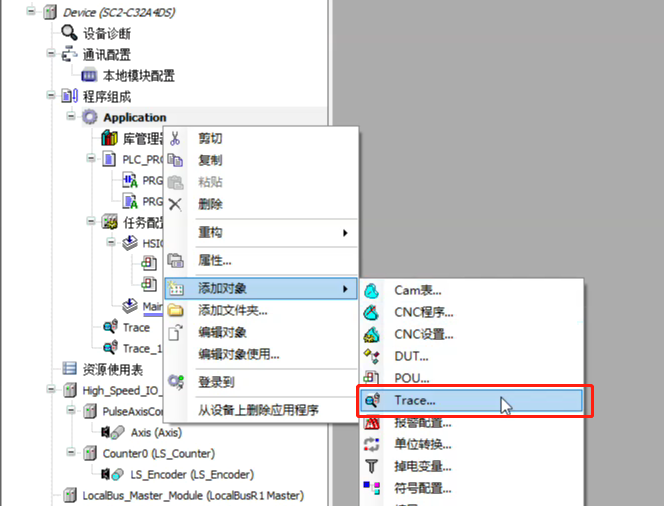
**恒速、点动、点位、回零运动**

恒速运动指令能控制轴以恒定的速度一直运动。点动运动（JOG运动）能根据操作者手动操作的时间进行一段指定方向和速度的与操作时间相同的运动。点位运动能实现电机的相对或绝对位置定位。回零运动能实现工作台的机械回原点。SC2-C轴控制程序应放到HSIOTask任务下调用。

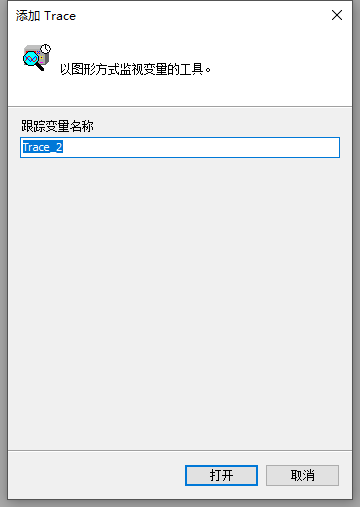
本文介绍常用的运动过程监测方法和SC2-C的恒速运动、点位运动、点动运动、回零运动的实现方法。

