



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 产品系列

Agito 控制器回零使用手册 v1.1



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	2023/5/15
1.1	修改 2.1-1-①中限位信号的描述，增加关于误差补偿功能的设置	2023/8/28
1.2	修改手册页脚	2023/9/6

※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，如需最新本本文档，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	回零简介	4
2	操作步骤	5
2.1	回零界面介绍	5
2.2	常用回零方式介绍	9
2.2.1	锁定 index 位置回零（带限位开关）	9
2.2.2	锁定 index 位置回零（带机械硬限位）	13
2.2.3	搜寻 index 位置回零（无限位）	14
2.2.4	硬限位回零（无 index 和限位开关）	16
2.2.5	限位开关信号回零	18
2.2.6	回零开关信号回零	20
2.3	关于误差补偿（Error Mapping）在回零步骤中设置	21
2.4	导出导入回零程序	22
3	回零相关关键字介绍	23
4	PCSuite IDE+编程环境中回零	24
5	常见问题解答（FAQ）	26

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 运动控制器的回零（或称回原点）使用方法。

手册中仅详细介绍与回零相关的配置内容，其他参数设置请参阅《Agito 快速入门手册》中的详细介绍，本文档将不再累述。

值得注意的是回零动作需要在基本参数设置完成并确保电机可以正常运动之后才能进行。

1.2 回零简介

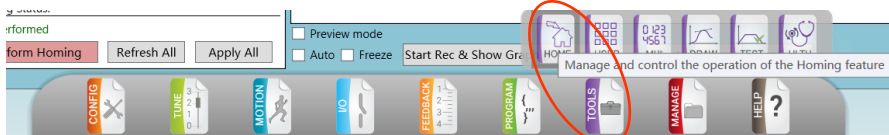
当使用增量编码器时，上电伺服无法获取电机位置，因此每次上电需要进行回零操作。Agito 运动控制器支持多种回零方式，用户可根据实际应用场景自定义回零步骤及方式。

2 操作步骤

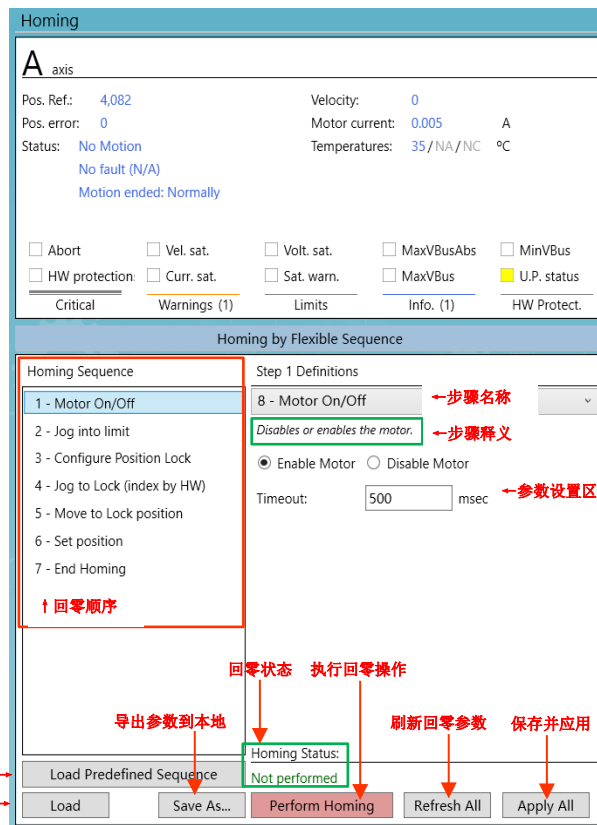
回零步骤通过 Agito PCSuite 中的 Homing 界面来设置相关参数，用户根据自己的需求选择合适的回零方式。执行回零之前确保电机相关参数已配置正确并已调参，电机可以正常运动。

2.1 回零界面介绍

1. 点击 PCSuite 菜单栏到 **Tools--->Homing**，切换到回零界面。

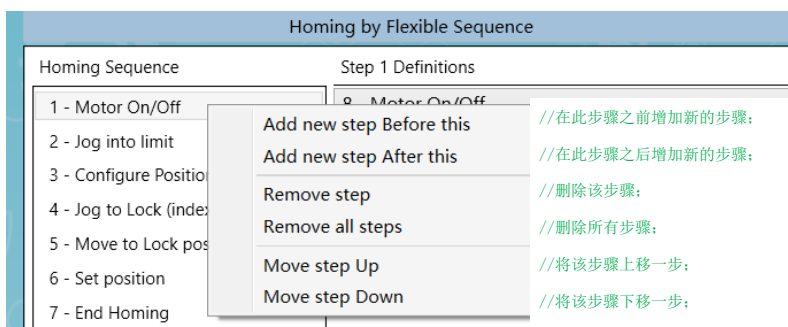


左侧用于用户定义回零步骤及顺序，右侧定义每一步的具体参数，在每一步骤下方都会给用户提供一个简要的解释供参考使用（如绿色方框标识处）。



2. 设定回零步骤：

- 鼠标右键点击左侧回零的某一步可以编辑回零步骤，最多可以设置 15 条步骤（最后一步必须是“End Homing”）。



- 鼠标左键点击左侧回零的某一步可以编辑每一步对应的回零参数。

共有 18 种类型可供选择（由于 PCSuite 版本和控制器型号及固件版本差异回零类型可能存在差异）：

①1-Jog into limit: 运动到限位开关位置（需要电机已安装物理限位开关，并且正确接线到数字输入口，该输入口配置为 RLS 或 FLS），正限位（FLS）或负限位（RLS），方向由速度方向决定，正方向（FLS）运动速度为正值，负方向（RLS）运动速度为负值。

1 - Jog into limit

Command the motor to move until hitting RLS or FLS. Motion direction is according to the Speed sign.

Speed:	<input type="text" value="10,000"/>	counts/sec	//运动速度，正负决定运动方向；
Accel/Decel:	<input type="text" value="100,000"/>	counts/sec ²	//加速度，只需设大小；
Emergency Decel:	<input type="text" value="1,000,000"/>	counts/sec ²	//急停减速度，只需设大小；
Timeout:	<input type="text" value="30,000"/>	msec	//本步骤执行延时，设置该步骤完成时间阈值，若在该时间段内未检测到限位信号停止，控制器将停止回零程序并输出报警，回零失败，因此设置时该时间必须大于运动到限位处的时间。

②2-Check that out of limits: 检查 RLS 和 FLS 是否都没有被激活。当某一个被激活时退出回零步骤。

2 - Check that out of limits

Check if indeed both RLS and FLS are not activated. Exit the homing process if one of them is activated.

③3-Relative PTP: 使电机相对于上一步位置走一段相位位移。

3 - Relative PTP

Command the motor to perform a relative point to point motion.

Speed:	<input type="text" value="5,000"/>	counts/sec	//运动速度，正负决定运动方向；
Accel/Decel:	<input type="text" value="500,000"/>	counts/sec ²	//加速度，只需设大小；
Relative distance:	<input type="text" value="200"/>	counts	//相对位移，正负决定位移方向（同时和速度正负相关）；
Timeout:	<input type="text" value="8,000"/>	msec	//本步骤执行延时，设置该步骤完成时间阈值，若在该时间段内未走完设定的相对位移距离，运动将停止并输出报警，退出回零程序，回零失败，因此设置时该时间必须大于运动到相对位移处的时间。

④4-Search for index: 以设定的速度方向去搜寻 index 信号（或称 Z 信号），值得注意的是该方式需要电机支持 index 信号（一般在编码器信号中）并正确接线。

4 - Search for index

Command the motor to move until detecting an index, and then to stop. Motion direction is according to the Speed sign.

Speed:	<input type="text" value="5,000"/>	counts/sec	//运动速度，正负决定运动方向，一般需要和前步骤中找限位运动方向相反；
Accel/Decel:	<input type="text" value="50,000"/>	counts/sec ²	//加速度，只需设大小；
Emergency Decel:	<input type="text" value="500,000"/>	counts/sec ²	//急停减速度，只需设大小；
Timeout:	<input type="text" value="100,000"/>	msec	//本步骤执行延时，设置该步骤完成时间阈值，若在该时间段内未搜寻到 index 信号，运动将停止并输出报警，退出回零程序，回零失败，因此该延时时间建议设定值为大于全行程的时间。

⑤5-Move to last pos. of index: 以设定的速度走绝对目标位置运动到最后一个定位到的 index 信号位置。

5 - Move to last pos. of index

Command the motor to perform an absolute point to point motion, to the latest captured position of the index.

