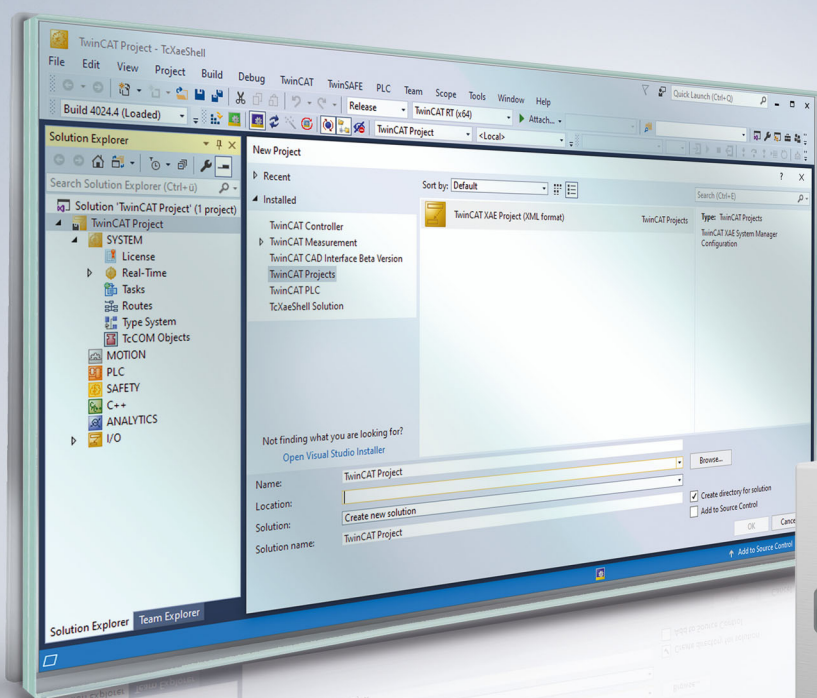


BECKHOFF New Automation Technology

手册 | ZH

TF5430

TwinCAT 3 | Planar Motion



目录

1 前言	5
1.1 文档说明	5
1.2 安全信息	5
1.3 信息安全说明	6
2 新功能概览	7
3 简介	8
4 状态和模式	9
4.1 Planar 对象状态图	9
4.2 Planar 动子命令器示意图	10
4.3 Planar 轨道运行模式	11
5 Parts	13
5.1 部件和坐标系	13
5.2 配置	14
5.3 具有ReferenceId的位置	15
6 Planar Motion 组件	16
6.1 Planar 动子	16
6.1.1 配置.....	16
6.1.2 创建 PLC.....	19
6.1.3 例如“创建和移动 Planar 动子”	19
6.1.4 示例“将 Planar 动子移动到 Planar 部件”	22
6.1.5 示例“创建并移动带辅助轴的 Planar 动子”	26
6.1.6 示例“利用外部设定点生成功能创建并移动 Planar 动子”	29
6.1.7 示例“将带有外部设定点生成功能的 Planar 动子移动到 Planar 部件”	31
6.1.8 示例“在 CRotationFreeMovement 模式下移动 Planar 动子”	36
6.1.9 运动命令的限值和选项.....	38
6.2 Planar 轨道	39
6.2.1 配置.....	39
6.2.2 轨道网络和防碰撞系统.....	42
6.2.3 几何定义.....	45
6.2.4 轨道和部件.....	46
6.2.5 示例“Planar 动子加入并在轨道上移动”	48
6.2.6 示例“在多 Planar 部件的轨道上移动 Planar 动子”	52
6.2.7 示例“将 Planar 动子耦合到轨道上,并在 CRotationOnTrack 模式下移动”	57
6.2.8 示例“将 Planar 动子耦合到轨道,并使用 AdoptTrackOrientation 移动它”	61
6.2.9 示例“Planar 动子在轨道上与外部轴同步”	64
6.2.10 示例:“使轨道上的 Planar 动子与另一个 Planar 动子同步”	70
6.2.11 示例“将 Planar 轨道与网络互联互通”	73
6.2.12 示例“将 Planar 轨道与 Planar 部件上的网络互联互通”	77
6.2.13 示例:“跟随 Planar 动子穿过 Track Network”	82
6.2.14 StartFromTrackAdvanced”和”EndAtTrackAdvanced”命令的选项.....	86
6.3 Planar 组	87
6.3.1 配置.....	87
6.3.2 示例“创建和移动具有防撞组功能的 Planar 动子”	89

6.4	Planar 环境	93
6.4.1	配置	94
6.4.2	示例“配置定子区域和边界”	96
6.5	示例：“使用轨道和组创建并移动 Planar 动子”	99
6.6	Planar 部件	103
6.6.1	示例“激活 Planar 部件位置并移动 Planar 动子”	104
6.7	Planar Feedback	110
6.7.1	示例“创建 Planar 动子和 Planar Feedback”	110
6.7.2	示例“Planar 运动组件：避免碰撞”	113
6.7.3	专门的反馈类型	117
6.8	Planar TrackTrail	118
6.8.1	示例“2 个 Planar 轨道上的同步运动”	118
7	PLC 功能库	123
7.1	插入功能库	123
7.2	Tc3_Mc3PlanarMotion API	123
7.2.1	Data Types	123
7.2.2	Function Blocks	133
8	技术支持和服务	174

1 前言

1.1 文档说明

本说明仅适用于熟悉国家标准且经过培训的控制和自动化工程专家。
在安装和调试组件时，必须遵循文档和以下说明及解释。
操作人员应具备相关资质，并始终使用最新的生效文档。

相关负责人员必须确保所述产品的应用或使用符合所有安全要求，包括所有相关法律、法规、准则和标准。

免责声明

本文档经过精心准备。然而，所述产品正在不断开发中。
我们保留随时修改和更改本文档的权利，恕不另行通知。
不得依据本文档中的数据、图表和说明对已供货产品的修改提出赔偿。

商标

Beckhoff®、TwinCAT®、TwinCAT/BSD®、TC/BSD®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、Safety over EtherCAT®、TwinSAFE®、XFC®、XTS® 和 XPlanar® 是德国倍福自动化有限公司的注册商标并由其授权使用。
本出版物中所使用的其它名称可能是商标名称，任何第三方出于其自身目的使用它们可能会侵犯商标所有者的权利。

专利权

EtherCAT 技术已申请并获得多项专利，包括但不限于：
EP1590927、EP1789857、EP1456722、EP2137893、DE102015105702
，并在其他多个国家进行了类似的专利申请或注册。

EtherCAT®

EtherCAT® 是德国倍福自动化有限公司的注册商标和专利技术。

版权所有

© 德国倍福自动化有限公司。
未经明确授权，不得复制、分发、使用和传播本文档内容。
违者将被追究赔偿责任。德国倍福自动化有限公司保留所有发明、实用新型和外观设计专利权。

第三方品牌

本文档中使用了第三方商标和文字商标。有关商标许可信息，可以访问：<https://www.beckhoff.com/trademarks>

1.2 安全信息

安全规范

为了确保您的使用安全，请务必仔细阅读
并遵守本文档中每个产品的安全使用说明。

责任免除

所有组件在供货时都配有适合应用的特定硬件和软件配置。严禁未按文档所述修改硬件或软件配置，否则，德国倍福自动化有限公司对由此产生的后果不承担责任。

人员资格

本说明仅供熟悉适用国家标准的控制、自动化和驱动工程专家使用。

警示性词语

文档中使用的警示信号词分类如下。为避免人身伤害和财产损失，请阅读并遵守安全和警告注意事项。

人身伤害警告

⚠ 危险

存在死亡或重伤的高度风险。

⚠ 警告

存在死亡或重伤的中度风险。

⚠ 谨慎

存在可能导致中度或轻度伤害的低度风险。

财产或环境损害警告

注意

可能会损坏环境、设备或数据。

操作产品的信息



这些信息包括：
有关产品的操作、帮助或进一步信息的建议。

1.3 信息安全说明

Beckhoff Automation GmbH & Co.KG (简称 Beckhoff) 的产品，只要可以在线访问，都配备了安全功能，支持工厂、系统、机器和网络的安全运行。尽管配备了安全功能，但为了保护相应的工厂、系统、机器和网络免受网络威胁，必须建立、实施和不断更新整个操作安全概念。Beckhoff 所销售的产品只是整个安全概念的一部分。客户有责任防止第三方未经授权访问其设备、系统、机器和网络。它们只有在采取了适当的保护措施的情况下，方可与公司网络或互联网连接。

此外，还应遵守 Beckhoff 关于采取适当保护措施的建议。关于信息安全和工业安全的更多信息，请访问本公司网站 <https://www.beckhoff.com/secguide>。

Beckhoff 的产品和解决方案持续进行改进。这也适用于安全功能。鉴于持续进行改进，Beckhoff 明确建议始终保持产品的最新状态，并在产品更新可用后马上进行安装。使用过时的或不支持的产品版本可能会增加网络威胁的风险。

如需了解 Beckhoff 产品信息安全的信息，请订阅 <https://www.beckhoff.com/secinfo> 上的 RSS 源。

2 新功能概览

版本 V3.3.39 及以上:

