恒速控制及点动功能说明文档

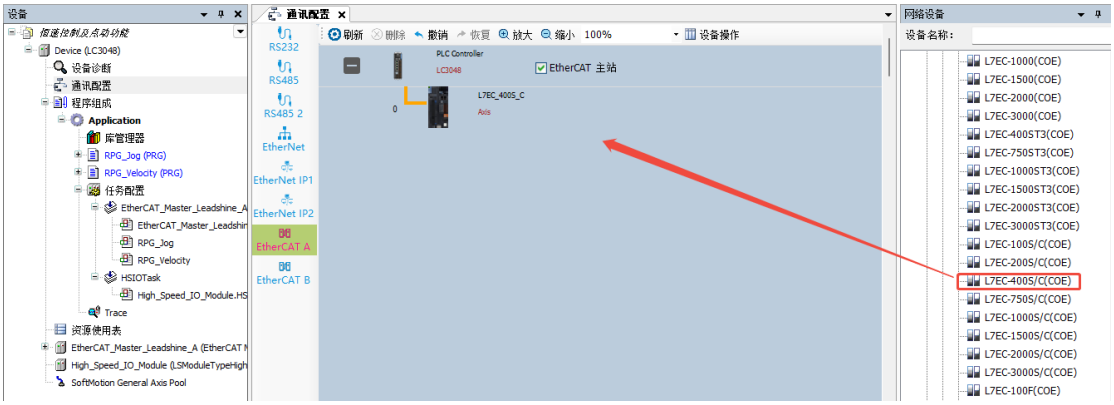
1、概要

恒速运动指令目是控制指定驱动轴，保持恒速运动控制。点动指令是实现手 动控制轴能够朝指定方向运动。

（1）功能运行前的准备：1 个控制器（LC3048），1 台电脑，2 台驱动器 L7EC-400S

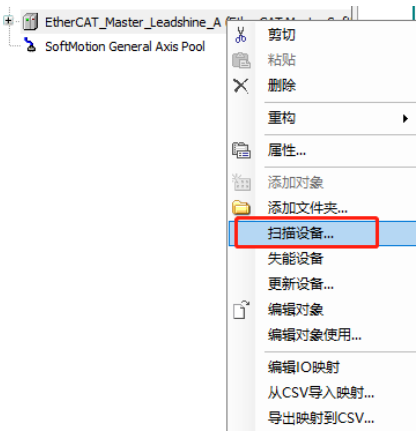
2、轴和 Trace 曲线配置

（1）轴设置 点击“目录树”下 【通讯配置】，双击 L7EC-400S，即可把驱动器和总线轴添加至 EtherCAT_Master_Leadshine_A 中，其设备组态如图 一所示。

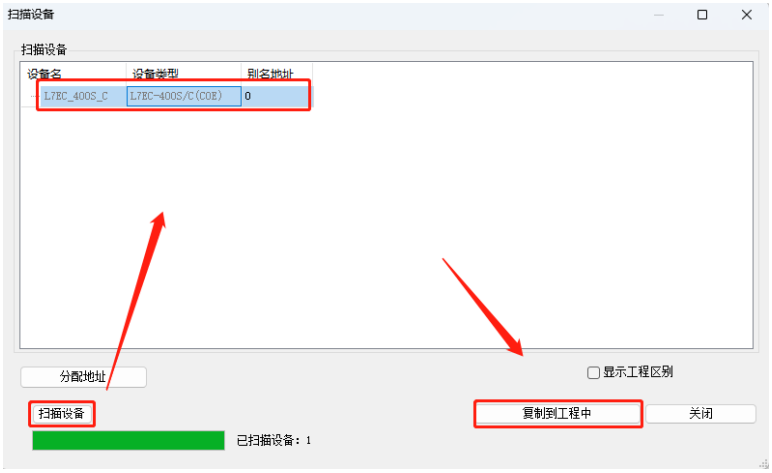


图一

或者右击目录树下“EtherCAT_Master_Leadshine_A”，选择扫描设备，再打 开的界面中选择“复制到工程中”，即可完成添加，分别如下图二、图三所示

