MotorOn 指令用于控制电机上下使能，电机下使能时，电源不会加到电机上，也无法进行运动控制。因为故障（请参阅 ConFlt），驱动器可能会在内部下使能，重新上使能时，ConFlt 会被清除。如果导致下使能的故障未清除，电机将仍保持下使能状态。

某些命令只能在下使能时使用。

关键字	MotorOn	
类型	参数	
语法格式	读：Axis_index+MotorOn; 写：Axis_index+MotorOn=Value;	
值范围	0、1	
默认值	0	
访问	读/写	
是否允许运动时读取	是	
是否允许使能时读取	是	
是否保存到 flash	否	
是否轴相关	是	
值含义	值	含义
	0	下使能
	1	上使能

▪ Pos

Pos 以 UsrUnits 为返回编码器读数。如 UsrUnits=65536，则 Pos 以 Counts 为单位。

关键字	Pos
类型	参数
语法格式	读：Axis_index+Pos; 写：Axis_index+Pos=Value;
值范围	-2,147,483,648~2,147,483,647
默认值	0
访问	只读
是否允许运动时读取	否
是否允许使能时读取	否
是否保存到 flash	否
是否轴相关	是

▪ PosRef

PosRef 以 UsrUnits 表示用户位置指令。

关键字	PosRef
类型	参数
语法格式	读：Axis_index+PosRef; 写：Axis_index+PosRef=Value;
值范围	-2,147,483,648~2,147,483,647
默认值	0
访问	只读

是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ PosErr

PosErr 以 UsrUnits 为单位返回位置误差值，是位置指令和实际位置之间的差值。如 PosErr 值超过 MaxPosErr 所定义值，则电机将会下使能。

关键字	PosErr
类型	参数
语法格式	读: Axis_index+PosErr; 写: Axis_index+PosErr=Value;
值范围	-2,147,483,648~2,147,483,647
默认值	0
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ RelTrgt

RelTrgt 定义了当前位置和目标位置的距离，用于 PTP 运动和往复 PTP 运动。如 RelTrgt 不为 0，则忽略 AbsTrgt，使用 RelTrgt 来确定下一次运动的实际目标。RelTrgt 可以在运动中改变并立即生效。

关键字	RelTrgt
类型	参数
语法格式	读: Axis_index+RelTrgt; 写: Axis_index+RelTrgt=Value;
值范围	-2,147,483,648~2,147,483,647
默认值	0
访问	读/写
是否允许运动时读写	是
是否允许使能时读写	是
是否保存到 flash	否
是否轴相关	是

另请参阅: [AbsTrgt](#)

▪ Speed

Speed 用于定义目标速度，单位为 UsrUnits/Sec。在运动中可以实时改变速度。如当前运动处于加速和匀速阶段，将改变速度为更新后的值。

关键字	Speed
类型	参数
语法格式	读: Axis_index+Speed; 写: Axis_index+Speed=Value;

值范围	-60,000,000~60,000,000
默认值	0
访问	读/写
是否允许运动时读写	是
是否允许使能时读写	是
是否保存到 flash	是
是否轴相关	是

▪ Vel

Vel[]数组以三种不同格式返回编码器速度反馈，单位为 UsrUnits / Sec。

Vel[1]是经过滤波器之后的速度值，滤波器参数根据 VelFilt 定义；

Vel[2]为滤波前原始速度值；

Vel[3]为 16 个采样周期的平均值，约 1ms。

关键字	Vel
类型	参数
语法格式	读： Axis_index+Vel[Array_index]；
值范围	-2,147,483,648~2,147,483,647
默认值	0
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ SpeedChgNew

SpeedChgNew 定义了速度变量，当满足触发条件时，将会把 SpeedChgNew 值赋给 Speed。

关键字	SpeedChgNew
类型	参数
语法格式	读： Axis_index+SpeedChgNew； 写： Axis_index+SpeedChgNew=Value；
值范围	-60,000,000~60,000,000
默认值	0
访问	读/写
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	是
是否轴相关	是

▪ Stop

Stop 命令将按照设定的减速度停止当前运动，如果在没有在运动中，则没有任何效果。

关键字	Stop
类型	命令
语法格式	Axis_index+Stop；

访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

另请参阅: [Abort](#)

▪ StopCNCA

StopCNCA 命令将停止当前 A 组 CNC 运动。

关键字	StopCNCA
类型	命令
语法格式	Axis_index+ StopCNCA;
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ StopRep

StopRep 命令将停止往复运动。如在其他运动过程中发送该指令，则指令将会被忽略。在往复运动过程中发送 StopRep 命令，运动不会立即减速停止，而是在当前运动结束后在下一次运动的起点处停止。如果需要立即停止运动，请使用“Stop”命令。

关键字	StopRep
类型	命令
语法格式	Axis_index+ StopRep;
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

4.1.4 电机参数相关

▪ EncRes

EncRes 用于定义编码器分辨率。

关键字	EncRes
类型	参数
语法格式	Axis_index+ EncRes;
访问	读/写
是否允许运动时写入	否
是否允许使能时写入	否
是否保存到 flash	是
是否轴相关	是

▪ Ia

Ia 用于返回 A 相电流值，单位 mA。

关键字	Ia
类型	参数
语法格式	Axis_index+ Ia;
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ Ib

Ib 用于返回 B 相电流值，单位 mA。

关键字	Ib
类型	参数
语法格式	Axis_index+ Ib;
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ MotorCurr

MotorCurr 是电机的总电流（所有相的总和），单位 mA。

关键字	Ia
类型	参数
语法格式	Axis_index+ Ia;
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ Va

Va 用于返回施加到 A 相电压千分比

关键字	Va
类型	参数
语法格式	Axis_index+ Va;
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ Vb

Vb 用于返回施加到 B 相电压千分比。

关键字	Vb
类型	参数
语法格式	Axis_index+ Vb;
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ Vc

Vc 用于返回施加到 C 相电压千分比。

关键字	Vc
类型	参数
语法格式	Axis_index+ Vc;
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

4.1.5 I/O 相关

▪ AInPort

AInPort 用于返回各模拟输入通道模拟量值，单位与 AInMode 相关，不同控制器模拟输入通道数量（Port_index）有差异。

关键字	AInPort
类型	参数
语法格式	Axis_index+ AInPort[Port_index];
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ AOutPort

AOutPort 用于返回各模拟输出通道模拟量值，与 AOutMode 相关，不同控制器模拟输出通道数量（Port_index）有差异。

关键字	AOutPort
类型	参数
语法格式	读：Axis_index+ AOutPort[Port_index];