Speed:	<input type="text" value="5,000"/>	counts/sec	//运动速度，只需设大小；
Accel/Decel:	<input type="text" value="50,000"/>	counts/sec ²	//加速度，只需设大小；
Emergency Decel:	<input type="text" value="500,000"/>	counts/sec ²	//急停减速度，只需设大小；
Timeout:	<input type="text" value="10,000"/>	msec	//本步骤执行延时，设置该步骤完成时间阈值，若在该时间段内未运动到该 index 位置，运动将停止并输出报警，退出回零程序，回零失败。

⑥6-Set position: 定义当前位置值，典型的应用是在回零结束前将当前位置置 0;

6 - Set position

Set the current position of the motor to a user defined value. Typically used as the last step of an homing sequence.

Set Position value: counts //定义当前位置值;

Timeout: msec //步骤执行延时。

⑦7-Wait time:等待延时。

7 - Wait time

Wait time to wait until moving to the next step.

Time to wait msec //定义延时时间。

⑧8-Motor On/Off: 电机上使能或下使能操作。

8 - Motor On/Off

Disables or enables the motor.

Enable Motor Disable Motor //选择上使能或下使能;

Timeout: msec //步骤执行延时。

⑨9-Hard stop by motor stuck: 通过 Jog 运动时的运动状态（堵转速度和电流）判断电机是否堵转到达硬限位，一般用于有硬限位的电机回零。

9 - Hard stop by motor stuck

Jog until hard stop is detected by motor stuck. Assign home position at this point.

Speed: counts/sec //设定 Jog 运动到硬限位的速度，正负决定运动方向;

Accel/Decel: counts/sec² //设定 Jog 运动的加/减速度，仅设置大小;

Emergency Decel: counts/sec² //设定急停减速度，仅设置大小;

Stuck vel. Threshold: counts/sec //判断是否到达硬限位的速度阈值（最小值），和电流阈值、堵转时间同时满足时判断电机到达硬限位;

Stuck curr. Threshold: mA //判断是否达到硬限位的电流阈值（最大值），和速度阈值、堵转时间同时满足时判断电机到达硬限位;

Stuck time: msec //判断堵转时间（最小值），和速度阈值、电流阈值同时满足时判断电机到达硬限位;

Set position value: counts //设定当前位置值;

Timeout: msec //本步骤执行延时，若在该时间内判断未到达硬限位，运动将停止并输出报警，退出回零程序，回零失败。

⑩10-Hard stop by high error: 通过 Jog 运动时的运动状态（位置误差）判断电机是否堵转到达硬限位，一般用于有硬限位的电机回零。值得注意的是所设定的位置误差不能过小（小于正常运动过程中的 PosErr），建议先通过 PCSuite 的示波器先监测该速度下正常运动过程中的速度误差，避免误判硬限位。

10 - Hard stop by high error

Jog until hard stop is detected by high position error. Assign home position at this point.

Speed: counts/sec //设定 Jog 运动到硬限位的速度，正负决定运动方向;

Accel/Decel: counts/sec² //设定 Jog 运动的加/减速度，仅设置大小;

Emergency Decel: counts/sec² //设定急停减速度，仅设置大小;

Pos. error Threshold: counts //判断是否到达硬限位的位置误差阈值（最大值），当位置误差超过该设定值时判断电机到达硬限位;

Set position value: counts //将判断电机到达硬限位处的位置设置为自定义设定值;

Timeout: msec //本步骤执行延时，若在该时间内判断未到达硬限位，运动将停止并输出报警，退出回零程序，回零失败。

⑪11-Jog to Home switch change: Jog 运动到回零开关位置处（需电机装有回零开关）。

11 - Jog to Home switch change

Jog until home switch toggles its state. Direction defined by Speed sign and Home state.

Speed: counts/sec //设定 Jog 运动速度，正负决定运动方向;

Accel/Decel: counts/sec² //设定 Jog 运动的加/减速度，仅设置大小;

Emergency Decel: counts/sec² //设定急停减速度，仅设置大小;

Timeout: msec //本步骤执行延时，若在该时间内未检测到回零开关信号变化，运动将停止并输出报警，退出回零程序，回零失败。

⑫ 12-Absolute PTP: 给电机一个绝对目标位置指令，让电机走绝对 PTP 运动。

12 - Absolute PTP

Command the motor to perform an absolute point to point motion.

Speed: counts/sec //设定绝对 PTP 运动速度，仅设置大小；
Accel/Decel: counts/sec² //设定绝对 PTP 运动的加/减速度，仅设置大小；
Absolute target: counts //设定绝对目标位置；
Timeout: msec //本步骤执行延时，若在该时间内未到达绝对目标位置，运动将停止并输出报警，退出回零程序，回零失败。

⑬ 13-Set pos. software limits: 设置正负软限位位置（绝对位置），一般设在回零之后。

13 - Set pos. software limits

Provides the means to optionally set the position software limits (RevPLim and/or FwdPLim).

Set Reverse position limit at: counts //设定负软限位；
 Set Forward position limit at: counts //设定正软限位；

⑭ 14-Configure Position Lock: 设置位置捕捉（或称探针）信号源，常用于配置 index 信号回零。

14 - Configure Position Lock

Configures the Position Lock feature. Refer to 'Feedback/Lock' window for 'Source' value and more details.

Enable Lock //勾选开启位置锁定；
Polarity: Rising Edge Falling Edge //选择信号上升沿或下降沿触发；
Lock Source: Show Sources //选择信号源，点击“Show Sources”可显示信号源序号，值得注意的是信号源需要在“Feedback Lock”界面设定好信号源。
Note! 'Lock Source' differs between products and versions! Please verify the correct value at Feedback/Lock window.
Timeout: msec //本步骤延时。

⑮ 15-Jog to Lock(index by HW): Jog 运动过程中捕捉 index 信号（或称 Z 信号）位置的位置。

15 - Jog to Lock (index by HW)

Jog with the provided motion parameters till Lock happens and Lock position is latched. Then decelerate and stop.

Speed: counts/sec //设定 Jog 运动的速度，正负决定运动方向；
Accel/Decel: counts/sec² //设定 Jog 运动加/减速度，仅需设置大小；
Emergency Decel: counts/sec² //设定急停减速度，仅需设置大小；
Timeout: msec //本步骤执行延时，若在该时间内未监测到触发位置捕捉，运动将停止并输出报警，退出回零程序，回零失败。

⑯ 16-Move to Lock position: 电机走绝对目标位置运动到⑮步捕捉到的位置。

16 - Move to Lock position

Move (using the provided motion parameters) to the last recorded Lock position (LockVal).

Speed: counts/sec //设定电机运动的速度，仅设置大小；
Accel/Decel: counts/sec² //设定电机运动的加/减速度，仅设置大小；
Emergency Decel: counts/sec² //设定急停减速度，仅设置大小；
Timeout: msec //本步骤执行延时，若在该时间内未运动到⑮步捕捉到的位置，运动将停止并输出报警，退出回零程序，回零失败。

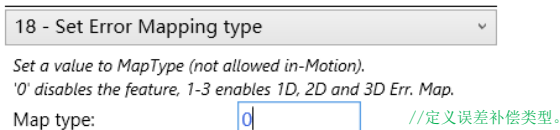
⑰ 17-Set Motion Mode: 设定电机运动模式，具体参数值定义可查阅“MotionMode”关键字解释，值得注意的是运动过程中及齿轮模式中不允许更改运动模式。

17 - Set Motion Mode

Set a value to MotionMode (not allowed in-Motion). (not allowed to use the Gearing motion modes).