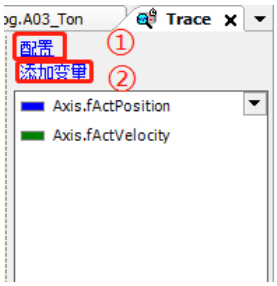


图二



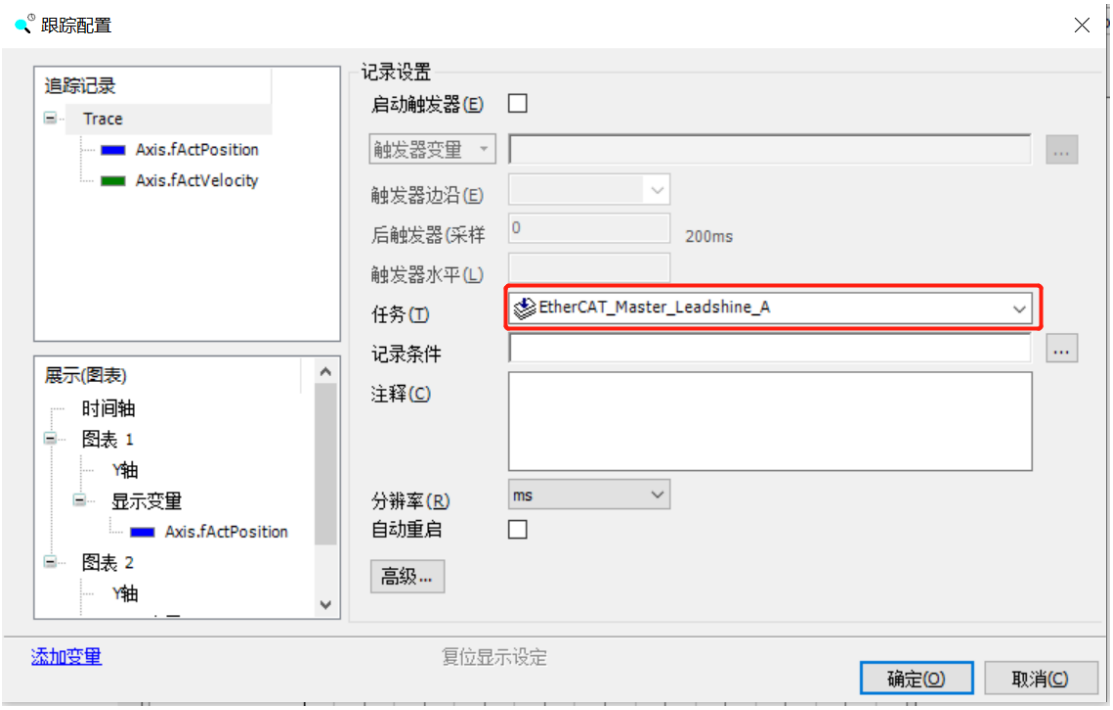
图三

（2）Trace 曲线设置



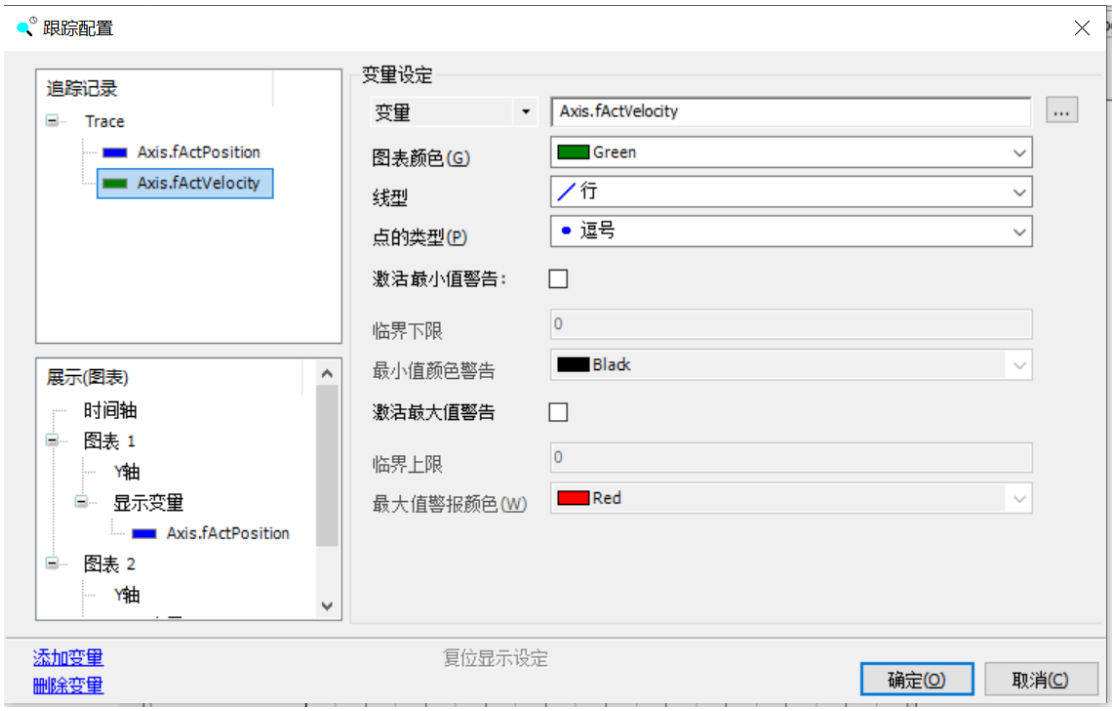
图四

点击图四中 1“配置”，打开跟踪配置界面，如图五所示，选择相对应的跟 踪任务



图五

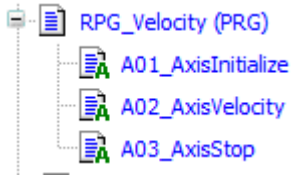
点击图四中 2“添加变量”，打开跟踪配置，如图六所示，添加 2 个变量， Axis.fActPosition（轴位置）、Axis.fActVelocity（轴速度）



图六

3、速度指令说明

（1）程序说明（详细参考例程速度、点动指令），恒速运动指令目录树如图 七所示



图七

（2）变量声明如图八所示

```
PROGRAM RPG_Velocity
VAR
    //********实例化功能块********
    MC_MoveVelocity_0: MC_MoveVelocity;      //总线轴恒速运动功能块
    MC_Power_0: MC_Power;                    //总线轴使能功能块
    MC_Stop_0: MC_Stop;                      //总线轴停止功能块

    //********总线轴功能块参数定义********
    xStartAxis:BOOL;                         //调用总线轴模块启动信号
    xExeAxis: BOOL;                          //总线轴恒速运动启动信号
    xStop:BOOL;                              //总线轴停止信号
    fVelAxis: LREAL:=5;                     //总线轴恒速运动速度
    fAccAxis: LREAL:=100;                    //总线轴恒速运动加速度
    fDecAxis: LREAL:=100;                    //总线轴恒速运动减速度
    fJerkAxis: LREAL:=1000;                  // 总线轴恒速运动加加速度

END_VAR
```

图八

主程序如图九所示

```
RPG_Velocity.A01_AxisInitialize();                    //调用轴使能模块
RPG_Velocity.A03_AxisStop();                          //调用总线轴停止模块

IF xStartAxis THEN
RPG_Velocity.A02_AxisVelocity();                      //调用总线轴恒速运动模块
xExeAxis:=TRUE;
END_IF


IF xStop THEN                                         //轴停止模块
    xExeAxis:=FALSE;
    xStartAxis:=FALSE;
    IF MC_Stop_0.Done THEN
    xStop:=FALSE;
    END_IF
END_IF
```