	写: Axis_index+ AOutPort[Port_index]=Value;
值范围	-12,000~12,000
访问	读/写
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	是
是否轴相关	是

▪ DInPort

DInPort 返回一个十进制值用于表示数字输入端口状态，不同控制器数字输出通道数量有差异；

当输入端口 Inputs 状态为“On”时， $DInPort = \sum_0^i 2^i$, i=bit, 逻辑为“Off”时不参与运算；

DInPort 返回值可以转换成二进制以查看每个单独输入状态；

Bit_0 (LBS)对应 Digital_Input_1, 依次类推；

值得注意的是 DInLog 会控制输入的逻辑，

例如：

如果 DInLog=4 (仅将 Digital_Input_3 逻辑取反)，并且输入状态如下：

Digital_Input_1=On

Digital_Input_2=Off

Digital_Input_3=Off

其他输入端口都为 Off

此时 DInPort 在 bit_0 返回“1”，bit_1 返回“0”，bit_2 返回“1”（因为 Digital_Input_3 虽实际为 Off，但因为逻辑取反，所以返回状态“1”）， $DInPort = \sum_0^2 2^i = 2^0 + 2^2 = 5$ ；

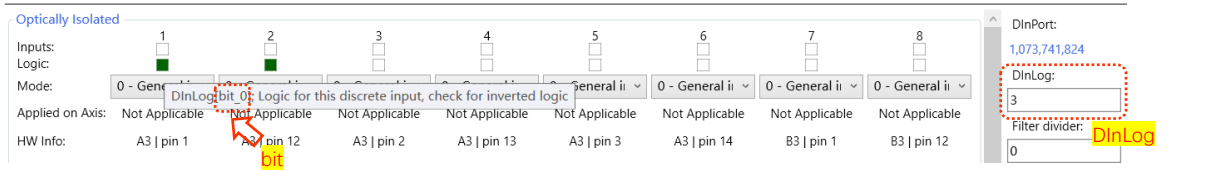
关键字	AOutPort
类型	参数
语法格式	读: Axis_index+ DInPort;
值范围	-12,000~12,000
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

▪ DInLog

DInLog 用于定义各数字输入通道逻辑。如 DInLog=0, 则当输入端口输入为 1 时，DInPort 返回为 1; DInLog=1 时，则当输入端口输入为 1 时，DInPort 返回为 0;

当输入端口 Logic 为“On”时， $DInLog = \sum_0^i 2^i$, i=bit, 逻辑为“Off”时不参与运算；

例如仅开启 Digital_Input_1 和 Digital_Input_2, $DInLog = 2^0 + 2^1 = 3$.



关键字	AInPort
类型	参数
语法格式	Axis_index+ AInPort[Port_index];
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

■ DInPortHigh

DInPortHigh 返回一个十进制值用于表示高速数字输入端口状态，不同控制器数字输出通道数量有差异；

当输入端口 Inputs 状态为“On”时， $DInPortHigh = \sum_0^i 2^i$, $i=bit$ ；逻辑为“Off”时不参与运算；

DInPortHigh 返回值可以转换成二进制以查看每个单独输入状态；

Bit_0 (LBS)对应 DInPortHigh Input_1，依次类推；

值得注意的是 DInLogHigh 会控制输入的逻辑，

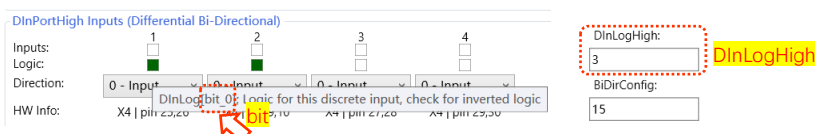
关键字	DInPortHigh
类型	参数
语法格式	读：Axis_index+ DInPortHigh[DInPortHigh_index];
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	是

■ DInLogHigh

DInLogHigh 用于定义各数字输入通道逻辑。如 DInLogHigh=0，则当输入端口输入为 1 时，DInPort 返回为 1；DInLogHigh=1 时，则当输入端口输入为 1 时，DInPort 返回为 0；

当输入端口 Logic 为“On”时， $DInLogHigh = \sum_0^i 2^i$, $i=bit$ ，逻辑为“Off”时不参与运算；

例如仅开启 DInPortHigh Input_1 和 DInPortHigh Input_2， $DInLogHigh = 2^0 + 2^1 = 3$ 。



关键字	AInPort
类型	参数
语法格式	Axis_index+ AInPort[Port_index];
访问	只读
是否允许运动时读取	是
是否允许使能时读取	是

是否保存到 flash	否
是否轴相关	是

▪ DOutPort

DOutPort 返回一个十进制值用于表示数字输出端口状态，不同控制器数字输出通道数量有差异；

当输出端口 Outputs 状态为“On”时， $DOutPort = \sum_0^i 2^i$ ，逻辑为“Off”时不参与运算；

DOutPort 返回值可以转换成二进制以查看每个单独输入状态；

实际的输出状态还取决于其他相关参数：

DOutPort 的 bit_i 与 DOutLog 的 bit_i 进行异或 (XOR) 运算， $i=0,1,2,\dots$ ；

若 $DOutMode[Port_index] \neq 0$ ，则与输出相关的输出状态将不受 DOutMode 对应位的值影响，输出将仅反映特殊功能。

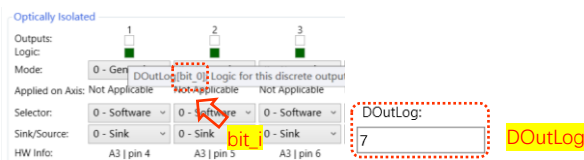
关键字	AOutPort
类型	参数
语法格式	读：Axis_index+ DOutPort; 写：Axis_index+ DOutPort=Value;
访问	读/写
是否允许运动时读写	是
是否允许使能时读写	是
是否保存到 flash	是
是否轴相关	是

▪ DOutLog

DOutLog 用于定义数字输出通道逻辑。如 $DOutLog=0$ ，则当输入端口输出为 1 时，DOutPort 返回为 1； $DOutLog=1$ 时，则当输入端口输入为 1 时，DInPort 返回为 0；

当 $DOutLog \neq 0$ (即开启) 时， $DOutLog = \sum_0^i 2^i$, $i=bit$ ，

例如仅将 Digital_Output_1、Digital_Output_2、Digital_Output_3 的 Logic 置 1，其他均置 0 时， $DOutLog = 2^0 + 2^1 + 2^2 = 7$ 。



关键字	AlnPort
类型	参数
语法格式	读：Axis_index+ DOutLog; 写：Axis_index+ DOutLog=Value
访问	读/写
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	是
是否轴相关	是