- 新功能: 支持边长为 160 mm 和/或 320 mm 的平面模块
- 新功能: MoveOnTrack 命令的附加选项 (“GapMode”)
- 新功能: PLC 中 MC_PlanarMover 的 “MoverHardwareOid” 属性和 “GetTrackTrailInformation” 方法

版本 V3.3.19 及以上:

- 新功能: Planar 动子和 Planar 部件的 Planar 状态机的状态切换 (Enabling、Disabling、Resetting) 超时
- 新功能: Planar 动子的大部分初始参数可在 Enabled 状态下更改 (先前只能在 Disabled 状态下更改)

版本 V3.2.60 及以上:

- 新功能: 动子行走的表面被划分为 Planar 部件, 可在运行时移动。动子可以穿越互联互通或相邻 Planar 部件之间的边界。
- 需要 TwinCAT V3.1.4024.40 或更高版本

版本 V3.1.10.63 及以上:

- 需要 TwinCAT V3.1.4024.24 或更高版本

版本 V3.1.10.51 及以上:

- 新功能: AdoptTrackOrientation 可旋转轨道上的动子, 使其与轨道方向保持一致。这将把 C 坐标模式从独立模式改为从属模式。
- 高级: MoveC 现在始终适用于轨道上静止的动子。这可能会将 C 坐标模式从从属模式改为独立模式。

版本 V3.1.10.44 及以上:

- 新功能: GearInPosOnTrack 和 GearInPosOnTrackWithMasterMover 命令, 分别用于将 Planar 动子耦合到主轴或主动子上
- 高级: Planar 轨道 TcCOM 模块参数
- 需要 TwinCAT V3.1.4024.17 或更高版本

版本 V3.1.10.30 及以上:

- 新功能: 带有模数定位功能的 CRotation 命令器模式 (360° 旋转)
- 新功能: 约束作为一个新变量可以限制运动命令的最大动态
- 高级: “闭环”轨道上的模数定位参数

版本 V3.1.10.11 及以上:

- 第一版 Planar Motion 发布
- 需要 TwinCAT V3.1.4024.12 或更高版本

3 简介

TwinCAT 3 Planar Motion 软件包 TF5430 在安装软件包 TF5400 时会一起安装。

目标系统

Windows 10 (仅 64-bit)

TwinCAT 3 Planar Motion 基础

TF5430 TwinCAT 3 Planar Motion 软件结合了控制 XPlanar 动子的多种功能组件，能够高效、智能地实现 XPlanar 的个性化应用。TF5430 TwinCAT 3 Planar Motion 是 TF5890 TwinCAT 3 XPlanar 的一部分。所有相关功能块都包含在 Tc3_Mc3PlanarMotion 库中，该库应与 [Tc3_Physics](#) 库结合使用。

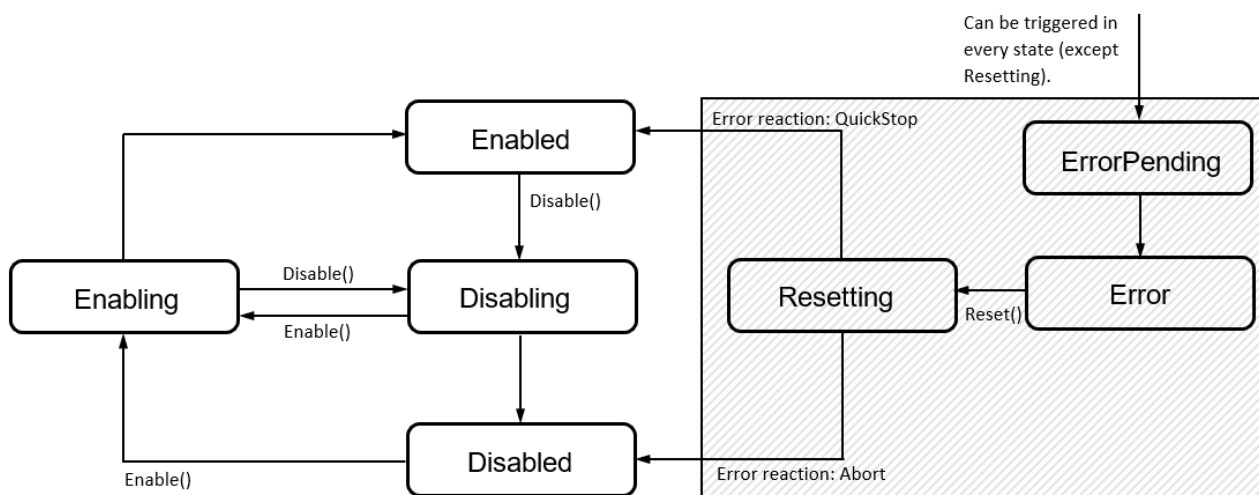
额外的授权要求

TF5430 TwinCAT 3 Planar Motion 需要 TC1250 授权。

4 状态和模式

4.1 Planar 对象状态图

Planar 动子、Planar 轨道和 Planar 组均可使用 Planar 状态机。所有这些组件都可以处于 7 种 Planar 状态：Enabling、Enabled、Disabling、Disabled、Resetting、ErrorPending、Error。



错误反应取决于错误的严重程度。对于轻微错误，正常的错误反应是 QuickStop。在这种情况下，用户也可以通过 Disable() 强制终止错误状态。命令必须在 3 种状态中的 1 种状态下发送：ErrorPending、Error 或 Resetting。

Enabling

在 Enabling 状态下，执行 Enable 命令。该命令结束时，组件处于 Enable 状态。在 Enabling 状态下，可以发送 Disable 命令，取消 Enable 命令并使状态变为 Disabling。

Enabled

在 Enabled 状态下，组件功能齐全，用户可以使用。在这种状态下，可以发送 Disable 命令。然后状态切换为 Disabling。

Disabling

在 Disabling 状态下，将执行 Disable 命令。该命令结束时，组件处于 Disabled 状态。在 Disabling 状态下，可以发送 1 条 Enable 命令，取消 Disable 命令并使状态变为 Enabling。

Disabled

系统启动后，组件处于 Disabled 状态。可以使用 Enable 命令将它们置于 Enabling 状态。组件在 Disabled 状态下无法运行。

Resetting

该组件正在纠正错误。根据错误反应，它将处于 Enabled 或 Disabled 状态。

ErrorPending

当发生错误时，组件会从除 Resetting 状态外的所有其他状态达到 ErrorPending 状态。一旦错误得到正确处理，状态就会切换为 Error。

Error

Error 状态表示发生了错误，此时可使用 Reset 命令将组件置于 Resetting 状态，以便校正错误。

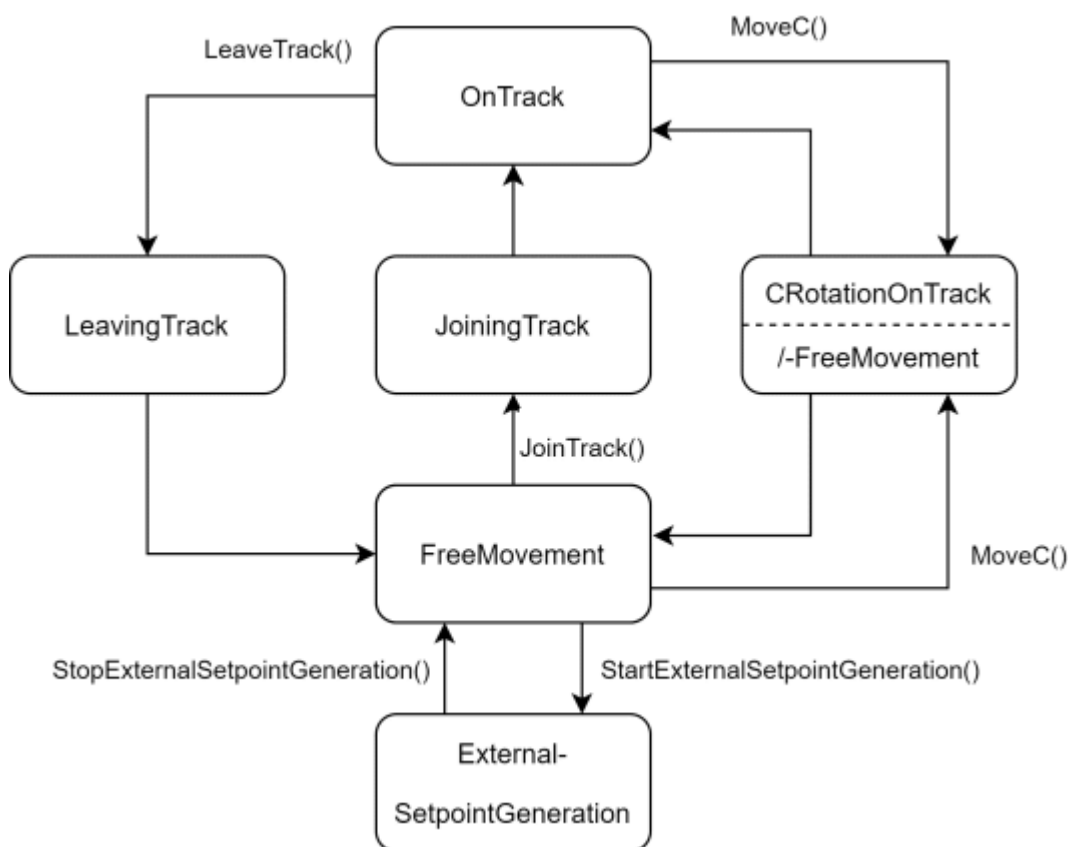
过渡状态超时

版本 V3.3.19 及以上：超时功能可用于 Planar 动子和 Planar 部件。

Enabling、Disabling 和 Resetting 等过渡状态的超时可调。如果此类过渡状态由用户触发，且未在超时时段内退出，那么系统会切换为 ErrorPending 状态，同时取消转换命令（Enable/Disable/Reset）（超时错误）。