图九

A01 为轴使能模块，如图十所示

```
MC_Power_0(
    Axis:=Axis, Enable:=TRUE, bRegulatorOn:=TRUE, bDriveStart:=TRUE,
    Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=> );
//总线轴使能
```

图十

A02 为轴恒速运动模块，如图十一所示

```
MC_MoveVelocity_0(
    Axis:=Axis, Execute:=xExeAxis, Velocity:=fVelAxis, Acceleration:=fAccAxis,
    Deceleration:=fDecAxis, Jerk:=fJerkAxis, Direction:=1, BufferMode:= ,
    InVelocity=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=> );
//总线轴恒速运动
```

图十一

A03 为轴停止模块，如图十二所示

```
MC_Stop_0(
    Axis:=Axis, Execute:=xStop, Deceleration:=3000, Jerk:= ,
    Done=> , Busy=> , Error=> , ErrorID=> );
    //轴停止
```

图十二

（3）程序运行 当置 xStartAxis 为 TRUE 时轴开始运动，如图十三所示

表达式	类型	值	预备值	地址	注释
MC_MoveVelocity_0	MC_MoveVelocity				总线轴恒速运动功能块
MC_Power_0	MC_Power				总线轴使能功能块
MC_Stop_0	MC_Stop				总线轴停止功能块
xStartAxis	BOOL	TRUE			调用总线轴模块启动信号
xExeAxis	BOOL	TRUE			总线轴恒速运动启动信号
xStop	BOOL	FALSE			总线轴停止信号
fVelAxis	LREAL	5			总线轴恒速运动速度
fAccAxis	LREAL	100			总线轴恒速运动加速度
fDecAxis	LREAL	100			总线轴恒速运动减速度
fJerkAxis	LREAL	1000			总线轴恒速运动加加速度

```
RPG_Velocity.A01_AxisInitialize();                    //调用轴使能模块
RPG_Velocity.A03_AxisStop();                          //调用总线轴停止模块

IF xStartAxis TRUE THEN
RPG_Velocity.A02_AxisVelocity();                      //调用总线轴恒速运动模块
xExeAxis TRUE :=TRUE;
END_IF

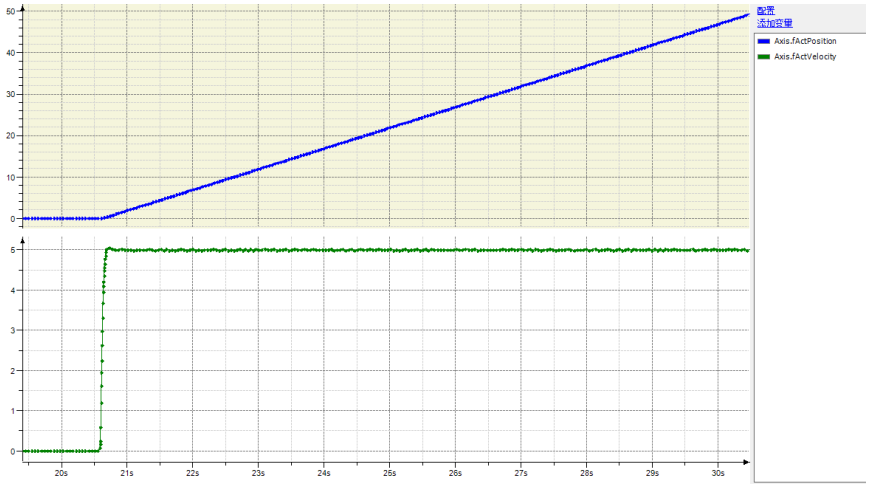

IF xStop FALSE THEN                                   //轴停止模块
    xExeAxis TRUE :=FALSE;
    xStartAxis TRUE :=FALSE;
    IF MC_Stop_0.Done FALSE THEN
    xStop FALSE :=FALSE;
    END_IF
END_IF




RETURN
```