4.1.6 系统相关

▪ WaitStatus

WaitStatus 是 UserProgram 低层关键字，与“Compare”相关的语法通常由 PCSuite 在编译过程中自动生成。等待直到满足所定义的状态，用户程序才会继续执行。

值得注意的是该关键字仅允许在 IDE 程序中使用。

关键字	WaitStatus	
类型	命令	
语法格式	Axis_index+ WaitStatus[Status type], status value;	
访问	读/写	
是否允许运动时读取	是	
是否允许使能时读取	是	
是否保存到 flash	否	
是否轴相关	是	
值含义	Status type	含义
	1	COUNTERDOWN1: Wait until CounterDown[1] reaches the value
	2	COUNTERDOWN2: Wait until CounterDown[2] reaches the value
	3	COUNTERDOWN3: Wait until CounterDown[3] reaches the value
	4	COUNTERDOWN4: Wait until CounterDown[4] reaches the value
	5	COUNTERUP1: Wait until CounterUp[1] reaches the value
	6	COUNTERUP2: Wait until CounterUp[2] reaches the value
	7	IN_MOTION: Wait until MotionStat in motion bit reaches the value
	8	IN_RPT_WAIT: Wait until MotionStat in repeat wait bit reaches the value
	9	IN_RPT_STOP: Wait until MotionStat stop bit reaches the value
	10	IN_STOP_REQUEST: Wait until MotionStat stop request bit reaches the value
	11	IN_ACCELERATION: Wait until MotionStat in acceleration bit reaches the value
	12	IN_DECELERATION: Wait until MotionStat in deceleration bit reaches the value
	13	IN_WAIT_END_SMOOTH: Wait until MotionStat smoothing ended bit reaches the value
	14	IN_ECAM_STOP: Wait until MotionStat in ECAM stop bit reaches the value
	15	IN_FIFO_STOP: Wait until MotionStat in FIFO stop bit reaches the value
	16	COMMUTATION: Wait until StatReg commutation bit reaches the value

17	IN_TARGET: Wait until StatReg in target bit reaches the value
18	REC_TRIG_DETECTED_STATUS: Wait until RecStat trigger detected reaches the value
19	REC_COMPLETE_STATUS: Wait until RecStat recording complete reaches the value
20	DIN1: Wait until digital input 1 reaches the value
21	DIN2: Wait until digital input 2 reaches the value
22	DIN3: Wait until digital input 3 reaches the value
23	DIN4: Wait until digital input 4 reaches the value
24	DIN5: Wait until digital input 5 reaches the value
25	DIN6: Wait until digital input 6 reaches the value
26	DIN7: Wait until digital input 7 reaches the value
27	DIN8: Wait until digital input 8 reaches the value
28	DIN9: Wait until digital input 9 reaches the value
29	DIN10: Wait until digital input 10 reaches the value
30	DIN11: Wait until digital input 11 reaches the value
31	DIN12: Wait until digital input 12 reaches the value
32	DIN13: Wait until digital input 13 reaches the value
33	DIN14: Wait until digital input 14 reaches the value
34	MOTOR_ON_INPUT_STATUS: Wait until the input designated as motor on reaches the value
35	VEL_GAIN_CHANGE_STATUS: Wait until the input designated as gain change reaches the value
36	CLEAR_ABS_ENC_STATUS: Wait until the input designated as clear absolute encoder on reaches the value
37	CLEAR_IN_PULSES_STATUS: Wait until the input designated clear input pulses on reaches the value
38	REV_LIMIT_STATUS: Wait until the input designated as reverse limit reaches the value
39	FWD_LIMIT_STATUS: Wait until the input designated as forward limit reaches the value
40	TORQUE_LIMIT_ON_STATUS: Wait until the input designated as torque limit reaches the value
41	ALARM_RESET_STATUS: Wait until the input designated as alarm reset reaches the value
42	ABORT_INPUT_STATUS: Wait until the input designated as abort reaches the value
43	MODE_SWITCH_VEL_POS_STATUS: Wait until the input designated as vel / pos switch reaches the value
44	MODE_SWITCH_VEL_CUR_STATUS: Wait until the input designated as vel/cur switch reaches the value

	45	MODE_SWITCH_POS_CUR_STATUS: Wait until the input designated as pos/cur switch reaches the value
	46	ADD_FILTER_STATUS: Wait until the input designated as add filter reaches the value

▪ GenData

GenData 是分配给用户的通用数组，在这个数组中，用户可以存储整型数据，通常用于存储 ECAM 等表格。

关键字	GenData
类型	参数
语法格式	读: Axis_index+ GenData[Array_index]; 写: Axis_index+ GenData[Array_index]=Value
访问	读/写
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	是
是否轴相关	否

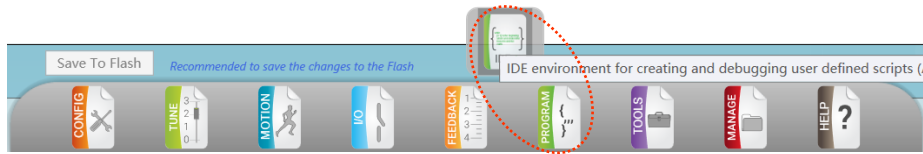
▪ WaitTime

WaitTime 用于定义继续执行下一行语句前的等待时间，以 ms 为单位。

值得注意的是该关键字仅允许在 IDE 编程环境中使用。

关键字	WaitTime
类型	参数
语法格式	写: Axis_index+ WaitTime, Value;
访问	读/写
是否允许运动时读取	是
是否允许使能时读取	是
是否保存到 flash	否
是否轴相关	否