Motion mode: //定义 motion mode 参数值，可查阅“MotionMode”关键字解释。

18-Set Error Mapping type: 设置位置误差补偿类型，值为 0 表示关闭位置误差补偿，1~3 分别表示开启 1D,2D,3D 误差补偿。**值得注意的是在回零开始前须关闭误差补偿功能**，见 2.3 章节。



执行回零操作过程中可以在回零界面右下角实时观察到当前执行步骤。

2.2 常用回零方式介绍

以下介绍几种 Agito 常用的回零设置方式，用户根据需求及应用方案选择合适的回零方式。值得注意的是不同的回零方式需要不同的外部硬件信号支持，对回零顺序和位置将会产生影响。

序号	回零方式	适用电机类型	所需硬件支持
1	锁定 index 位置回零（带限位开关）	旋转、直线、音圈	Index 信号、限位开关
2	锁定 index 位置回零（带机械限位）	旋转、直线、音圈	Index 信号、机械限位
3	搜寻 index 位置回零（无限位）	旋转	Index 信号
4	硬限位回零（无 index 和限位开关）	旋转、直线、音圈	机械限位
5	限位开关信号回零	旋转、直线、音圈	限位开关（RLS、FLS）
6	回零开关信号回零	旋转、直线、音圈	回零开关

2.2.1 锁定 index 位置回零（带限位开关）

通过锁定 index 位置（带限位开关）回零的工作机制是定义一个 index 信号（一般是当前运动轴）作为位置捕捉（或称探针）触发信号，以该物理位置作为零位参考，该方式常用于**高精高速回零**过程，适用于**各类带有 index 信号及限位开关信号的电机**，为推荐回零方式。

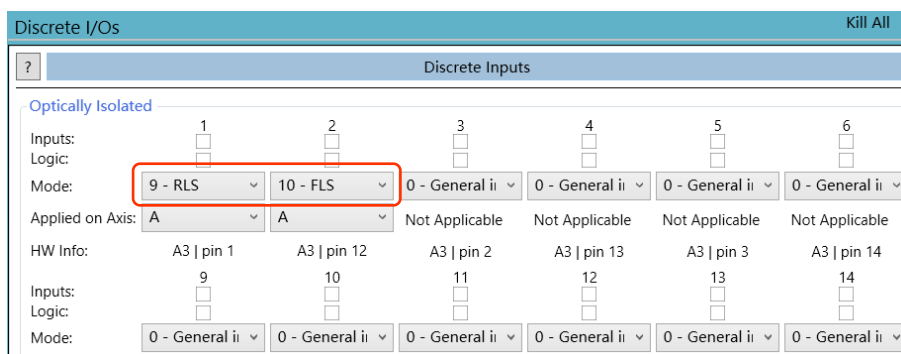
1.外部电气接线:

①index 信号（或称 Z 信号）接线:

电机编码器接口	Agito 控制器(主)编码器口	
Index+（或 Z+）	9	Encoder_4+
Index-（或 Z-）	10	Encoder_4-

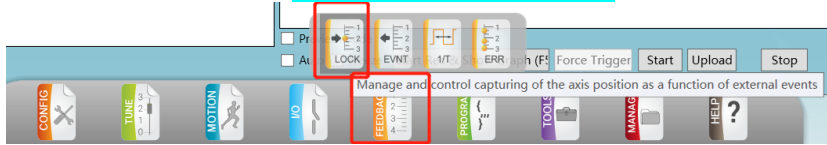
②限位信号：接正限位（FLS）、负限位（RLS）开关

根据所使用对应的硬件手册将限位开关信号接到对应数字输入口。此处以 AGD301 控制器 DI_1#和 DI_2#为例，将左右限位信号分别接到数字输入 1 和 2 通讯，并分别设置为 RLS 和 FLS。

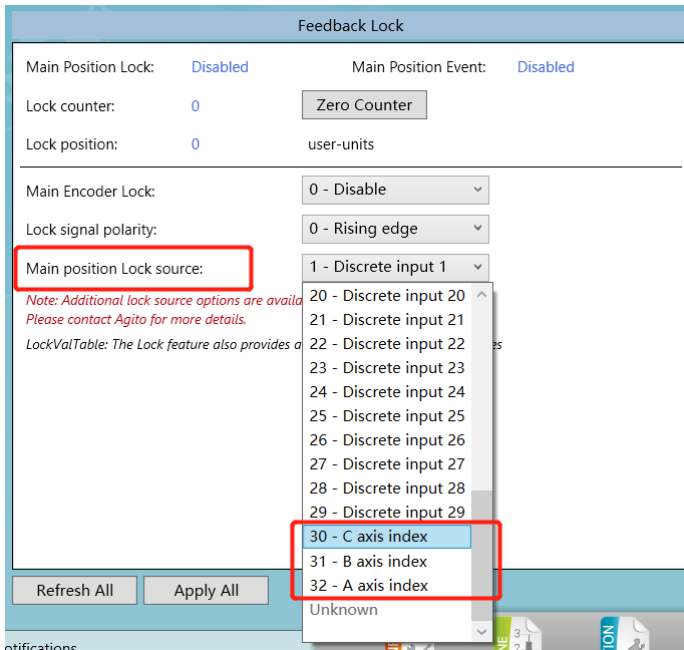


2.PCSuite 设置:

①切换 PCSuite 菜单栏到 **Feedback-->Feedback lock**，切换到位置捕捉界面。

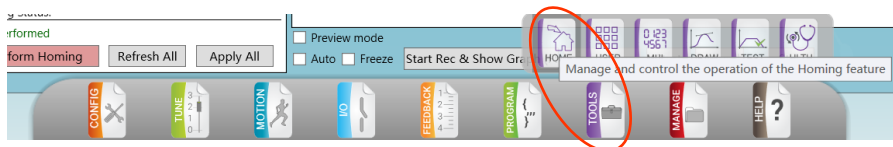


②在“Main position Lock source”下拉框中，选择当前轴对应的 index，如 A 轴 index 是编号 32



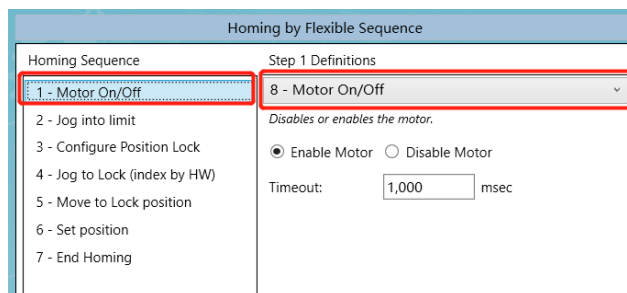
③点击上图中“Apply All”（或按键盘回车键）启用上述配置。

④点击 PCSuite 菜单栏到 **Tools-->Homing**，切换到回零界面。



⑤设定回零步骤:

- **第 1 步: 电机使能 (类型: 8 - Motor On/Off) ;**



- 第 2 步：运动到限位开关（类型：1 - Jog into limit）。本例以负方向限位（RLS）为例，所以 Speed 的值是负数值，即朝负方向运动，直到到达 RLS；如运动到正限位（FLS），则 Speed 应输入正数值；

- 第 3 步：启用位置锁定（类型：14 - Configure Position Lock）。勾选“Enable Lock”启用位置“锁定”，选中 Rising edge（上升沿），输入 Lock Source 值 32（本例，对应 A 轴 index）。如果是 B 轴 index，则输入 Lock Source 值 31；如果是 C 轴 index，则输入 Lock Source 值 30（点击“Show Sources”可以显示对应的触发信号源编号）。

- 第 4 步：运动并捕获 index【类型：15 - Jog to Lock (index by HW)】。本例第 2 步是运动到负限位（RLS）方向，到达 RLS 后，应朝正方向运动去捕获 index，因此此处 Speed 的值是正数值。若第 2 步是运动到正限位（FLS），则此处 Speed 应输入负数值。

- 第 5 步：运动到捕获的 index 位置（类型：16 - Move to Lock position）。上一步 Index 信号触发位置“锁定”功能后，以绝对运动方式回到捕获的 index 位置，所以此处的速度值只需输入正数值。

- 第 6 步：设定当前位置数值（类型：6 - Set position）。设定当前位置即上一步到达的 index 位置，Set Position value 输入 0，即设定当前位置为零点。