1. 轴运动状态的常用监测方法——采集Trace曲线
   1. 添加Trace

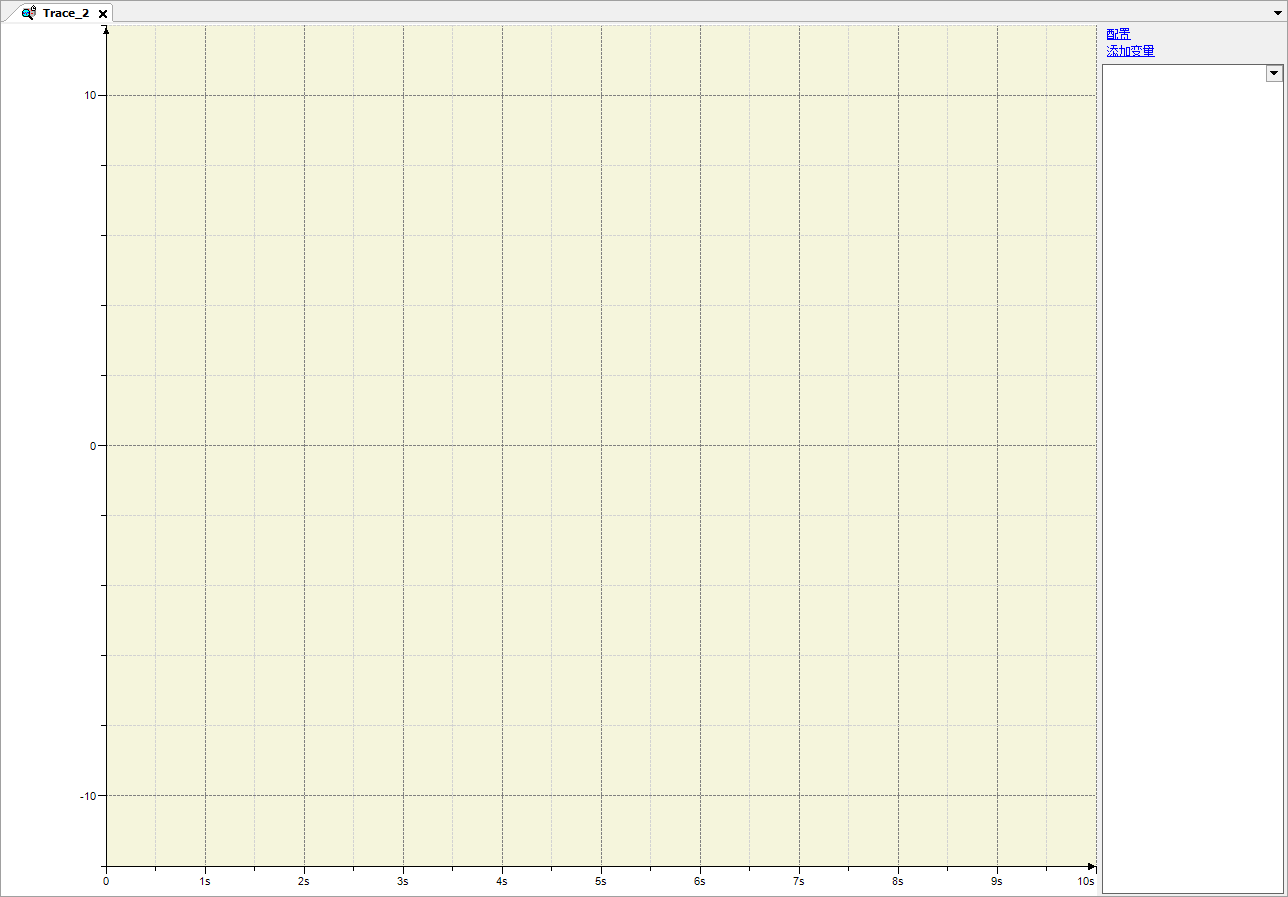
右键【Applivation】-【添加对象】-【Trace】。



给添加的Trace命名后，点击【打开】。



新建完成的Trace如下。

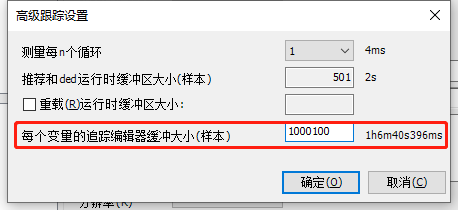
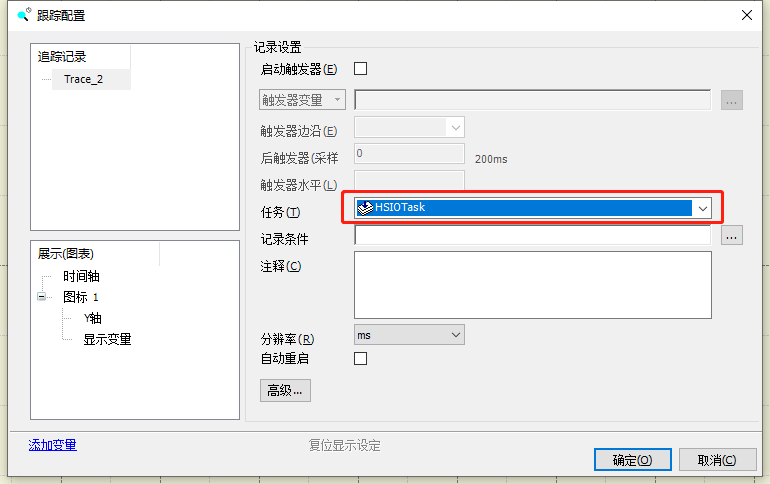


* 1. 配置Trace

点击Trace界面右上角的【配置】，打开跟踪配置窗口。



选择Trace采集的任务，若希望采集频率高，则选择任务周期短、任务优先级高的任务。若希望增长采集数据的时间，则点击【高级】，打开【高级跟踪设置】，将【每个变量的跟踪编辑器缓存大小（样本）】设置更大值。



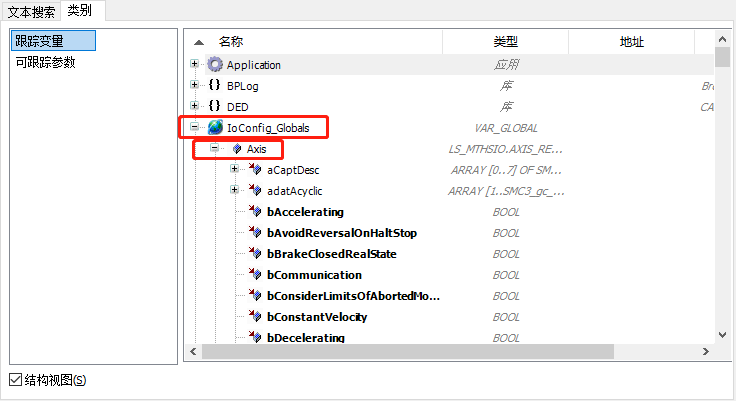
添加要采集的变量。右键Trace名称（这里Trace名称是Trace\_2），【添加变量】。



