**电子齿轮、电子凸轮**

电子齿轮用来建立主轴与从轴之间的特定线性速度比率（齿轮比）关系，该功能可用于两个相互配合的传送带的驱动电机。当一个产品从一个传送带传到另一个，两个传送带必须定时定点交接产品，所以主从传送带的速度必须保持同步，使用电子齿轮功能即可简单地实现量传送带的同步运动。SC2-C系列PLC提供电子齿轮功能。

凸轮是一个具有曲线轮廓的构件，通常作等速转动；从动件在凸轮的驱动下，按照凸轮轮廓曲线定义的运动规律上下运动。SC2-C系列PLC提供电子凸轮功能。用PLC代替凸轮机构，不但可降低硬件成本，而且改变凸轮运动规律简单、方便。

本文介绍SC2-C系列PLC电子齿轮和电子凸轮的实现方法。

**1.电子齿轮**

1.1电子齿轮相关指令

电子齿轮相关指令一览。

|  |  |
| --- | --- |
| **名称** | **功能** |
| MC\_GearIn | 设置主从轴、齿轮比，并启动电子齿轮 |
| MC\_GearInPos | 设置主从轴、同步距离、齿轮比，并启动电子齿轮 |
| MC\_GearOut | 断开电子齿轮 |
| MC\_Phasing | 从动轴跟随主动轴 |

1 齿轮耦合指令MC\_GearIn

该指令用于设置电子齿轮比及相关参数，并启动电子齿轮。其格式如下。

MC\_GearIn (Master:=主动轴号, Slave:=从动轴号, Execute:=启动信号,

RatioNumerator:=齿轮比分子, RatioDenominator:=齿轮比分母,

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

BufferMode:=缓存模式, InGear=>齿轮啮合完成, Busy=>齿轮工作中,

Active=>轴在工作, CommandAborted=>指令被中断,

Error=>出错, ErrorID=>错误码 );

功能说明：

MC\_GearIn是速度同步类指令，执行该指令后，从动轴将处于同步状态；要解除耦合，必须调用GearOut指令。

当从动轴的速度到达设定速度时，InGear信号变为TRUE，此后，

从动轴的位移量 = 主动轴的位移量 × RatioNumerator / RatioDenominator

MC\_GearIn指令处于运行状态时，重新触发MC\_GearIn指令就可改变电子齿轮比，不需要调用MC\_GearOut指令暂停电子齿轮运动。

注意：执行该指令过程中，请不要使用MC\_SetPosition指令，以免引起电机转速突变而发生事故。

2 齿轮平滑耦合指令MC\_GearInPos

该指令与MC\_GearIn指令略有不同。它可以指定开始同步时的主动轴位置、从动轴位置、主动轴开始同步的距离，并以此来完成切入电子齿轮的动作。其格式如下。

MC\_GearInPos( Master:= 主动轴号, Slave:= 从动轴号, Execute:= 启动信号,

RatioNumerator:= 齿轮比分子, RatioDenominator:= 齿轮比分母,

MasterSyncPosition:= 主动轴同步位置, SlaveSyncPosition:= 从动轴同步位置,

MasterStartDistance:= 主动轴开始距离, BufferMode:= 缓存模式，

AvoidReversal:= 禁止反转， StartSync=> 同步开始，

InSync=> 同步完成, Busy=> 齿轮工作中, Active=> 轴在工作,

CommandAborted=> 指令被中断, Error=> 出错, ErrorID=> 错误码 );

3 齿轮脱离指令MC\_GearOut

该指令将从动轴与主动轴之间的耦合断开，但对主动轴的动作无影响。其格式如下。

MC\_GearOut( Slave：= 从动轴号， Execute：= 启动信号， Done=> 执行完成，

Busy=> 执行中， Error=> 出错, ErrorID=> 错误码 );

**注意：**齿轮脱离完成后，此时从轴的速度为脱离前的速度，所以需配合MC\_Stop指令停止从轴。

4 位置差指令MC\_Phasing

该指令将结束MC\_GearIn或MC\_CamIn指令，但并不停止从动轴的运动，而是以设置的速度、加速度和加加速度调整从动轴的运动，使得从动轴与主动轴保持一个固定的位置差。其格式如下。