- 第 7 步：回零结束（类型：0 - End Homing）。回零步骤的最后一步必须是以“0-End Homing”结束回零程序。

⑥ 点击“Apply All”（或按键盘回车键）启用上述设置。

⑦ 点击“Perform Homing”将开始按顺序执行回零步骤。回零结束后提示“Completed successfully”。

2.2.2 锁定 index 位置回零（带机械硬限位）

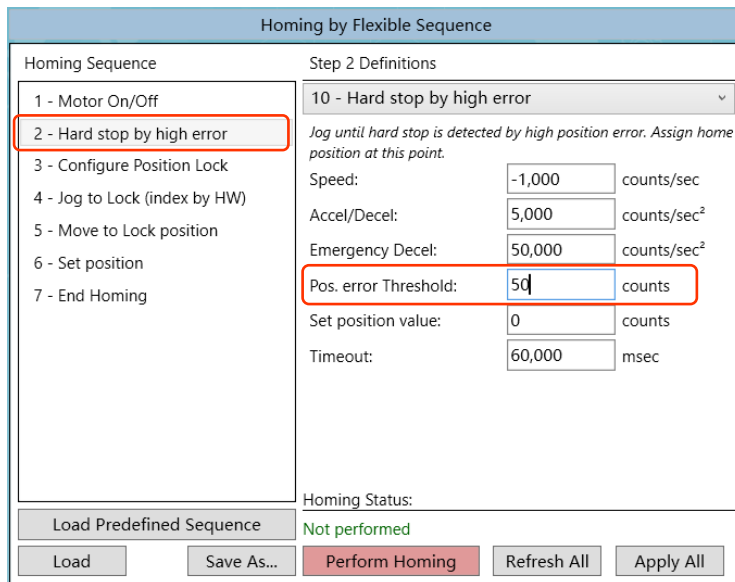
通过锁定 index 位置（带机械限位）回零的工作机制是，当所控制电机只有 Index 信号，没有限位开关，但是有硬限位，如直线电机、音圈电机以及有机械限制转角的旋转电机等，可以通过 Jog 运动碰撞寻找机械硬限位，然后再反向运动捕捉 index 信号位置来实现回零。主要步骤和 2.2.1 章节中所介绍“通过锁定 index 位置回零（带限位开关）”回零方式类似，主要在运动到限位处方式有差异，以下内容仅展开解释差异步骤，其他相同步骤将不再累述。

1.外部接线：编码器 index 信号连接到控制器主编码器口；

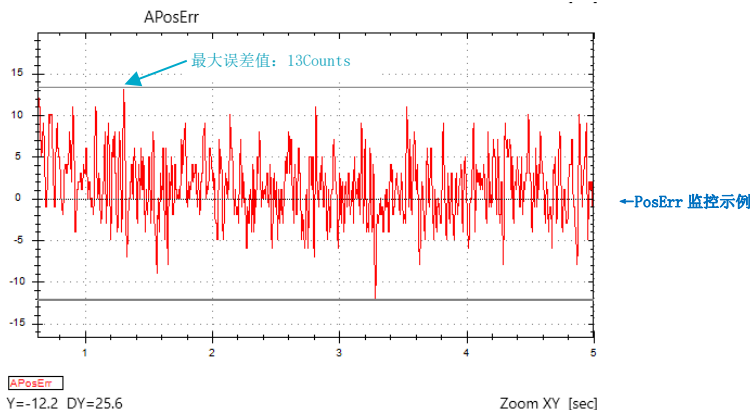
2.Feedback Lock 设置（参阅 2.2.1 章节对应 Feedback Lock 设置）；

3.回零步骤：其他步骤同 2.2.1 章节中所介绍“通过锁定 index 位置回零（带限位开关）”方式，仅第 2 步有差异。根据判断机械硬限位的方式不同此处又细分为两种：方式一：判断位置误差是否满足条件；方式二：判断堵转速度、电流和持续时间是否满足条件。

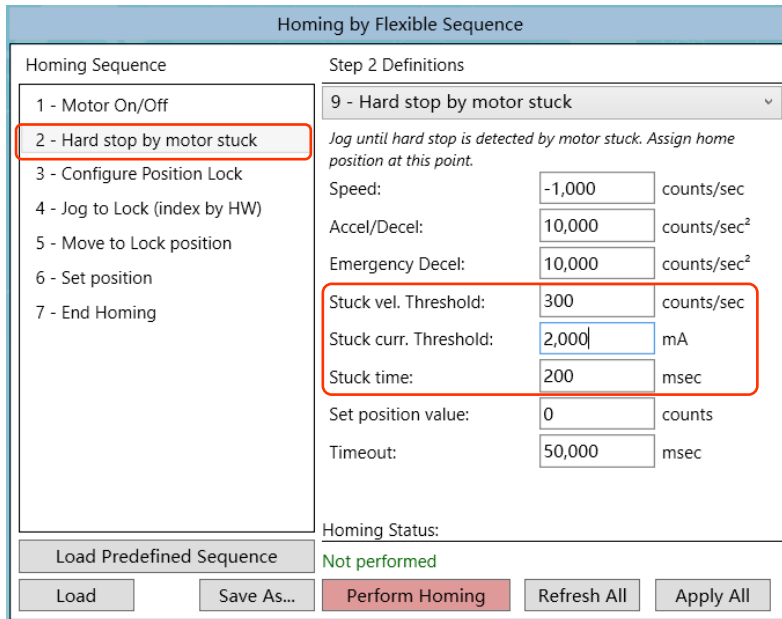
➤ 方式一：判断位置误差。



该方式判断机械硬限位的关键参数是设置位置误差阈值（Pos. error Threshold），设置太大可能会产生剧烈机械撞击声，太小会可能导致误判机械限位，因此设置此值前建议先通过 PCSuite 示波器功能监测按以上设定速度等运动参数走 PTP 或 Jog 运动过程的 PosErr 作为参考值来设置。例如下图中所监测到的运动误差值（绝对值）最大为 13Counts，推荐位置误差阈值（Pos. error Threshold）设置为所监测到 PosErr 最大值的 5~10 倍之间。



➤ 方式二：判断堵转速度/电流。



该方式判断机械硬限位的关键参数是设置堵转速度阈值（Stuck vel. Threshold）、堵转电流阈值（Stuck curr. Threshold）、堵转时间（Stuck time），设定值不合适可能会产生机械噪声或误判，堵转速度阈值建议设定为运动速度（本步骤 Speed 值）的 30% 左右，堵转电流阈值设置为电机持续电流的 150% 左右（该值不可大于电机安全电流），堵转持续时间推荐 100~500ms 之间。

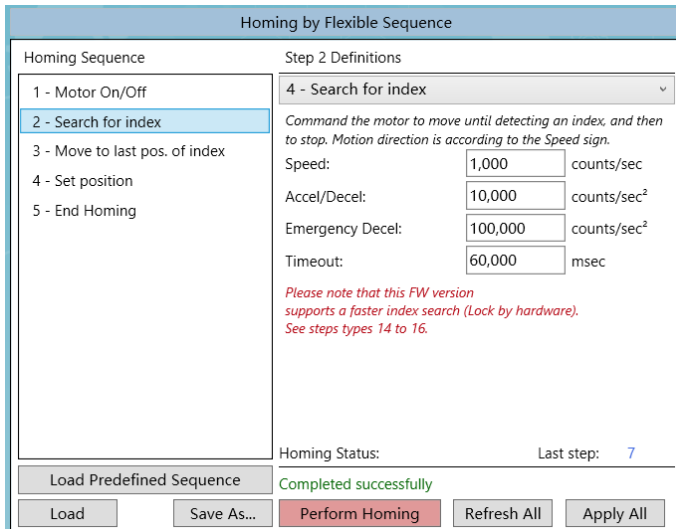
2.2.3 搜寻 index 位置回零（无限位）

通过搜寻 index 位置（无限位）回零的工作机制是，当所控制电机仅有 Index 信号，没有物理限位的时候，如旋转电机，可以使电机按所定义方向走 Jog 运动去搜寻 index 位置来回零。