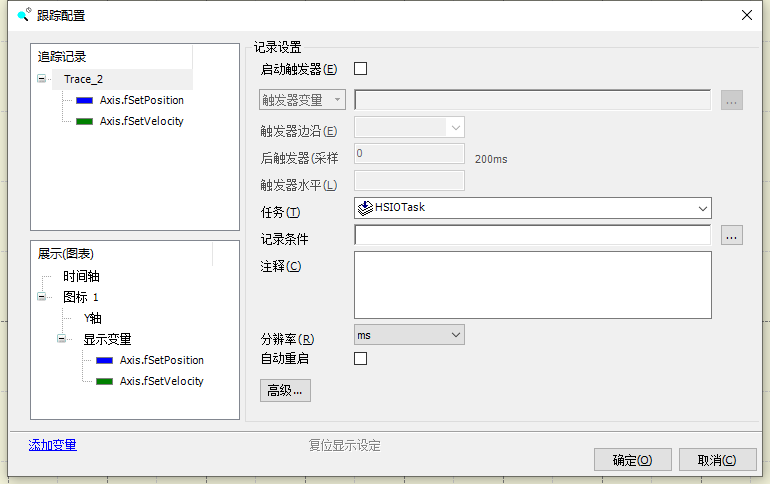
点击刚添加的空变量，点击变量栏后面的按钮，打开输入助手。



以添加轴的指令位置变量为例。展开【IoConfig\_Globals】，再展开【Axis】，找到并双击【fSetPosition】，即完成了轴的指令位置变量的添加。

