



Agito 搭配步进电机使用手册 v2.1



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	2023/4/27
1.1	修改第二篇中的“Stepper resolution”描述	2023/7/27
2.0	更新整体排版格式，增加部分内容	2023/11/6
2.1	更新闭环步进调参步骤	2024/1/19

※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	4
1.1	关于手册	4
1.2	Agito 控制器控制步进简介	4
2	操作步骤	5
2.1	PD 控制（脉冲方向）	5
2.2	直驱开环步进	9
2.3	直驱闭环步进	12
2.4	特殊 PD 模式	15
3	相关关键字介绍	19

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 Agito 控制器搭配步进电机使用方法，手册中仅详细介绍与步进电机相关的配置内容，其他参数设置或功能可参阅《Agito 快速入门手册》及其他相关功能手册中的详细介绍，本文档将不再赘述。

1.2 Agito 控制器控制步进简介

步进电机由于成本低、控制简单，在自动化系统中应用广泛。Agito 控制器支持常用的脉冲方向（PD）方式控制步进驱动器驱动步进电机运动，同时也支持直接驱动步进电机运动（纯控制器需搭配驱动器使用），用户根据系统应用需求选择对应的控制方式。

2 操作步骤

不同的控制方式系统接线将会有区别，用户根据应用需求选择相应的硬件及接线。

2.1 PD 控制（脉冲方向）

PD 模式为目前最常用的步进电机控制方式，该模式需要外接第三方步进驱动器支持，但此方式步进电机选型灵活，可根据步进驱动器支持的电机类型以及输出扭力自由选择用户所需的电机型号。

值得注意的是：PD 控制模式需要占用控制器运动轴，尽管该轴主编码器和动力接口未接任何实际电机，而是使用虚拟控制轴，该轴不能再连接其他电机同时使用，在 CNC 或齿轮等模式下可实现多轴协调运动。

以下将详述使用 Agito 控制器控制步进电机的步骤：

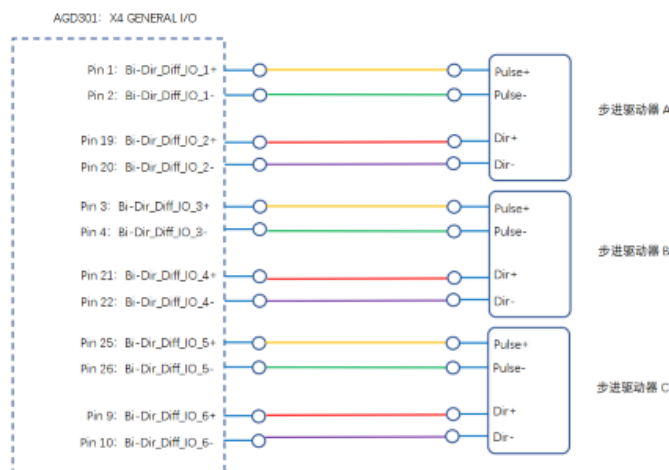
- **第 1 步：电气接线**

（以下以 AGD301 作为示例，其他可参考）

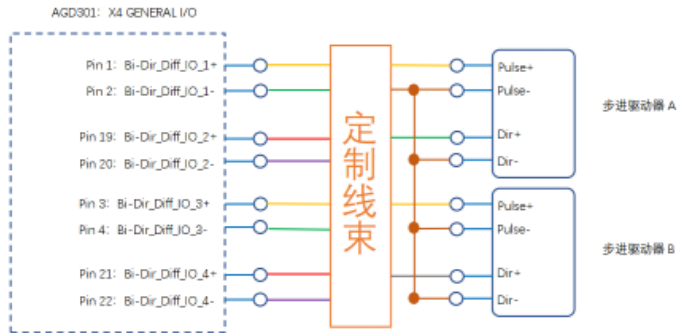


控制器差分口输出为±2.5V 差分脉冲方向信号，如用户端步进驱动器是单端 0/5V 单端数字信号有效，则需要使用定制线束以适配单端信号的使用。

- ◆ **标准±2.5V 差分信号：**



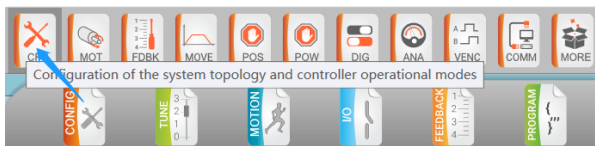
◆ 单端 0/5V 数字信号



注：如需使用定制线束请联系 Agito-Akribis 获取相关支持

• 第 2 步：参数配置

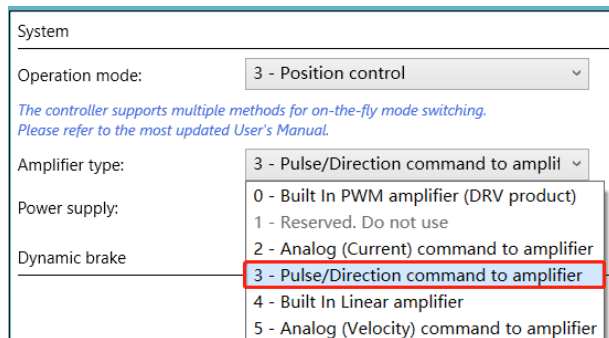
◆ 控制模式



打开 PCSuite，切换菜单栏到 CONFIG ---> Basic Configuration 界面：

Operation mode: 选择 “3-Position control” ，

Amplifier type: 选择 “3-Pulse/Direction command to amplifier” ；



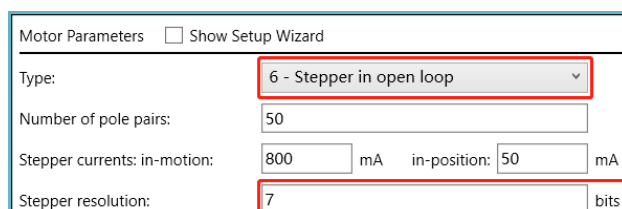
◆ 电机参数

Type: 电机类型选择 “6-Stepper in open loop” ；

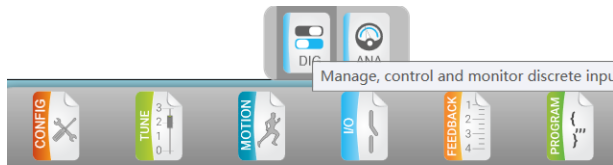
Stepper resolution: 细分位数，如设值为 7bits，表示 6,400counts/Rev，即电机转动一圈走 6,400 步($2^7 \text{bits} * 50 \text{poles} = 6,400$)，最大可设置为 16bits；