4.2 Planar 动子命令器示意图

Planar 动子有六种不同的命令模式，分别表示动子执行的命令类型：OnTrack、LeavingTrack、JoiningTrack、ExternalSetpointGeneration 和 CRotationFreeMovement/-OnTrack（从 V3.1.10.51 版开始）。在除 ExternalSetpointGeneration 模式之外的所有模式中，当动子处于 1 个组中时，防碰撞功能都会激活。



OnTrack

在 OnTrack 模式下，动子加入轨道并可在轨道上移动（MoveOnTrack）。动子还可以再次离开轨道（LeaveTrack），这样模式就变为 LeavingTrack。如果有必要，MoveC 命令器会切换到 CRotationOnTrack 模式。

LeavingTrack

在 LeavingTrack 模式下，动子不会接受任何命令。当动子结束 LeaveTrack 命令后，模式自动退出。此时，动子处于 FreeMovement 模式。

JoiningTrack

在 JoiningTrack 模式下，动子不接受任何其他命令。当动子结束 JoinTrack 命令后，模式会自动退出。此时动子处于 OnTrack 模式。

FreeMovement

启用动子后，会自动进入该命令模式。当动子扭曲过大时，则动子在启动后进入 CRotationFreeMovement 模式。可以通过 MoveToPosition 命令自由移动动子。如果用户通过命令器启动外部设定点生成，则模式切换为 ExternalSetpointGeneration。JoinTrack 命令也可以将模式改为 JoiningTrack。MoveC 命令可能会导致模式改变为 CRotationFreeMovement。

CRotationFreeMovement/-OnTrack

如果正在进行的 C 运动没有完全在 C 位置窗口内进行，则通过 MoveC 命令启动该模式。窗口由动子 C 轴的位置限值定义，存在 4 个窗口，每个窗口旋转 90°（90° 旋转是动子对称性的结果：例如，限值 $\pm 15^\circ$ -> 窗口 1. $[-15^\circ, +15^\circ]$, 2. $[75^\circ, 105^\circ]$, 3. $[165^\circ, 195^\circ]$, 4. $[255^\circ, 285^\circ]$ ）。根据您之前是在 FreeMovement 模式还是 OnTrack 模式，模式将相应变为 CRotationFreeMovement 或 CRotationOnTrack。当 C 运动完成且终点位置位于 4 个窗口之一内时，模式结束。然后，动子就会自动返回到之前的模式。这样，动子就从 CRotationFreeMovement 变成 FreeMovement，或者从 CRotationOnTrack 变成 OnTrack。否则，它将保持 CRotationFreeMovement/-OnTrack 模式。在这 2 种 CRotation 模式下，动子的 X 轴和 Y 轴都不能移动。如果动子在启动时方向已经在 4 个窗口之外，则会立即进入 CRotationFreeMovement 模式，而不是 FreeMovement 模式。

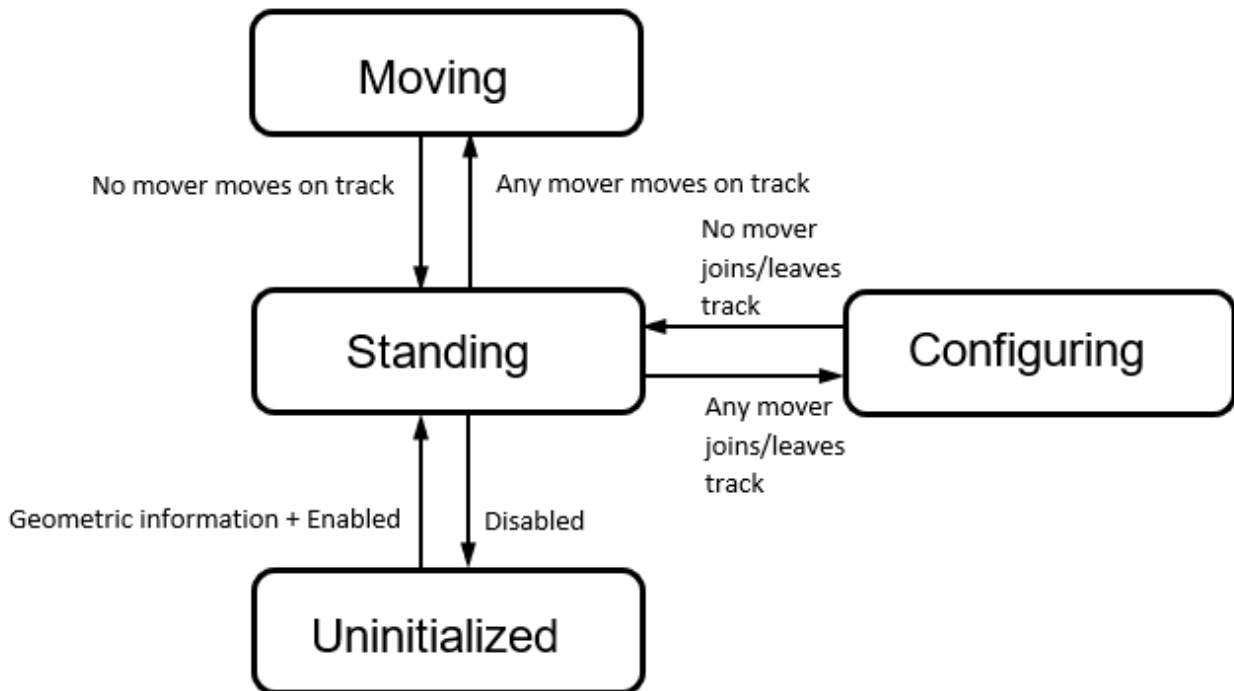
ExternalSetpointGeneration

在 ExternalSetpointGeneration 模式下，动子执行相应的命令。该模式以相应命令器的开始（或结束）为开始（或结束）。在 ExternalSetpointGeneration 模式下，动子按照用户提供的外部设定点循环运行。

ExternalSetpointGeneration 可与其他模式结合使用。在这种情况下，外部设定点只需作为其他模式设定点的相对偏移量添加即可。但此时动子并不处于 ExternalSetpointGeneration 模式。

4.3 Planar 轨道运行模式

Planar 轨道有 4 种不同的运行模式，分别表示轨道是否以及如何执行或能够执行其作为“Street for Movers”的功能：“Street for Movers”：Moving、Standing、Configuring 和 Uninitialized。



Moving

在 Moving 模式下，1 个或多个动子即将在轨道上移动 (MoveOnTrack)。第 1 个以 Standing 模式在轨道上开始运动的动子，会自动将 Standing 模式改为 Moving 模式。相应地，完成移动的最后 1 个动子将模式变回 Standing。轨道处于 Moving 模式时，任何动子都不得执行 JoinTrack 或 LeaveTrack 命令。如果轨道位于 Planar 组中，则会阻挡其表面。

Standing

在 Standing 模式下，轨道可供动子使用。轨道上的所有动子都处于 Standing 状态，等待行驶命令。在此模式下，允许对动子发出 JoinTrack、LeaveTrack 和 MoveOnTrack 命令。每条命令都会结束轨道的 Standing 模式。如果轨道位于 Planar 组中，则不会遮挡其表面。

Configuring

在 Configuring 模式下，1 个或多个动子即将离开轨道 (LeaveTrack) 或加入轨道 (JoinTrack)。第 1 个以 Standing 模式离开 (或加入) 轨道的动子自动将模式从 Standing 改为 Configuring。相应地，最后 1 个完成离开或加入的动子将模式变回 Standing。当轨道处于 Configuring 模式时，任何动子都不得执行 MoveOnTrack 命令。如果轨道位于 Planar 组中，则不会遮挡其表面。

Uninitialized

在 Uninitialized 模式下，动子无法使用轨道。它还没有完整的几何描述。当用户创建并启用该几何描述时，轨道会切换到 Standing 模式。

5 Parts

5.1 部件和坐标系

版本 V3.2.60 及以上：可使用 Part 功能，这也是本节的主题。

XPlanar 动子移动的表面由定子组成。例如，边长为 240 mm 的正方形或边长为 160 或 320 mm 的长方形/正方形。整个基底区域分为 1 个或多个部件，每个部件由 1 个或多个定子组成。1 个定子只能属于 1 个部件，即各部件不能重叠。部件的所有定子必须装配在 1 个连续的表面。