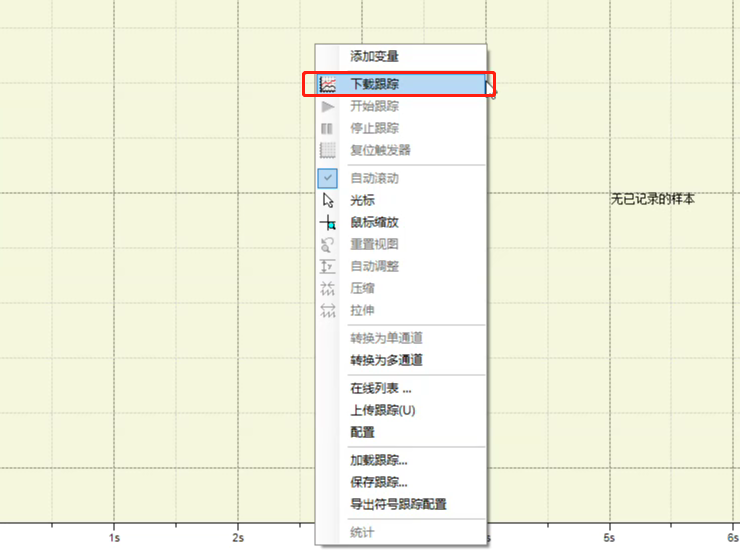


以相同方法添加轴的指令速度变量，添加结果如下。



* 1. 启动Trace

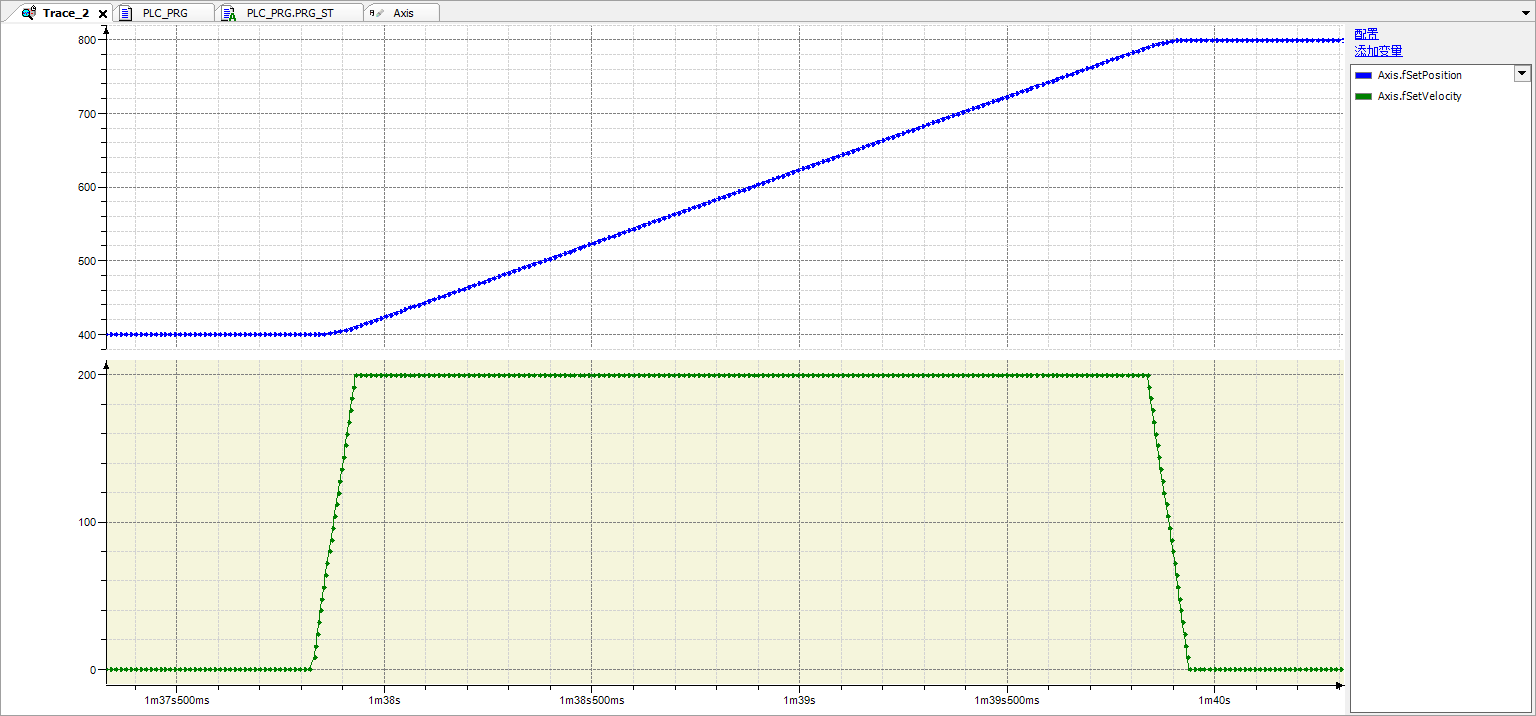
右键Trace图标任意位置，【下载跟踪】，此时Trace已经开始采样。



监测一段运动结果如下。



Trace支持多图表功能，可以右键Trace图标任意位置，【转换为多通道】，结果如下。



1. 恒速运动

2.1 恒速运动指令

MC\_MoveVelocity(

Axis:= 轴号, Execute:= 启动信号, Velocity:= 速度, Acceleration:= 加速度,

Deceleration:= 减速度, Jerk:= 加加速度, Direction:= 方向, BufferMode:= 缓存模式,

InVelocity=> 到达设定速度, Busy=> 运动中, Active=> 轴工作中,

CommandAborted=> 指令中断, Error=> 错误, ErrorID=> 错误代码);