(注：步进细分数= $2^{\text{StepBits}} * \text{PolePrs}$)

值得注意的是：此处的细分数并不影响电机的实际运动，但只有当该处的细分数和步进驱动器细分设置一致时（注：通常步进驱动器的细分通过拨码开关设置），控制器和电机的运行周期才能同步；

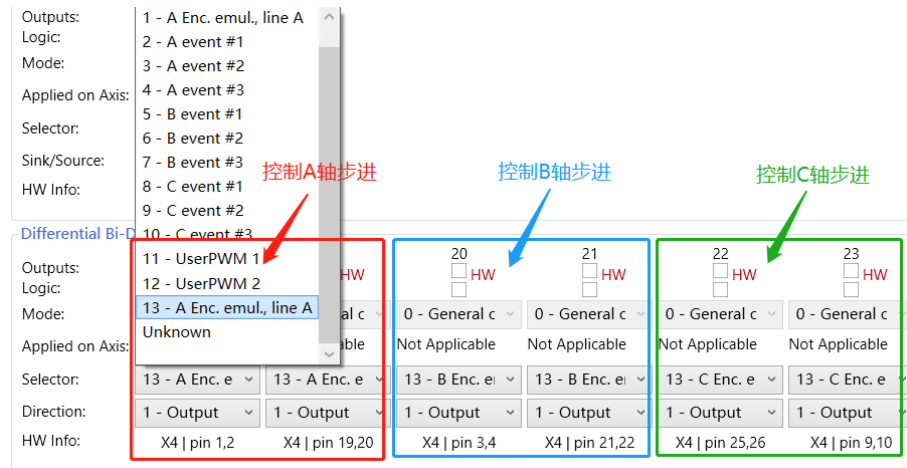


◆ I/O 设置



切换菜单栏到 I/O --->Discrete Outputs 界面（以下以 AGD301 为示例）：

将对应双向差分数字输出设置为 Output 模式，然后根据输出轴选择对应的输出模式为“13- [Axis] Enc.emul,line A 或 B”，



HW: This output pin is now used for dedicated hardware function (see Selector). The state of the output is not affected by DOutPort[].

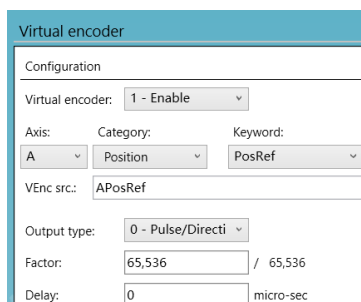
被控驱动器	所双向差分口	方向	功能选项
A 轴驱动器	18(Bi-Dir_Diff_IO_1)	1-Output	13-A Enc.emul line A
	19(Bi-Dir_Diff_IO_2)	1-Output	13-A Enc.emul line B
B 轴驱动器	20(Bi-Dir_Diff_IO_3)	1-Output	13-B Enc.emul line A
	21(Bi-Dir_Diff_IO_4)	1-Output	13-B Enc.emul line B
C 轴驱动器	22(Bi-Dir_Diff_IO_5)	1-Output	13-C Enc.emul line A
	23(Bi-Dir_Diff_IO_6)	1-Output	13-C Enc.emul line B

◆ 虚拟编码器设置

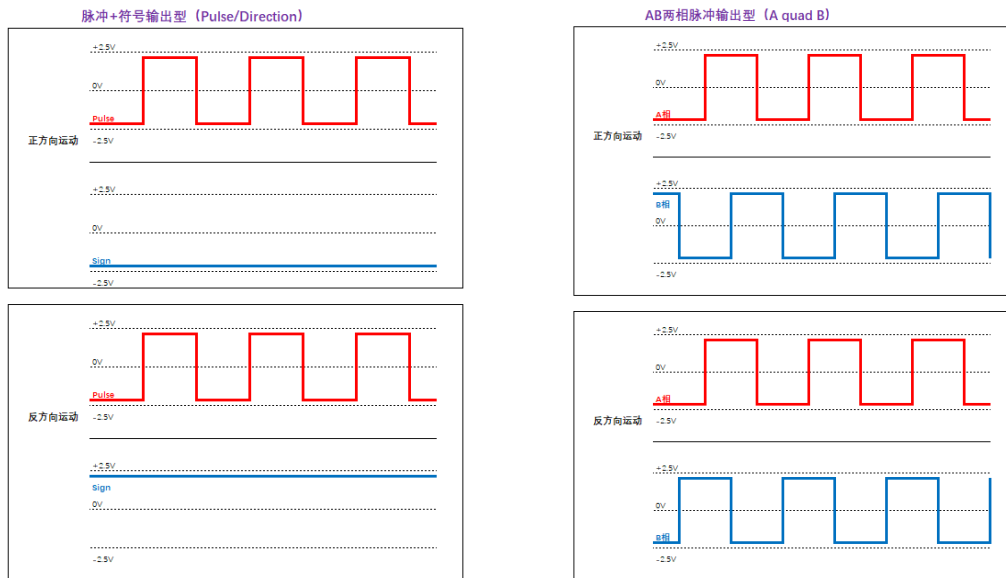


切换菜单栏到 CONFIG--->Virtual encoder 界面（以下以 AGD301 为示例）

切换到虚拟编码器配置界面，VEnc source 设置为“APosRef”（B、C 轴依次类推），Output type 可以选择“0-Pulse/Direction”或者“1-A quad B”（根据用户步进驱动器可接受的控制信号类型选择，见下页信号类型），然后将 Virtual encoder 选择为“1-Enable”开启虚拟编码器功能：