1.外部接线：外部 index 信号（或称 Z 信号）接线：

电机编码器接口	Agito 控制器(主)编码器口	
Index+（或 Z+）	9	Encoder_4+
Index-（或 Z-）	10	Encoder_4-

2.设定回零步骤：捕捉 index 位置：



- **第 1 步：电机使能（类型：8 - Motor On/Off）；**

Homing Sequence	Step 1 Definitions
1 - Motor On/Off	8 - Motor On/Off
2 - Search for index	<i>Disables or enables the motor.</i>
3 - Move to last pos. of index	<input checked="" type="radio"/> Enable Motor <input type="radio"/> Disable Motor
4 - Set position	Timeout: <input type="text" value="1,000"/> msec
5 - End Homing	

- **第 2 步：搜寻 index 位置（类型：4-Search for index）；**

按设定的速度运动搜寻 index 信号，注意速度不宜过大，推荐速度(Speed)为 1/5rev/s 左右，该步骤执行延时时间(Timeout)需大于该速度下满行程所需时间。

Homing Sequence	Step 2 Definitions
1 - Motor On/Off	4 - Search for index
2 - Search for index	<i>Command the motor to move until detecting an index, and then to stop. Motion direction is according to the Speed sign.</i>
3 - Move to last pos. of index	Speed: <input type="text" value="1,000"/> counts/sec
4 - Set position	Accel/Decel: <input type="text" value="10,000"/> counts/sec ²
5 - End Homing	Emergency Decel: <input type="text" value="100,000"/> counts/sec ²
	Timeout: <input type="text" value="60,000"/> msec

- **第 3 步：运动到搜寻到的最后一个 index 位置（5 - Move to last pos.of index）**

按设定速度走绝对运动模式运动到上一步所搜寻到的 index 位置。

Homing Sequence	Step 3 Definitions
1 - Motor On/Off	5 - Move to last pos. of index
2 - Search for index	<i>Command the motor to perform an absolute point to point motion, to the latest captured position of the index.</i>
3 - Move to last pos. of index	Speed: <input type="text" value="1,000"/> counts/sec
4 - Set position	Accel/Decel: <input type="text" value="10,000"/> counts/sec ²
5 - End Homing	Emergency Decel: <input type="text" value="100,000"/> counts/sec ²
	Timeout: <input type="text" value="60,000"/> msec

- **第 4 步：设定当前位置值（6- Set position）**

设定当前位置即上一步到达的 index 位置值，Set Position value 输入 0，即设定当前位置为零点。

Homing Sequence	Step 4 Definitions
1 - Motor On/Off	6 - Set position
2 - Search for index	<i>Set the current position of the motor to a user defined value. Typically used as the last step of an homing sequence.</i>
3 - Move to last pos. of index	Set Position value: <input type="text" value="0"/> counts
4 - Set position	Timeout: <input type="text" value="1,000"/> msec
5 - End Homing	

- **第 5 步：回零结束（0 - End Homing）**

回零步骤的最后一步必须是以“0-End Homing”结束回零程序。

Homing Sequence	Step 5 Definitions
1 - Motor On/Off	0 - End Homing
2 - Search for index	
3 - Move to last pos. of index	
4 - Set position	
5 - End Homing	

3. 点击 “Apply All”（或按键盘回车键）启用上述设置。

4. 点击 “Perform Homing” 将开始按顺序执行回零步骤。回零结束后提示 “Completed successfully”。

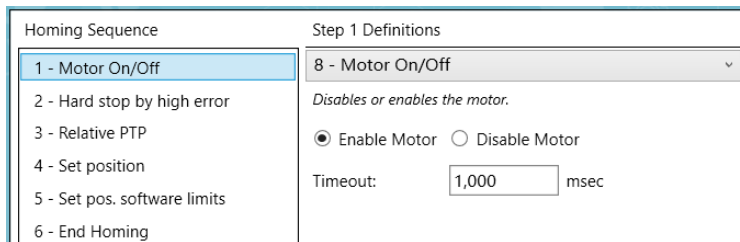


2.2.4 硬限位回零（无 index 和限位开关）

通过碰撞机械硬限位（无 index 和限位开关）回零的工作机制是，当所控制电机无 Index 信号和限位开关时，仅有机硬限位的时候，可以使电机按所定义方向走 Jog 运动碰撞机械硬限位，将机械限位作为参考设置零点。根据判断机械硬限位的方式不同此处又细分为两种：方式一：判断位置误差是否满足条件；方式二：判断堵转速度、电流和持续时间是否满足条件。该方式适用于有机硬限位的直线电机、音圈电机及转角限位的旋转电机，并且允许多次回零重复性存在微小位置偏差（此偏差是由于用户负载动量的变化判断硬限位位置可能会存在微小差异导致）。

1. 设定回零步骤：判断硬限位；

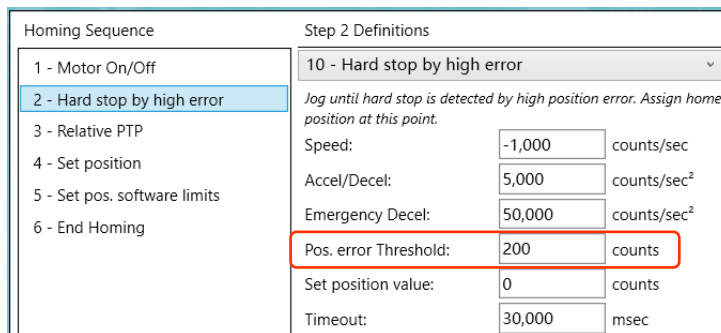
- 第 1 步：电机使能（类型：8 - Motor On/Off）；



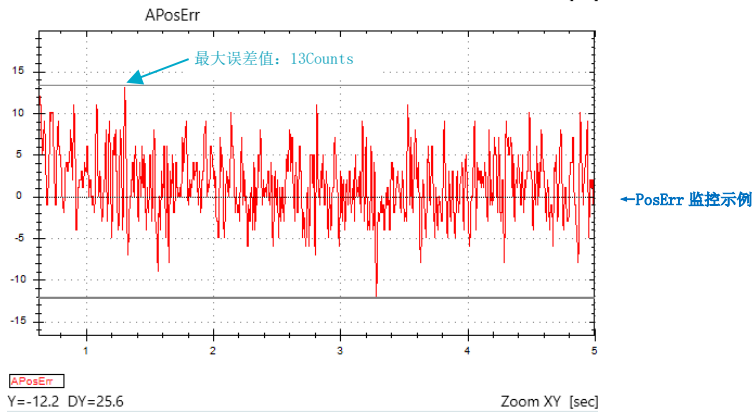
- 第 2 步：运动到机械硬限位（类型：10 - Hard stop by high error）；

根据判断机械硬限位的方式不同此处又细分为两种：

- 方式一：判断位置误差。



该方式判断机械硬限位的关键参数是设置位置误差阈值（Pos. error Threshold），设置太大可能会产生剧烈机械撞击声，太小会可能导致误判机械限位，因此设置此值前建议先通过 PCSuite 示波器功能监测按以上设定速度等运动参数走 PTP 或 Jog 运动过程的 PosErr 作为参考值来设置。例如下图中所监测到的运动误差值（绝对值）最大为 13Counts，推荐位置误差阈值（Pos. error Threshold）设置为所监测到 PosErr 最大值的 5~10 倍之间。



➤ 方式二：判断堵转速度/电流。

Homing Sequence	Step 2 Definitions
1 - Motor On/Off	9 - Hard stop by motor stuck
2 - Hard stop by motor stuck	Jog until hard stop is detected by motor stuck. Assign home position at this point.
3 - Relative PTP	Speed: <input type="text" value="-1,000"/> counts/sec
4 - Set position	Accel/Decel: <input type="text" value="5,000"/> counts/sec ²
5 - Set pos. software limits	Emergency Decel: <input type="text" value="50,000"/> counts/sec ²
6 - End Homing	Stuck vel. Threshold: <input type="text" value="300"/> counts/sec
	Stuck curr. Threshold: <input type="text" value="2,000"/> mA
	Stuck time: <input type="text" value="200"/> msec
	Set position value: <input type="text" value="0"/> counts
	Timeout: <input type="text" value="50,000"/> msec