MC\_Phasing( Master:=主动轴号, Slave:=从动轴号, Execute:= 启动信号, PhaseShift:=位置差,

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

Done=>指令完成, Busy=>指令进行中, CommandAborted=>指令被中断,

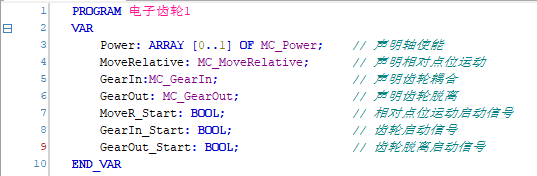
Error=>出错, ErrorID=>错误码 );

1.2电子齿轮例程1

编码器轴LS\_Encoder作电子齿轮的主轴，脉冲轴Axis\_1作电子齿轮的从轴，当电子齿轮啮合时，脉冲轴Axis\_1按1：3的减速比跟随编码器轴LS\_Encoder运动。使用脉冲轴Axis作为编码器轴LS\_Encoder的脉冲输入源，需要短接out0、out1和in0、in1端口。

1.2.1 例程代码

编写如图1代码。



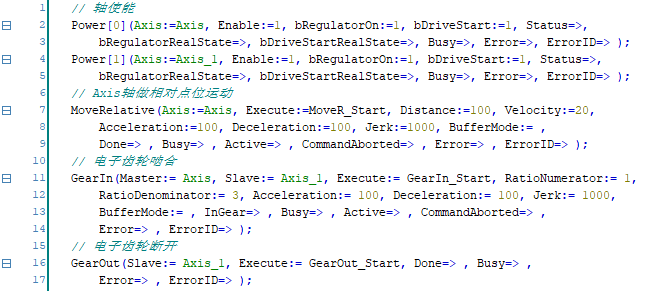


图1 电子齿轮例程1代码

1.2.2 例程效果

例程1效果如下（操作步骤如图2所示）：

①使能计数器，开始编码器轴LS\_Encoder的计数。

②开启编码器轴LS\_Encoder与脉冲轴Axis\_1的电子齿轮啮合。

③开始脉冲轴Axis的点位运动。因为脉冲轴0输出是短接到计数器1的，所以此时编码器轴数值开始增加。通过trace监控LS\_Encoder数值和Axis\_1位置的变化如图3所示。