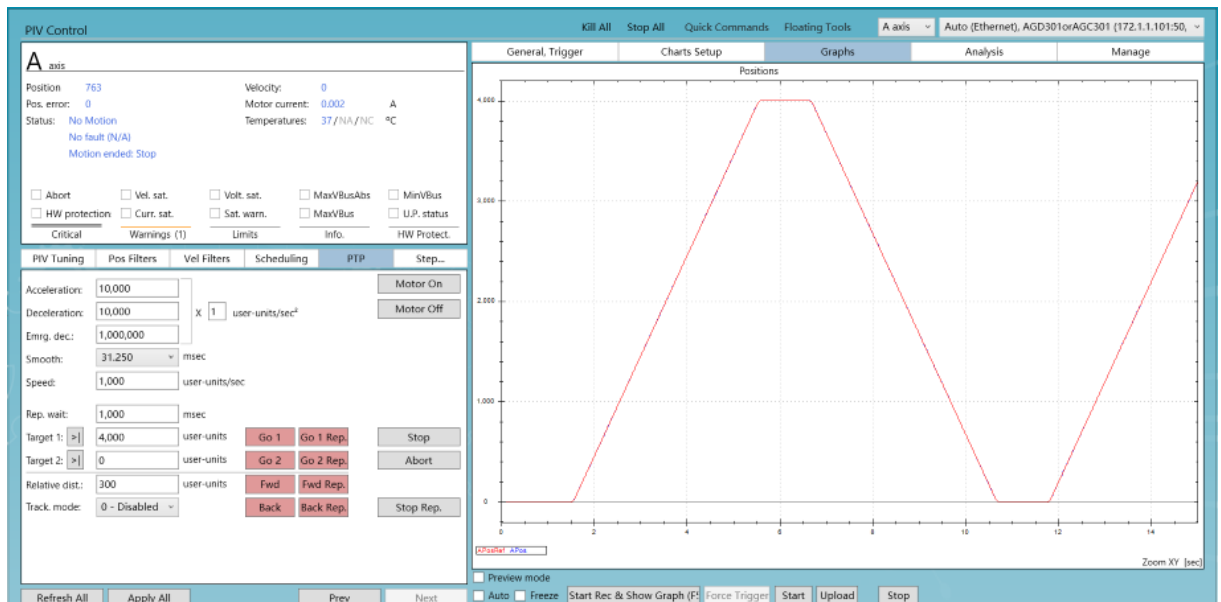


PD 信号输出类型 (Output Type) :



• 第 3 步：运动控制

完后以上设置后即可进行相应的运动控制；



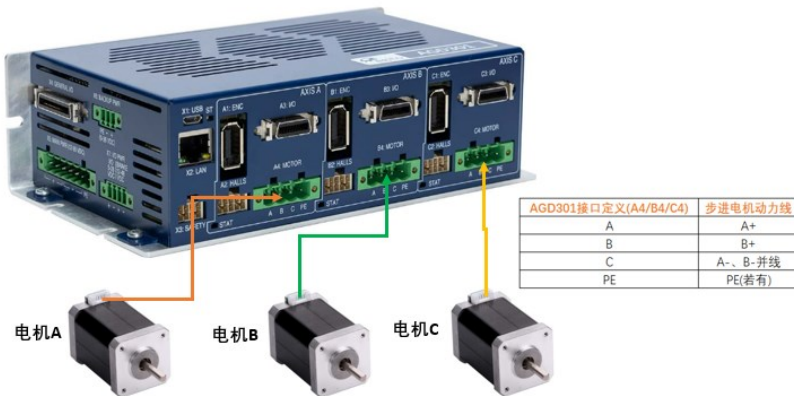
2.2 直驱开环步进

Agito 大部分系列控制器支持直接驱动步进电机（AGD 系列和 AGA 系列，其中 AGA 系列需要搭配 AGM800 控制器使用），当用户速度和精度要求不高的情况下可以使用该驱动方式。

- **第 1 步：系统框图**

（以下以 AGD301 作为示例，其他可参考）

AGD301控制器



注意：目前仅支持 2 相步进电机

- **第 2 步：参数配置**

- ◆ 控制器模式

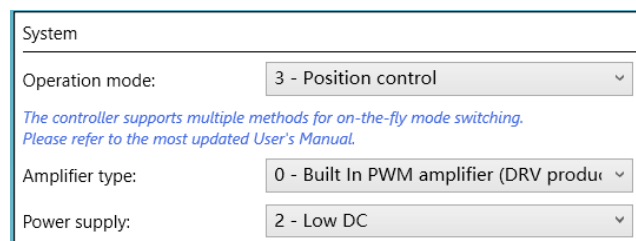


打开 PCSuite，切换菜单栏到 **CONFIG ---> Basic Configuration** 界面：

Operation mode: 选择“3-Position control”，

Amplifier type: 选择“0 - Built In PWM amplifier (DRV product)”，

Power supply: 选择“2- Low DC”；



- ◆ 电机参数

Type: 电机类型，选择“6-Stepper in open loop”；

Number of pole pairs: 电机极对数，根据步进电机手册来设置，典型步距角为 1.8° 的步进电机极对数为 50Pairs（注：步距角=360° / (2*N*P)，其中 N 为极对数，P 为电机相数）；

Stepper currents: in-motion:表示电机运动时的电流，即额定电流，注意为了保护电机请勿超过电机手册中额定电流值；in-position:表示电机空闲时的保持电流，常见设为额定电流值的 90%、50%、30%，根据实际应用设置；

Stepper resolution: 细分位数, 例如值设为 7bits, 表示 6,400counts/Rev, 即电机运动一圈走 6,400 步($2^{7bits} * 50poles = 6,400$), 最大值为 16bits;

(注: 步进细分数 = $2^{StepBits} * PolePrs$)

Motor Parameters <input type="checkbox"/> Show Setup Wizard	
Type:	6 - Stepper in open loop
Number of pole pairs:	50
Stepper currents: in-motion:	200 mA in-position: 50 mA
Stepper resolution:	7 bits

◆ 电机保护

切换菜单栏到 **CONFIG ---> Protections** 界面:

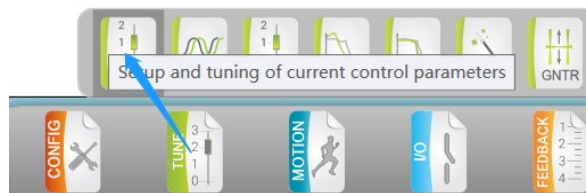
保护参数的详细设置请参阅《Agito 快速入门手册》中的内容介绍, 此处仅介绍与开环步进功能相关的设置。由于是开环控制未接入物理编码器, 因此需将主、辅编码器信息屏蔽:

Mask protections	
<input checked="" type="checkbox"/> Mask main encoder errors	<input checked="" type="checkbox"/> Mask auxiliary encoder errors (if exist)

• 第 3 步: 电流环调参

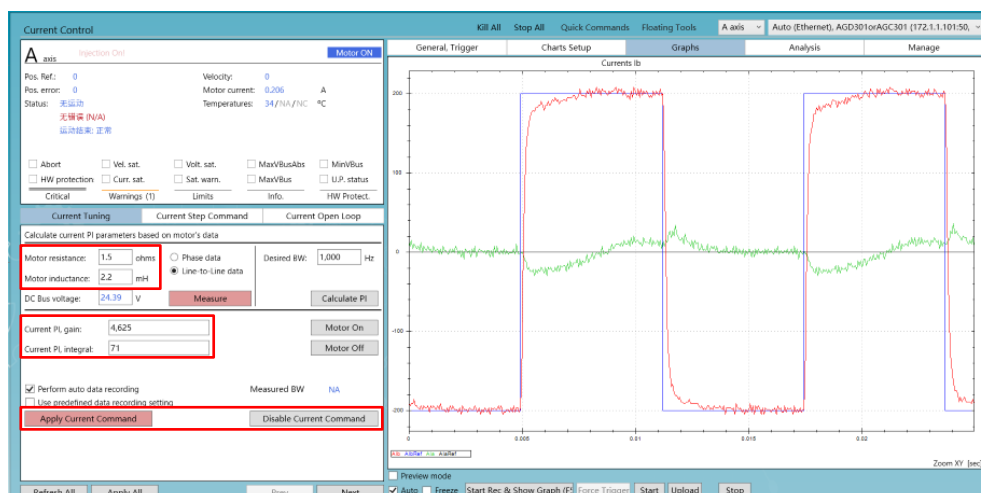
为了保证电机具有良好的运动性能, 需要对电流环进行调参;

切换菜单栏到 **TUNE ---> Current Tuning** 界面:



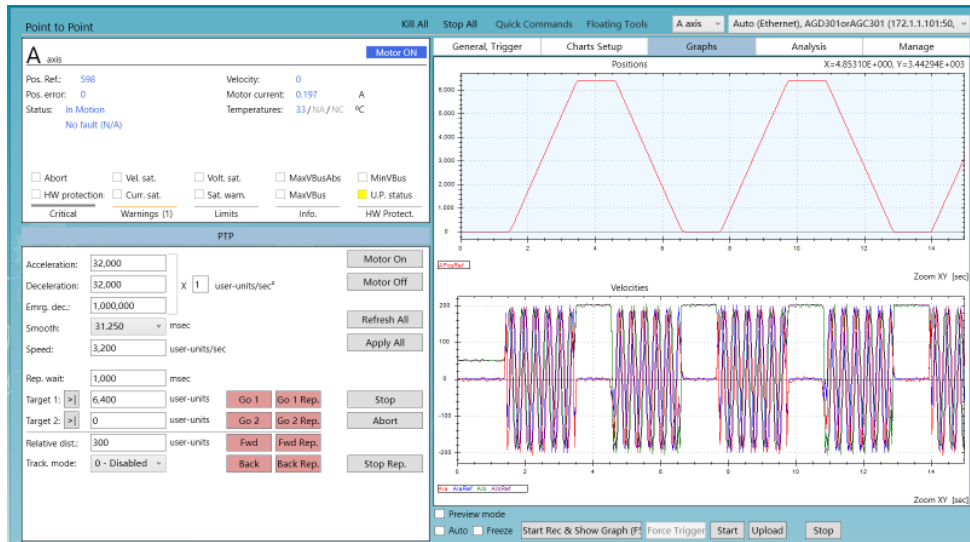
Motor resistance/Motor inductance: 电机电阻/电感, 根据步进电机手册填入;

Current PI, gain/Current PI, integral: 填入电流环 PI 参数 (推荐值从 100、10 开始以避免增益过大电机产生啸叫声), 设定好参数后, 点击 “Apply Current Command” 开始注入电流, 滚动鼠标滚轮并点击回车键可实时改变参数值并动态响应, 调整参数的同时观察右侧图像中激励电流和响应电流(Ia/IaRef、Ib/IbRef), 使其最大程度相吻合, 点击 “Disable Current Command” 可停止注入电流;



- 第 4 步：运动控制

完后以上设置后即可进行相应的运动控制；



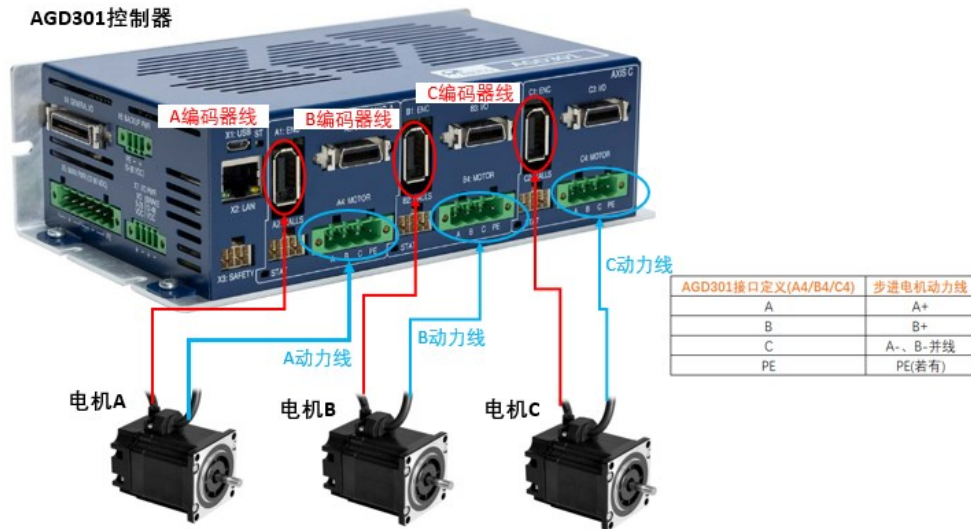
2.3 直驱闭环步进

Agito 大部分系列控制器都支持直接驱动闭环步进电机（AGD 系列和 AGA 系列，其中 AGA 系列需要搭配 AGM800 控制器使用），当用户速度和精度要求不高的情况下可以使用该驱动方式。