恒速运动指令启动后，将一直不停的运动，若需要停止，则使用MC\_Stop轴停止指令。

2.2 恒速运动例程

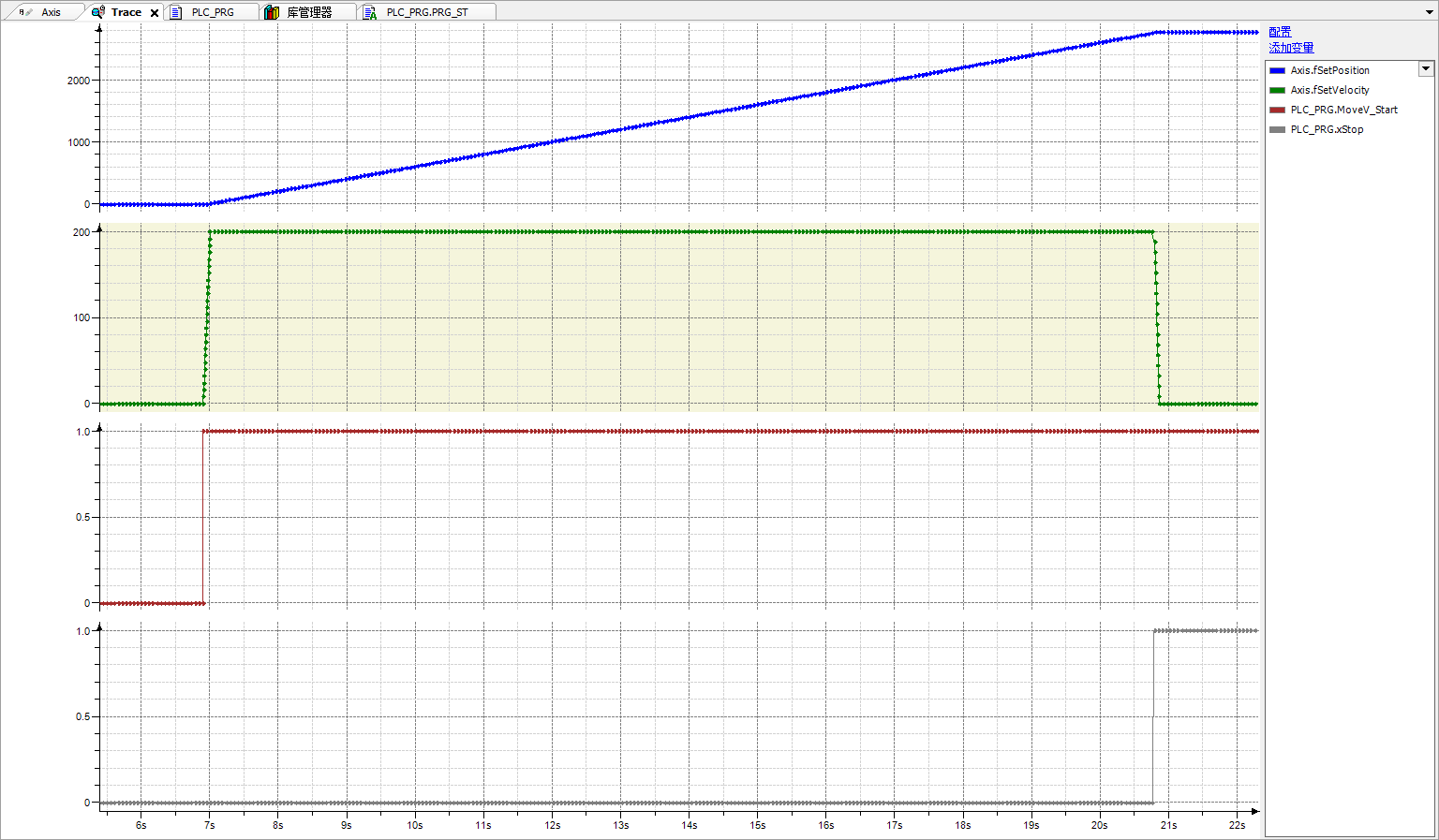
相关程序代码：

|  |
| --- |
| PROGRAM PLC\_PRG  VAR  MC\_Power\_0: MC\_Power; //轴使能  MC\_Stop\_0: MC\_Stop; //轴停止  xStop: BOOL; //轴停止信号  MC\_MoveVelocity\_0: MC\_MoveVelocity; //恒速运动  MoveV\_Start: BOOL; //恒速运动启动信号  END\_VAR |

|  |
| --- |
| //轴使能  MC\_Power\_0(  Axis:=Axis, Enable:=1, bRegulatorOn:=1, bDriveStart:=1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  //轴停止：减速度3000unit/s2,加加速度30000unit/s3  MC\_Stop\_0(  Axis:=Axis, Execute:=xStop, Deceleration:=3000, Jerk:= 30000,  Done=> , Busy=> , Error=> , ErrorID=> );  //恒速运动：速度200unit/s,加减速度2000unit/s2,加加速度20000unit/s3，正向  MC\_MoveVelocity\_0(  Axis:=Axis, Execute:=MoveV\_Start, Velocity:=200, Acceleration:=2000,  Deceleration:=2000, Jerk:=20000, Direction:=1, BufferMode:= ,  InVelocity=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=> ); |

2.3 恒速运动例程执行结果

当变量MoveV\_Start由FALSE变为TRUE时，轴开始以200unit/s的速度正向运动。当变量xStop由FALSE变为TRUE时，轴减速停止。



1. 点动运动（JOG运动）
   1. JOG运动指令

MC\_Jog(

Axis:= 轴号, JogForward:= 正向JOG, JogBackward:= 反向JOG, Velocity:= 速度,

Acceleration:= 加速度, Deceleration:= 减速度, Jerk:= 加加速度, Busy=> 运动中,

CommandAborted=> 指令被中断, Error=> 错误, ErrorId=> 错误代码);

