CoDeSys SoftMotion

CoDeSys SoftMotion – 逻辑控制与运动控制的完美集成

将运动控制功能集成在 CoDeSys 编程系统和 CoDeSys 实时运行系统中，便形成了 CoDeSys SoftMotion 工具包。从简单的运动控制到复杂 CNC 控制的多种复杂应用，都可以使用德国 3S 软件公司的运动控制解决方案 CoDeSys SoftMotion 来进行编程和控制。通常，传统的运动控制解决方案是使用单片机及其相关的硬件来实现的。CoDeSys SoftMotion 则提供了一种完全与众不同的优质高效方案：集成在 PLC 编程系统中的工具包 CoDeSys SoftMotion 提供了运动控制所必需的全部功能。这个工具包集成了符合 IEC 61131-3 国际标准的编程语言。用户可以用一个抽象的数据结构（现场总线和独立的硬件制造商）来实现对轴的各种操作。而现场总线的通信则是通过驱动接口来实现的。这些驱动接口可以通过 CoDeSys 工程树中的设备配置器来进行配置。

3S 软件公司为以下运动控制模式提供了开发工具包：

- 使用 PLCopen 运动控制单元（POUs）的单轴与多轴运动控制
- 电子凸轮传动控制
- 电子齿轮传动控制
- 多轴的 CNC 控制

CoDeSys SoftMotion 包括：

- 包含基于 IEC61131-3 标准的所有运动控制功能块的 POU 库函数，此标准是由 PLCopen 国际组织定义的。
- 附带所有必要组件的 CNC POU 库函数，可开发出多种不同的运动控制系统：从一个插补器到轨迹生成的 POU。此设计使得所有的程序组合单元（POUs）可以实现无缝集成，而且可以将整个应用开发清晰地划分到不同的任务中去。
- 通用接口的驱动如 CAN、EtherCAT、SERCOS、模拟量等等。
- 适用于目前众多的硬件驱动系统，如博世力士乐、伦茨、丹纳赫、KEB、施奈德、倍福等等。
- 用来规划和编辑运动控制的图形化编辑器：

  —图形化的 DIN 66025 编辑器（支持 G-Code）

  —CAM 编辑器

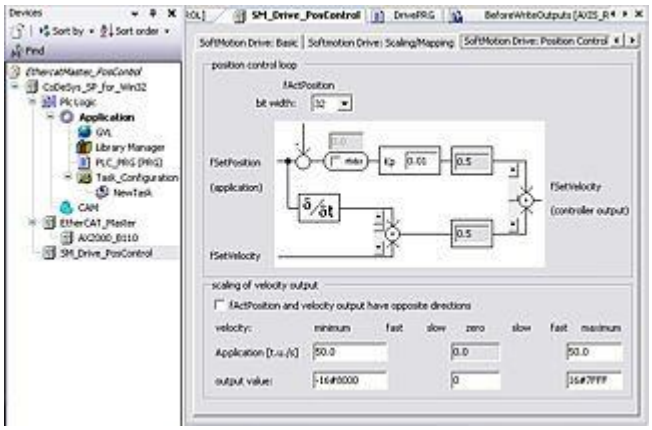
- 所有库功能中的复杂可视化组件均可用于操作和测试界面的快速生成。

CoDeSys SoftMotion 适用于所有 32 位 CPU 和支持浮点运算的不同的 CoDeSys 工具包。与 CoDeSys Control RTE 相结合后，CoDeSys SoftMotion 可以运行在任何一台标准的 PC 机上。

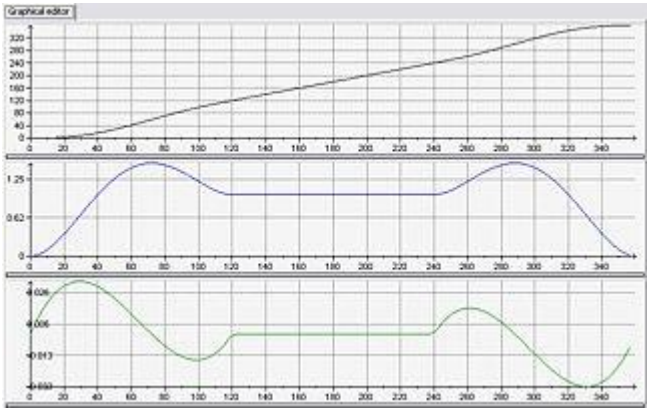
支持的驱动器：

标准的 **3S** 公司软件接口目前支持下列驱动：

- **CAN/CANopen:** JAT Ecovario, KEB F5, Nanotec SMCI47S, **Schneider Electric** Lexium05/Lexium32 and SD-3
- **EtherCAT: Beckhoff** EL2521 u. EL5101, **Control Techniques** (Digitax, Mentor, Unidrive), **Copley** Accelnet, **Danaher** Servostar 300, 400, 600, 700, KEB F5, **Stöber** Posidrive



PLC 中的闭环位置

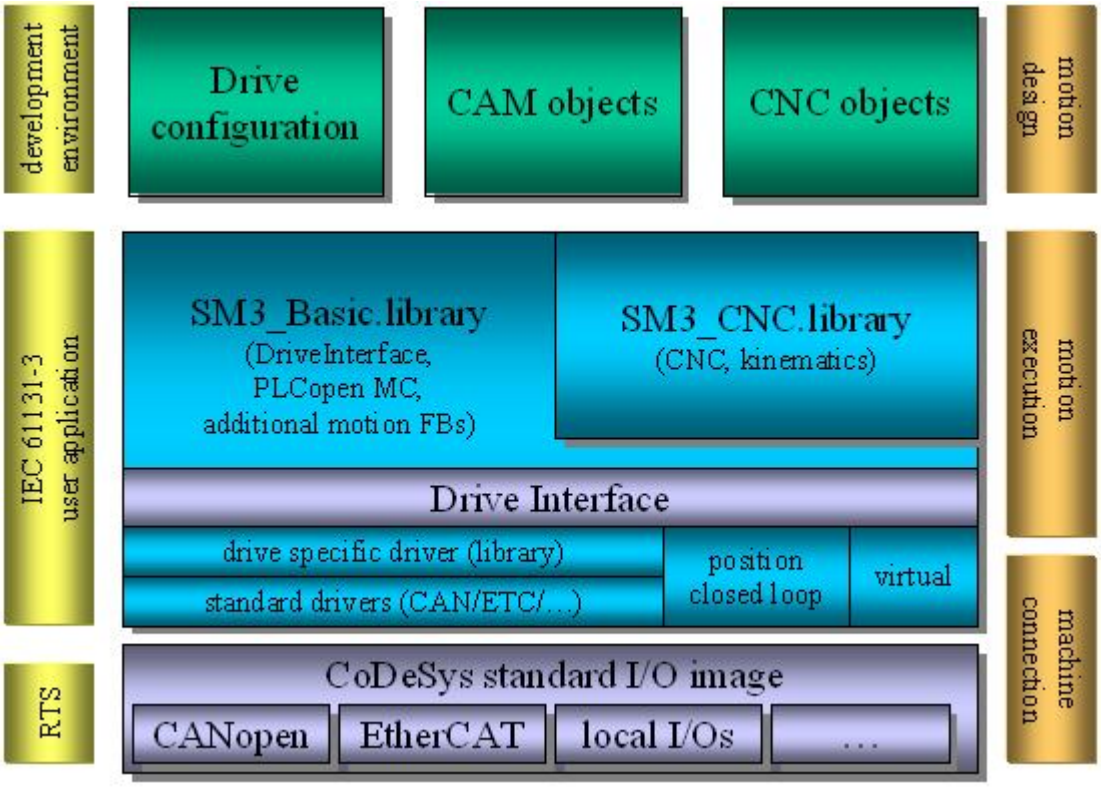


集成在 **CAM** 编辑器中的运动规划

CoDeSys SoftMotion 组件化的概念

CoDeSys SoftMotion 能够使用户高效地实现运动控制而无需了解其烦琐的底层细节：从简单的单轴运动或电子凸轮到多维的复杂运动控制。主要的应用不仅可集中在运动功能特性，而且还集中在序列和过程控制以及相关的功能，使得运动控制应用与 **CoDeSys** 开发环境融为一体。程序逻辑完全由 PLC 程序处理完成，其中纯粹的运动控制则由调用库函数执行。

CoDeSys SoftMotion 是一个软件工具包，它是运动控制器开发及实时的运行环境。除了包含 **CoDeSys** 和 **CoDeSys** Control 的标准特性外，其还包含了运动控制器所特有的部分:



▪ **驱动配置**

该配置提供了一个编辑器用来导入的结构和驱动-硬件配置到 **CoDeSys** 用户界面中。因此，驱动界面库中的函数将会创建 IEC 数据结构作为该驱动的抽象表示。例如，IEC 程序员不需要做任何的额外工作，驱动界面都会自动地与驱动器设备进行通讯，并且能处理该驱动器数据结构及完成传输更新数据。为了控制驱动器，IEC 程序可以有两种方式访问其抽象的数据结构，通过使用 **SoftMotion** 库的标准模块（**SM3_Basic.library, SM3_CNC.library**），或者通过使用由 IEC 程序员为此目的而创建的模块。由此，目标值会被周期性地输出；这就意味着，在每个 IEC 任务中会在每次循环中都计算一次目标值（如位置，速度，加速度等），并将目标值从驱动界面传输给驱动器。去“教会”驱动（例如只给定一个最终位置，驱动将会主动地运动并报告它的到达）是不可能的。原因是当一个指令被执行时，中央控制器对于驱动是没有任何影响的。因此，它不太可能实现例如对多轴的协调动作控制。

▪ **电子凸轮编辑器**是一个独立的 **CoDeSys** 插件，其可以被集成到程序界面内。电子凸轮被用来控制多个驱动轴。在该编辑器内，一个电子凸轮可以通过图形或表格的方式来实现。至此，一个包含对象描述的全局数据结构实例将会被隐含创建，并且被发送至该应用，在此处，它可以通过一个对应的 POU 来访问。

▪ **CNC 编辑器**是一个独立的 **CoDeSys** 插件。它支持多维运动的编程，其可以通过与驱动-硬件相连接的驱动界面进行调度和控制。该编辑器工作类似于 CNC 语言 DIN66025，是由一个图形编辑器和一个与之同步的文本编辑器构成。基本上，其可以实现 9 维运动，但这里只能实现 2 维的非线性插补。因此，在 2 维系统内，直线、圆弧、平行线、椭圆及样条曲线都可以被编程，而另外维度方向上则只能实现线性插补。对于每个已经设计好了的轨迹，**CoDeSys** 都会自动地为其创建一个数据结构，其可以在 IEC 编程时使用。

▪ **库 SM3_Basic.library** 是一个基本元素，其必须被包含在 **SoftMotion** 应用中。由以下元素组成：

- PLCopen 功能块依照 PLCopen 标准，可以简单地实现一个单轴运动的控制，也可以使得两个轴的同步运动。除了库元素状态检测、参数化及一般操作外，还有些功能块用来设定轴相关的速度定义及加速度参数等。如果两个轴需要同步，一个轴做为主轴并按照一定的规则控制第二个轴（从轴）。这个规则既可以是一个使用相关 POU 的电子凸轮，使从轴连接到主轴。此外，有些功能块支持电子齿轮功能和相位移动功能。
- 几个特有的运动控制功能块。
- 驱动界面基本驱动器功能块 (AXIS_REF, AXIS_REF_VIRTUAL, AXIS_REF_MAPPING)
- 通用功能：例如文件复位或错误报告
- 实现特殊功能驱动器的功能块：基本功能块可以被更多的特殊驱动器扩展，其可以与一个特定的驱动类型进行通讯。

▪ **库 SM3_CNC.library** 基于 SM3_Basic.library。除了一些功能块用于运动学的转换外，它还提供所有 POU 所必需的创建、执行以及可视化的 CNC 运动。

▪ **驱动界面**是 SM3_Basic.library 库的一部分，其负责 IEC 程序与驱动之间的通讯。它还为扩展基本驱动功能块到特殊驱动器类型提供了可能性。这些特殊驱动功能块（以 AXIS_REF_*命名）被特意地组合为一个分离的库，该库可以做为相关设备的设备描述参考。还有，驱动界面包括总线特定的库，其用来处理 I/O 映射。

可移植性：除了这些直接复位硬件元件的驱动器外，所有的 **SoftMotion** 实时组件可以用 IEC61131-3 编程。因此，其有很好的平台可移植性。

CoDeSys SoftMotion 驱动界面

CoDeSys SoftMotion 驱动界面是一个标准化的界面，允许在 IEC 程序内包括、配置以及定位一个驱动硬件的抽象影像。在一个通用界面内的各种硬件映射不仅可以简单地进行驱动更改及 IEC 程序复用，还可以使得 PLC 和各种驱动间或在驱动设备间进行通讯。这个界面

在应用中（AXIS_REF）是连接驱动与其表达式的界面，对于驱动控制来说其负责更新及传递必须的运动数据。

驱动界面是由下面几项组成：：

- **SoftMotion** 设备的设备描述在设备树中进行表示。
- 设备描述所参考的库其依照特殊设备类型的需要，来扩展或重载基本的 AXIS_REF 功能块。
- 包括功能块的库，其用于一个非循环数据的读/写，还用于对现场总线栈的基本函数进行包装。

基本库导入到库管理器也可以通过 **SoftMotion** PLC 提供一个 **SoftMotion** 一般驱动池的应用来实现。接着可以有两种方式添加驱动设备：

▪作为自由驱动设备的配置

在目标树中，这些驱动设备不能被确定到另一个设备中，但是可以收集在每个 SoftMotion PLC 所配备的 SoftMotion 一般驱动池中。

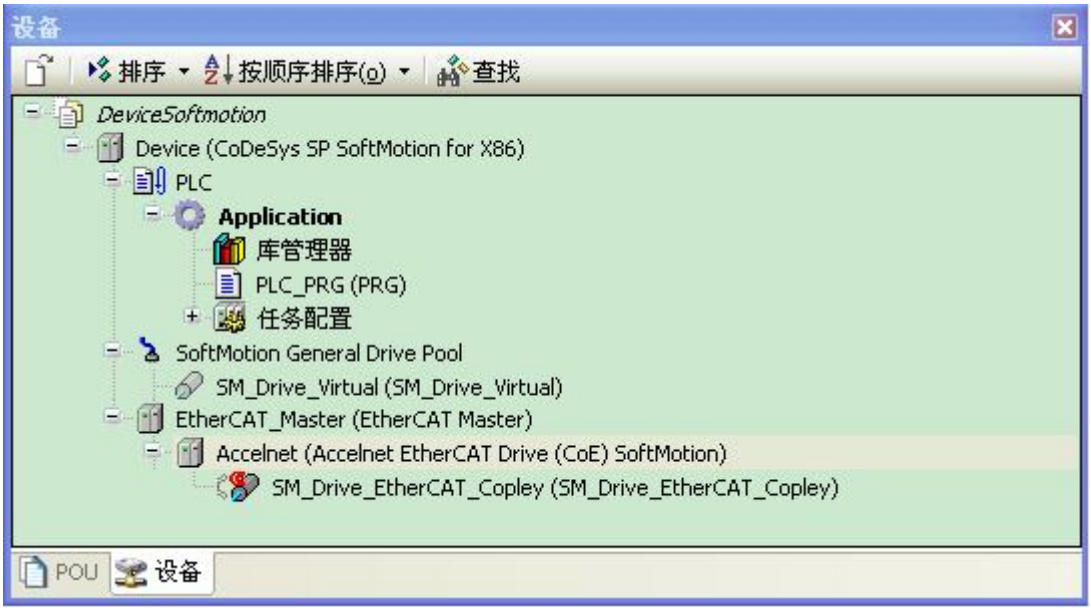
▪例如虚拟驱动 是自由驱动设备，因此它们不能被连接到一个别的现场总线设备；还有位置控制的设备 可以做为自由设备，所以对于该现场总线设备给它们提供输入/输出并不是必须的。

自由驱动设备的典型配置

▪ 已连接的驱动设备配置

这些驱动设备在现场总线拓扑图内，采用另外一种设备表示驱动的方式代替一对一方式表示的相互关联关系。与这些驱动设备相关联的图标由一些小符号来扩充，这些符号可以表示其相连接现场总线类型。例如，任何标准的伺服驱动设备可以通过一个现场总线连接到控制器。

已连接驱动设备的典型配置



电子凸轮（CAM）编辑器

在 **CoDeSys** 开发环境中集成了 **SoftMotion** 电子凸轮盘。在电子凸轮编辑器中可以通过图形或列表的方式实现电子凸轮盘(或者凸轮开关功能)。当根据相关的应用程序产生代码时，将创建各种全局数据结构 (CAM 数据)，这些数据结构能够被 IEC 程序访问。因此在添加 **SoftMotion** 驱动时 **SM3_Basic** 库将会被自动包含。