图十三

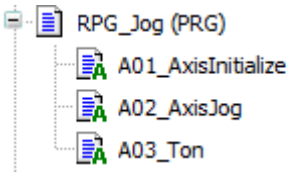
（4）Trace 曲线 当轴进行恒速运动时，trace 抓取的曲线如图十四所示



图十四

4、点动指令说明

（1）程序说明（详细参考例程恒速和点动指令），点动程序的目录树如图十 五所示



图十五

（2）变量声明如图十六所示

```
PROGRAM RPG_Jog
VAR
    //********实例化功能块********
    MC_Power_0: MC_Power;                   //总线轴使能功能块
    MC_Jog_0: MC_Jog;                       //总线轴点动功能块
    TON_0: TON;                             //延时器功能块
    MC_Stop_0: MC_Stop;                     //总线轴停止功能块

    //********总线轴功能块参数定义********
    xForAxis: BOOL;                        //总线轴正向点动启动信号
    xBackAxis: BOOL;                       //总线轴反向点动启动信号
    fVelAxis: LREAL:=1;                   //总线轴点动运动速度
    fAccAxis: LREAL:=100;                  //总线轴点动运动加速度
    fDecAxis: LREAL:=100;                  //总线轴点动运动减速度
    fJerkAxis: LREAL;                    // 总线轴点动运动加加速度
    xStop:BOOL;                            //总线轴停止信号

    //********其他参数定义********
    xStartTime:BOOL;                        //开始延时
    timTime:TIME:=T#30S;                   //延时时间
    iAxisState:INT;                         //轴运动状态

END_VAR
```

图十六

（3）主程序如图十七所示

```
RPG_Jog.A01_AxisInitialize();              //调用轴初始化模块
RPG_Jog.A02_AxisJog();                     //调用总线轴点动模块
RPG_Jog.A03_Ton();                         //调用延时器模块

//********程序说明********
//1、将所有变量复位
//2、总线轴正转30s
//3、总线轴反转30s
//4、脉冲轴正转30s
//5、脉冲轴反转30s，结束后返回到第一步

CASE iAxisState OF       //将iAxisState置为1，开始执行
1:
    xForAxis:=FALSE;
    xBackAxis:=FALSE;
    xStartTime:=FALSE;
    iAxisState:=2;

2:
    xForAxis:=TRUE;
    xStartTime:=TRUE;
    IF TON_0.Q THEN
    xForAxis:=FALSE;
    xStartTime:=FALSE;
    iAxisState:=3;
    END_IF
3:
    xBackAxis:=TRUE;
    xStartTime:=TRUE;
    IF TON_0.Q THEN
    xBackAxis:=FALSE;
    xStartTime:=FALSE;
    iAxisState:=0;
    END_IF

END_CASE
IF xStop THEN                                    //轴停止模块
    IF MC_Stop_0.Done THEN
    iAxisState:=0;
    END_IF
END_IF
```

图十七

A01 轴使能模块，如图十八所示

```
MC_Power_0(
    Axis:=Axis, Enable:=TRUE, bRegulatorOn:=TRUE,
     bDriveStart:=TRUE,
    Status=> , bRegulatorRealState=> , bDriveStartRealState=> ,
    Busy=> , Error=> , ErrorID=> );
    //总线轴使能

```

图十八

A02 轴点动和停止模块，如图十九所示

```
MC_Jog_0(
    Axis:=Axis, JogForward:=xForAxis, JogBackward:=xBackAxis,
    Velocity:=fVelAxis, Acceleration:=fAccAxis, Deceleration:=fDecAxis,
    Jerk:=fJerkAxis,
    Busy=> , CommandAborted=> , Error=> , ErrorId=> );
    //轴点动
MC_Stop_0(
    Axis:=Axis, Execute:=xStop, Deceleration:=3000, Jerk:= ,
    Done=> , Busy=> , Error=> , ErrorID=> );
    //轴停止
```