3.2 JOG运动例程

相关程序代码：

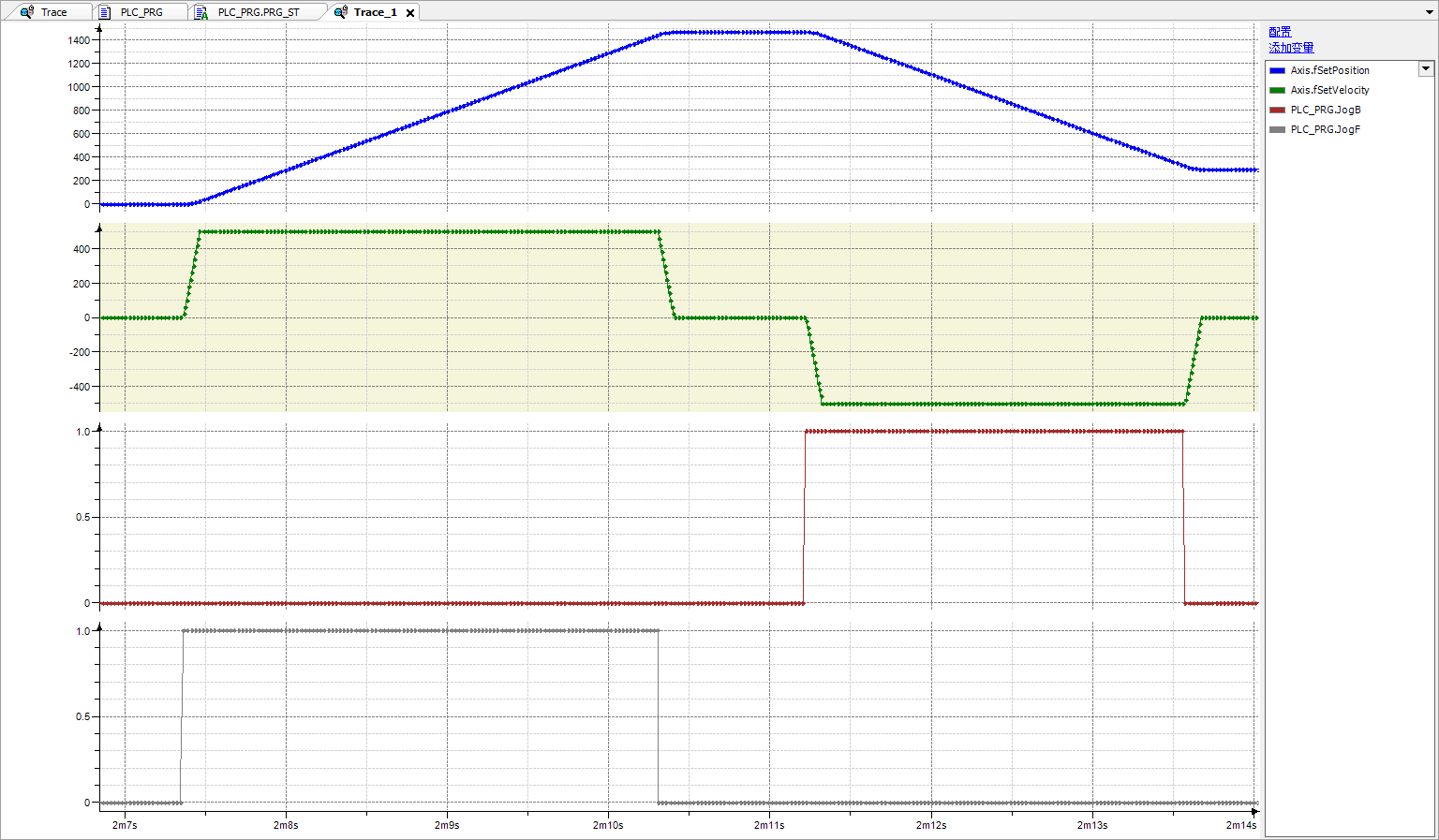
|  |
| --- |
| PROGRAM PLC\_PRG  VAR  MC\_Power\_0: MC\_Power; //轴使能  MC\_Jog\_0:MC\_Jog; //Jog运动  JogF:BOOL; //正向Jog运动信号  JogB:BOOL; //反向Jog运动信号  END\_VAR |

|  |
| --- |
| //轴使能  MC\_Power\_0(  Axis:=Axis, Enable:=1, bRegulatorOn:=1, bDriveStart:=1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  //Jog运动：速度500unit/s,加减速度5000unit/s2,加加速度50000unit/s3  MC\_Jog\_0(  Axis:= Axis, JogForward:= JogF, JogBackward:= JogB, Velocity:= 500,  Acceleration:= 5000, Deceleration:= 5000, Jerk:= 50000, Busy=> ,  CommandAborted=> , Error=> , ErrorId=> ); |

3.3 JOG运动例程执行结果

当变量JogF由FALSE变为TRUE时，轴开始以500unit/s的速度正向运动。当变量JogF由TRUE变为FALSE时，轴开始以5000unit/s的减速度减速停止。

当变量JogB由FALSE变为TRUE时，轴开始以500unit/s的速度反向运动。当变量JogB由TRUE变为FALSE时，轴开始以5000unit/s的减速度减速停止。



1. 点位运动

4.1绝对点位运动指令

MC\_MoveAbsolute(

Axis:= 轴号, Execute:= 启动信号, Position:= 目标位置, Velocity:= 速度,

Acceleration:= 加速度, Deceleration:= 减速度, Jerk:= 加加速度,

Direction:= 方向, BufferMode:= 缓存模式, Done=> 运动完成, Busy=> 运动中,

Active=> 轴工作中, CommandAborted=> 指令被中断,

Error=> 错误, ErrorID=> 错误代码);

4.2相对点位运动指令

MC\_MoveRelative(

Axis:= 轴号, Execute:= 启动信号, Distance:= 运动距离, Velocity:= 速度,

Acceleration:= 加速度, Deceleration:= 减速度, Jerk:= 加加速度,

BufferMode:= 缓存模式, Done=> 运动完成, Busy=> 运动中,

Active=> 轴工作中, CommandAborted=> 指令被中断,

Error=> 错误, ErrorID=> 错误代码);