本章分如下四部分：

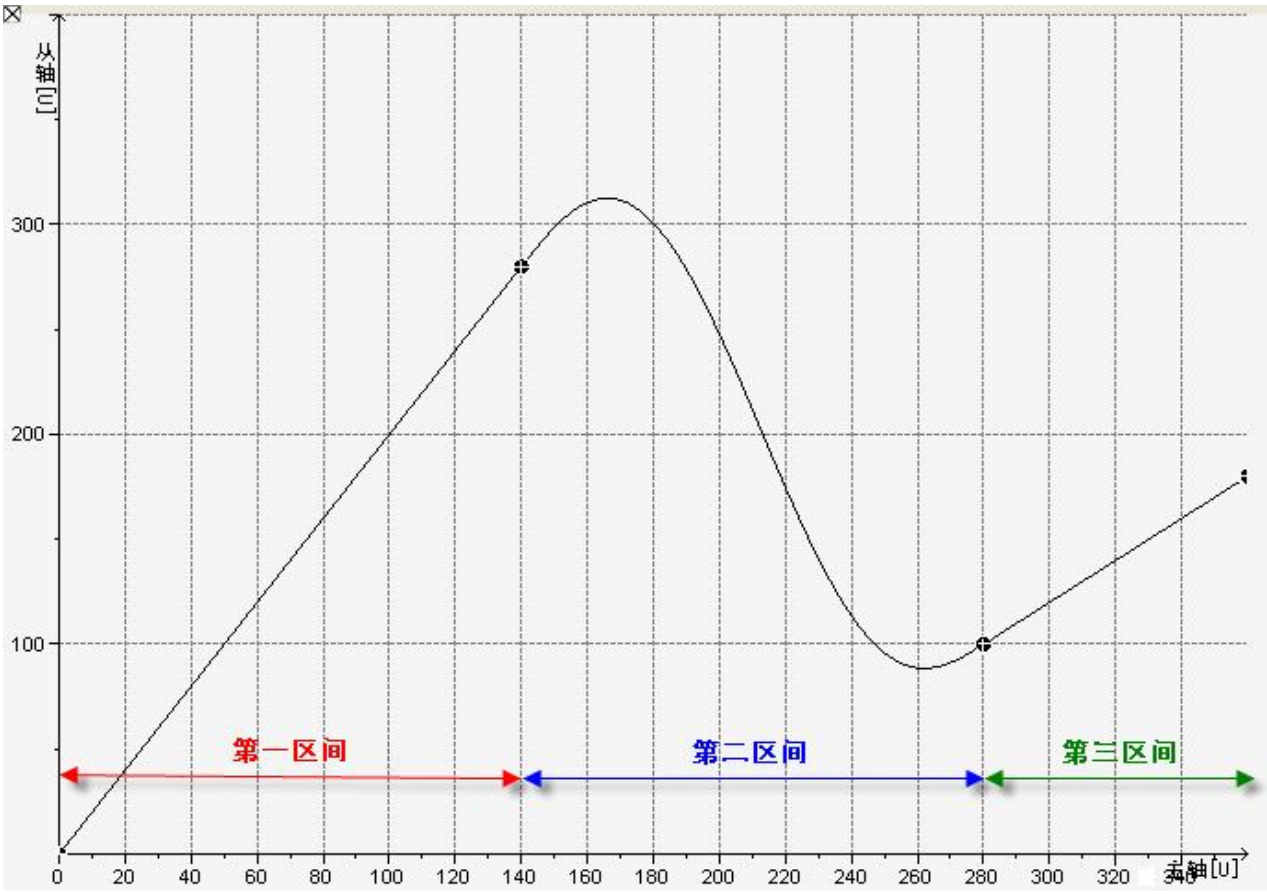
- **SoftMotion** 电子凸轮盘的定义
- 创建电子凸轮盘
- 编辑电子凸轮
- 编译电子凸轮

SoftMotion 电子凸轮盘的定义

电子凸轮盘规定了主从轴参数间的功能依赖关系。这个依赖关系可以通过主从轴值之间的数值映射关系反应出来。更准确的讲，对主轴细分若干个区间，在每个区间中，从轴与主轴的映射关系，由直线或者五次多项式曲线（5 阶可导）构成。

示例：电子凸轮盘图形

一个电子凸轮盘图形：横轴代表主轴，纵轴代表从轴。



例如，主轴的数值范围从 0 到 360。将这个范围被划分为三个区间：

1.
   - 左边的第一个区间： [0, 140]
   - 中间的第二个区间： [140, 280]
   - 右边的第三个区间： [280, 360]

以主轴位置为参考，是从轴的运动与主轴的位置构成该映射关系，其一阶导数即为从轴相对于主轴运动的速度，其二阶导数即为从轴相对于主轴运动的加速度。

采用上述物理分析和解释，可以明显的得出映射必须连续的结论，即从图形上看不能有突变。特别是，两个相邻区间的图形需要光滑连接，并且其上面的每一个点都不能突变。此外，通常一阶导数及二阶导数也需要连续。（事实上，该举例中，要满足三重连续性要求，就决定了在两段直线之间的区间，其起点和终点间必须是唯一的插入一条五阶可导的多项式曲线。）

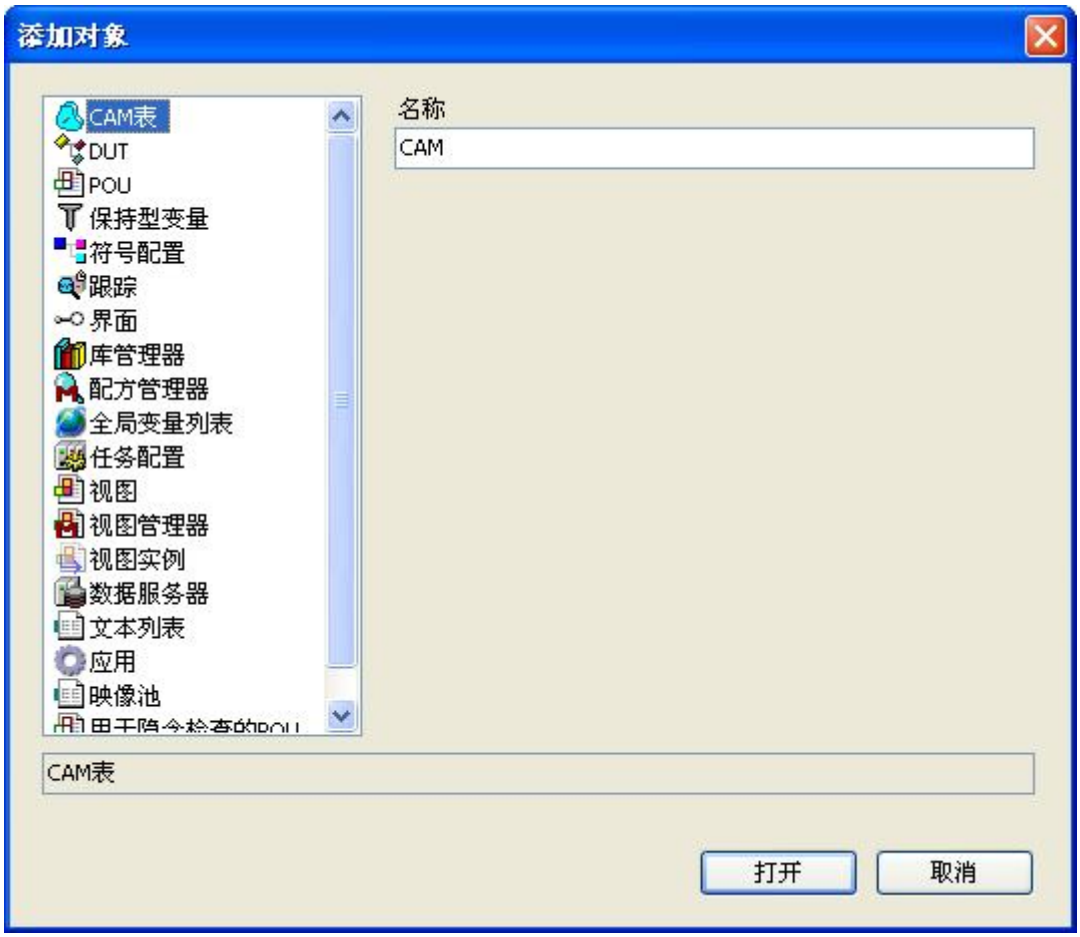
另外，如果电子凸轮盘定义为周期变化，映射值和从轴终点的一阶及二阶导数必须符合其起点的相应值。（在终点的映射值必须符合在起点值的倍数）。

此外，凸轮挺杆,即二进制开关，可以被添加到电子凸轮盘的任意位置。特别是，可以创建一种只实现开关控制的**电子凸轮表**，方法是将整个主轴的行程内对应的从轴位置值设为零，而表中仅含有凸轮挺杆（开关）信息。

创建电子凸轮盘

电子凸轮盘作为一个对象可以被添加到工程或应用程序中。因此选择工程（可以在设备窗口中，也可以在 POU 窗口中）或应用程序，激活菜单中的“添加对象……”菜单项，将会弹出“添加对象”对话框。

添加对象对话框



在窗口中点击“电子凸轮表”选项，输入名称（默认为“CAM”），点击“打开”确定。电子凸轮盘被插入到设备树（或 POU 树）中，并且电子凸轮编辑器会打开这个新创建并采用默认设置方式的电子凸轮盘。

编辑电子凸轮

电子凸轮编辑器由四部分组成，其中前两部分在创建电子凸轮时会被自动打开:

- **工具箱**中提供了各种选择及插入点工具。
- **主编辑器**的标题为电子凸轮的路径名。（例如：CAM [PLCWinNT:Plc Logic:Application]）。该窗口中包含 4 个属性页，通过图形或表格方式对电子凸轮进行设置。

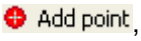
- 在**成员属性视图**  上点击一下后打开成员特定属性编辑器。

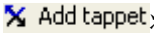
- 对象属性编辑器中包含电子凸轮的一般信息。其可以在设备窗口中，通过点击电子凸轮对象菜单内容中的“**属性…**”打开，也可以通过类别视图中的“**属性…** () 命令打开。

工具箱

根据主编辑器中实际上所选的属性页，工具箱中包含如下各种电子凸轮编辑器工具:

- **选择工具**总是有效的。通过点击  ,将其激活，在主编辑器中点击鼠标选择点或采用电子凸轮图形描述的映射段，将描述内容记录在表格中。通过图形表示法所选的点或映射段将会加粗显示，被选中的表项为窄蓝色边框，此外其所在的行呈现浅蓝色。t

- 只有在属性页的标题为“CAM”时，**添加点工具**才有效。通过点击  ，将其激活，在主编辑器中的速度或加速度映射图的某个位置上点击鼠标，一个新的点将会被插入到该位置。注意，加加速度图形是无法编辑的。

- 只有在属性页的标题为“Tappets”时，**添加凸轮挺杆**工具才有效。通过点击  将其激活，在主编辑器中轨迹的某个位置上点击鼠标一个新的凸轮挺杆将会被插入到其相应位置上。

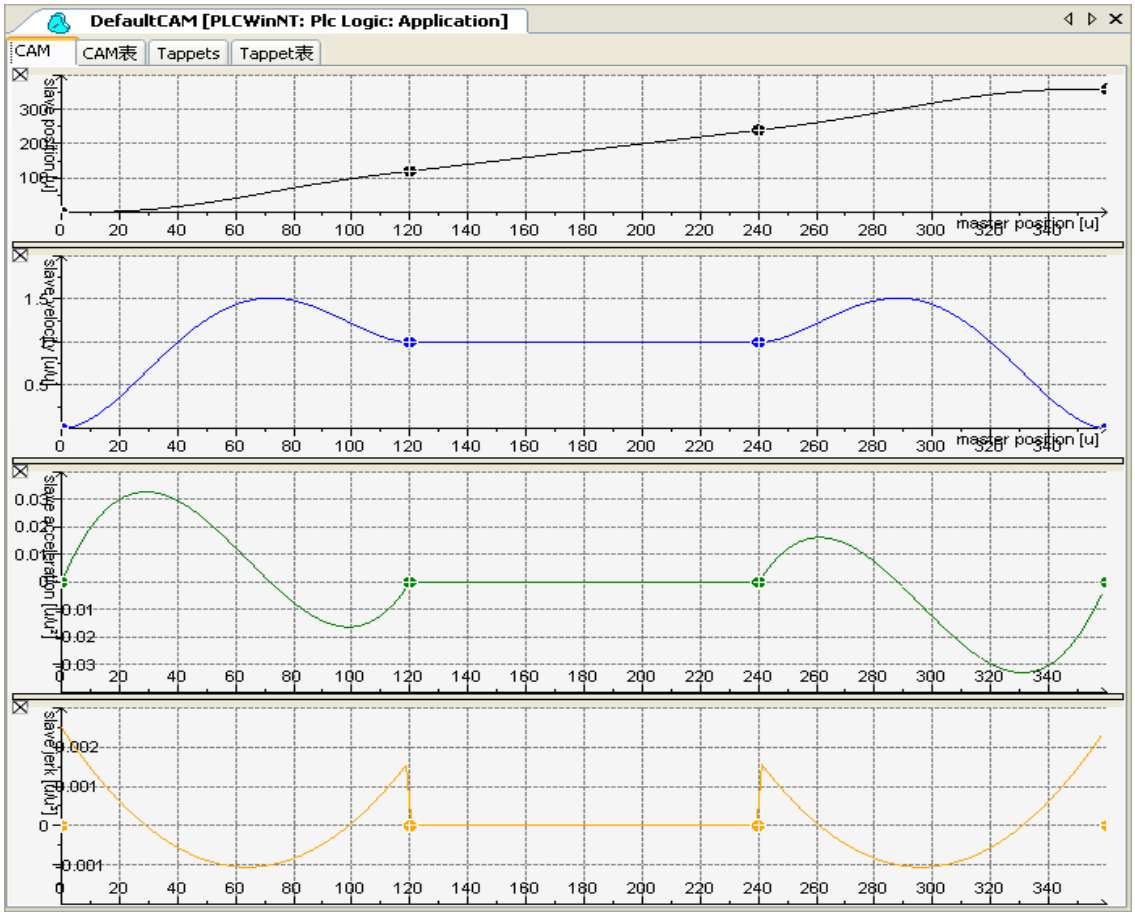
关于主编辑器中各种工具更详细的用法描述如下：

在主编辑器中编辑

在主编辑器中包含**四个不同的属性页**用来显示映射图及一阶导数图或者采用**图形**或**表格**帮助方式添加凸轮挺杆。图形编辑器可以代替表格编辑器进行修正，反之亦然。根据选择不同的属性页来确定相应的编辑模式：

- **标题为“CAM”的属性页：**用**图形编辑器**定义一个电子凸轮；该属性页从上到下被分成的四个子窗口中显示的图形依次为： 从轴位置图（黑色），从轴速度图（黄色），从轴加速度图（绿色），从轴加加速度图（黄色）。

▶主编辑器：CAM



通过按下鼠标左键不释放调整子窗口的高度来调整横坐标。也可以通过点击子窗口左上角的 ☒ 来将其关闭。然后，被关闭的子窗口的位置上将显示一行字符串（**位置坐标、速度、加速度**或**加加速度**）。点击相应的字符串将会再次打开对应的子窗口。

在每个子窗口中，坐标系的横轴对应着**主轴的范围**（默认为[0,360]）。在这个位置关系图中，纵轴的幅值将根据预先在属性编辑器 中设定的数值范围来确定,但是，纵轴的幅值也会根据对应图表中速度、加速度或加加速度的数值范围来自动进行调整。

除了加加速度图外，其它图形都是可编辑的。正如速度、加速度以及加加速度都是由上级函数求导而得出的，所以对位置、速度、加速度任意一个进行修正，都会影响到另外三个图形的显示。

一个新建的电子凸轮，缺省条件下会用四个点定义出三个区间，这三个区间为[0,120], [120,240]和[240,360]。**缺省的映射**关系为指定的从轴位置值与主轴位置值相等，速度、加速度及加加速度值为 0，此外第二和第三点的速度被指定为 1。此时，在各区间的映射关系为五次多项式的函数类型。

可以通过下面的途径改变默认的映射方式：

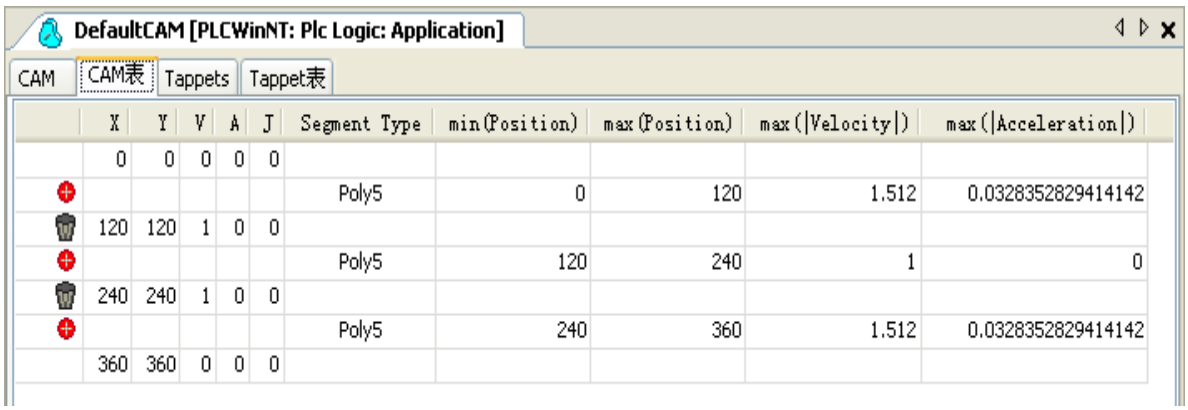
> 从工具箱中选择添加点工具插入**一个远点**。在上述三个图形的任何一个的坐标系中通过单击鼠标左键指定其到适当的位置。在包含新点的区间中，依据新点将主轴分成两部分，并在这两个部分分别创建两个新的映射段（用五次多项式类型），并会调整其起点及终点的值。

使用工具箱中的选择工具在对应的图形上选择一个点来**更新其位置、速度或加速度，**在此过程中保持按下鼠标左键且不释放。另外，也可以使用成员特殊属性编辑器 实现上述操作。