4.2 IDE 简介



4.3 常用 IDE Demo 程序

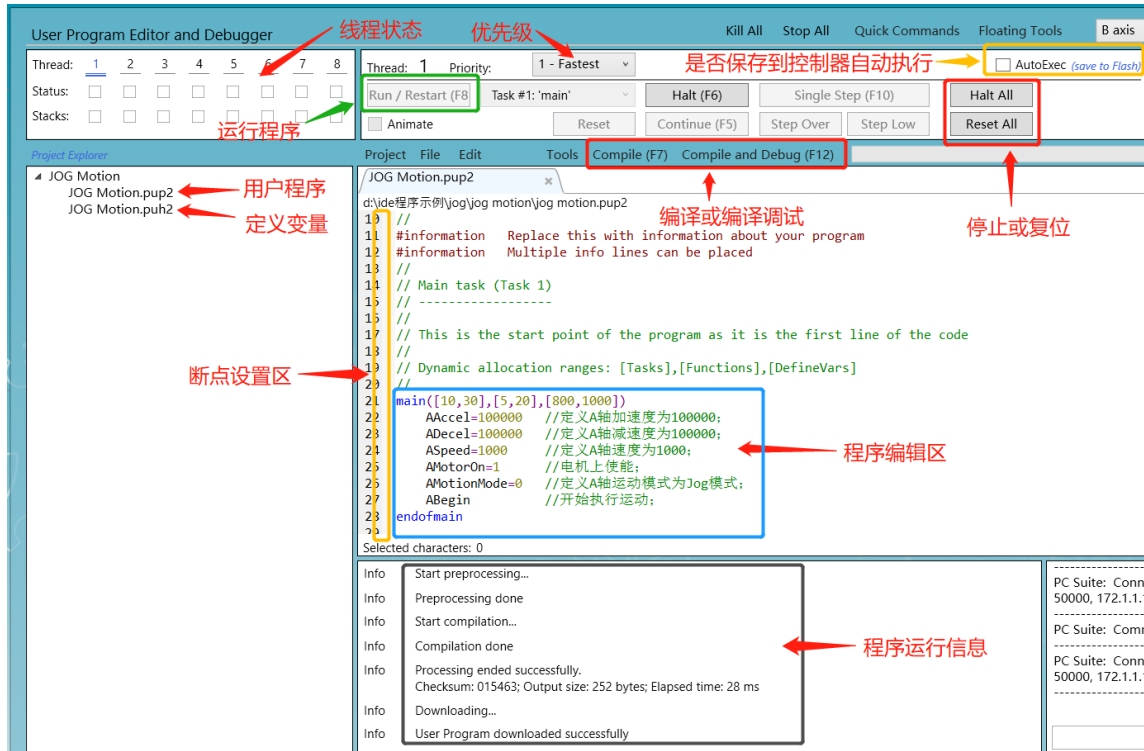
本章节主要介绍一些常用的 IDE User Program 程序以帮助用户快速了解 Agito IDE 编程环境。

IDE+为 Agito PCSuite 集成开发环境，用户可以在此环境下编译下载用户程序到控制器中，使其可以在用户发送请求下或控制器内的预定义事件的触发下独立运行。用户程序可以在控制器内部独立运行，无需通过通信接收命令。

■ IDE 编程环境基本语法：

- **关键字：**所有关键字前需要加轴号（即使该关键字为轴无关参数），关键字不区分大小写；
- **结束标志：**语句以回车、换行或回车+换行结束，每行只能包含一条语句（可以附加行尾注释），且语句必须包含在一行中（没有连续行）；
- **大小写：**程序语言区分大小写；
- **线程：**控制器支持多线程多任务，同时也支持线程间的优先级设置；
- **变量：**用户可以自定义变量名（`#define MY_EXAMPLE1 10000`、`#definevar MyExampleVar1`）、指针（`#defineptr MY_Pointer1 init_pointer`）等；
- **注释：**支持单行注释（`//...`）和段注释（`/*...*/`）；
- **运算符：**算术运算符、关系运算符、逻辑运算符、位运算符、赋值运算符等；
- **判断语句：**①`if...end` ②`if...else if...end` ③`switch...case...break...end`；
- **循环语句：**①`while...end` ②`for/foreach...end` ③`do...while...end`，且支持多层循环嵌套使用；
- **函数：**定义：`function:<MyFuncName>(<para1>,<para2>)...endoffunc`，调用：`<MyFuncName>(<argu1>,<argu2>)`，值得注意的是`<para>`和`<argu>`可为空；
- **任务：**方式 1：定义：`task:<MyTaskName>...endoftask`，调用：`runtask(<MyTaskName>,<Thread#>,<Priority>)`；方式 2：`task:<MyTaskName>{<Task#>}...endoftask`，调用：`AProgRun[<Thread#>,<Task#>]`；
- **断点：**支持 breakpoint 断点以及单步运行；

IDE 编程环境界面简介



- 程序运行基本流程：编写程序→Compile and Debug→Run/Restart;
- 程序运行中停止程序：Halt All→Reset All;
- 用户程序在.pup2 中编辑，变量或指针在.puh2 中定义;
- 线程状态区显示当前线程运行状态：空表示当前线程无程序在运行，绿色表示该线程有程序在正常运行，红色表示该线程程序出错，鼠标放在对应线程方框处会提示错误原因;
- Main 主函数默认在线程 1 中运行，不能更改，同一线程可放置多个任务（Task），但同时只能执行一个任务，建议 1 个线程只放一个任务，多个任务分配在多个不同的线程中执行;
- 断点设置需要在 Compile and Debug 之后才能点击对应行断点设置区进行设置;
- 勾选“Animate”，可以实时显示程序当前执行行数，配合断点可方便调试;
- IDE 程序中注意加阻塞语句以避免在不合适的状态下发送运动指令，如在发送“Begin”指令前必须保证电机此时处于非运动状态，可以在发送 Begin 命令前，加上阻塞语句“while(AMotionStat!=0) ←┘ end”;
- 如用户希望程序在上电时自动运行可以勾选“AutoExec”并“Save to flash”保存到控制器，当重新上电或重启时控制器将在内部自动执行程序。

4.3.1 PTP 运动

绝对目标位置运动:

```

21 main([10,30],[5,20],[800,1000])
22   AAccel=100000 //定义A轴加速度为100000;
23   ADecel=100000 //定义A轴减速度为100000;
24   ASpeed=1000 //定义A轴速度为1000;
25   ARelTrgt=0 //定义A轴相对目标位置为0;
26   AAbsTrgt=2000 //定义A轴绝对目标位置为2000;
27   AMotorOn=1 //电机上使能;
28   AMotionMode=1 //定义A轴运动模式为PTP运动;
29   ABegin //开始执行运动;
30 endofmain

```


相对目标位置运动:

```

21 main([10,30],[5,20],[800,1000])
22   AAccel=100000 //定义A轴加速度为100000;
23   ADecel=100000 //定义A轴减速度为100000;
24   ASpeed=1000 //定义A轴速度为1000;
25   ARelTrgt=3000 //定义A轴相对目标位置为3000;
26   AAbsTrgt=0 //定义A轴绝对目标位置为0;
27   AMotorOn=1 //电机上使能;
28   AMotionMode=1 //定义A轴运动模式为PTP运动;
29   ABegin //开始执行运动;
30 endofmain

```

4.3.2 Jog 运动

```

21 main([10,30],[5,20],[800,1000])
22   AAccel=100000 //定义A轴加速度为100000;
23   ADecel=100000 //定义A轴减速度为100000;
24   ASpeed=1000 //定义A轴速度为1000;
25   AMotorOn=1 //电机上使能;
26   AMotionMode=0 //定义A轴运动模式为Jog模式;
27   ABegin //开始执行运动;
28 endofmain