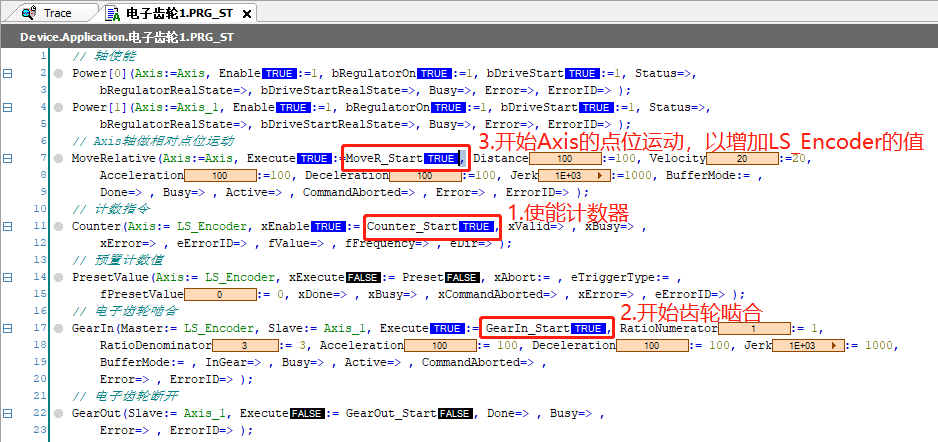


图2 电子齿轮例程1操作步骤

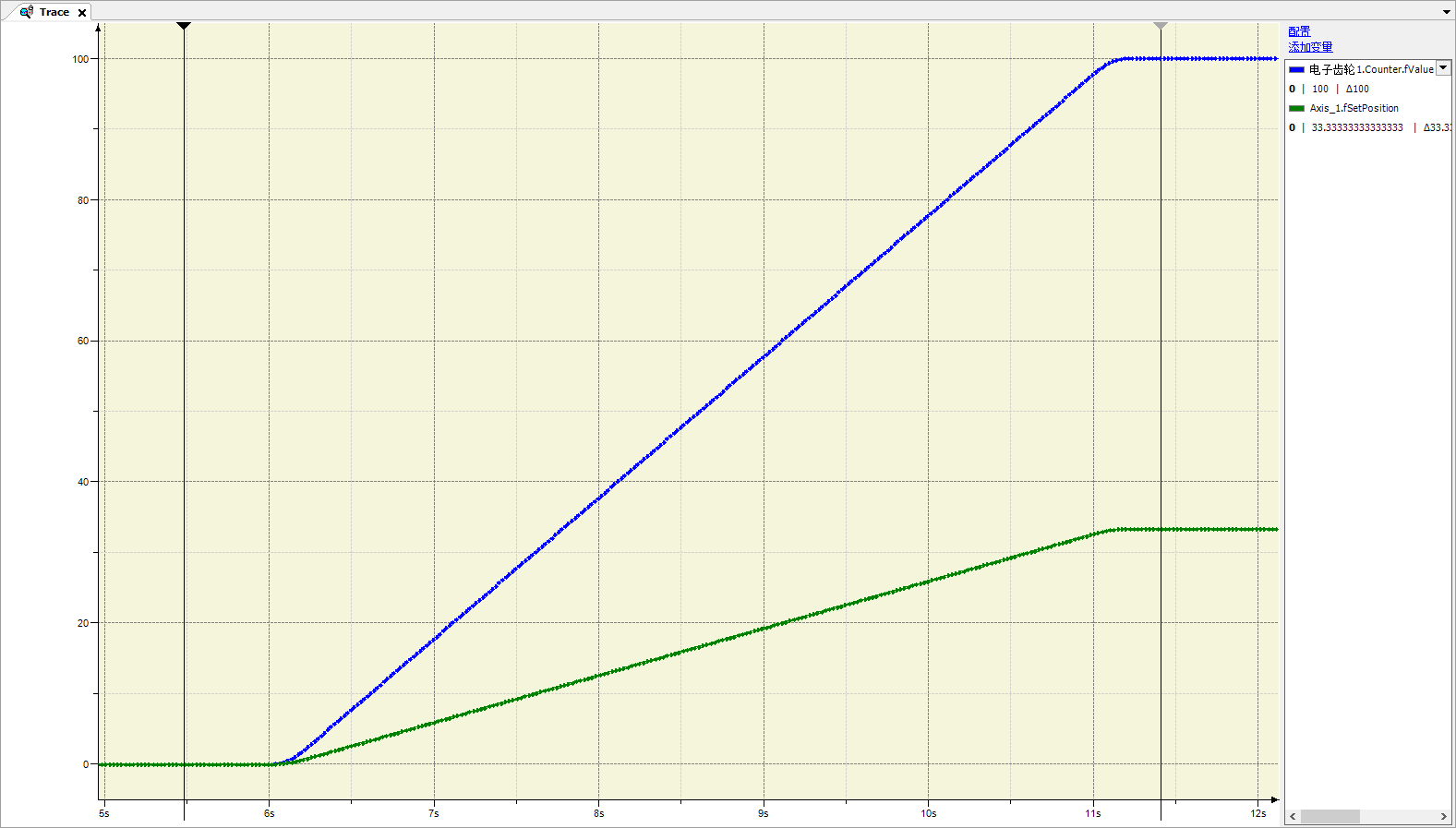


图3 电子齿轮啮合时主轴与从轴的变化

图3显示，主轴LS\_Encoder从位置0作相对点位运动到达100位置处，从轴Axis\_1作同步运动从0位置运动到33.333位置处。

1.3电子齿轮例程2

Axis\_2作为主轴进行恒速运动，当变量IN0为TRUE时，Axis\_3作为从轴电子齿轮啮合作同步运动，变量IN0为FALSE时，Axis\_3断开电子齿轮并停止。