图十九

A03 定时器模块，如图二十所示

```
1   TON_0(IN:=xStartTime, PT:=timTime, Q=> , ET=> );
2   //延时器功能块
```

图二十

（3）程序运行

将 iAxisState 置为 1，开始执行，如图二十一所示

```
CASE iAxisState 3 OF        //将iAxisState置为1，开始执行
1:
    xForAxis FALSE :=FALSE;
    xBackAxis TRUE :=FALSE;
    xStartTime TRUE :=FALSE;
    iAxisState 3 :=2;

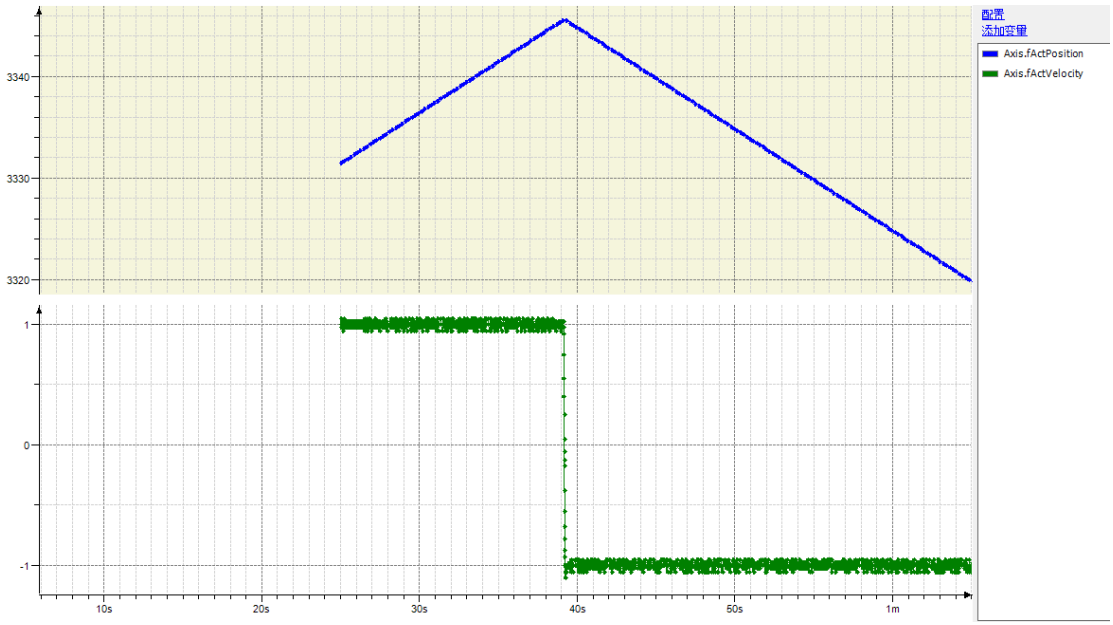
2:
    xForAxis FALSE :=TRUE;
    xStartTime TRUE :=TRUE;
    IF TON_0.Q FALSE THEN
    xForAxis FALSE :=FALSE;
    xStartTime TRUE :=FALSE;
    iAxisState 3 :=3;
    END_IF
3:
    xBackAxis TRUE :=TRUE;
    xStartTime TRUE :=TRUE;
    IF TON_0.Q FALSE THEN
    xBackAxis TRUE :=FALSE;
    xStartTime TRUE :=FALSE;
    iAxisState 3 :=0;
    END_IF

END_CASE
```

图二十一

（4）Trace 曲线

trace 抓取的曲线如图所示二十二所示



图二十二