```

4.3.3 数字 I/O

判断数字输入状态:

```

24 //判断数字输入端口状态, AGenData[1000]赋值仅用于示例说明, 无实际意义
25 if ((ADInPort & 4)==4) //4=2^2, bit_2, 表示Digital_Input_3,依此类推;
26   AGenData[1000]=1 //当Digital_Input_3为“On”时, 将AGenData[1000]置1;
27 else if ((ADInPort & 4)==0)
28   AGenData[1000]=2 //当Digital_Input_3为“Off”时, 将AGenData[1000]置2;
29 end

```

数字输入逻辑取反:

```

32 ADInLog=ADInLog | 8 //bit_3, 将Digital_Input_4为Logic设为“On”
33 //（无论此时Digital_Input_4是“On”还是“Off”, 都将置为“On”）;

34 if ((ADInLog & 8)==8)
35   ADInLog=ADInLog-8 //bit_3, 将Digital_Input_4为Logic设为“Off”;
36 end

```

判断数字输出状态:

```

39 //判断数字输出端口状态, AGenData[900]赋值仅用于示例说明, 无实际意义
40 if ((ADOutPort & 16)==16) //16=2^4, bit_4, 表示Digital_Output_5, 依此类推;
41   AGenData[900]=1 //当Digital_Output_5为“On”时, 将AGenData[900]置1;
42 else if ((ADOutPort & 16)==0)
43   AGenData[900]=2 //当Digital_Output_5为“Off”时, 将AGenData[900]置2;
44 end

```

改变数字输出状态:

```

47     ADOutPort=ADOutPort | 32    //bit_5, 将Digital_Output_6为Logic设为 “On”;
48                                     // (无论此时Digital_Input_6是“On”还是“Off”,都将置为“On”);

49     if ((ADOutPort & 32)==32)
50         ADOutPort=ADOutPort-32 //bit_5, 将Digital_Output_6为Logic设为 “Off”;
51     end

```

数字输出逻辑取反:

```

54     ADOutLog=ADInLog | 64      //bit_6, 将Digital_Input_7为Logic设为 “On”
55                                     // (无论此时Digital_Input_7是“On”还是“Off”,都将置为“On”);

56     if ((ADOutLog & 64)==64)
57         ADOutLog=ADOutLog-64    //bit_6, 将Digital_Input_7为Logic设为 “Off”;
58     end

```

4.3.4 飞拍 (Position Event)

本例中以指针的方式设置 EventTable 值,

在 `.pup2` 头文件中定义一个指针:

```
11 #defineptr AEventTable_ptr AEventTable[1]
```

在 `.pup2` 程序文件 main 函数中调用封装的 SetEventTable function:

```
25     SetEventTable(2,1000) //示例中将 AEventTable[2]的值设为 1000;
```

将 EventTable 封装到 SetEventTable function 中:

```

29 function:SetEventTable(index,value)
30     *(AEventTable_ptr+(index-1)*65536)=value
31 endoffunc

```

4.3.5 误差补偿 (Err Mapping)

本例中以指针的方式设置 MapTable 值,

在 `.pup2` 头文件中定义一个指针:

```
13 #defineptr AErrMap_ptr AMapTable[1]
```

在 `.pup2` 程序文件 main 函数中调用封装的 SetAMapTable function:

```

21 main([10,30],[5,20],[800,1000])
22
23     SetAMapTable(3,100) //示例中将A轴MapTable[3]的补偿值设为100;
24
25 endofmain
26
27
28 function:SetAMapTable(index,value) //将A轴MapTable封装到SetAMapTable()函数中;
29     *(AErrMap_ptr+(index-1)*65536)=value
30 endoffunc

```

4.3.6 多线程 (Multi-Thread)

```

21 main([10,30],[5,20],[800,1000])
22
23   AProgRun[5],2 //将2#_task绑定thread_5
24   AwaitTime, 500
25   AProgRun[6],3 //将3#_task绑定thread_6
26   AwaitTime, 500
27   AProgRun[7],4 //将4#_task绑定thread_7
28   AwaitTime, 500
29
30 endofmain
31
32 task: A_AxisMotion{2} //将第1个任务绑定到2#_task, 注意1#_task为主线程保留task, 不允许自定义使用
33   AMotorOn=1
34 endoftask
35
36 task: B_AxisMotion{3} //将第2个任务绑定线程3#_task
37   BMotorOn=1
38 endoftask
39
40 task: AGenData_Def{4} //将第3个任务绑定线程4#_task
41   AGenData[1000]=100
42 endoftask

```

4.3.7 插补运动 (CNC)

通常情况下 CNC 程序从 CNC 调试界面导出，并添加到 IDE 主程序中调用。

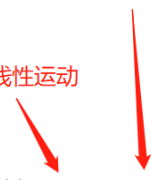
```

10 // Member axes: A, B, C
11 // Encoder ratios
12 //
13 ACNCAEncRatio = 256
14 BCNCAEncRatio = 256
15 CCNCAEncRatio = 256
16 //
17 // Vector speed percents
18 //
19 ACNCA Percents = 100
20 //
21 // Clear CNCA FIFO
22 //
23 ACNCA Clear
24 //
25 // Vector Motion Continuity Mode
26 //
27 ACNCA EndSegMod=0
28 //
29 // Segment index: 1
30 //
31 // Push Linear motion, involved: A, B, C
32 // Go to: A=300000, B=300000, C=-350000
33 // Velocity: 800000
34 // End Velocity: 0
35 //
36 ACNCA PushType,16855039 //参数类型, 为32位数, 高8位代表运动类型, 如1代表线性运动, 低24位代表关联轴, 如16855039(D)-->>1 000000010010111111111111
37 ACNCA PushParam,300000
38 ACNCA PushParam,300000
39 ACNCA PushParam,-350000
40 ACNCA PushParam,800000