XPlanar 系统的这种几何配置通常是静态的；在系统运行时不会改变。要创建动态配置，用户必须通过分配多个位置来移动或重新定位单独的部件，并在运行时激活它们。

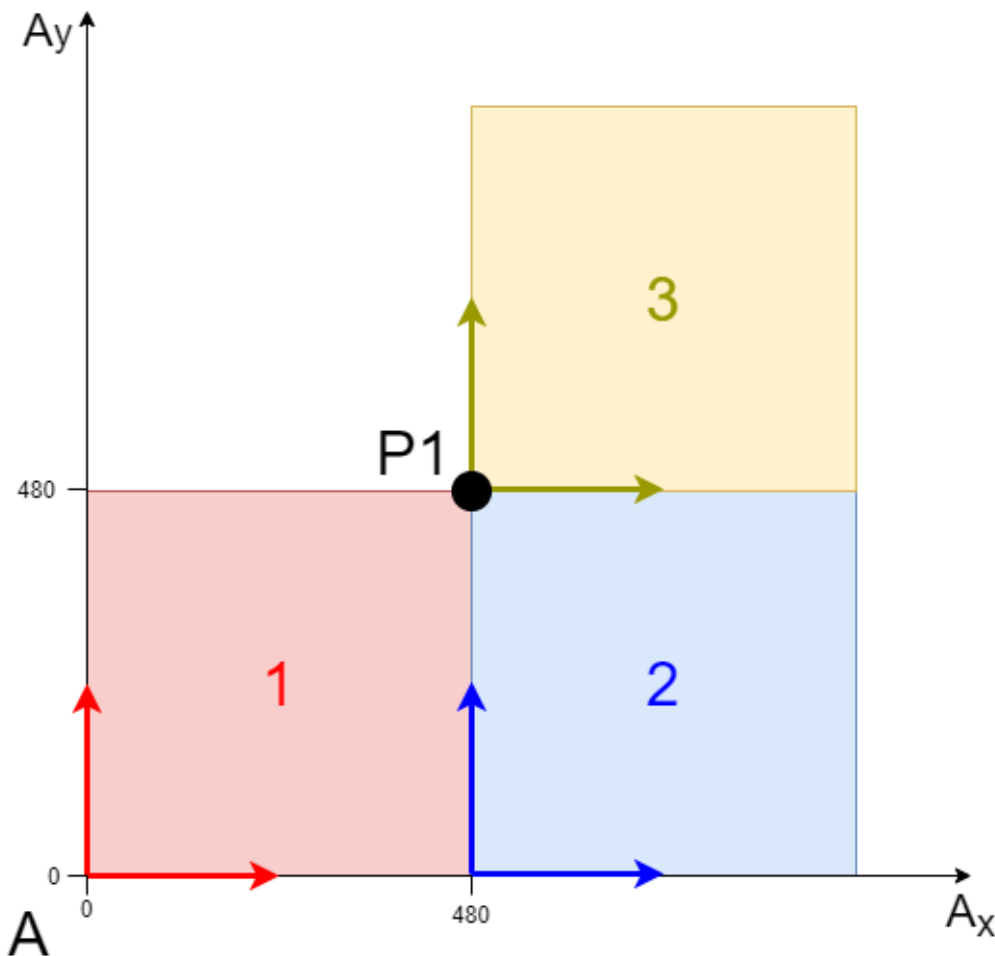
一般来说，会出现这样 1 个问题：位置不再唯一，或者没有绝对坐标系。用户可以定义多个二维（2D）坐标系，并将部件放置在这些坐标系中。



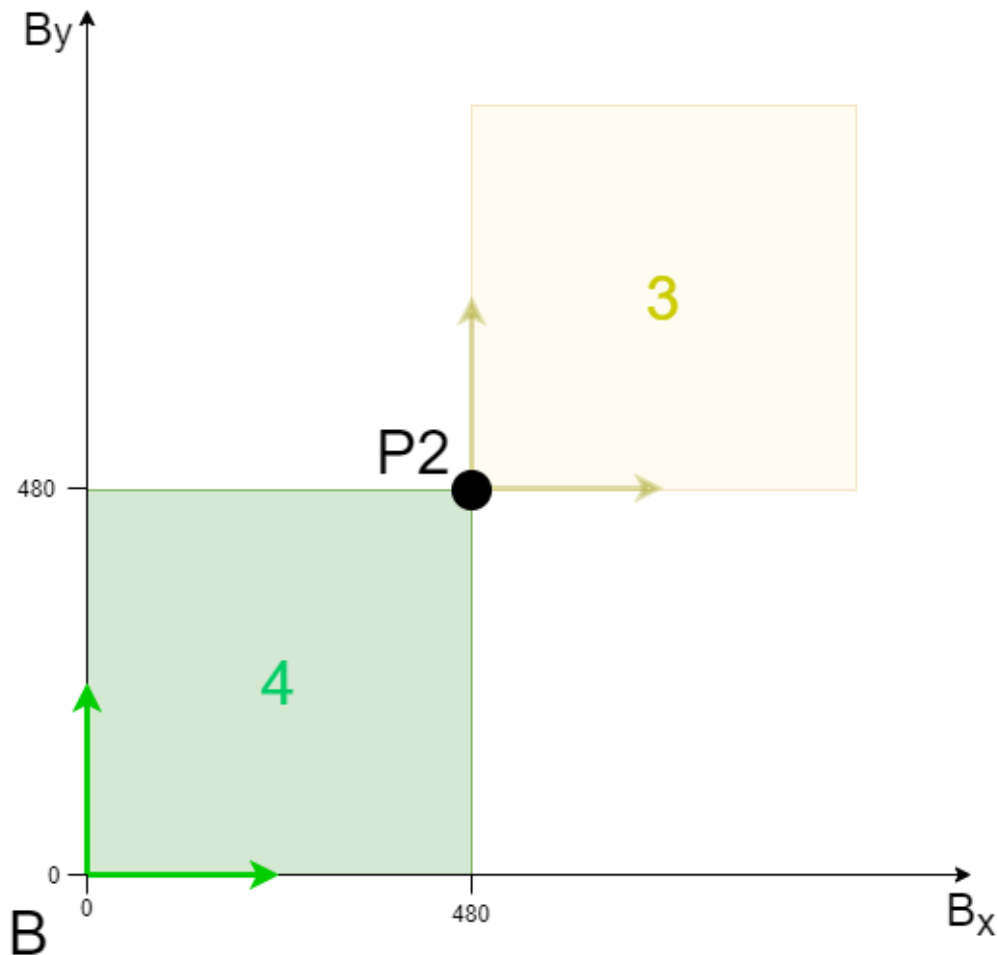
二维定位（例如动子或轨道的定位）如果没有标明其所处的坐标系，现在是不完整的。

示例

共有 4 个部件（1-4）和 2 个坐标系（A 和 B）。部件 3 位于坐标系 A 或 B 中。其他部件则被永久分配到 1 个坐标系中。位置 P1 位于坐标系 A 中的 $X=480$ 和 $y=480$ 处（简称 $P1=(480, 480, A)$ ）。



位置 P2 位于坐标系 B 中的 $X=480$ 和 $y=480$ 处（简称 $P2=(480, 480, B)$ ）。



如果不指定坐标系 A，位置 P1 与坐标系 B (480, 480, B) 中的位置 P2 就无法区分。然而，这 2 个位置是不同的，甚至不在同一个坐标系中，即它们之间没有几何上的互联互通（例如，动子不可能从 P1 移动到 P2）。

除了 A 坐标系和 B 坐标系外，部件 1-4 中的每一个还有局部部件坐标系。这些坐标系的原点都位于各自部件的左下角。这意味着在部件 1-3 的部件坐标系中， $P1=(0, 0, 3)$ 、 $=(0, 480, 2)$ 、 $=(480, 480, 1)$ 。

只有当部件 3 位于坐标系 A 中时才是如此。部件 3 在坐标系 B 中的透明表示表明部件 3 可以从坐标系 A 变换到坐标系 B。部件的位置与其坐标系的原点相同。因此，部件 3 将从位置 P1（坐标系 A 中）移动到位置 P2，现在部件 3 的坐标系中无法再指定 P1。

相反，可以在部件 3（和 4）的坐标系中指定位置 $P2=(0, 0, 3)$ 、 $=(480, 480, 4)$ 。总之，如果部件随其坐标系移动，部件坐标系中的位置就不是静态的。

5.2 配置

从 V3.2.60 版开始：可使用 Part 功能，这也是本节的主题。

在 XPlanar 驱动程序中而不是在 MC Configuration 中配置零件的定子（即几何范围）和零件在不同坐标系中的位置。因此，我们在此不再介绍如何创建此配置。

不过，这些信息对 MC Configuration 和 PLC 控制都很重要，因此在激活 XPlanar 驱动程序时，Planar 环境会从该驱动程序中读取这些信息，并将其分配给 MC Configuration 中的所有 Planar 对象。在 PLC 中，使用单独的 MC PlanarPart [► 158] 对象来控制零件的位置和状态。