• 第 1 步：系统框图

（以下以 AGD301 作为示例，其他可参考）

◆ 场景 1：步进电机+旋转编码器

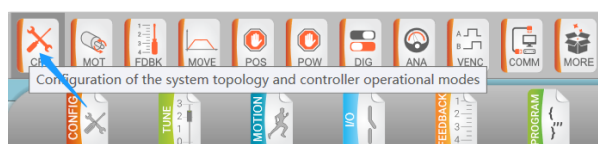


◆ 场景 2：步进电机+丝杆光栅



• 第 2 步：参数配置

◆ 控制模式



打开 PCSuite，切换菜单栏到 **CONFIG ---> Basic Configuration** 界面：

Operation mode: 选择 “3-Position control” ，

Amplifier type: 选择 “0 - Built In PWM amplifier (DRV product)” ，

Power supply: 选择 “2- Low DC” ；

System	
Operation mode:	3 - Position control
<i>The controller supports multiple methods for on-the-fly mode switching. Please refer to the most updated User's Manual.</i>	
Amplifier type:	0 - Built In PWM amplifier (DRV produ
Power supply:	2 - Low DC

◆ 电机参数

点击上一步 “Apply and next” 进入电机参数设置页面：

Type: 电机类型选择 “7-Stepper in closed loop” ；

Number of pole pairs: 步进电机极对数，根据步进电机手册设置，典型两相步距角为 1.8° 步进电机的极对数为 50Pairs；

（注：步距角= $360^\circ / (2*N*P)$ ，其中 N 为极对数，P 为电机相数）

Stepper resolution: 闭环模式下 Stepper resolution 作为数字细分，其值越大电机运动将会越平滑，最大为 16bits（注意：在电流环调参时请将其值设为 2bits）；

Type:	7 - Stepper in closed loop		
Number of pole pairs:	50		
Stepper currents: in-motion:	800 mA	in-position:	50 mA
Stepper resolution:	16 bits		

◆ 编码器反馈

编码器类型支持 ABI 增量式、EnDat 2.2 和 BISS-C 绝对式及 SinCos 模拟式，详细参数设定请参阅《Agito 快速入门手册》文档中 2.1.2 章节中关于编码器部分的详细描述，此处仅简要介绍关于分辨率（Resolution）参数的设定，通常为以下 2 种使用场景：

步进电机+旋转编码器：**Resolution=步进一圈对应的脉冲数**

步进电机+丝杆光栅（直线或圆弧）：**Resolution=步进电机走一圈对应的线位移/光栅分辨率；**

Dual loop			
Dual Loop mode:	0 - No dual loop	?	
Main encoder	Reading: 4,082	Last index: 3,301	
Type:	1 - Incremental	<i>Please properly configure max. speed limitation at Config/Pos window</i>	
Sub Type:	0 - A quad B encoder		
Resolution:	4,000	counts / rotation (or pitch)	?
Invert direction:	0 - No	Max. speed:	10 - 5,000,000 ?
Modulus range:	0	user-units	
Emulation divider value:	0	Direction:	Normal
User units' factor:	65,536	/ 65536	counts/user-units

- ◆ 电机保护

切换菜单栏到 **CONFIG ---> Protections** 界面:

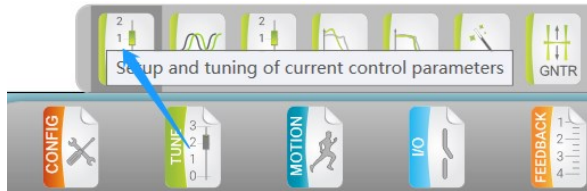
保护参数的详细设置请参阅《Agito 快速入门手册》中的内容介绍:

- **第 3 步: PIV 调参**

为了保证电机具有良好的运动性能, 需要对运动控制参数进行调整:

- ◆ 电流环

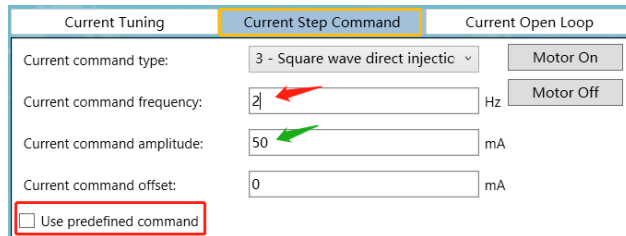
切换菜单栏到 **TUNE ---> Current Tuning** 界面:



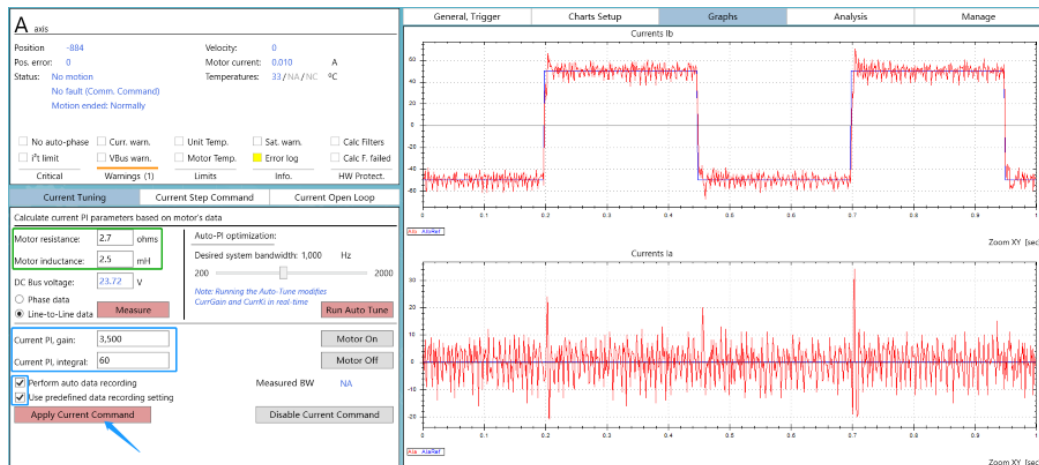
Motor resistance/Motor inductance: 电机电阻/电感, 根据步进电机手册填入;