```

低24位转为16进制为012FFF, 即表示CNC关联轴为A、B、C

高位“1”表示线性运动



5 常见故障代码及解决办法

故障代码 (关键字: ConFlt)	故障描述 (英文)	故障描述 (中文)	可能的原因	处理方法
0	No fault has occurred	无故障		
1001	Abort signal was detected	检测到中止信号		
1002	Short from motor phase to ground	电机相线与地短路	(1) 接线错误, 误将地线接到驱动器 ABC 相线接口	纠正接线错误
			(2) 电机相线绝缘保护层损坏, 导致与地 (或金属外壳) 发生碰触	检查电动力线, 更换线缆, 保证用万用表导通档测量时, 电机相线和地线之间是不导通的
			(3) 电机自身内部对地绝缘损坏	更换电机
1003	Encoder error (disconnected or other errors)	编码器错误 (断开连接或其他错误)	(1) 编码器线焊错针脚	检查编码器接线, 纠正焊线错误
			(2) 接头未插紧或意外脱落	插紧接头。如有螺钉固定, 则拧紧固定螺钉
			(3) 编码器类型配置错误	在 PCSuite 调试软件中, 配置正确的编码器类型
1004	FPGA watchdog not received	没收到 FPGA 看门狗程序		
1005	PWM dead time too short	PWM 死区时间太短		
1006	Hall input disconnected	霍尔输入断开连接	(1) 霍尔线焊错针脚	检查霍尔接线, 纠正焊线错误
			(2) 接头未插紧或意外脱落	插紧接头。如有螺钉固定, 则拧紧固定螺钉
			(3) 霍尔配置错误	在 PCSuite 调试软件中, 正确配置霍尔对应的数字量输入
1007	Motor is stuck	电机堵转	(1) 设置的运动距离过长, 导致撞到了机械限位	设置合适的运动距离
			(2) 在回零步骤“Hard stop by high error”中, 检测 Hard stop (机械限位) 是否到达的跟随误差阈值过高	设置合理的跟随误差阈值
			(3) 其它机械部件干涉导致	检查机械

1008	Bus Voltage too high	母线电压过高	(1) 电源供电电压过高	根据驱动器型号, 选择符合规格的电源
			(2) 发生在高速运动的减速制动阶段	外接合适的再生制动电阻
			(3) EMI 电磁干扰 (更多出现在高速运动时)	采取抑制电磁干扰的措施, 如接地、屏蔽、在线缆上套磁环等
1009	Bus Voltage too low	母线电压过低	(1) 电源供电线路异常	检查电源输入接线
			(2) 电源供电异常断开	检查供电电源是否正常工作
			(3) 选用的供电电源电压过低	选用合适的供电电源
1010	Logic Voltage too high	逻辑电压过高	(1) 控制器内部电路异常	请联系技术支持或销售, 返厂维修
			(2) EMI 电磁干扰 (更多出现在高速运动时)	采取抑制电磁干扰的措施, 如接地、屏蔽、在线缆上套磁环等
1011	Logic Voltage too low	逻辑电压过低	(1) 逻辑电源供电线路异常	检查逻辑电源输入接线
			(2) 逻辑电源供电异常断开	检查供电电源是否正常工作
1012	Bus current too high	母线电流过高		
1013	Phase A current too high	A 相电流过高	(1) 相电流限制值设置得过小	在 PCSuite 调试软件中, 配置正确的相电流限制值
			(2) 电机类型设置错误	在 PCSuite 调试软件中, 配置正确的电机类型
			(3) 电机绕组异常	用万用表测量电机绕组各相线之间的电阻, 如果电阻值与电机规格书中的额定值差异较大, 则需更换电机
1014	Phase B current too high	B 相电流过高	(1) 相电流限制值设置得过小	在 PCSuite 调试软件中, 配置正确的相电流限制值
			(2) 电机类型设置错误	在 PCSuite 调试软件中, 配置正确的电机类型
			(3) 电机绕组异常	用万用表测量电机绕组各相线之间的电阻, 如果电阻值与电机规格书中的额定值差异较大, 则需更换电机

1015	Phase C current too high	C 相电流过高	(1) 相电流限制值设置得过小	在 PCSuite 调试软件中，配置正确的相电流限制值
			(2) 电机类型设置错误	在 PCSuite 调试软件中，配置正确的电机类型
			(3) 电机绕组异常	用万用表测量电机绕组各相线之间的电阻，如果电阻值与电机规格书中的额定值差异较大，则需更换电机
1016	Motor current too high	电机电流过高	(1) 相电流限制值设置得过小	在 PCSuite 调试软件中，配置正确的相电流限制值
			(2) 电机类型设置错误	在 PCSuite 调试软件中，配置正确的电机类型
			(3) 电机绕组异常	用万用表测量电机绕组各相线之间的电阻，如果电阻值与电机规格书中的额定值差异较大，则需更换电机
1017	Driver power exceeds limit	驱动器功率超过限制	驱动器功率过小	换用更大功率的驱动器
1018	IPM temperature too high	IPM 温度过高	(1) 连续运行节拍过快	降低加速度和速度，或增加停顿时间
			(2) 运行环境温度过高	改善驱动器所在机柜的散热
			(3) 驱动器之间的安装间距过小	增加驱动器之间的间距
			(4) 测温传感器损坏	请联系技术支持或销售，返厂维修
1019	Velocity too high	速度过高	(1) 速度限制值设置得过小	在 PCSuite 调试软件中，配置所需要的速度限制值
			(2) 反馈编码器信号弱或短暂异常	检查编码器安装，保证信号为最佳状态
1020	Position error exceeds limit	位置误差超过限制	(1) 位置误差允许值设定得过小	在 PCSuite 调试软件中，配置合理的位置误差允许值
			(2) 电机换相错误	纠正电机相序，或在 PCSuite 调试软件中，重新执行自动换相
			(3) 位置控制 PIV 参数不合适	在 PCSuite 调试软件中，调试合适的 PIV 参数
1021	Velocity error exceeds limit	速度误差超过限制	(1) 速度误差允许值设定得过小	在 PCSuite 调试软件中，配置合理的速度误差允许值