4.3运动例程

相关程序代码：

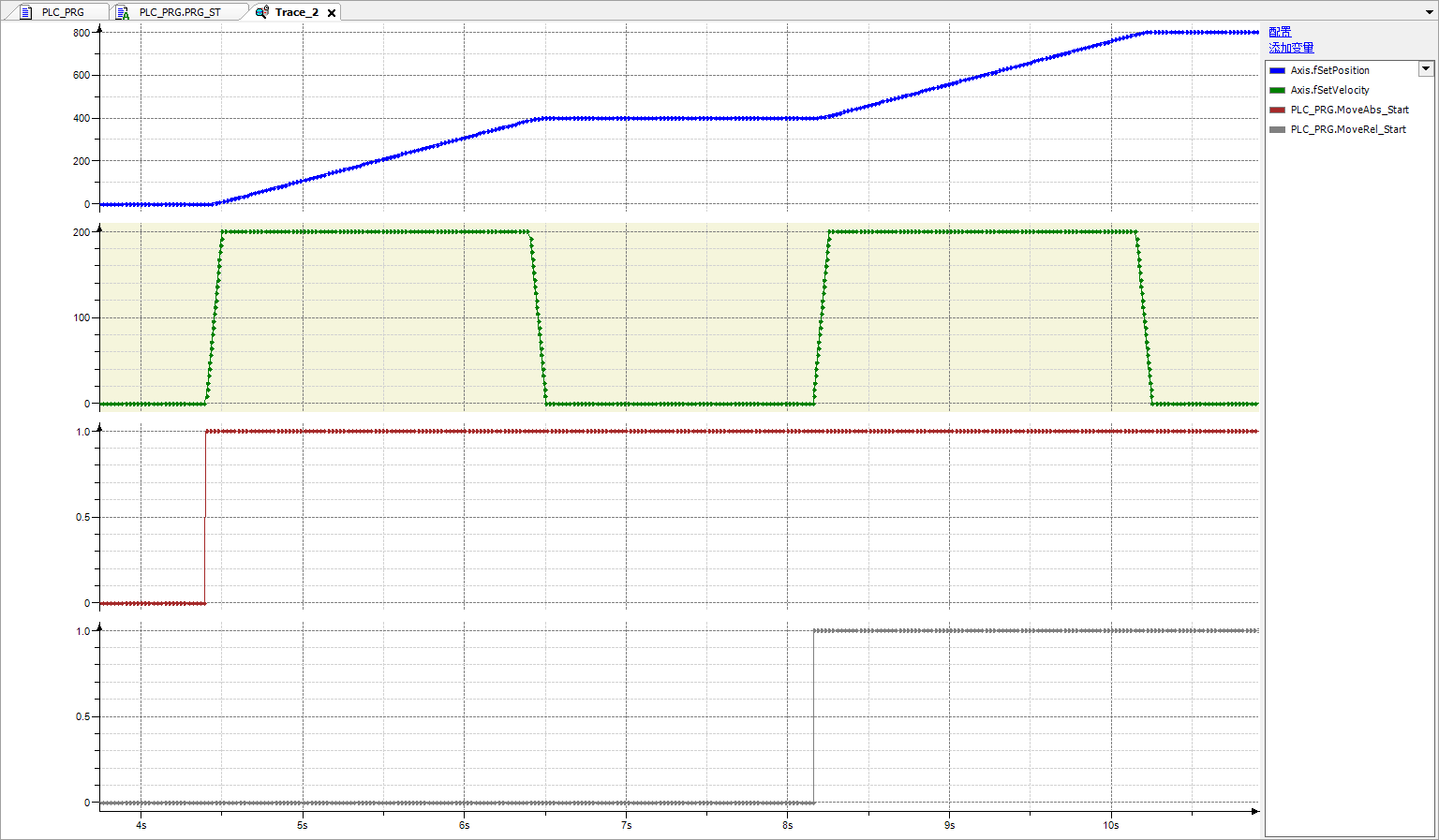
|  |
| --- |
| PROGRAM PLC\_PRG  VAR  MC\_Power\_0: MC\_Power; //轴使能  MC\_MoveAbsolute\_0: MC\_MoveAbsolute; //绝对定位运动  MoveAbs\_Start: BOOL; //绝对定位运动启动信号  MC\_MoveRelative\_0: MC\_MoveRelative; //相对定位运动  MoveRel\_Start: BOOL; //相对定位运动启动信号  END\_VAR |

|  |
| --- |
| //轴使能  MC\_Power\_0(  Axis:=Axis, Enable:=1, bRegulatorOn:=1, bDriveStart:=1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  //绝对点位运动：定位位置400unit,速度200unit/s,加减速度2000unit/s2,加加速度20000unit/s3  MC\_MoveAbsolute\_0(  Axis:= Axis, Execute:= MoveAbs\_Start, Position:= 400, Velocity:= 200, Acceleration:= 2000,  Deceleration:= 2000, Jerk:= 20000, Direction:= , BufferMode:= , Done=> , Busy=> ,  Active=> , CommandAborted=> , Error=> , ErrorID=> );  //相对点位运动：定位位置400unit,速度200unit/s,加减速度2000unit/s2,加加速度20000unit/s3  MC\_MoveRelative\_0(  Axis:= Axis, Execute:= MoveRel\_Start, Distance:= 400, Velocity:= 200, Acceleration:= 2000,  Deceleration:= 2000, Jerk:= 20000, BufferMode:= , Done=> , Busy=> ,  Active=> , CommandAborted=> , Error=> , ErrorID=> ); |