(注意: 在电流环调参时, 请将上页中电机参数的“Stepper Resolution”值设为 2bits)

设置抓图参数: 先将“Use predefined command”勾选去掉, 然后将电流频率设为 2Hz, 电流设为 in-motion 电流的 5~10%之间;

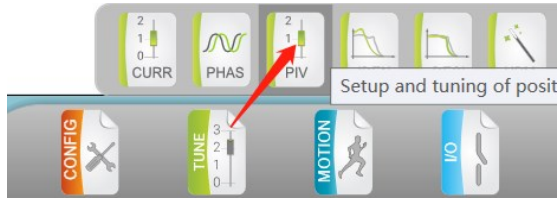


Current PI, gain/Current PI, integral: 填入电流环 PI 参数 (推荐值从 100、10 开始以避免增益过大电机产生啸叫声), 设定好参数后, 点击“Apply Current Command”开始给入电流指令, 滚动鼠标滚轮并点击回车键可实时改变参数值并动态响应, 调整参数的同时观察右侧图像中激励电流和响应电流 (I_a/I_{aRef} 或 I_b/I_{bRef}), 使其最大程度相吻合, 点击“Disable Current Command”可停止给入电流;

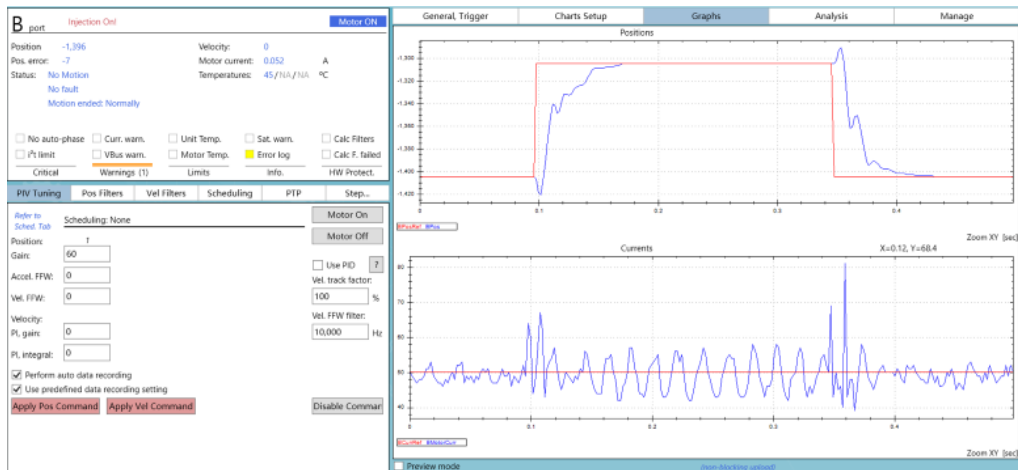


- ◆ 位置环

切换菜单栏到 **TUNE ---> PIV Tuning** 界面:

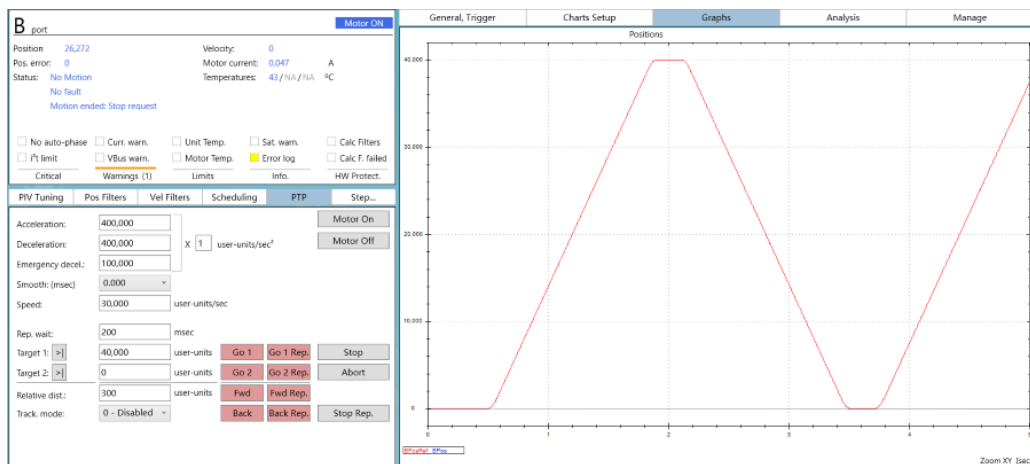


在调参之前确保电流环时修改的“Stepper Resolution”值为用户所需的 bit 数，将速度环参数 PI gain 和 PI integral 都设为“0”，然后调整 Position 中的 Gain 值，点击“Apply Pos Command”开始给入位置指令激励，滚动鼠标滚轮并点击回车键可实时改变参数值并动态响应，调整参数的同时观察右侧图像中位置指令(PosRef)和位置响应(Pos)，使其最大程度相吻合，点击“Disable Command”可停止给入指令；



• 第 4 步：运动控制

完后以上设置后即可进行相应的运动控制；

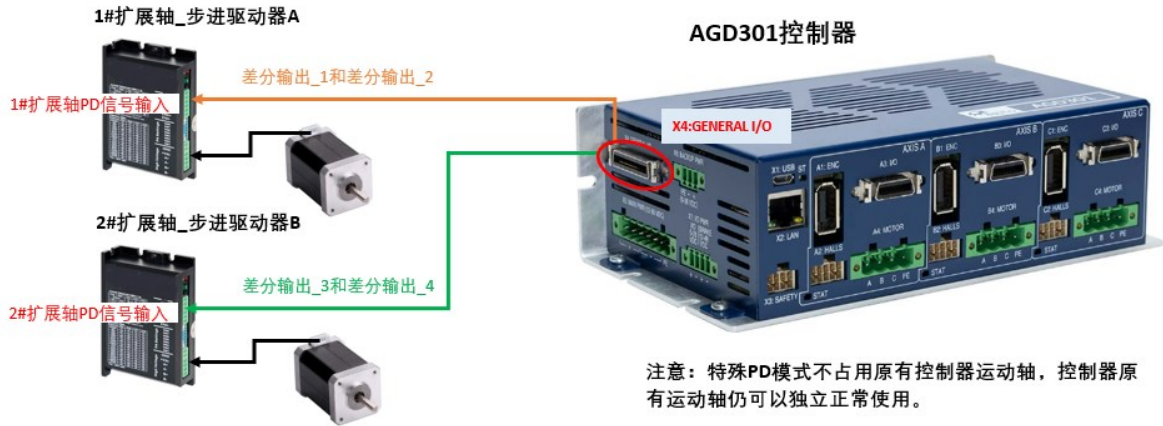


2.4 特殊 PD 模式

AGD301 控制器支持一种特殊扩展 PD 模式，该模式在不占用原有的 3 个运动轴的前提下，可使用 4 路差分数字输出信号最大再扩展 2 路独立的 PD 信号用于控制步进驱动器，目前仅支持 AGD301-ET-2D05 型号。