1.3.1 例程代码

编写如图4代码。





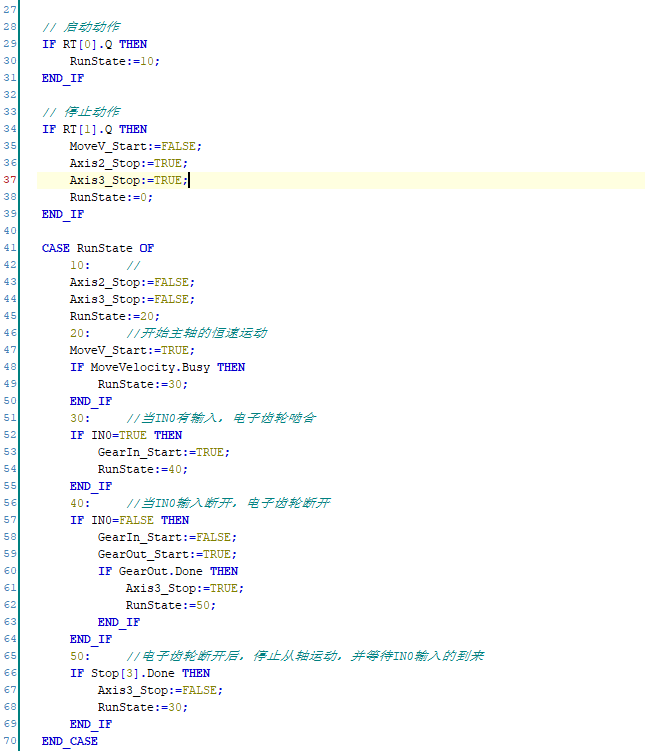


图4 电子齿轮例程2代码

1.3.2 例程效果

例程2效果如下（如图5所示）：

①将变量“例程启动标志”置为TRUE，Axis\_2开始恒速运动。

②当变量“IN”为TRUE时，电子齿轮啮合，Axis\_3以1：10的减速比反向运动。

③当变量“IN”为FALSE时，电子齿轮断开，Axis\_2继续恒速运动，Axis\_3停止。

④变量“例程停止标志”变为TRUE时，Axis\_2和Axis\_3停止。

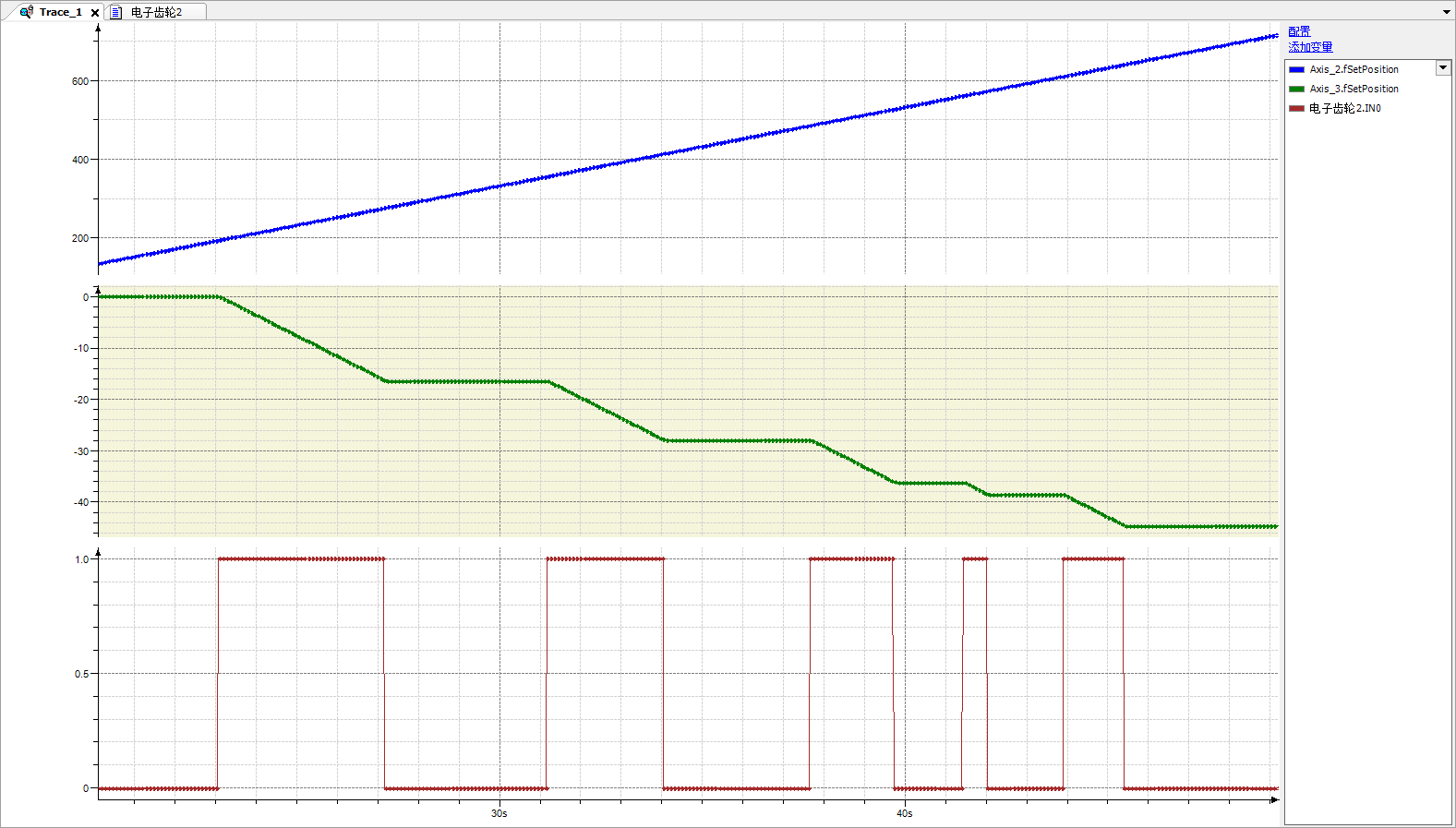


图5 电子齿轮例程2效果

**2.电子凸轮**

2.1电子凸轮相关指令

电子凸轮相关指令一览。

|  |  |
| --- | --- |
| **名称** | **功能** |
| MC\_CAM\_REF | 定义凸轮表参数 |
| MC\_CamTableSelect | 选择cam表 |
| MC\_CamIn | 设置主从轴参数并启动电子凸轮 |
| MC\_CamOut | 断开电子凸轮 |

1 凸轮表参数定义指令MC\_CAM\_REF

如果使用程序生成凸轮表，需要调用该指令；如果使用LeadSys软件手动生成凸轮表，就不用调用该指令。

MC\_CAM\_REF( wCamStructID:=CamStruct的ID, byType:=凸轮轮廓类型,

byVarType:=凸轮轮廓数据类型, xStart:=主动轴开始位置, xEnd:=主动轴结束位置,

nElements:=凸轮轮廓数据点个数, nTappets:=, pce:=凸轮地址指针,

pt:= , dwTappetActiveBits:= ,strCAMName:=,

byInterpolationQuality:=插补质量, byCompatibilityMode:= ,

bChangedOnline:= , xPartofLM:= );

其中：

凸轮轮廓类型：0 − 五阶多项式；

1 − 等距，根据从动轴位置表；

2 − 非等距，根据主从轴相对位置表；