可以通过删除键将选中的**点删除**（当然被删除的点不能是主轴的起点或终点）。

▪ **标题为"CAM 表"的属性页：** 采用**表格编辑器**定义一个电子凸轮：前面的五列包含相关点的属性（即横坐标 **X** 和纵坐标的位置 **Y**，速度 **V**，加速度 **A** 和加加速度 **J**）。接下来为映射段的五个属性（区间类型，从轴位置**极值**（**最小值,最大值**），以及在相关区间内的**速度**和**加速度**的最大值）。保留表格中第一行的值作为主轴的**起始位置**（以及相关的从轴数值），最后一行作为终点位置。奇数行（最少一行）以交替的顺序确定映射段或点。

主编辑器：电子凸轮表



	X	Y	V	A	J	Segment Type	min(Position)	max(Position)	max(|Velocity|)	max(|Acceleration|)
	0	0	0	0	0					
⊕						Poly5	0	120	1.512	0.0328352829414142
🗑	120	120	1	0	0					
⊕						Poly5	120	240	1	0
🗑	240	240	1	0	0					
⊕						Poly5	240	360	1.512	0.0328352829414142
	360	360	0	0	0					

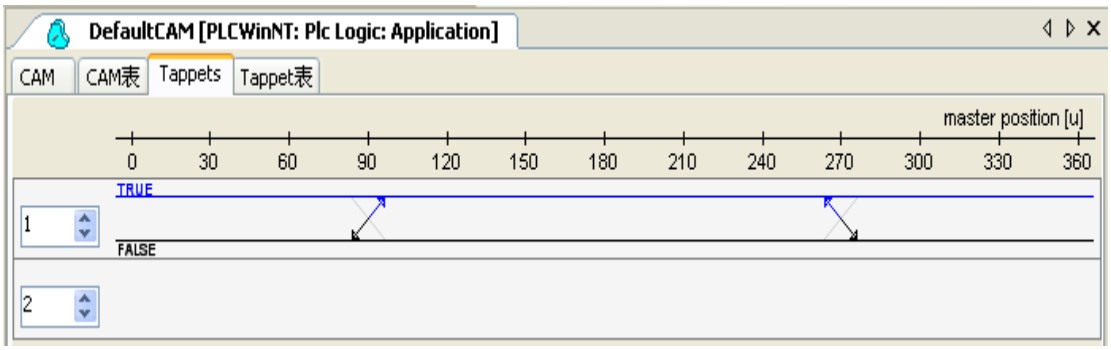
在一个已经存在的区间中，通过点击映射段属性行前的⊕ 符号**添加一个点**。在此之后将插入两行新内容，第一行内容是相应区间的中点的入口值，第二行内容是下一步映射段的属性。无论整个区间添加前的段类型如何，添加后的两部分均被默认设置为五次多项式类型。

通过点击鼠标来**更新选中的表项**。注意，不能编辑最后 4 列中设置的相关映射段的最大值。选中后，通过键盘输出入，或再次点击进入表项，从下拉列表中选择适当的内容来修改所选表项的段类型。必须注意曲线连续性的要求。否则错误的修改将不会被应用，或者被自动根据其他表中数据修正。

通过点击欲删除的点所在行前面的🗑 符号可以**删除该点**（只要该点不是主轴的起点或终点）。该行及随后的一行将会被从表格中移除。区间中左边的点将会被移除，其它部分相应的将会与区间中右边的点相接合。无论之前两个单独的段是何类型，新形成的映射段为五次多项式类型。

▪ **标题为"凸轮挺杆"的属性页：** **在图形编辑器**中定义凸轮挺杆轨迹。凸轮挺杆轨迹中定义了一个或多个具有相同制动开关的由主轴位置坐标决定的凸轮挺杆（即凸轮挺杆事件）。横轴上反应了主轴的行程。可以通过调节右侧箭头 增加或减少 ID 号来控制接下来轨迹的走势（默认情况下有两条轨迹，第一条轨迹已经由两个凸轮挺杆的行为所决定）。

主编辑器：凸轮挺杆



从工具箱中选择**添加凸轮挺杆**工具来添加一个新的凸轮挺杆。可以在需要的轨迹处单击鼠标左键来指定一个适当的位置，在此处将插入一个新的凸轮挺杆，并且此时选择工具将会代替添加凸轮挺杆工具自动生效。通过点击其中心选中一个凸轮挺杆后，可以通过鼠标拖动来移动其位置（在移动的过程中保持鼠标处于按下状态）或者通过**删除键**将其删除。

正向（主轴位置值增加的方向）或负向经过凸轮挺杆标记时，可以赋予不同的事件（“打开开关”,“关闭开关”,“转换开关状态”,“无操作”）。请看列表中的各种组合情况。一个轨迹中各凸轮挺杆的各种属性，即相同的 ID 号，指向同一**挺杆位**，其为布尔型变量。默认情况下，电子凸轮在正转情况下具有“打开开关”的属性，在反转  情况下具有“关闭开关”属性。因此该布尔变量在正转时的值被设置为 TRUE，当反转时被设置为 FALSE。“转换开关状态”事件将会使该布尔变量的值取反，“无操作”则对布尔变量的值无影响。通过单击可以使  中的交叉线打开或关闭，用来改变对**应凸轮的各种属性**。

凸轮挺杆属性组合表

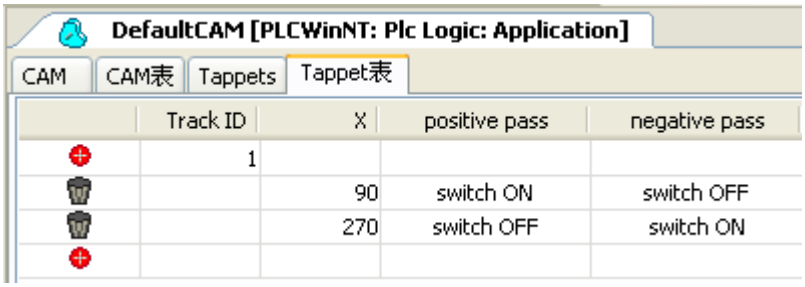
凸轮挺杆符号	正转	反转
	无操作	无操作
	打开开关	无操作
	关闭开关	无操作
	无操作	打开开关
	无操作	关闭开关
	打开开关	关闭开关
	打开开关	关闭开关

	关闭开关	打开开关
	关闭开关	关闭开关
	转换开关状态	无操作
	无操作	转换开关状态
	打开开关	转换开关状态
	转换开关状态	打开开关
	转换开关状态	关闭开关
	关闭开关	转换开关状态
	转换开关状态	转换开关状态

1.

- **标题为“凸轮挺杆表”的属性页：** 在**表格编辑器**中可以定义凸轮挺杆轨迹。在显示出轨迹 ID 号的行（行前有 ⊕符号）下面，该轨迹有多少凸轮挺杆就有多少行显示。 符号之前的这些行中每一行都代表一个凸轮挺杆事件。下表中的各列内容均是有效的：轨迹 ID 号（凸轮挺杆轨迹 ID 号）；X：主轴的相应位置坐标；正转/ 反转：从正向或反向通过主轴位置坐标的事件定义。

主编辑器：凸轮挺杆表



一个轨迹中的各凸轮挺杆的属性即相同的 ID 号，涉及到相同的从动件,其为布尔型变量。默认情况下，在正转情况下电子凸轮的属性为“打开开关”， 在反转 情况下电子凸轮的属性为"关闭开关",因此在正转情况下布尔型变量的值设置为 TRUE,在反转情况下设置为 FALSE。“转换开关状态”事件对该布尔型变量的值取反，但“无操作”事件对布尔型变量的值没有影响。

通过点击轨迹 ID 号前面的⊕ 符号，可以向一个已经存在的轨迹上添加一个**凸轮挺杆**。在此，一个新的行将会被插入到紧跟主轴位置 0 之后，其内容为默认属性。

单击选中表项可以**修改其内容**。然后，可以输入一个新的轨迹 ID 号或者主轴的位置坐标，或者通过进一步点击相应表项从下拉列表中的正转/反转中选择属性。

通过点击表中相应行前面的  符号**删除凸轮挺**杆。

点击表格中的最后一行中的 符号**添加一个新的轨迹**。在此，  将会插入两行新的内容，第一行包含一个尚未分配的轨迹号，第二行提供主轴位置 0 上默认的凸轮挺杆及默认的属性。一旦与轨迹对应的所有的凸轮挺杆全被移除，轨迹本身将会被移除。

在成员属性编辑器中进行编辑

在图形编辑器中，当前选中的电子凸轮或凸轮挺杆的**成员属性**，可以通过紧邻工具箱窗口的“属性”窗口查看并编辑，其每个默认值均有效。

在电子凸轮对象属性编辑器中进行编辑

在设备树中，当前被选中的电子凸轮对象，其基本设定可以在对象属性对话框中进行定义。

编译电子凸轮

在编译的过程中将创建 **MC_CAM_REF** 类型的结构体变量。这些结构体变量中包含的每个映射部分都是用来描述该电子凸轮。关于该类数据结构被用来作为 MC_CamIn 模块的输入参数。该结构是 **SM3 Basic.library** 库的一部分。