扩展了项目对象 PositionXYC 和 PositionXY，以便唯一指定项目。现在，除了坐标值外，它们还包含坐标系的 ID (ReferenceId)。由于位置不再唯一，现在可以通过 2 种方式指定动子的位置：

1. 作为（动子所在的部件所处的）上层坐标系中的位置。
2. 作为部件坐标系中的 1 个位置。上一级坐标系中的位置在循环界面中指定。此外，还可以使用 [GetPositionOnCurrentPart \[► 157\]](#) 方法查询其在零件坐标系中的位置。

根据部件的位置，轨道可以与其他各种轨道互联互通。可以使用 [StartFromTrackAdvanced \[▶_166\]](#) 和 [EndAtTrackAdvanced \[▶_167\]](#) 这 2 种方法进行这些互联互通。生成外部设定点时，还必须指定外部设定点所在的坐标系。这是通过 `SetExternalSetpointReferenceId` 方法实现的。



下文将详细介绍 **MC Configuration** 的这些功能。

5.3 具有ReferenceId的位置

从 V3.2.60 版开始：可使用 `Part` 功能，这也是本节的主题。

随着部件和坐标系的引入，如果不说明位置所在的参考系，位置就不再有意义。为此，已为 `PositionXYC` 和 `PositionXY` 对象添加了“`ReferenceId`”属性。现在还可以保存参考系统的 ID。

现在，`PositionXYC` 和 `PositionXY` 这 2 个位置对象可以（也应该）始终与参照系统的显式 ID 一起使用。说明 1 个具体的参照系统更为安全，因为这样可以明确指出所指的是哪个系统。

不指定参考系或指定“零”参考系只有在特殊情况下才允许，即只有 1 个静态坐标系，在该坐标系中所有部件都有固定位置。然后在内部将 ID “零”转换为唯一坐标系的 ID。在所有其他情况下，参考系“零”都会被拒绝。除“零”以外的无效参照系统（无效部件/坐标系对象 ID）总是被拒绝。

6 Planar Motion 组件

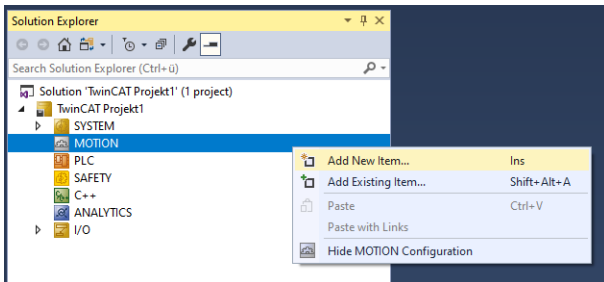
6.1 Planar 动子

Planar 动子是表示 XPlanar 动子的软件对象。它为用户总结了实际动子的状态（位置、速度等）。此外，用户还可以通过 Planar 动子来影响或控制真实动子的状态。

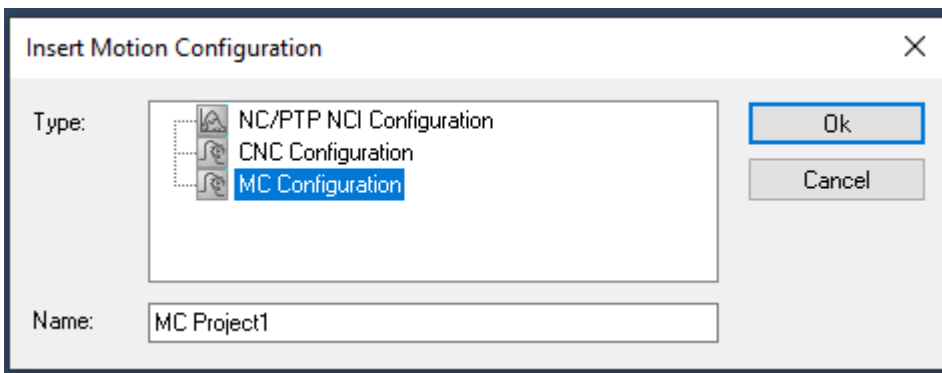
6.1.1 配置

✓ 要创建 Planar 动子，必须先创建 MC Configuration。

1. 选择 MOTION > Add New Item...

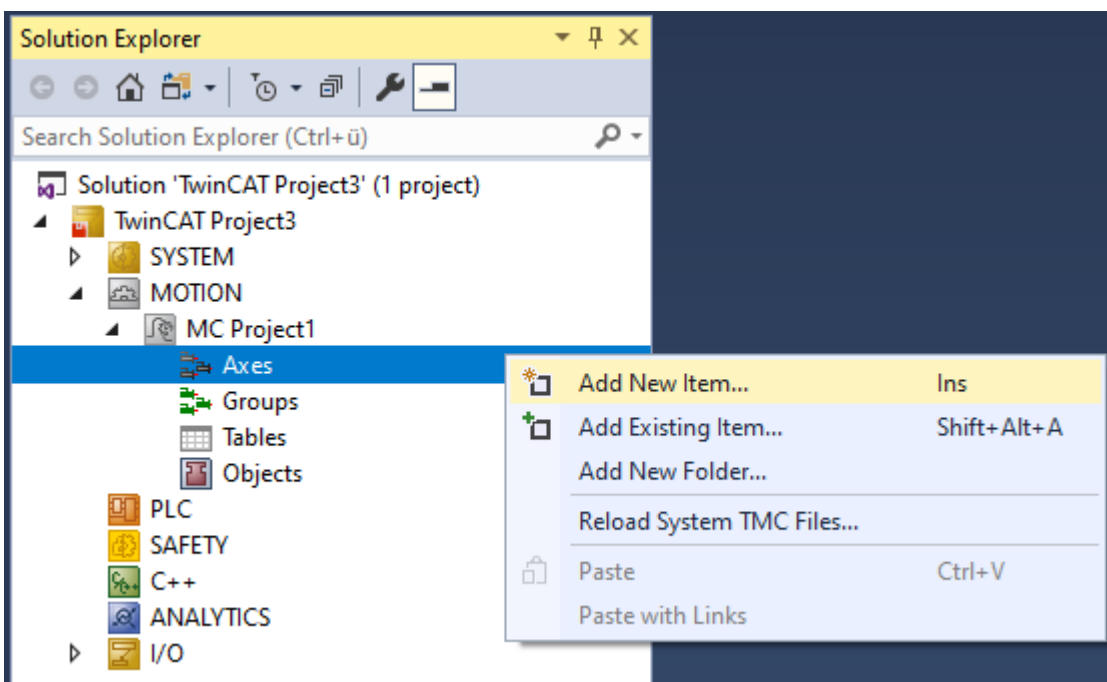


2. 在以下对话框中，选择 MC Configuration 并按 OK 确认。

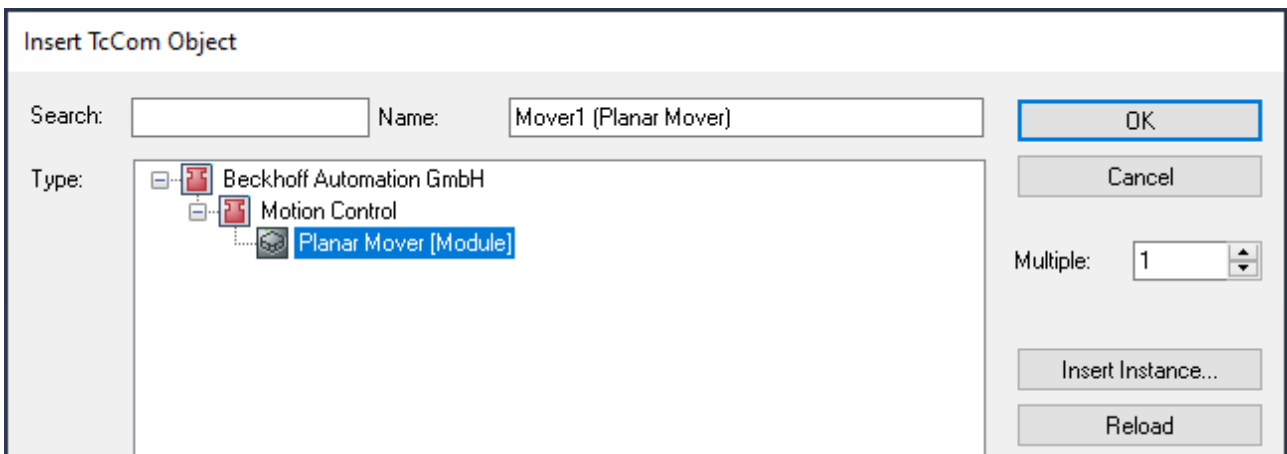


⇒ 您创建了 1 个 MC Project。

3. 选择 MC Project > Axes > Add New Item...