3 − 多项式，根据主动轴位置，从动轴位置、速度和加速度。

凸轮轮廓数据类型：1 − INT，2 − UINT，3 − DINT，4 − UDINT，5 − REAL，6 − LREAL

插补质量：1 −直线插补，3 − 三次插补

2 选择凸轮表指令MC\_CamTableSelect

该指令用于凸轮表选择，输出凸轮表ID号；需要和指令MC\_CamIn配合使用。

MC\_CamTableSelect( Master：=主动轴轴号， Slave：=从动轴轴号， CamTable：=凸轮表名，

Execute：=启动信号（上升沿有效）， Periodic：=重复模式，

MasterAbsolute：=主动轴位置模式， SlaveAbsolute：=从动轴位置模式，

Done=>执行完成， Busy=>执行中， Error=>出错， ErrorID=>错误代码，

CamTableID=>凸轮表ID );

其中：

重复模式：TRUE − 凸轮重复转动；FALSE − 凸轮转一周。

主动轴位置模式：TRUE − 绝对坐标，FALSE − 相对坐标

从动轴位置模式：TRUE − 绝对坐标，FALSE − 相对坐标

3 启动凸轮指令MC\_CamIn

设置主动轴和从动轴的参数并启动电子凸轮。

MC\_CamIn( Master：=主动轴轴号， Slave：=从动轴轴号， Execute：=启动信号（上升沿有效），

MasterOffset：=主动轴位置偏移， SlaveOffset：=从动轴位置偏移，

MasterScaling：=主动轴位置比例， SlaveScaling：=从动轴位置比例，

StartMode：=从动轴啮合方式， CamTableID：=凸轮表ID，

VelocityDiff：=速度， Acceleration：=加速度， Deceleration：=减速度，

Jerk：=加加速度， TappetHysteresis：=挺杆滞后位置，

InSync=>同步中， Busy=>执行中， CommandAborted=>指令被中断， Error=>出错，

ErrorID=>错误代码， EndOfProfile=>凸轮曲线结束， Tappets=>挺杆表 );