使用电子凸轮-常用提示

SM3_Basic.library 库提供合适的模块来处理电子凸轮，可以通过库管理器的**“添加库…”**菜单命令将该库插入到工程中。否则，也会在 **SoftMotion** 驱动添加到设备树中时，自动插入。

下面的内容将会对具体参数（例如周期、偏移等）的含义作一些深入的解释，并说明将一个电子凸轮叠加到另一个电子凸轮上的可能性：

- 模块参数的作用
- 两个电子凸轮间的切换

模块参数的作用

电子凸轮属性（电子凸轮属性 对话框）：

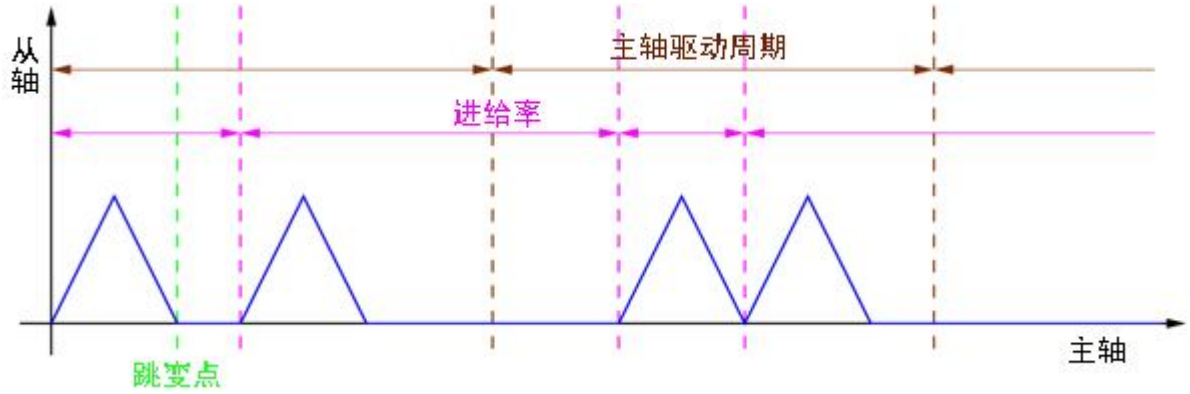
- **周期变化，周期**

  一个周期性电子凸轮可以从一个周期到下一个周期重复运行而不出现“跳变”现象。从轴的位置与主轴位置相关，当选项“周期重复”前面的选择框被选定☑ ，从轴相对于主轴终止位置的位置值与从轴相对于主轴开始的位置值是成倍数关系，但是，从轴的速度和加速度的值还等于他们在起始处的值。周期或进给率用从轴幅值的单位来测算。即使“周期重复”选项没有被选中，相关的电子凸轮也可以连续的工作。在这种情况下，用户必须确保在转换处的动作与预想的一致。

MC_CAMTableSelect 模块：

- **周期**

这个参数决定了主轴到终止位置后电子凸轮是否要再次进行。下面是参数设为 TURE 时的情形说明。下图显示了周期间的明显差别：这个例子，你可以把它想象成一个传送带传送一些相同物料的应用场景。主轴的值从 0 到所有物料对象的长度。在这个范围内，一个物料对象传送过来，其左后方会一个个的都传送过来，用一个工具（例如，用来给物料打印标记）其位置是由从轴来驱动的（蓝色的图）。



当然，一个电子凸轮主轴的范围（也就是所定义的主轴的整个行程值）并不等于主轴驱动的周期距离（即在例子中，皮带传送一个物料对象的距离）。因此，等式“SlavePosition = CAM（MasterPosition）”，指的是在电子凸轮关系中，从轴位置的定义依赖于主轴的位置，并且只在电子凸轮第一个循环是有效的。当有新的物料对象到来时，电子凸轮驱动设备开始一个新的循环。由于皮带上的物料对象之间有间距（并且间距距离可能还不同），以及生产速度的影响，也就是说在连续的两个电子凸轮之间的间隔时间不同，因此每个循环的电子凸轮主轴范围并不一致。如果参数设置为 FASLE，主轴到达终止位置时，模块 CamIn 的输出变量 EndOfProfile 将置为 TRUE，并且从轴将固定停止在它的当前实际位置。注意，即使主轴超出定义的范围，电子凸轮的动作也不会结束。当主轴重新进入有效范围时，从轴仍会根据电子凸轮进行控制。

- **主轴绝对位置**

如果 MasterAbsolute 输入为 TRUE，电子凸轮将在当前主轴位置处开始，该点也可能位于电子凸轮过程的中间部分。但是如果该点在电子凸轮指定的范围之外，将会弹出一个错误信息。如果输入为 FALSE，电子凸轮运动将在当前位置处开始，也就是说，电子凸轮会以当前主轴所在的位置做为电子凸轮主轴运动所设定的零点。电子凸轮的主轴范围如果不包括“0”，就不能在该模式下使用，否则将会弹出一个错误信息（“…主轴超出指定范围…”）。

- **从轴绝对位置**

参数 CamTableSelect.SlaveAbsolute 影响从轴的启动模式（StartMode）（启动模式主要由参数 CamIn.StartMode 控制）。

MC_CamIn 模块

▪ **主轴偏移，主轴幅值**

输入量 MasterOffset 和 MasterScaling 按照下面公式变换主轴位置，并且电子凸轮会用变换后的位置 X 进行计算：

X = MasterScaling*MasterPosition + MasterOffset

因此，如果 MasterScaling 的值大于 1，电子凸轮将在较高速度下运行；如果值小于 1，电子凸轮将在较低速度运行。

▪ **从轴偏移，从轴幅值**

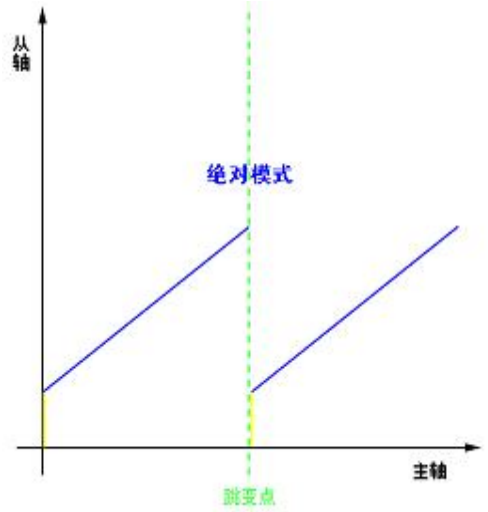
SlaveOffset 参数会使电子凸轮纵向（从轴方向）移动，SlaveScaling 参数则会在从轴方向拉伸电子凸轮。根据下述公式说明可知，第一步电子凸轮先进行拉伸动作，然后再进行移动：

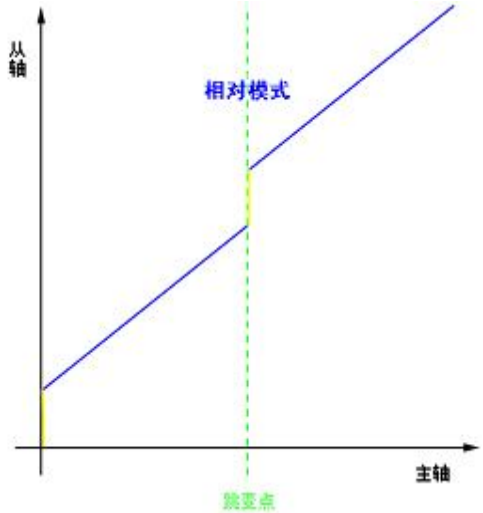
Y = SlaveScaling*CAM(X) + SlaveOffset

A SlaveScaling >1 ，拉伸电子凸轮，从轴运动范围增大；同理，SlaveScaling<1，缩小电子凸轮从轴运动范围。

▪ **启动模式：绝对/相对/斜坡输入/正向斜坡输入/反向斜坡输入**

下面的图片展示是在 “绝对”与“相对”模式下对一个典型电子凸轮的不同作用。该电子凸轮定义在主轴范围为：从左侧纵坐标开始到右侧绿色垂直虚线之间的范围。因为我们关心的是周期运动的电子凸轮，因此在主轴位置超出范围后，它会继续重复运行。





绝对模式：在新的电子凸轮循环开始时，电子凸轮的计算与当前从轴的位置无关。如果从轴相对于主轴起始位置不同于从轴相对于主轴的终止位置，其将造成一个跳变。

相对模式：新的电子凸轮会根据当前从轴的位置进行改变；也就是说，从轴在上一个电子凸轮循环结束的位置，会被现在的电子凸轮运动作为“从轴偏移”进行位置相加的计算。但是，如果在电子凸轮定义中，与主轴起始位置对应的从轴位置不是 0，其将造成一个跳变。

斜坡输入：通过增加一个补偿运动（根据极限值 VelocityDiff、加速度、减速度得出的运动）来防止电子凸轮在开始时的潜在跳变。因此，只要从轴是旋转的方式，正向**斜坡输入**选项则只进行正向补偿，而**反向斜坡输入**则只进行反向补偿。对于线性运动的从轴，补偿方向可以自动实现，也就是说，正向**斜坡输入**和**反向斜坡输入**可以被用**斜坡输入**的方式进行解释。).