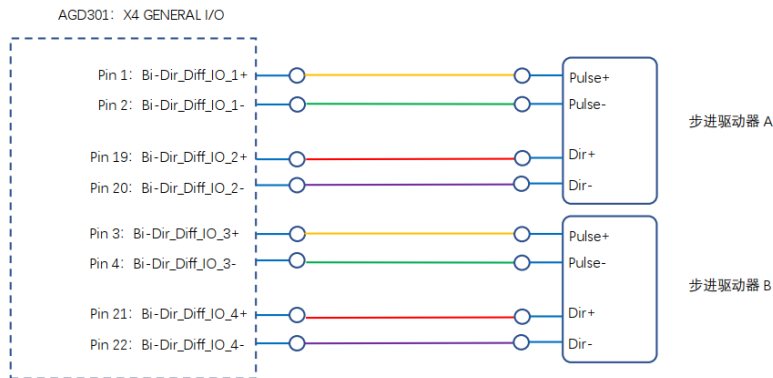
值得注意的是：该特殊 PD 模式需要特殊版本固件支持，如需使用请联系 Agito-Akribis 获取相关支持。

• 第 1 步：系统框图

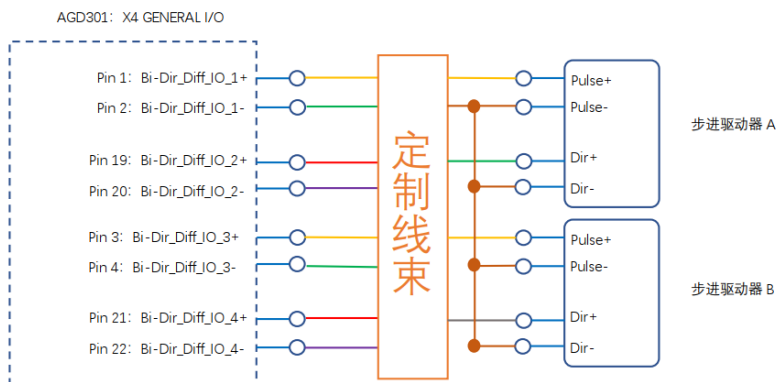


控制器输出为±2.5V 差分脉冲方向信号，如用户端步进驱动器是单端 0/5V 单端数字信号，需要使用定制线束以适配单端信号的使用。

◆ **标准±2.5V 差分信号：**



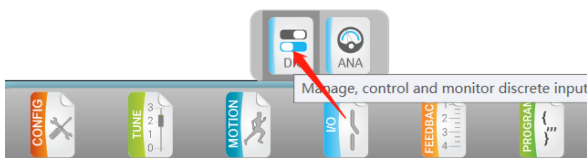
◆ **单端 0/5V 数字信号**



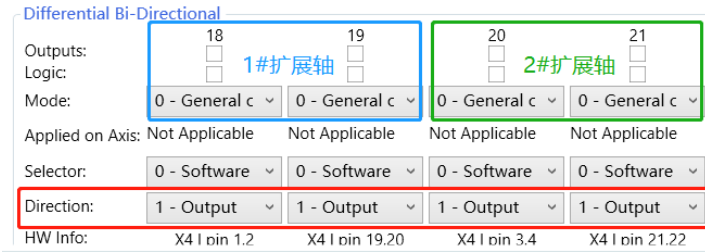
注：如需使用定制线束请联系 Agito-Akribis 获取相关支持

• **第 2 步：参数配置**

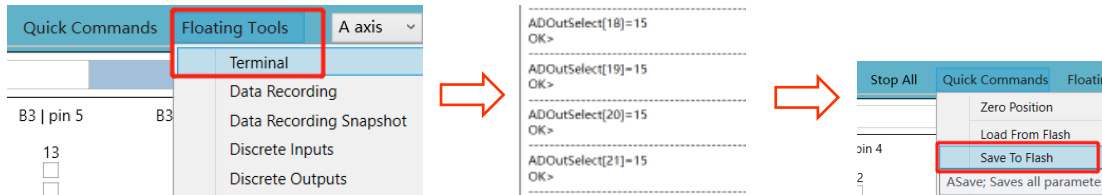
切换菜单栏到 I/O --->Discrete Outputs 界面：



将对应双向差分数字口设置为 Output 模式，扩展 1#步进轴对应 Bi-Dir_Diff_IO_1+/-、Bi-Dir_Diff_IO_2+/-，扩展 2#步进轴对应 Bi-Dir_Diff_IO_3+/-、Bi-Dir_Diff_IO_4+/-；



打开 Terminal，分别将对应数字输出口的 Selector 值设置为 15（如上图，18、19 对应 1# 扩展轴，20、21 对应 2# 扩展轴），设置完成后，点击菜单栏“Save To Flash”保存参数到控制器，并断电重启生效；



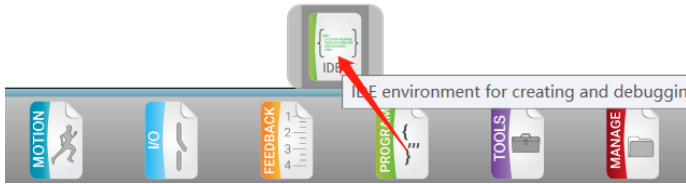
• 第 3 步：运动控制

扩展轴 PD 模式在 PCSuite 中无运动控制 UI 界面，只能在 IDE 编程环境中或 ACSII 通讯通过操作 FPGA 对应寄存器中的值来进行运动控制；