4. 在以下对话框中，创建 1 个（或多个）Planar 动子，并按 OK 确认。



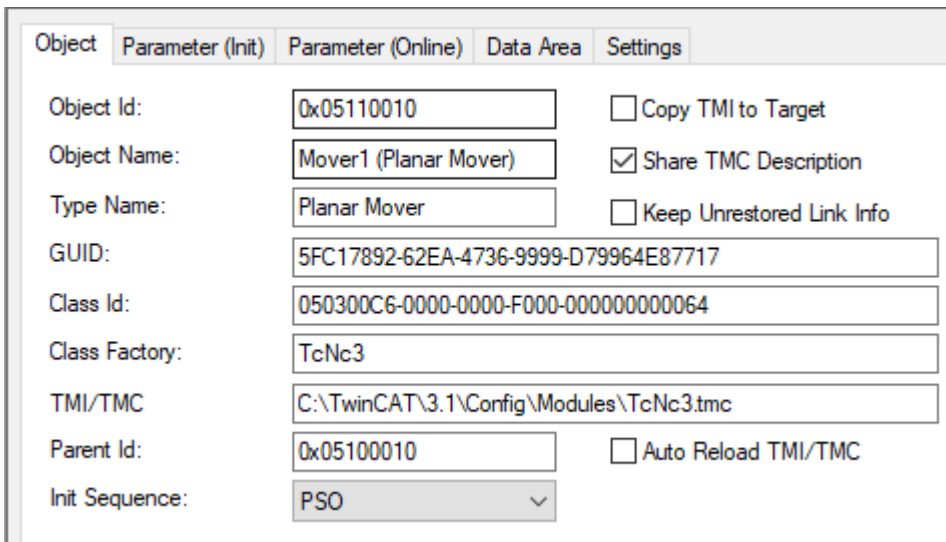
⇒ Planar 动子现已创建，可以进行参数设置。

打开详细描述

- 在树中选择 Planar 动子并双击。

各标签的用途

对象：一般信息（名称、类型、ID 等）可在此显示。



参数 (Init)：指定用户可以更改的初始参数，以影响动子的行为。



如果没有链接硬件驱动程序，在参数设置前应将参数 (Init) 置入仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

Object	Parameter (Init)	Parameter (Online)	Data Area	Settings	Name	Value	CS	Unit	Type	PTCID	Comment
- General											
					Mover width	155.0	<input type="checkbox"/>	mm	LREAL	0x050...	Width that is u...
					Mover height	155.0	<input type="checkbox"/>	mm	LREAL	0x050...	Height that is u...
					C coordinate modulus	360.0	<input type="checkbox"/>	°	LREAL	0x050...	C coordinate ...
					C coordinate modulo tolerance window	0.0	<input type="checkbox"/>	°	LREAL	0x050...	C coordinate ...
- Maximum Dynamics											
					+ Maximum dynamic XY	...	<input type="checkbox"/>	mm per s, s ² , or...		0x050...	Maximum dyn...
					+ Maximum dynamic C	...	<input type="checkbox"/>	° per s, s ² , or s ³		0x050...	Maximum dyn...
					+ Maximum dynamic Z	...	<input type="checkbox"/>	mm per s, s ² , or...		0x050...	Maximum dyn...
					+ Maximum dynamic A	...	<input type="checkbox"/>	° per s, s ² , or s ³		0x050...	Maximum dyn...
					+ Maximum dynamic B	...	<input type="checkbox"/>	° per s, s ² , or s ³		0x050...	Maximum dyn...
- Default Dynamics											
					+ Default dynamic XY	...	<input type="checkbox"/>	mm per s, s ² , or...		0x050...	Default dynami...
					+ Default dynamic C	...	<input type="checkbox"/>	° per s, s ² , or s ³		0x050...	Default dynami...
					+ Default dynamic Z	...	<input type="checkbox"/>	mm per s, s ² , or...		0x050...	Default dynami...
					+ Default dynamic A	...	<input type="checkbox"/>	° per s, s ² , or s ³		0x050...	Default dynami...
					+ Default dynamic B	...	<input type="checkbox"/>	° per s, s ² , or s ³		0x050...	Default dynami...
- Monitoring											
					Position Lag Monitoring Enabled	TRUE	<input checked="" type="checkbox"/>		BOOL	0x050...	
					+ Maximum Position Lag Value	...	<input type="checkbox"/>	mm, °		0x050...	The maximum ...
					+ Maximum Position Lag Filter Time	...	<input type="checkbox"/>	s		0x050...	The maximum ...
					Timeout for state machine	5.0	<input type="checkbox"/>	s	LREAL	0x050...	Timout for ena...

初始参数的初始设置是为了让 Planar 动子（已准备好链接）能与硬件一起移动。如果用户希望在没有硬件的情况下移动，则必须将“Simulation Mode”参数设置为 TRUE。在仿真模式下，应设置“Initial Position”和“PartOID”参数（版本 V3.2.60 及以上）。如果实际动子没有标准尺寸，则必须调整“Mover width”和“Mover height”参数。

版本 V3.1.10.30 及以上：隐藏的“Minimum/Maximum Position”参数用于定义动子何时切换到 C 轴的 CRotation 命令模式。对于 C 运动的所有目标位置，“C coordinate modulus”和“C coordinate modulo tolerance window”参数（后者用于模数定位）定义了到绝对目标位置的转换。详见 [Modulo positioning](#)。

版本 V3.1.10.51 及以上：AdoptTrackOrientation 也是 1 个 C 运动，因此受“C coordinate modulus”和“C coordinate modulo tolerance window”的影响。详见 [AdoptTrackOrientation \[► 61\]](#)。

版本 V3.2.60 及以上：“PartOID”参数用于指定仿真模式下“Initial Position”所在的部件。“PartOID”、“Simulation Mode”和“Initial Position”参数都是隐藏的，并在各自的“Simulation”分组中。

版本 V3.3.19 及以上：增加了以下参数的默认值：C 轴的“最小/最大位置”（从 10° 增至 11°）和“最大动态 C”参数（从 [10, 100, 100, 1000] 增至 [7000, 7000, 7000, 35000]）。“状态机超时”参数规定了动子保持在 Enabling、Disabling 或 Resetting 状态下的最长时间，请参见 [Planar 对象状态图 \[► 9\]](#) 中的“过渡状态超时”。“最小/最大位置”、“默认/最大动子”、“C 坐标模数”、“C 坐标模数公差窗口”和所有“监控”参数（位置监控/超时）都可以在动子 Enabled 状态下更改（之前只能在 Disabled 状态下更改）。

其他参数包括“Maximum Dynamic(s)”和“Default Dynamic(s)”。此外，还有“monitoring”参数，用于激活对实际动子的位置监控或进行参数设置。

参数（在线）：显示对象运行时的动子状态。当前的预设值位置（“SetPos”）和实际位置（“ActPos”）与状态信息一起显示。

Object	Parameter (Init)	Parameter (Online)	Data Area	Settings	Name	Online	CS	Unit	Type	PTCID	Comment
					+ SetPos	--	<input type="checkbox"/>	mm, °		0x050300D2	Mover position, read only.
					CoordinateSystemOID	00000000	<input type="checkbox"/>		OTCID	0x05030126	The coordinate system the mover is on, set and act position are given in this system.
					GroupOID	00000000	<input type="checkbox"/>		OTCID	0x050300C3	Object id of the PlanarGroup the mover is in, read only.
					State	Enabled	<input type="checkbox"/>		M.C.MC_PLANAR_STATE	0x050300B6	State, read only.
					Command mode	FreeMovement	<input type="checkbox"/>		M.C.MC_PLANAR_MOVER_COMMAND_MODE	0x050300B7	Command mode, read only.
					+ ActPos	--	<input type="checkbox"/>			0x050300C1	Mover hardware position, read only.
					+ Mover SetPos on track	--	<input type="checkbox"/>			0x050300C2	Mover state on track, read only.
					External setpoint generation mode	None	<input type="checkbox"/>		M.C.MC_EXTERNAL_SET_POSITION_MODE	0x050300DB	Indicates which external setpoint generation mode is active.