电子凸轮间的切换

基本上，两个电子凸轮在任何时候都可以切换，但是需要考虑一些情况：

在电子凸轮编辑器中，从轴的位置定义为电子凸轮函数的计算输出，电子凸轮函数是以在主轴范围内的一个主轴位置为计算条件，由此可以用下述简单的公式来进行表达说明：

SlavePosition = CAM(MasterPosition)

因为主轴驱动的实际周期一般与电子凸轮定义的主轴范围是不同的，所以为满足电子凸轮函数的正确输入，主轴位置必须被按比例调整到函数定义域内：

SlavePosition = CAM(MasterScale*MasterPosition + MasterOffset)

类似的情况，如果一个电子凸轮在绝对值模式下启动，产生了一个向上的跳变，函数输出（也就是虚拟从轴位置）也会按比例被修正：SlavePosition = SlaveScale*CAM(MasterPosition) + SlaveOffset

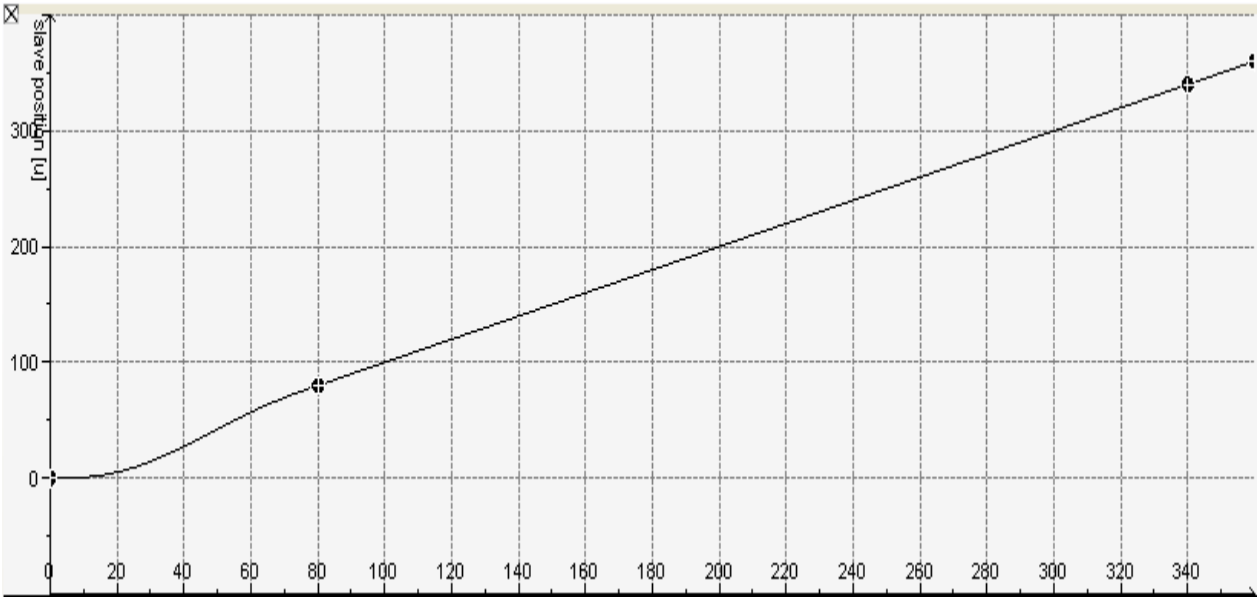
最坏情况下，这两种比例修正都必须被采用，所以，事实上从轴位置（SlavePosition）是由更为复杂的公式计算得出的：Slaveposition = SlaveScale*CAM(MasterScale*Masterposition + MasterOffset) + SlaveOffset

在每个电子凸轮周期结束时，比例和偏移可以改变以获得更为合适的参数。

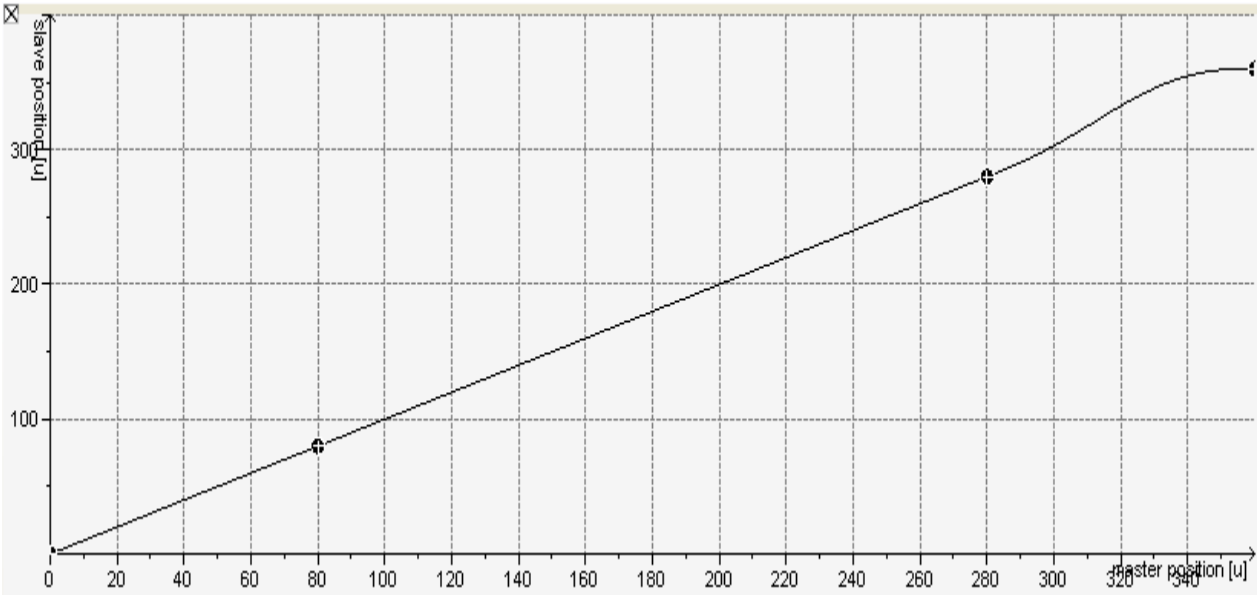
遗憾的是，电子凸轮的 MC_CamIn 模块的重新启动，将会删除它的内存并且包括比例值和偏移值。因此，所定义的电子凸轮函数会适应于一般各种不同的从轴数值。基于这个原因，建议只有在需要处理另外一个不同的电子凸轮时，才去重启 MC_CamIn-FB。