			(2) 位置控制 PIV 参数不合适	在 PCSuite 调试软件中, 调试合适的 PIV 参数
1022	CPU temperature too high	CPU 温度过高	(1) 连续运行节拍过快	降低加速度和速度, 或增加停顿时间
			(2) 运行环境温度过高	改善驱动器所在机柜的散热
			(3) 驱动器之间的安装间距过小	增加驱动器之间的间距
			(4) 测温传感器损坏	请联系技术支持或销售, 返厂维修
1023	Bus Voltage too high - exceed abs. limit	母线电压过高 - 超过绝对限制值	(1) 电源供电电压过高	根据驱动器型号, 选择符合规格的电源
			(2) 发生在高速运动的减速制动阶段	外接合适的再生制动电阻
1024	STO1 activated	STO1 被激活	(1) 出于安全, 拍下急停开关, 断开了 STO 信号 (通常发生在电机使能 MotorOn=1 的情况下)	松开急停开关, 恢复 STO 信号
			(2) STO 接线错误	检查 STO 接线
1025	Over current detected on motor A	在 A 轴电机上检测到电流过流	(1) A 轴相电流限制值设置得过小	在 PCSuite 调试软件中, 对 A 轴配置正确的相电流限制值
			(2) A 轴电机类型设置错误	在 PCSuite 调试软件中, 对 A 轴配置正确的电机类型
			(3) A 轴电机绕组异常	用万用表测量电机绕组各相线之间的电阻, 如果电阻值与电机规格书中的额定值差异较大, 则需更换电机
1026	Auxiliary encoder error (disconnected or other errors)	辅助编码器错误 (断开连接或其他错误)	(1) 辅助编码器线焊错针脚	检查辅助编码器接线, 纠正焊线错误
			(2) 接头未插紧或意外脱落	插紧接头。如有螺钉固定, 则拧紧固定螺钉
			(3) 辅助编码器类型配置错误	在 PCSuite 调试软件中, 配置正确的辅助编码器类型
1027	IPM fault	IPM 故障	控制器内部电路异常	请联系技术支持或销售, 返厂维修
1028	Selected encoder type is not supported	不支持选定的编码器类型	编码器类型配置错误	在 PCSuite 调试软件中, 配置正确的编码器类型
1029	Selected aux. encoder type is not supported	不支持选定的辅助编码器类型	辅助编码器类型配置错误	在 PCSuite 调试软件中, 配置正确的辅助编码器类型
1030	ECAM master changed too much	ECAM 主轴变化太大		

1031	FIFO segment with too low acceleration	FIFO 程序段的加速度过低		
1032	Encoder modulus used with input shaping	编码器模量和输入整形同时使用	在启用编码器模量功能的情况下，欲再启用输入整形滤波器；或者，在启用输入整形滤波器的情况下，欲再启用编码器模量功能	在 PCSuite 调试软件中，要么启用编码器模量功能，要么启用输入整形滤波器，两者不可同时启用
1033	Fault was detected during power on BIT	在 BIT 上电时检测到故障		
1034	STO2 activated or internal amp. power failure	STO2 被激活或内部放大器电源故障	(1) 出于安全，拍下急停开关，断开了 STO 信号（通常发生在控制器上电过程中）	松开急停开关，恢复 STO 信号
			(2) STO 接线错误	检查 STO 接线
1035	AC Power supply to the control was cut off	用于控制的交流输入电源被切断	(1) 逻辑电源供电线路异常	检查逻辑电源输入接线
			(2) 逻辑电源供电异常断开	检查供电电源是否正常工作
1036	Over current detected on motor B	在 B 轴电机上检测到电流过流	(1) B 轴相电流限制值设置得过小	在 PCSuite 调试软件中，对 B 轴配置正确的相电流限制值
			(2) B 轴电机类型设置错误	在 PCSuite 调试软件中，对 B 轴配置正确的电机类型
			(3) B 轴电机绕组异常	用万用表测量电机绕组各相线之间的电阻，如果电阻值与电机规格书中的额定值差异较大，则需更换电机
1037	Over continuous current at Linear Amplifier	线性放大器的连续电流过高	(1) 连续电流限制值设置得过小	在 PCSuite 调试软件中，配置正确的连续电流限制值
			(2) 连续运行节拍过快	降低加速度和速度，或增加停顿时间
			(3) 线性放大器的持续电流不够	换用更高持续电流的线性放大器
1038	Over voltage at Linear Amplifier	线性放大器的电压过高	(1) 电源供电电压过高	根据驱动器型号，选择符合规格的电源
			(2) 发生在高速运动的减速制动阶段	外接合适的再生制动电阻
			(3) EMI 电磁干扰（更多出现在高速运动时）	采取抑制电磁干扰的措施，如接地、屏蔽、在线缆上套磁环等
1039	Conflicting values in AmpType of axes	轴相关的 AmpType 取值冲突	配置的放大器类型 (AmpType) 错误	在 PCSuite 调试软件中，配置正确的放大器类型 (AmpType)
1040	Motor temperature too high	电机温度过高	(1) 连续运行节拍过快	降低加速度和速度，或增加停顿时间

			(2) 测温传感器接线错误	纠正测温传感器接线
			(3) 在无温度传感器接线的情况下, 启用了电机过温检测功能	在 PCSuite 调试软件中, 关闭电机过温检测功能
1041	Unexpected error code. Please consult PCSuite provider	意外错误代码。请咨询 PCSuite 提供商	PCSuite 版本过低	安装更新 PCSuite 软件
1042	Amplifier isolated 5v fault	放大器 5v 隔离电路故障	控制器内部电路异常	请联系技术支持或销售, 返厂维修
1043	CI Communication was disconnected	CI 通讯断开连接	(1) 网线损坏	更换网线
			(2) 网线未插紧或意外脱落	插紧网线接头
			(3) 远程放大器掉电	检查远程放大器的电源供电情况
1044	Motor current over I2T	电机电流超过 I2T	(1) 连续运行节拍过快	降低加速度和速度, 或增加停顿时间
			(2) 连续电流和最大电流参数设置不合理	在 PCSuite 调试软件中, 设置合理的连续电流和最大电流参数
			(3) 电机连续电流不满足工况要求	换用更高持续电流的电机
1045	Force error exceeds limit	力误差超过限制	(1) 力误差允许值设定得过小	在 PCSuite 调试软件中, 配置合理的力误差允许值
			(2) 闭环力控压力传感器压力反馈异常	检查压力传感器压力反馈模拟量输入的接线和读数
			(3) 闭环力控 PID 参数不合适	在 PCSuite 调试软件中, 调试合适的闭环力控 PID 参数
			(4) 闭环力控 IDE 程序中设置的目标位置不合适	目标位置应设置为大于接触点处的位置 (控制器会根据设定的压力阈值自动从位置模式切换到力控模式)
1046	Force control: analog force feedback not defined	力控制: 未定义模拟量力反馈	未配置压力传感器接线对应的模拟量输入	在 PCSuite 调试软件中, 将压力传感器接线对应的模拟量输入配置为 Force Feedback 模式
1047	A 5v pin is overloaded. Check your HW connections!	5v 引脚电流过载。检查您的硬件连接!	(1) 给编码器供电的 +5v 和 0v 针脚接反	纠正编码器接线错误
			(2) 工作电流超过控制器 5v 电源最大带载能力 (最大 0.5A)	将超额的负载接到外部 5v 电源
1048	Other member axis got motor off	其他参与轴的电机下使能	CNC 或矢量运动模式下, 某个参与轴发生报警	查看 CNC 或矢量运动参与轴的故障信息, 排查原因