4 断开凸轮指令MC\_CamOut

该指令解除指定的从动轴与主动轴的凸轮耦合关系。

MC\_CamOut( Slave：=从动轴轴号， Execute：=启动信号（上升沿有效）， Done=>执行完成，

Busy=>执行中， Error=>出错， ErrorID=>错误代码 );

**注意：**如果从动轴在执行该指令前速度不为0，即使凸轮耦合断开，从动轴仍然按照耦合时的速度运行。故MC\_CamOut指令通常会与MC\_Stop指令配合使用。

2.2电子凸轮例程

X轴作为电子凸轮的主轴，Y轴作为从轴。X轴作恒速运动，电子凸轮耦合后，当X轴旋转到120°时，Y轴开始伸出，当X轴旋转到150°时，Y轴开始缩回，当X轴旋转到180°时，Y轴停止。

2.2.1 创建cam表

创建凸轮表的步骤如下：

①鼠标右键单击工程设备栏中的“Application”，选择“添加对象”-“cam表”，如图6所示；命名为Cam1，并打开凸轮表；

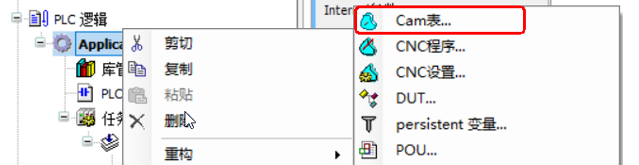


图6 新建Cam表

②在X、Y轴的位置填入如图7所示凸轮表的值，凸轮表会自动生成速度V。



图7 设置cam表参数

③点击“Cam”子窗体查看凸轮曲线，如图8所示。可以用鼠标拖动凸轮位移曲线上的符号⊕，修改凸轮曲线。

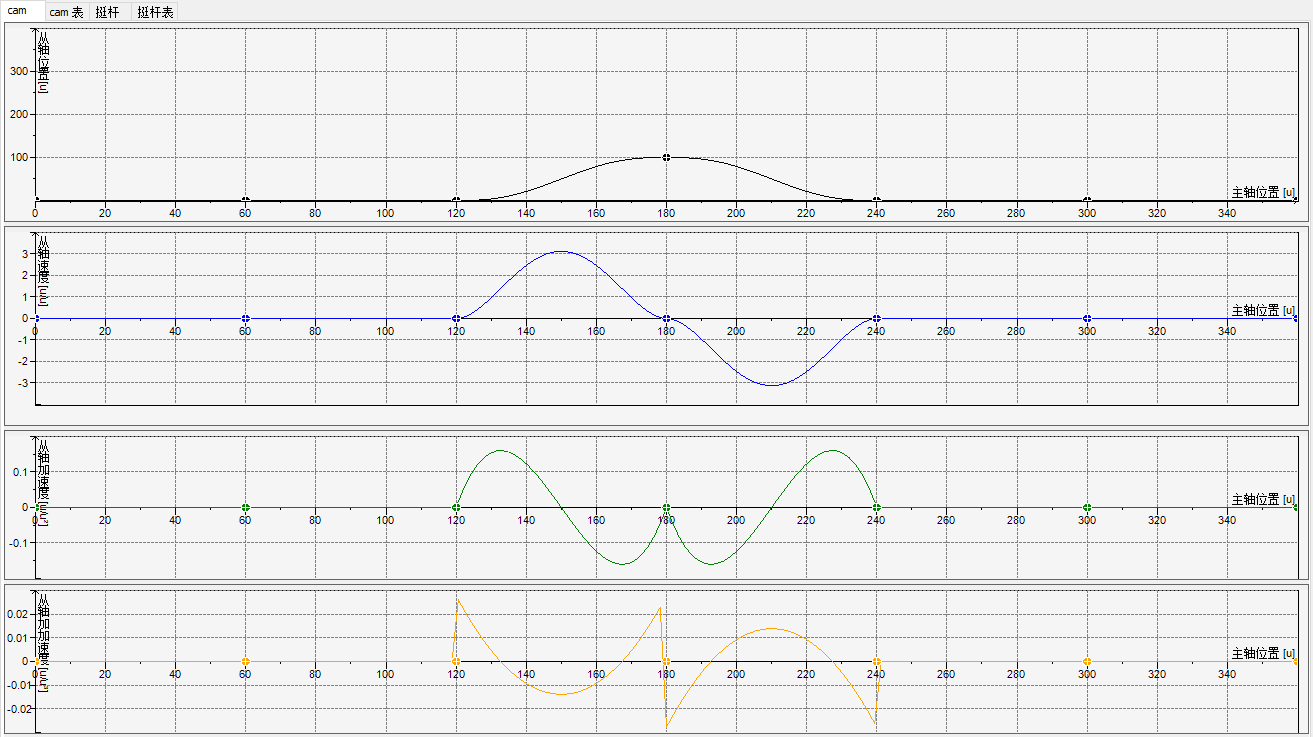
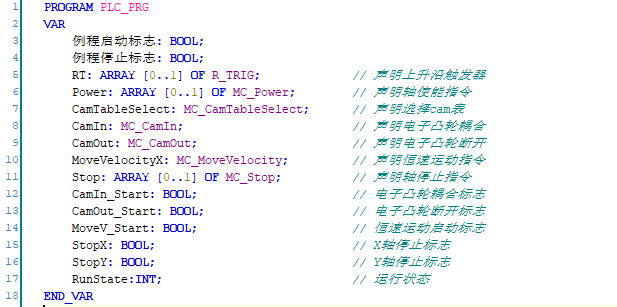
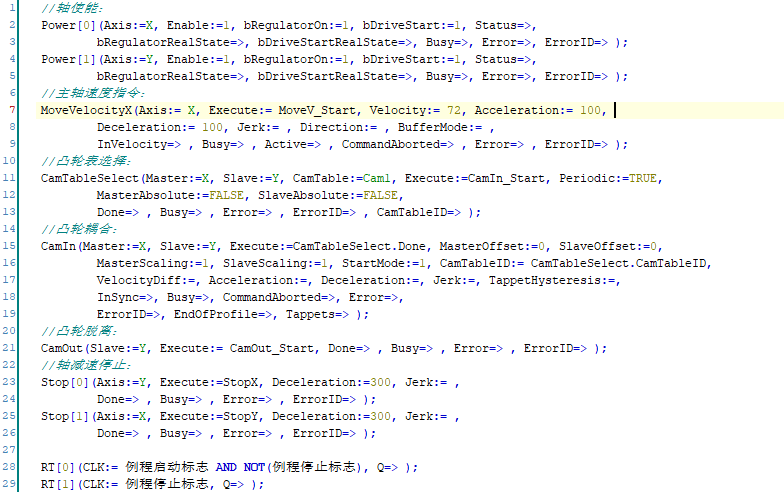