十进制指令地址表：

寄存器	1#扩展轴地址	2#扩展轴地址	访问类型	Default Value	Description
脉宽	66	118	只写	50(1us)	脉宽=0.02μs*value
脉冲间隔	21	277	只写	50(1us)	脉冲间隔=0.02μs*value
目标位置： (高位)	67	119	只写	0	DSP 使用关键字： PDOutTrgtPos
目标位置： (低位)	68	120	只写	0	DSP 使用关键字： PDOutBTgtPos
当前位置： (高位)	69	123	只读	0	DSP 使用关键字： PDOutFBAbsPos
目标位置： (低位)	70	124	只读	0	DSP 使用关键字： PDOutBAbsPos
Command Reg	71	121	只写	0	值=0，空闲状态， 值=1，为跟随模式， 值=2，将目标位置置 0；
方向延时	72	122	只写	0	Delay time=0.02μs*value
PDConfig ^[1]	23(14 th bit)	279(14 th bit)	只写	0	值=0 输出为脉冲方向形式， 值=1 输出为 AqB 形式；
Dir Polarity	23(15 th bit)	279(15 th bit)	Write Only	0	
Pulse Polarity	23(16 th bit)	279(16 th bit)	Write Only	0	

以下介绍在 IDE 编程环境中控制电机运动的示例：



◆ 扩展关键字：

关键字	描述
PDOOutTrgtPos	1#扩展轴写入目标位置
PDOOutFBAbsPos	1#扩展轴读取当前位置
PDOOutBTgtPos	2#扩展轴写入目标位置
PDOOutBAbsPos	2#扩展轴读取当前位置

◆ IDE 程序

在 FPGA 对应地址中写入值需要在 DebugData 里写入 3 次，示例程序中将该步骤封装成一个函数（function），在使用时只要调用该函数即可将值写入 FPGA 对应地址；

以下程序实现了 1#、2#扩展轴分别运动到 20000 的位置处：

```
main([10,30],[5,20],[800,1000])

    writeToFPGA(66,50000) //将1#扩展轴脉宽设为1ms (Pulse High Duration), 脉宽=0.02μs*50000=1ms;
    writeToFPGA(21,50000) //将1#扩展轴脉冲间隔设为1ms (Pulse Low Duration), 通常将占空比设为50%;
    writeToFPGA(71,1) //将1#扩展轴设为运动状态（跟随模式）;
    APDOutTrgtPos = 20000 //设定1#扩展轴目标位置, 跟随模式下给定目标位置后电机即开始运动;

    AWaitTime, 1000
    writeToFPGA(118,50000) //将2#扩展轴脉宽设为1ms (Pulse High Duration);
    writeToFPGA(277,50000) //将2#扩展轴脉冲间隔设为1ms (Pulse Low Duration), 通常将占空比设为50%;
    writeToFPGA(121,1) //将2#扩展轴设为运动状态（跟随模式）;
    APDOutBTgtPos = 20000 //设定2#扩展轴目标位置, 跟随模式下给定目标位置后电机即开始运动;

endofmain

function: writeToFPGA(writeAddress,writeValue) //将FPGA值写入操作封装成函数;
    ADebugData[34]=writeAddress
    AWaitTime, 1
    ADebugData[35]=writeValue
    AWaitTime, 1
    ADebugData[33]=1
    AWaitTime, 1
endoffunc
```

以上程序中将脉宽和脉冲间隔都设为 1ms，即将占空比设为 50%，

位置设定： *PDOOutTrgtPos* 和 *PDOOutBTgtPos* 分别为 1#扩展轴和 2#扩展轴的目标位置指令。当寄存器地址为 71 的值被赋为 1 时，即开启 1#扩展轴为跟随模式，此时只要给 *PDOOutTrgtPos* 变量写入对应目标位置值时，电机就开始运动，同样地，给 2#扩展轴对应地址 121 赋值为 1 时，开启 2#扩展轴为跟随模式；

速度设定： 步进电机速度由控制器输出脉冲频率决定，频率 $f = 1 / (\text{脉宽} * 2)$ ，占空比为 50%，为确保控制精度，建议输出频率为 10KHz 以下，即脉宽 50us 以上（50us 对应寄存器值为 2500 以上），换算成速度为 $Vel = f / \text{subdiv}$ ，（*vel* 单位为 rev/s，*f* 单位为 Hz，*subdiv* 为步进驱动器细分）；

值得注意的是：该模式无置 0 命令，即不能将当前位置置 0。

3 相关关键字介绍

Agito 关键字不仅可以在 PCSuite Terminal 终端及 IDE 编程环境使用，用户也可通过字符串或 ASCII 通讯使用，此处仅介绍与步进控制相关关键字。

关键字	描述
AmpType	驱动类型， AmpType =0 表示 PWM 输出， AmpType =3 表示 PD 输出；
StepBits	步进细分位数，步进细分数= $2^{StepBits} * PolePrs$ ，步进细分数表示步进电机一圈对应的步数；
PolePrs	电机极对数，典型两相步距角为 1.8° 步进电机的极对数为 50Pairs， (步距角 = $360^\circ / (2*N*P)$ ，其中 N 为极对数，P 为电机相数)；
StepInMotCurr	电机运动时的电流，即额定电流，注意为了保护电机请勿超过电机手册中额定电流值；
StepInPosCurr	电机使能的保持电流，常见为额定电流值的 90%、50%、30%；
DOutSelect	DOutSlect[i], i 为差分输出口编号，不同控制器有区别， DOutSelect[i]=13 表示启用标准 PD 模式（标准固件）， DOutSelect[i]=15 表示启用扩展 PD 模式（定制固件）；
VEncSrc	虚拟编码器信号源，通常使用 PosRef；
PDOutTrgtPos	扩展特殊 PD 模式下，1#扩展轴写入目标位置（定制固件）；
PDOutFBAbsPos	扩展特殊 PD 模式下，1#扩展轴读取当前位置（定制固件）；
PDOutBTgtPos	扩展特殊 PD 模式下，2#扩展轴写入目标位置（定制固件）；
PDOutBAbsPos	扩展特殊 PD 模式下，2#扩展轴读取当前位置（定制固件）；