1049	Dual Loop high difference between velocities	全闭环两路编码器速度反馈相差太大	(1) 辅助编码器分辨率设置错误	在 PCSuite 调试软件中, 配置正确的辅助编码器分辨率
			(2) 全闭环控制 PIV 参数不合适	在 PCSuite 调试软件中, 调试合适的全闭环控制 PIV 参数
1050	External fault input (for example: of external drive) is activated	外部故障输入 (例如: 来自外部驱动器) 被激活	配置了某个数字量输入点为“Fault (external drive)”且收到外部设备报警信号	检查外部设备报警信息并清除其报警信号输出
1051	Internal relay is still switched off. Try waiting more time after power on before enabling the motor	内部继电器仍处于断开状态。在使能电机之前, 请尝试在通电后等待更多时间		
1052	STO2 activated	STO2 被激活	(1) 出于安全, 拍下急停开关, 断开了 STO 信号 (通常发生在电机使能 MotorOn=1 的情况下)	松开急停开关, 恢复 STO 信号
			(2) STO 接线错误	检查 STO 接线
1053	AmpType value not allowed for this product	AmpType 的值不允许用于本产品	配置的放大器类型 (AmpType) 错误	在 PCSuite 调试软件中, 配置正确的放大器类型 (AmpType)
1054	At least one required phase of the AC Power inputs was cut off	交流电源输入其中至少一相被切断	(1) 电源供电线路异常	检查电源输入接线
			(2) 电源供电异常断开	检查供电电源是否正常工作
1055	Position error exceeds limit in open-loop	开环位置误差超过限制	(1) 开环位置误差允许值设定得过小	在 PCSuite 调试软件中, 配置合理的开环位置误差允许值
			(2) 电机换相错误	纠正电机相序, 或在 PCSuite 调试软件中, 重新执行自动换相
			(3) 在系统辨识时发生, 设定的开环保护值过小	在 PCSuite 调试软件中, 配置合理的开环保护值
1056	Velocity error exceeds limit in open-loop	开环速度误差超过限制	(1) 开环速度误差允许值设定得过小	在 PCSuite 调试软件中, 配置合理的开环速度误差允许值
			(2) 电机换相错误	纠正电机相序, 或在 PCSuite 调试软件中, 重新执行自动换相
			(3) 在系统辨识时发生, 设定的开环保护值过小	在 PCSuite 调试软件中, 配置合理的开环保护值
1057	Force error exceeds limit in open-loop	开环力误差超过限制	(1) 开环力误差允许值设定得过小	在 PCSuite 调试软件中, 配置合理的开环力误差允许值
			(2) 闭环力控压力传感器压力反馈异常	检查压力传感器压力反馈模拟量输入的接线和读数

			(3) 闭环力控 PID 参数不合适	在 PCSuite 调试软件中, 调试合适的闭环力控 PID 参数
			(4) 闭环力控 IDE 程序中设置的目标位置不合适	目标位置应设置为大于接触点处的位置 (控制器会根据设定的压力阈值自动从位置模式切换到力控模式)
1058	Undervoltage detected on 10.5v supply (8v threshold)	在 10.5v 电源 (8v 阈值) 上检测到电压不足	(1) 逻辑电源供电线路异常	检查逻辑电源输入接线
			(2) 控制器内部电路异常	请联系技术支持或销售, 返厂维修
1059	Over current detected on motor C	在 C 轴电机上检测到电流过流	(1) C 轴相电流限制值设置得过小	在 PCSuite 调试软件中, 对 C 轴配置正确的相电流限制值
			(2) C 轴电机类型设置错误	在 PCSuite 调试软件中, 对 C 轴配置正确的电机类型
			(3) C 轴电机绕组异常	用万用表测量电机绕组各相线之间的电阻, 如果电阻值与电机规格书中的额定值差异较大, 则需更换电机
1060	Product temperature (BoardTemp) too high	产品温度 (BoardTemp) 过高	(1) 连续运行节拍过快	降低加速度和速度, 或增加停顿时间
			(2) 运行环境温度过高	改善驱动器所在机柜的散热
			(3) 驱动器之间的安装间距过小	增加驱动器之间的间距
			(4) 测温传感器损坏	请联系技术支持或销售, 返厂维修
1061	Other gantry member axis got motor off	其他龙门参与轴电机下使能	龙门模式下, 某个参与轴发生报警	查看龙门模式参与轴的故障信息, 排查原因
1062	External request to Fault with Controlled Stop: stop ended and motor was disabled	外部请求的受控停止故障: 电机已停止并且下使能	配置了某个数字量输入点为“Fault (controlled stop first)”且收到外部设备报警信号	检查外部设备报警信息并清除其报警信号输出

6 附录

关于 Agito 各功能手册可参阅以下资料：

手册名称	语言	内容
《Agito-AG□xxx-Product-Manual-Rev.xx》	中/英	主要介绍各系列控制器硬件参数、基本配置流程等内容；
《Agito PCSuite 快速入门手册》	中	主要介绍 Agito PCSuite 基本使用方法，包含电机参数设置、PIV 调参等内容；
《(中文版)PCSuite UM_3.0-CN》	中/英	Agito PCSuite 说明文档；
《Akribis-Agito Controller Keywords Reference》	英	Agito 控制器关键字说明；
《Communication Protocols》	英	Agito 通讯协议说明文档；
《ErrLog User's Manual》	英	Agito 控制器报错信息说明文档
《Agito 搭配步进电机使用手册》	中	Agito 控制步进电机/驱动器使用文档；
《Agito 全闭环控制使用手册》	中	Agito 全闭环（双回路）控制使用方法；
《Agito 回零使用手册》	中	回零配置手册；
《Agito 误差补偿使用手册》	中	误差补偿配置手册；
《Agito CNC（插补运动）使用手册》	中	插补运动使用方法；
《Agito PEG & Position Lock 使用手册》	中	PEG（飞拍）和 Position Lock（探针）使用手册；
《Agito IDE+编程环境使用手册》	制作中	Agito PCSuite IDE+编程环境使用手册；
《Agito 力控使用手册》	制作中	力控功能使用手册；
《Agito 龙门模式使用手册》	制作中	龙门模式调试手册；
《Agito 模拟输入电流控制使用手册》	中	Agito 模拟输入电流控制使用手册；
《Agito 电子齿轮和凸轮使用手册》	制作中	电子齿轮（GEAR）和凸轮（ECAM）使用手册；
《Agito 脉冲方向控制使用手册》	制作中	Agito 脉冲方向控制使用手册；
《Agito UPM（超精模式）使用手册》	制作中	Agito 超精模式使用手册；
《AAMotion 手册》	中	API 接口使用手册；