图7 凸轮的位移、速度曲线

2.2.2 例程代码

编写如图8代码。





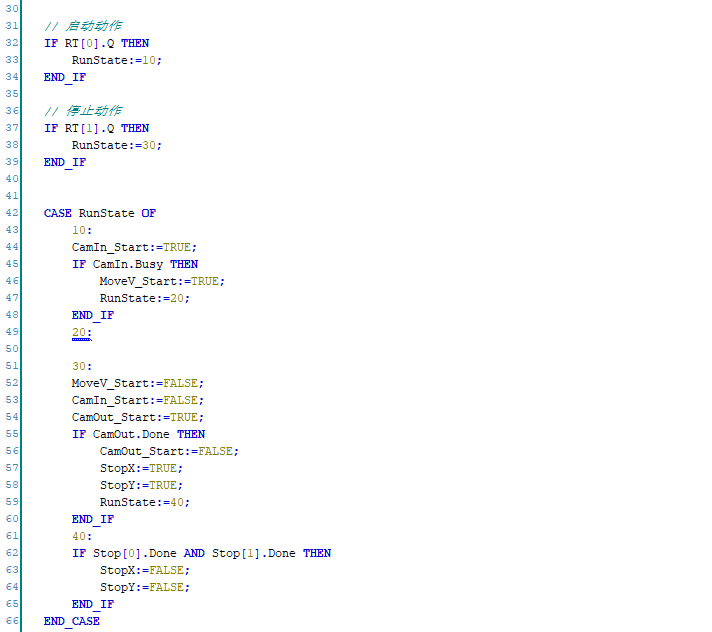


图8 电子凸轮例程代码

2.3.3 例程效果

①将变量“例程启动标志”置为TRUE，X轴开始恒速运动，Y轴开始电子凸轮的同步运动，如图9所示。

②变量“例程停止标志”变为TRUE时，电子凸轮断开，X、Y轴停止。

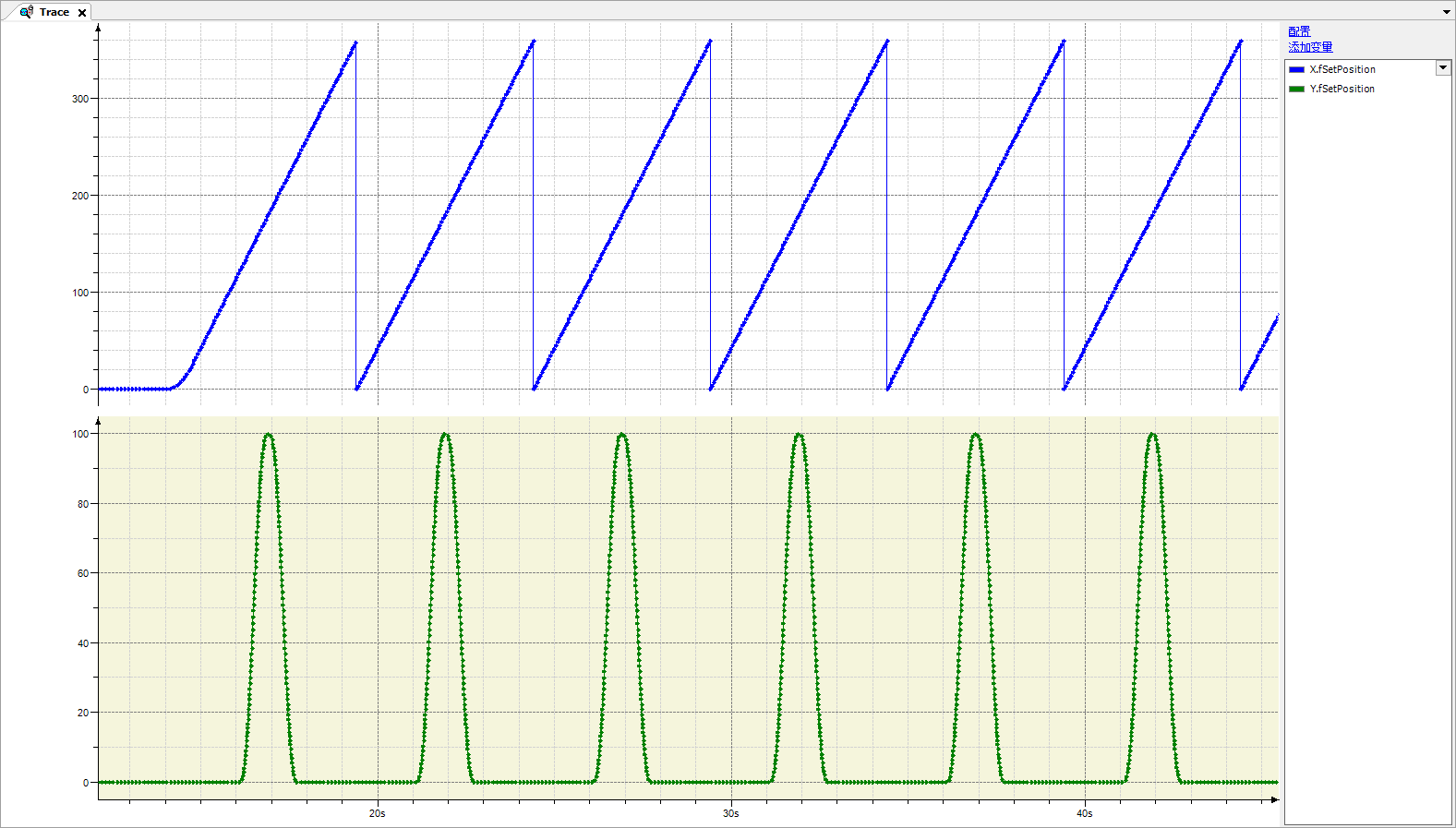


图9 电子凸轮例程运动监控