4.4 点位运动例程执行结果

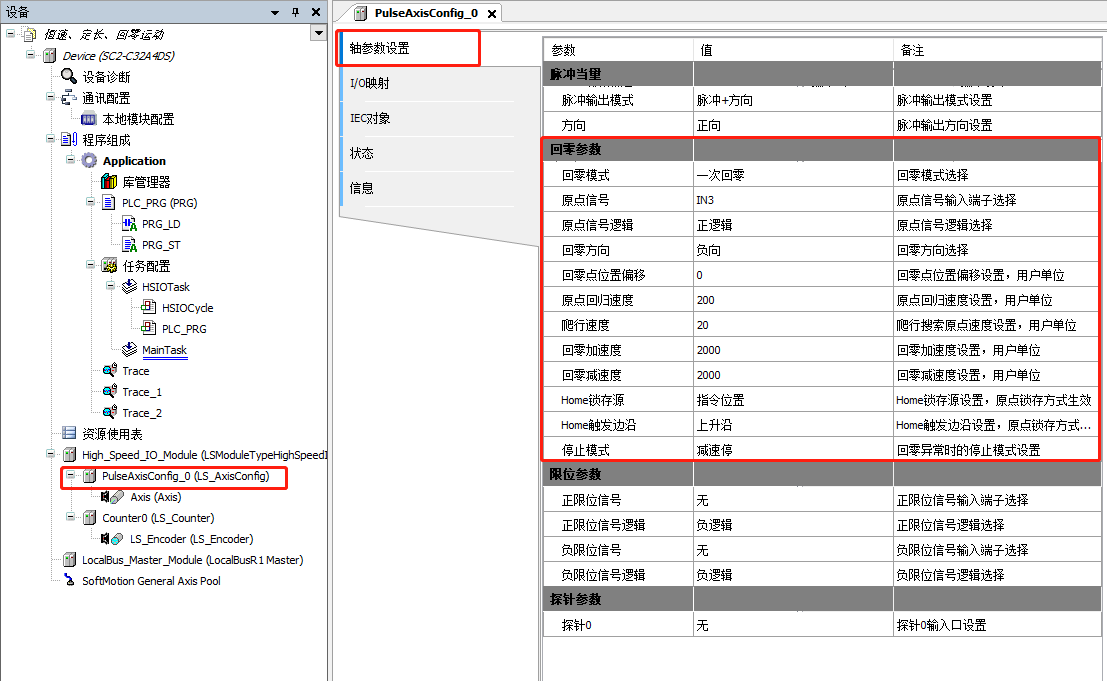
当变量MoveAbs\_Start触发上升沿时，轴开始以200unit/s的速度正向运动到400unit位置。

当变量MoveRel\_Start触发上升沿时，轴开始以200unit/s的速度正向运动400unit，到达800unit位置。



1. 回零运动
   1. 设置回零运动参数

打开“PulseAxisConfig\_0”，点击“轴参数设置”，设置回零参数栏内参数。



回零模式有：一次回零、一次回零加反找、二次回零、原点锁存。

原点信号在专用输入口未被占用时，可选择IN0-IN7，须选择原点位置传感器所安装的输入端口，否则无法实现工作台机械回原点。

* 1. SC2-C脉冲轴回零运动指令

MC\_HOME\_P(

Axis:= 轴号, xExecute:= 启动信号, xAbort:= 中止信号, xDone=> 回零完成,

xBusy=> 回零进行中, xCommandAborted=> 执行被中断,

xError=> 错误, eErrorID=> 错误代码);

注：该指令仿真不能完成回零。

5.3 回零运动例程

回零参数设置：



相关程序代码：

|  |
| --- |
| PROGRAM PLC\_PRG  VAR  MC\_Power\_0: MC\_Power; //轴使能  MC\_Home\_P\_0: MC\_Home\_P; //回零运动  Home\_Start: BOOL; //回零运动启动信号  END\_VAR |

|  |
| --- |
| //轴使能  MC\_Power\_0(  Axis:=Axis, Enable:=1, bRegulatorOn:=1, bDriveStart:=1,  Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );  //回零运动  MC\_Home\_P\_0(  Axis:= Axis, xExecute:= Home\_Start, xAbort:= , xDone=> , xBusy=> ,  xCommandAborted=> , xError=> , eErrorID=> ); |

5.4回零运动例程执行结果

当变量Home\_Start触发上升沿时，轴开始以200unit/s的速度反方向回零。当IN3（地址%IX0.3）原点信号输入口上升沿时，轴以2000unit/s的减速度减速停止，并自动将停止位置设置为0。