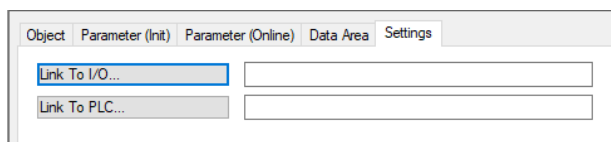
版本 V3.1.10.30 及以上：参数“External setpoint generation”指示动子是否遵循（绝对或相对）用户的外部设定点。

版本 V3.2.60 及以上：“CoordinateSystemOID”参数用于指定 SetPos 和 ActPos 所处的坐标系以及动子所处的坐标系。

数据区：显示内存区域，动子通过这些区域与其他对象相连并交换信息。

Object	Parameter (Init)	Parameter (Online)	Data Area	Settings		
Area No	Name	Type	Size	CS	CD / Elements	Owner
+ 4 (0)	IoToMc	InputDst	8	<input checked="" type="checkbox"/>	<input type="checkbox"/> 1 Symbols	
+ 0 (0)	PlcToMc	InputDst	8	<input checked="" type="checkbox"/>	<input type="checkbox"/> 1 Symbols	05100010, Offs: 0
+ 5 (0)	McToIo	OutputSrc	8	<input checked="" type="checkbox"/>	<input type="checkbox"/> 1 Symbols	
+ 1 (0)	McToPlc	OutputSrc	40	<input checked="" type="checkbox"/>	<input type="checkbox"/> 5 Symbols	05100010, Offs: 0

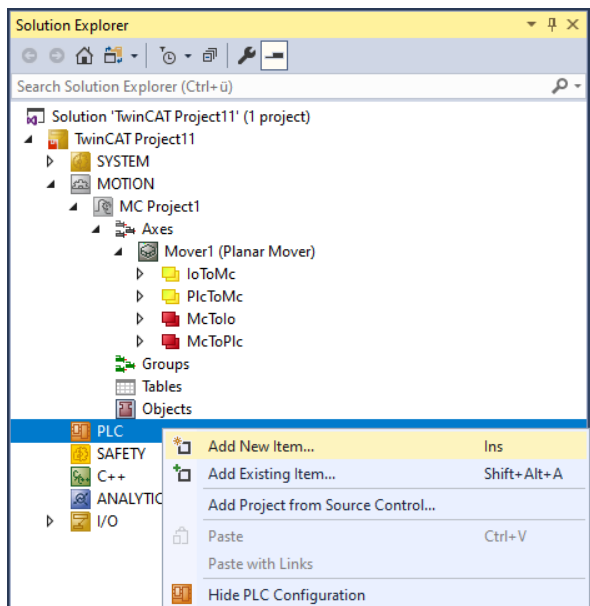
设置：用户可在此处建立链接。通过 2 个“Link To ...”按钮，可以将 XPlanar 驱动程序与 PLC 中的动子连接起来。



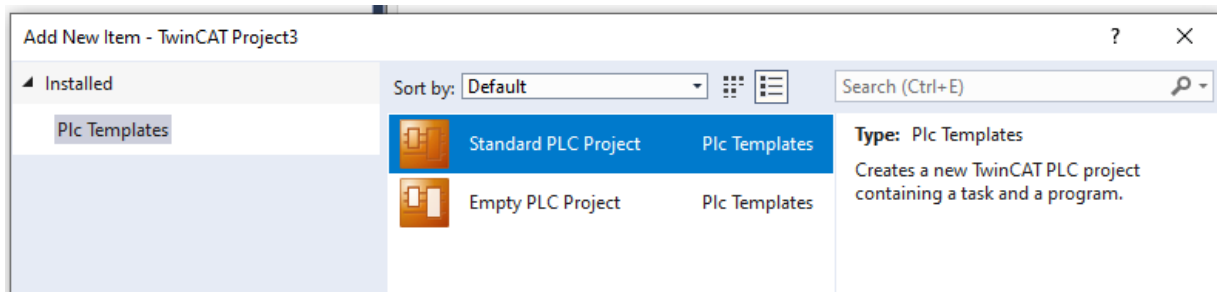
6.1.2 创建 PLC

✓ 必须创建 1 个 PLC 来控制动子、轨道或组，创建环境的几何图形或使用 Planar Feedback。

1. 选择 PLC > Add New Item...



2. 在以下对话框中，选择 **Standard PLC Project** 并按 **OK** 确认。



3. 将 “Tc3_Physics”和“Tc3_Mc3PlanarMotion”库添加到 PLC 项目中；请参阅插入功能库 [▶ 123]。

⇒ PLC 已创建，您可以按照以下示例对相应对象发出命令。

6.1.3 例如“创建和移动 Planar 动子”

通过本简短指南，您将创建 1 个包含 Planar 动子的 TwinCAT 项目，并以简单的方式移动它。

创建 Planar 动子

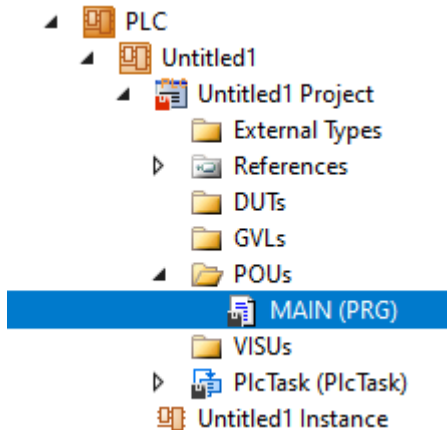
✓ 参阅 配置 [▶_16]。

1. 为此示例创建 1 个 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

创建 PLC

✓ 见初步步骤 创建 PLC [▶_19]。

1. 使用 MAIN 创建动子 (“MC_PlanarMover [▶_142]”) 如下。



⇒ 这/这些代表 MC Configuration 中的动子。

2. 创建 1 个 Planar 动子，为状态机创建 1 个任意状态变量，为动子的移动命令创建 1 个目标位置，如下图所示。

```
PROGRAM MAIN
VAR
    mover : MC_PlanarMover;
    state : UDINT;
    target_position : PositionXYC;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 这段程序代码激活了动子，并将其移动到 $x = 100$ 和 $y = 100$ 的位置。

```
CASE state OF
0:
    mover.Enable(0);
    state := 1;
1:
    IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    target_position.SetValuesXY(100, 100);
    mover.MoveToPosition(0, target_position, 0, 0);
    state := 3;
END_CASE
```

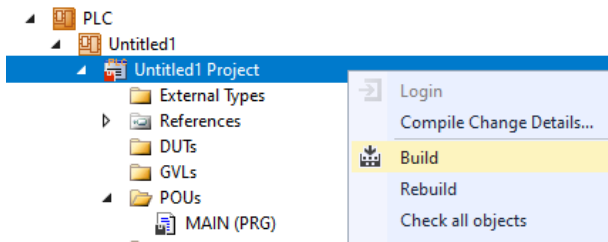
发送命令

4. 要发送命令，必须在 END_CASE 后循环调用动子的更新方法：

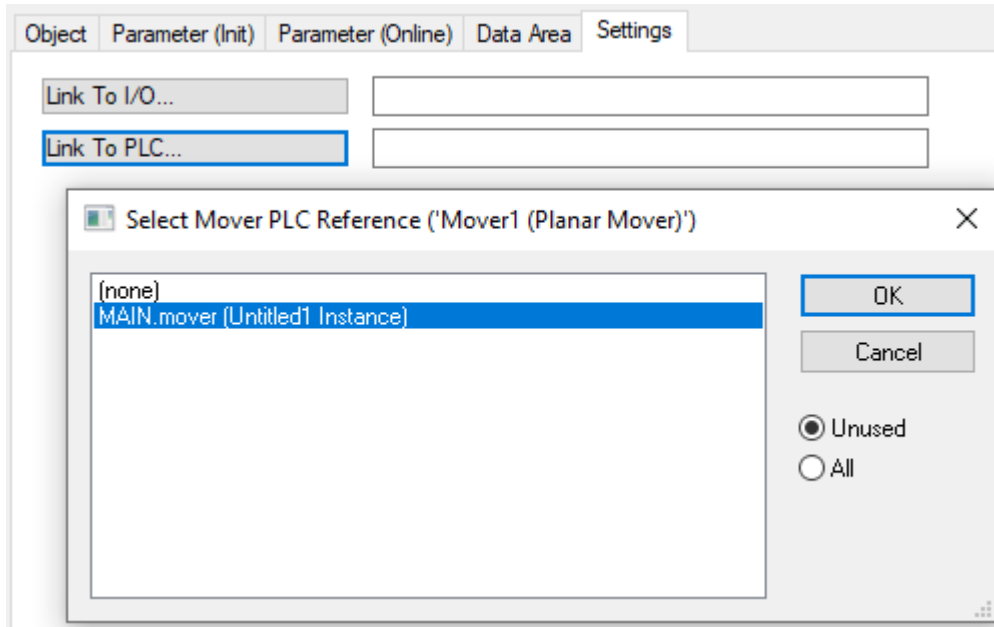
```
mover.Update();
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



⇒ 随后，可以通过 **Settings** 选项卡上的 **Link To PLC...** 按钮链接“MC Project”（双击）中的 Planar 动子。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机结束时（状态=3），动子处于理想位置。