该方式判断机械硬限位的关键参数是设置堵转速度阈值（Stuck vel. Threshold）、堵转电流阈值（Stuck curr. Threshold）、堵转时间（Stuck time），速度（Speed）和判断堵转参数设定值不合适可能会产生机械噪声或误判，堵转速度阈值建议设定为运动速度（本步骤 Speed 值）的 30% 左右，堵转电流阈值设置为电机持续电流的 150% 左右（该值不可大于电机安全电流），堵转持续时间推荐 100~500ms 之间。

■ **第 3 步：设定偏移量（类型：3- Relative PTP）；**

为避免在后续的运动过程中碰撞到硬限位导致机械磨损或意外保护下使能，一般在找到机械硬限位后朝反方向走一段相对运动以避开机械硬限位。

Homing Sequence	Step 3 Definitions
1 - Motor On/Off	3 - Relative PTP
2 - Hard stop by high error	Command the motor to perform a relative point to point motion.
3 - Relative PTP	Speed: <input type="text" value="5,000"/> counts/sec
4 - Set position	Accel/Decel: <input type="text" value="50,000"/> counts/sec ²
5 - Set pos. software limits	Relative distance: <input type="text" value="1,000"/> counts
6 - End Homing	Timeout: <input type="text" value="20,000"/> msec

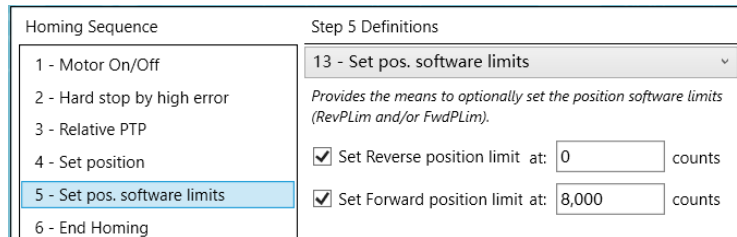
■ **第 4 步：设定当前位置值（类型：6- Set position）；**

将上一步偏移后的位置设置为 0。

Homing Sequence	Step 4 Definitions
1 - Motor On/Off	6 - Set position
2 - Hard stop by high error	Set the current position of the motor to a user defined value. Typically used as the last step of an homing sequence.
3 - Relative PTP	Set Position value: <input type="text" value="0"/> counts
4 - Set position	Timeout: <input type="text" value="1,000"/> msec
5 - Set pos. software limits	
6 - End Homing	

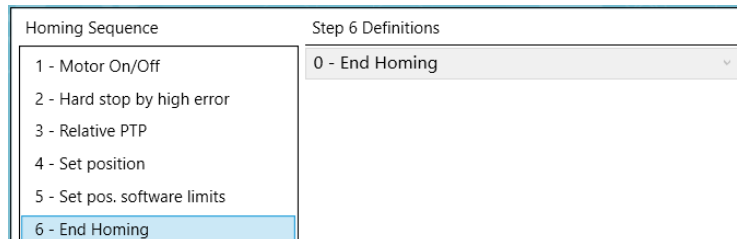
▪ **第 5 步：设定软限位（类型：13-Set pos. software limits）；**

设定软限位行程（此步骤为非必要步骤，但推荐设置以避免撞击机械限位），设置时需要注意零点处的方向，关系到软限位区间的正负，例如在以上步骤电机先向负方向（第 2 步中 Speed 值为负）寻机械限位，然后再向正方向（第 3 步中 Speed 值为正值且 Relative distance 值也为正值）偏移置 0，因此在本步骤设置限位时负方向为 0Counts，正方向为 8,000Counts（此处假设电机全行程为 10,000Counts，正负方向均偏移 1,000Counts，用户可根据实际应用需求设定）。



▪ **第 6 步：回零结束（类型：0-End Homing）。**

回零步骤的最后一步必须是以“0-End Homing”结束回零程序。



2. 点击“Apply All”（或按键盘回车键）启用上述设置。

3. 点击“Perform Homing”将开始按顺序执行回零步骤。回零结束后提示“Completed successfully”。

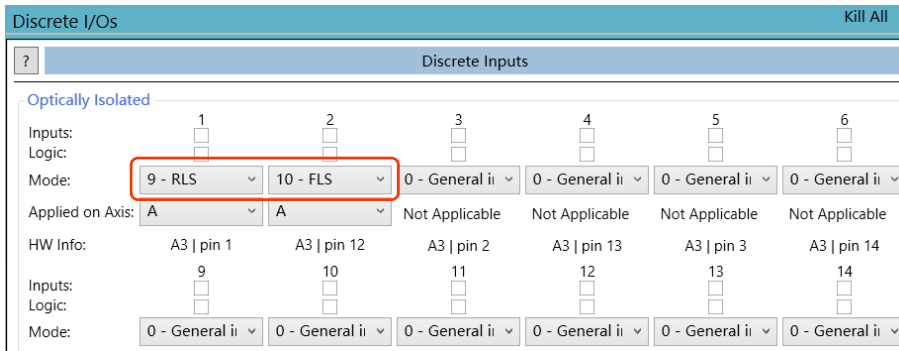


2.2.5 限位开关信号回零

通过限位开关信号回零的工作机制是，当所控制电机无 Index 信号和仅有限位开关时，可以使电机按所定义方向去搜寻限位信号（RLS 或 FLS），将限位开关信号作为参考设置零点。由于电机碰到限位信号后是按照急停减速度去停止，因此重复回零点的位置会存在微小的差异，此方式适用于有限位开关并允许存在微小误差的系统。

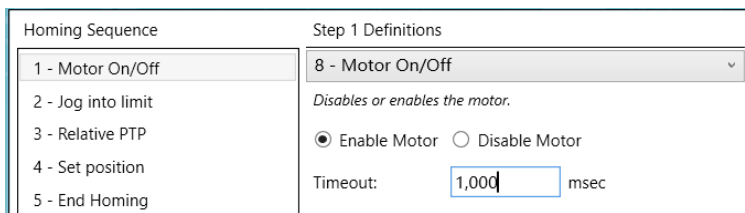
1. **外部接线：**接限位信号（RLS、FLS）到控制器；

根据所使用对应的硬件手册将限位开关信号接到对应数字输入口。此处以 AGD301 控制器 DI_1#和 DI_2#为例，将左右限位信号分别接到数字输入 1 和 2 通讯，并分别设置为 RLS 和 FLS，其他 DI 通道使用依次类推。

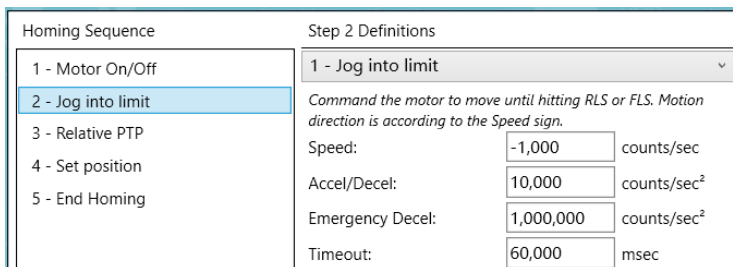


2. 设定回零步骤:

- 第 1 步: 电机使能 (类型: 8 - Motor On/Off) ;

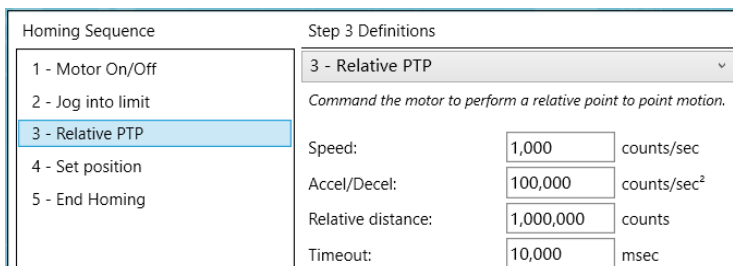


- 第 2 步: 运动到限位开关 (类型: 1 - Jog into limit) 。本例以负方向限位 (RLS) 为例, 所以 Speed 的值是负数值, 即朝负方向运动, 直到到达 RLS; 如运动到正限位 (FLS), 则 Speed 应输入正数值;