看下面从 CAM1 切换到 CAM2 的例子：

CAM1 是在两条直线之后而形成的五次多项式曲线构成。

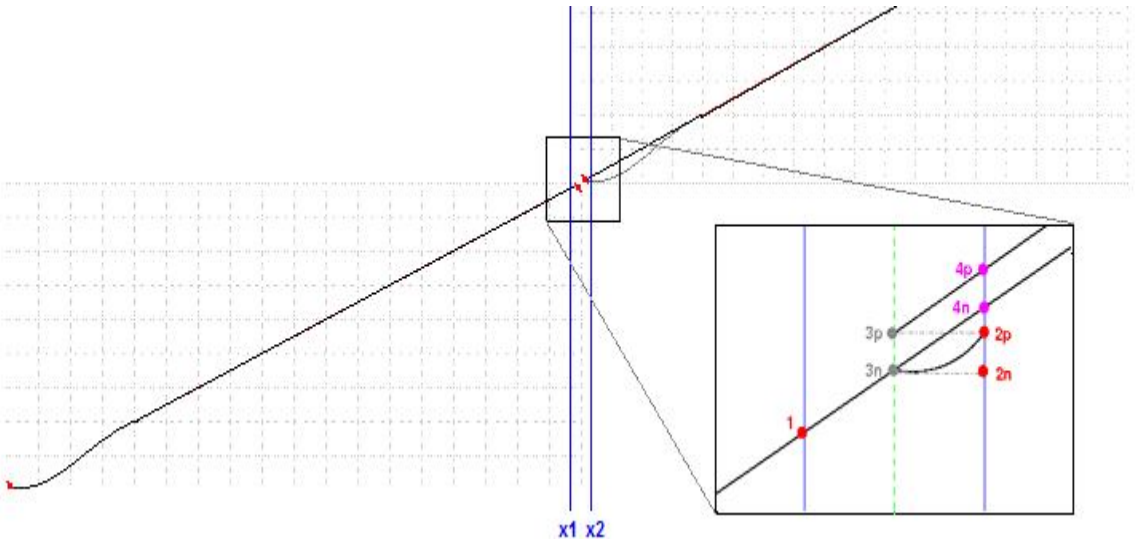


CAM2 是在五次多项式曲线后的两条直线组成。



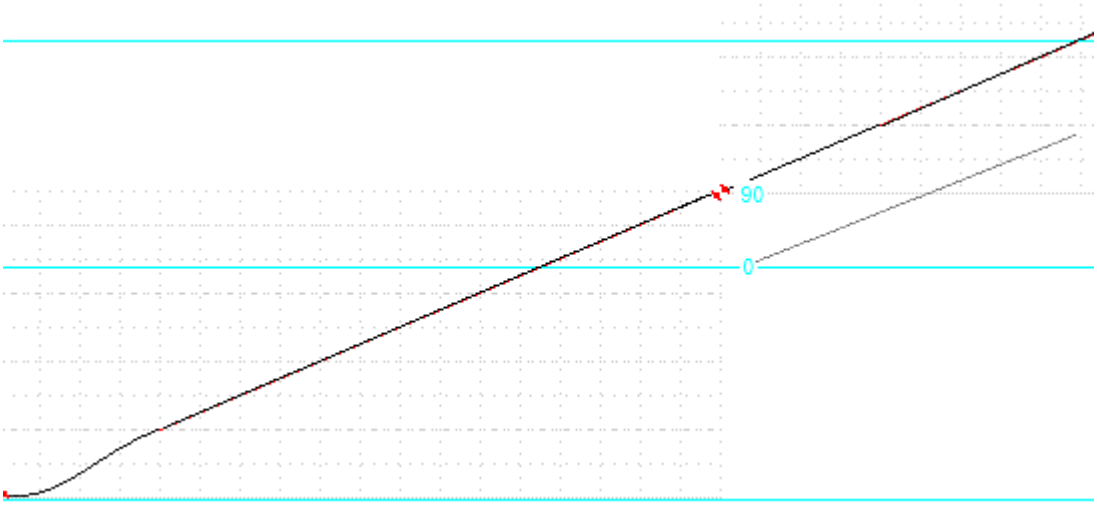
在两个电子凸轮间进行切换需要考虑下面的问题：

1. 为了避免跳变，第一个电子凸轮在结束点处的速度和加速度应该与第二个电子凸轮在开始点处的值对应相匹配。就现在这个例子来看，需要满足的条件为：CAM1 的结束点和 CAM2 的开始点，其速度都应为 1，加速度都应为 0。
2. 如果第二个电子凸轮从轴的位置值在开始时被指定为 0，那么应该是在相对模式下启动的。但是第一个电子凸轮必须是在非周期模式下运行。否则，如果第一个电子凸轮为周期性的，则其在完成运行第一个周期时，从轴的位置就不再是唯一确定的：



该图框的右侧展示了由 CAM1 变换到 CAM2 的细节。蓝线表示电子凸轮函数在主轴位置 x1 和 x2 上的计算值。

3. 在绝对模式下启动电子凸轮时需要注意，从轴要有合适的起始位置。不管电子凸轮是以周期模式还是非周期模式启动，如果主轴范围等于从轴的周期，切换起来就没有那么复杂了。 如果两个周期（主轴位置和从轴位置的周期）与 CAM2 主轴值域相一致，等于 360°，上述例子中的 CAM2 就可以在绝对模式下启动。否则，如果假如从轴周期是 270°（淡蓝色线条所示），绝对模式选项就不可以，除非进行一些调整。



在 CAM1 与 CAM2 转换时，从轴的位置为 90°的情况下。CAM2 用绝对模式启动，会在 0°造成一个跳变，用灰色线表示。当然，可以通过设定从轴偏移到合适的 90°值时，就可以恰当地处理这种情况。

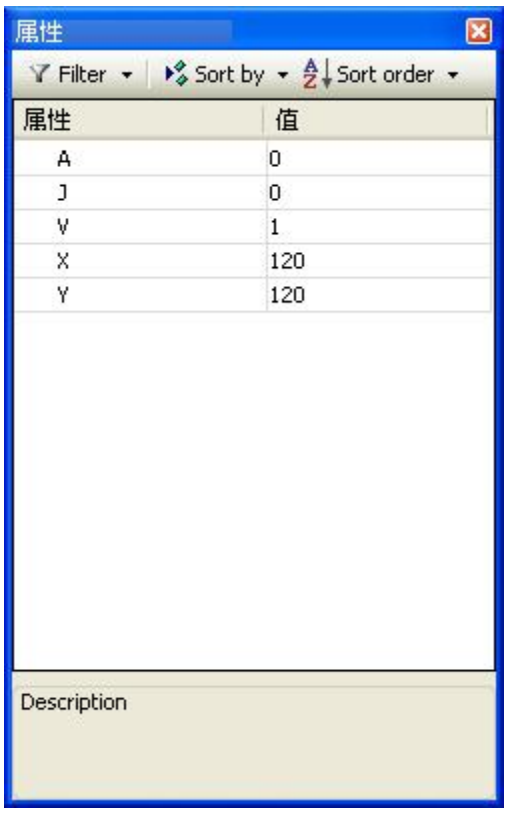
CAM：成员属性

电子凸轮编辑器中使用到的成员属性可以通过“属性”编辑窗体进行查看和修改。

在电子凸轮编辑器中，点击**成员属性标志**  **将**会出现**属性**窗体。如果当前没有成员（点，映射段，挺杆，……）被选中，那么编辑器的工作区为空，同时在窗体下部的“描述”区域中无任何内容。

另外，如果在电子凸轮编辑器的绘图选项卡（标题为“CAM”或“Tappets”）中点击一个成员，那么工作区将以表格的方式列出与被选成员相关的所有属性。

示例：显示被选点的属性

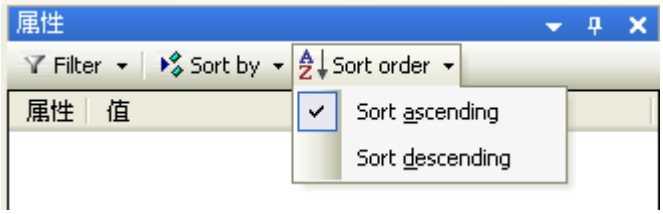


属性编辑器的顶端是一个小工具条。第一项是“过滤”命令。由于目前没有过滤功能，所以与选中成员相关的所有属性都显示出来。第二项是“排序”命令，可以通过选择“按类型”或者“按名字”来进行排序。

工具条： 按. . .排 序

第三项是关于升序和降序的“排序”命令。

工具条：排序



某个表项被**选中**时，该项会有一个窄蓝边框，并且它所在的行都会呈现浅蓝色。同时，相关属性的说明信息也会在窗体的 “描述”区域显示。

根据相关属性，被选表项的**值**可以通过键入适当值进行**修改**；或者通过单击表格单元（table cell），从出现的选择列表中进行选取。在取消选定表格单元后，修改将生效；即单击另外一个表项或空白处，都可使修改生效。

CAM 数据结构

编辑的电子凸轮数据在工程编译过程中自动转换为一张全局变量（CAM 数据）列表。其中，每个 CAM 由 MC_CAM_REF 类型的结构体变量描述，并且 IEC 程序、CAM 预处理函数和功能块都可以访问那些结构体变量。为了使编译过程不发生错误，必须在 IEC 程序中添加 SM3_Basic.library 库，同时定义适当的结构体变量。

此外，还将创建数据结构_SMC_CAM_LIST（指向 MC_CAM_REF 的指针数组）和一个指向所有可用电子凸轮的数据结构。数据结构_SMC_CAM_LIST 通过指针引用特定的电子凸轮。

当然，在运行时（run time）IEC 程序分别创建和填充相应的数据结构。因此，下面将给出所有用到的数据结构的详细描述信息。

数据结构简要介绍

编辑的电子凸轮数据 在工程编译过程中自动转换为一张全局变量列表。其中，每个 **CAM** 由 **MC_CAM_REF** 类型的结构体变量进行描述，并且 IEC 程序、CAM 预处理函数和功能块都可以访问那些结构体变量。

此外，还将创建数据结构_SMC_CAM_LIST（指向 MC_CAM_REF 的指针数组）和一个指向所有可用电子凸轮的数据结构。数据结构_SMC_CAM_LIST 通过指针引用特定的电子凸轮。

SM3_Basic.library 库提供了一些适当的数据类型。在设备树中添加 **SoftMotion** 驱动时，该库会自动包含到库管理器中。

当然，在运行时 IEC 程序会创建描述电子凸轮的变量，或者为其赋值。因此，下面将给出使用到的所有数据结构的具体描述：