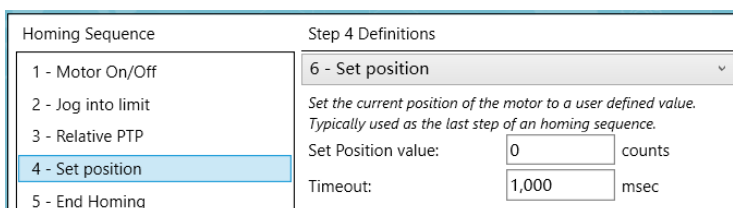
- 第 3 步: 设定偏移量 (类型: 3- Relative PTP) ;

反方向走一段相对运动以避免限位开关。

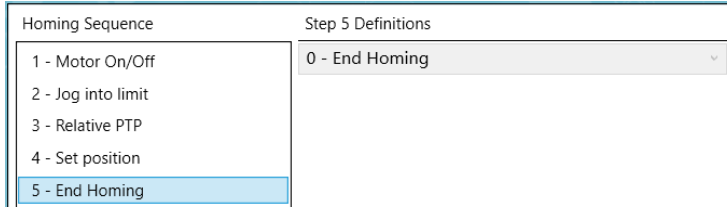


- 第 4 步: 设定当前位置值 (类型: 6- Set position) ;

将上一步运动到限位后减速停止的位置置 0。



- **第 5 步：回零结束（类型：0-End Homing）。**



3. 点击 **“Apply All”**（或按键盘回车键）启用上述设置。

4. 点击 **“Perform Homing”** 将开始按顺序执行回零步骤。回零结束后提示 **“Completed successfully”**。



2.2.6 回零开关信号回零

通过回零开关信号回零的工作机制是，当所控制电机安装有回零传感器（Home Switch），可以通过 Jog 运动过程中实时监测回零传感器信号状态，将状态变化位置作为回零参考零点。搜寻限位开关和机械限位步骤和上述 2.2.1~2.2.3 章节相应部分相同，用户可根据实际应用场景参考下表，该部分将不再累述，本章节仅介绍搜寻回零开关信号的步骤。

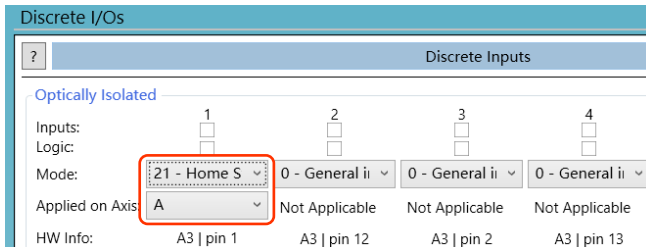
应用场景	参考章节	回零步骤
带限位开关、回零开关	2.2.1（第 9 页）	1 - Motor On/Off 2 - Jog into limit 3 - Jog to Home switch change 4 - Set position 5 - End Homing
带机械硬限位、回零开关	2.2.2（第 13 页）	1 - Motor On/Off 2 - Hard stop by high error 3 - Jog to Home switch change 4 - Set position 5 - End Homing
仅回零开关	2.2.2（第 14 页）	1 - Motor On/Off 2 - Jog to Home switch change 3 - Set position 4 - End Homing

↑ 表 2.2.6-1

1.外部接线：根据对应控制器使用手册将回零传感器信号到所使用的控制器 DI 口，如有 RLS、FLS 信号也接到相应的 DI 口。

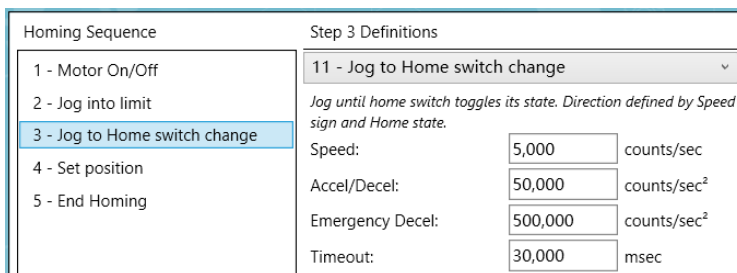
2.PC Suite 设置：

①切换 PC Suite 菜单到 **I/O --> Discrete Inputs** 界面。本例以 AGD301 控制器 DI_1#接回零传感器为例，将模式设置为“21-Home Switch”并选择对应运动轴“A”；



②切换 PC Suite 菜单到 **Tools--->Homing** 界面，设置回零程序；

详细步骤请参阅表 2.2.6-1，此处仅说明第 2/3 步：**Jog to Home switch change**，以有限位信号场景为例，电机监测到硬限位后，反向搜寻回零开关信号，因此此处速度值（Speed）应该和探寻机械硬限位方向相反。



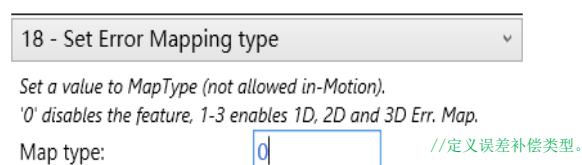
3.点击“Apply All”（或按键盘回车键）启用上述设置。

4.点击“Perform Homing”将开始按顺序执行回零步骤。回零结束后提示“Completed successfully”



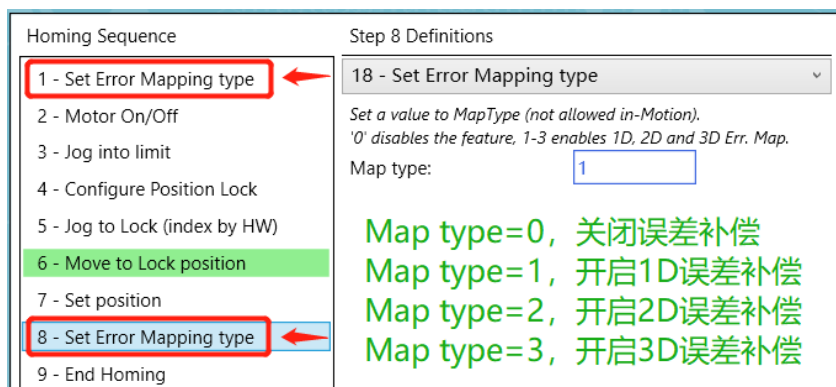
2.3 关于误差补偿（Error Mapping）在回零步骤中设置

误差补偿功能在回零过程中需要关闭，步骤内容见 2.1-2-⑱节中的描述



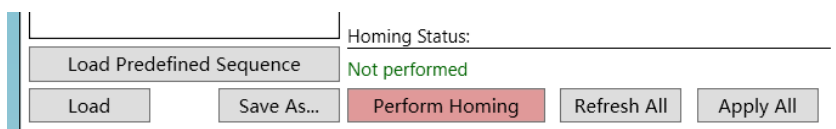
以下以回零方式 1（锁定 index 位置回零（带限位开关））中设置误差补偿功能为例，其他依此类推：

在回零开始和结束前都增加一步：**18-Set Error Mapping type**，其中第 1 步的 Map type 值设为 0 关闭误差补偿，倒数第 2 步 Map type 值根据实际使用的误差补偿类型设置。



2.4 导出导入回零程序

Agito 控制器支持将回零程序导出到本地文件（.hseq 格式）



- 点击“Save As...”，在弹窗中选择保存路径可以将回零程序保存到所选目录下；
- 点击“Load”，在弹窗中选择对应目标文件可以将本地文件导入到 PCSuite；
- 点击“Load Predefined Sequence”可以导入软件预定义的一些回零步骤；
- 点击“Apply All”将会应用当前回零界面程序。

3 回零相关关键字介绍

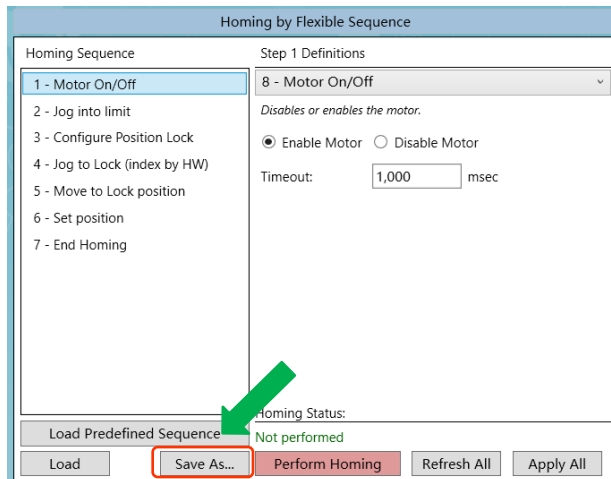
关键字可以用于终端（Terminal）输入指令、示波器采集图像、ASCII 通讯、IDE 编程、二次开发等使用，此处仅简要介绍和回零相关的部分关键字，用户可以通过 PCSuite 中的 Help --> Documentation 界面查询对应关键字来获取详细解释。

关键字	值类型	描述																								
HomingOn	读/写	<p>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。</p> <p>定义格式： 如启动 A 轴回零：AHomingOn=1</p>																								
HomingDef	读/写	<p>定义回零数组参数，数组范围为[1:150]。</p> <p>定义格式： 如将 A 轴回零程序第 1 步定义为上使能：AHomingDef[1]=8</p>																								
HomingStat	只读	<p>监测回零状态。</p> <p>返回值含义：</p> <table border="1"> <thead> <tr> <th>值</th> <th>含义</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>上电或重启后未执行回零。</td> </tr> <tr> <td>正值 (除 100 以外)</td> <td>回零进行中，HomingStat 值表示当前回零进行的步骤。</td> </tr> <tr> <td>-1</td> <td>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。</td> </tr> <tr> <td>-2</td> <td>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。</td> </tr> <tr> <td>-3</td> <td>由于回零过程中电机意外下使能导致回零失败。</td> </tr> <tr> <td>-4</td> <td>由于运动错误导致的回零失败，如未定义运动结束方式。</td> </tr> <tr> <td>-5</td> <td>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。</td> </tr> <tr> <td>-6</td> <td>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。</td> </tr> <tr> <td>-7</td> <td>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。</td> </tr> <tr> <td>-8</td> <td>由于未知的步骤类型导致的回零失败。</td> </tr> <tr> <td>100</td> <td>回零完成并成功。</td> </tr> </tbody> </table>	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。
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 PCSuite IDE+编程环境中回零

在 Agito PCSuite 上的 IDE+编程环境中可以方便的执行回零程序，以下以 A 轴捕获 index 位置回零（本文回零方式 2.2.1）作为示例来介绍回零程序。

■ 第 1 步：导出回零程序为.hseq 文件



名称

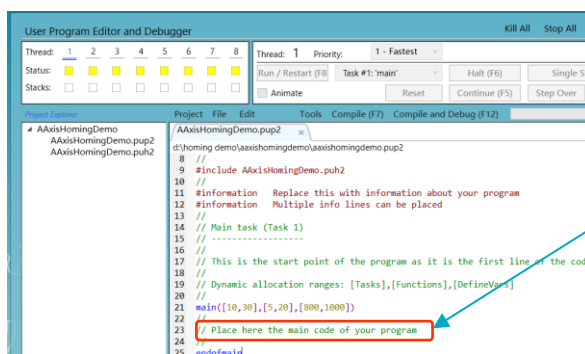
AAxisHoming.hseq

■ 第 2 步：使用文本格式打开导出的.hseq 文件，并将回零参数 HomingDef[*]前都加上轴号，如 A 轴就加“A”，以此类推。本步骤也可以直接在 IDE+编程环境中更改。

```
AAxisHoming.hseq
7 // Firmware version:
8 // FPGA version:
9 //
10 // File Checksum:
11 //
12 HomingDef[1]=8
13 HomingDef[2]=1
14 HomingDef[3]=16384
15 HomingDef[4]=0
16 HomingDef[5]=0
17 HomingDef[6]=0
18 HomingDef[7]=0
19 HomingDef[8]=0
20 HomingDef[9]=0
21 HomingDef[10]=0
22 HomingDef[11]=1
23 HomingDef[12]=5000
24 HomingDef[13]=100000
```

```
AAxisHoming.hseq
7 // Firmware version:
8 // FPGA version:
9 //
10 // File Checksum:
11 //
12 AHomingDef[1]=8
13 AHomingDef[2]=1
14 AHomingDef[3]=16384
15 AHomingDef[4]=0
16 AHomingDef[5]=0
17 AHomingDef[6]=0
18 AHomingDef[7]=0
19 AHomingDef[8]=0
20 AHomingDef[9]=0
21 AHomingDef[10]=0
22 AHomingDef[11]=1
23 AHomingDef[12]=5000
24 AHomingDef[13]=100000
```

■ 第 3 步：将改好的参数复制粘贴到 IDE+编程界面 main 函数中，当然也可以封装成函数在 main 函数中调用。



```
23 AHomingDef[1]=8
24 AHomingDef[2]=1
25 AHomingDef[3]=16384
26 AHomingDef[4]=0
27 AHomingDef[5]=0
28 AHomingDef[6]=0
29 AHomingDef[7]=0
30 AHomingDef[8]=0
31 AHomingDef[9]=0
32 AHomingDef[10]=0
33 AHomingDef[11]=1
34 AHomingDef[12]=5000
35 AHomingDef[13]=100000
36 AHomingDef[14]=1000000
37 AHomingDef[15]=983040
38 AHomingDef[16]=0
```

■ 第 4 步：增加一些状态判断以避免一些错误产生。

回零参数（HomingDef）即使不加载到 IDE 程序中，也可以在 IDE 程序中发送 “[AxisNum]HomingOn=1” 开始回零操作，但前提是需要是在回零界面已设置好并保存回零参数到控制器中。


```

main ([10,30],[5,20],[800,1000])
AWaitTime,200
while (AComtStatus[1]!=100)      //判断换相是否成功（如需要的情况下）
end

AHomingDef[1]=8
AHomingDef[2]=1
AHomingDef[3]=16384
AHomingDef[4]=0
AHomingDef[5]=0
..... //本处在示例程序中省略显示
AHomingDef[67]=0
AHomingDef[68]=0
AHomingDef[69]=0
AHomingDef[70]=0

AHomingOn=1
while (AHomingStat!=100) //判断回零完成并成功后，继续执行后续语句；
end
..... //回零完成后继续其他程序段
endofmain

```

} //回零参数定义

5 常见问题解答 (FAQ)

- **问题 1:** 通过捕获 index 位置方式回零时，电机为什么会碰到硬限位后下使能停止运动？
原因: 可能是因为搜寻 index 信号的方向和碰撞硬限位方向相同，导致控制器启动堵转保护下使能停止。
解决办法: 可以将搜寻 index 信号的速度取反来解决。
- **问题 2:** 通过判断位置误差方式（10-Hard top by high error）判断硬限位时，为什么电机还没碰到硬限位就开始反向运动？或者刚开始加速运动时就开始反向运动？
原因: 可能是位置误差阈值（Pos. error Threshold）设定过小，导致运动过程跟随误差超过所设定误差阈值误判为碰撞到硬限位；在刚开始加速运动时就开始反向运动可能是因为 PIV 参数刚性比较弱导致刚开始加速运动时位置误差超过所设定阈值导致误判碰撞到硬限位。
解决办法: 前者可以使用示波器抓图功能抓取运动过程中的 PosErr，推荐位置误差阈值设为运动过程中最大跟随误差值的 5~10 倍；后者可以调整 PIV 参数尽量抑制过冲来解决。
- **问题 3:** 通过判断位置误差方式（10-Hard top by high error）判断硬限位时，为什么电机碰撞到硬限位后电机出现震荡？
原因: 可能是位置误差阈值（Pos. error Threshold）设定过大，且系统 PIV 参数刚性太强。
解决办法: 可以通过减小位置误差阈值来解决。
- **问题 4:** 采用搜寻 index 位置回零（4-Search for index）时，为什么找不到 index 信号或者电机转多圈才找到 index 信号？
原因: 可能是搜寻 index 时速度太快导致错过信号点。
解决办法: 可以适当降低速度来解决。
- **问题 5:** 电机运动到限位信号（RLS、FLS）时，为什么电机没有停止？
原因: 可能是没有在 I/O 模式里设置 RLS 和 FLS，或者两者设反了，亦或者限位信号所应用的轴号和回零运动轴未对应上。
解决办法: 排查并更正以上错误。