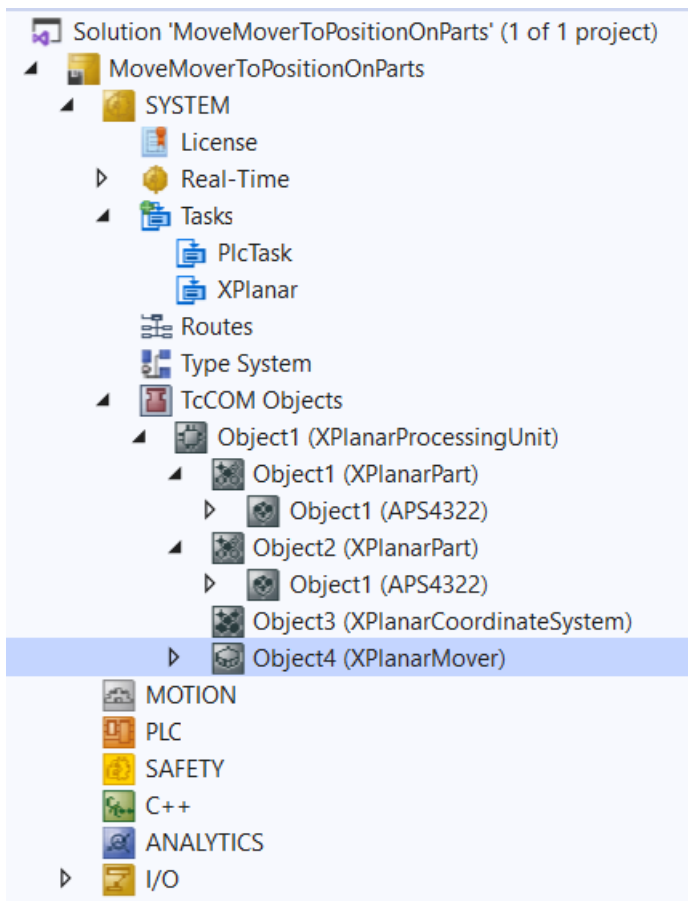
Expression	Type	Value	Prepared value	Address	Comment
mover	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLANAR_M...			%Q*	Mover data that is tra...rred from the Plana...
MCTOPLC	CDT_MCTOPLC_PLANAR_M...			%I*	Mover data that is tra...rred from the Plana...
STD	REFERENCE TO CDT_MCTO...			%IB*	Mover standard data t...is transferred from ...
SET	REFERENCE TO CDT_MCTO...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	100			X coordinate.
y	LREAL	100			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	0			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	66246161725...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT_MCTO...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT_MCTO...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT_MCTO...			%IB*	Mover busy informatio...at is transferred fro...
Error	BOOL	FALSE			Flag indicating a PlanarMover error.
ErrorId	UDINT	0			Error id indicating the PlanarMover error type.
state	UDINT	3			
target_position	PositionXYC				

6.1.4 示例“将 Planar 动子移动到 Planar 部件”

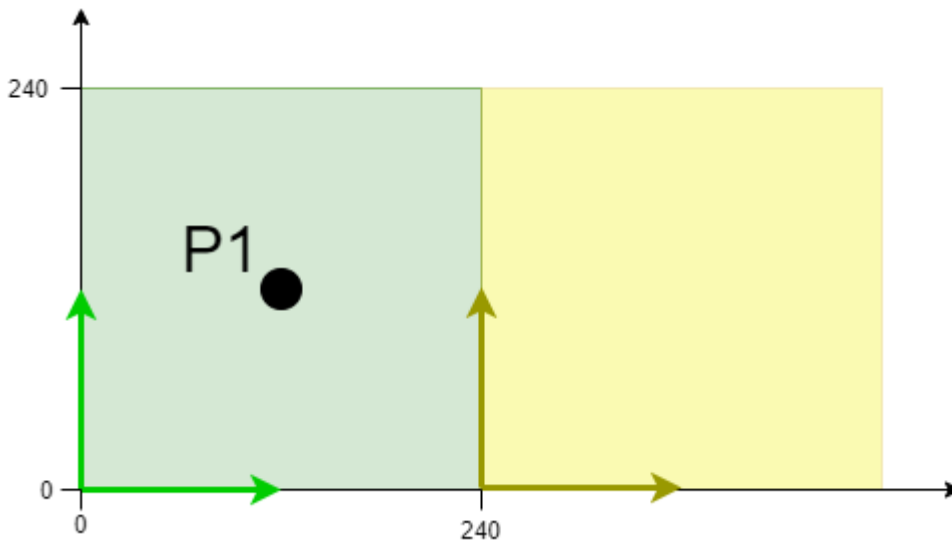
在这个例子中，1 个 Planar 动子被移动到 2 个 Planar 部件上。

起点

您可以从包含 1 个完全配置好的 XPlanar Processing Unit 的解决方案开始。在 XPlanar Processing Unit 下创建了 2 个部分，1 个坐标系和 1 个动子。2 个部分下各创建 1 个 Planar 模块。



设定的几何情况如下：2 部分相邻，动子从左边部分的中间开始（位置 P1）。这 2 个部件都是不可移动的，因此配置是静态的。



本示例就是在这种配置的基础上开发的。



XPlanar Processing Unit 的文档中介绍了初始化的步骤。

创建 Planar 动子和 Planar 环境

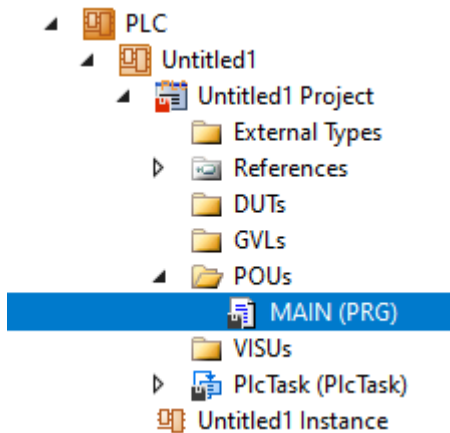
1. 为本示例创建 Planar 动子，请参阅配置 [▶_16]。
2. 创建 Planar 环境，请参阅配置 [▶_94]。

- 将初始参数 XPlanar Processing Unit OID 设置为 XPlanar Processing Unit 的对象 ID。这将激活所有 MC Configuration 对象（尤其是已创建的 Planar 动子）Part 的功能。

创建 PLC

✓ 见初步步骤 [创建 PLC \[19\]](#)。

- 使用 MAIN 创建动子（“MC PlanarMover [142]”）如下。



⇒ 这/这些代表 MC Configuration 中的动子。

- 创建 1 个 Planar 动子，为状态机创建 1 个任意状态变量，为动子的移动命令创建 1 个目标位置，如下图所示。

```
PROGRAM MAIN
VAR
    mover : MC_PlanarMover;
    state : UDINT;
    target_position : PositionXYC;
END_VAR
```

- 然后在 MAIN 中编程 1 个序列。

⇒ 该程序代码激活动子并将其移动到 x=100 和 y=100 的位置，该位置在右零件的零件坐标系中指定（其对象 Id 为 16#01010030）。

```
CASE state OF
0:
    mover.Enable(0);
    state := 1;
1:
    IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    target_position.SetValuesXYCReferenceId(100, 100, 0, 16#01010030);
    mover.MoveToPosition(0, target_position, 0, 0);
    state := 3;
END_CASE
```

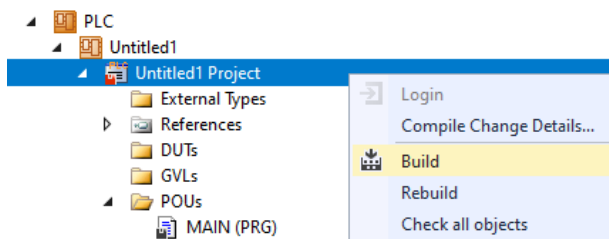
发送命令

- 要发送命令，必须在 END_CASE 后循环调用动子的更新方法：

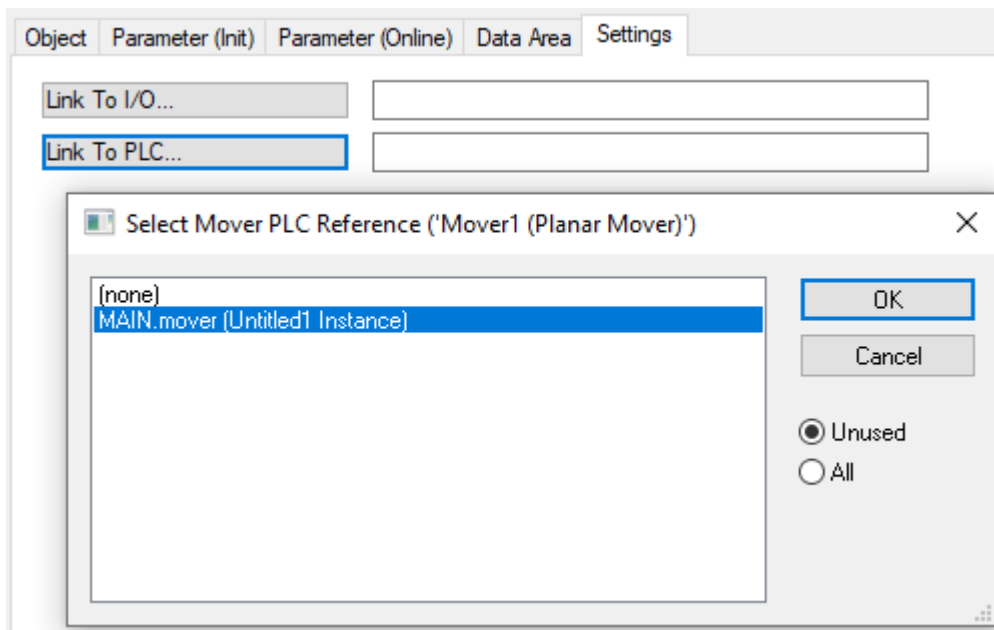
```
mover.Update();
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

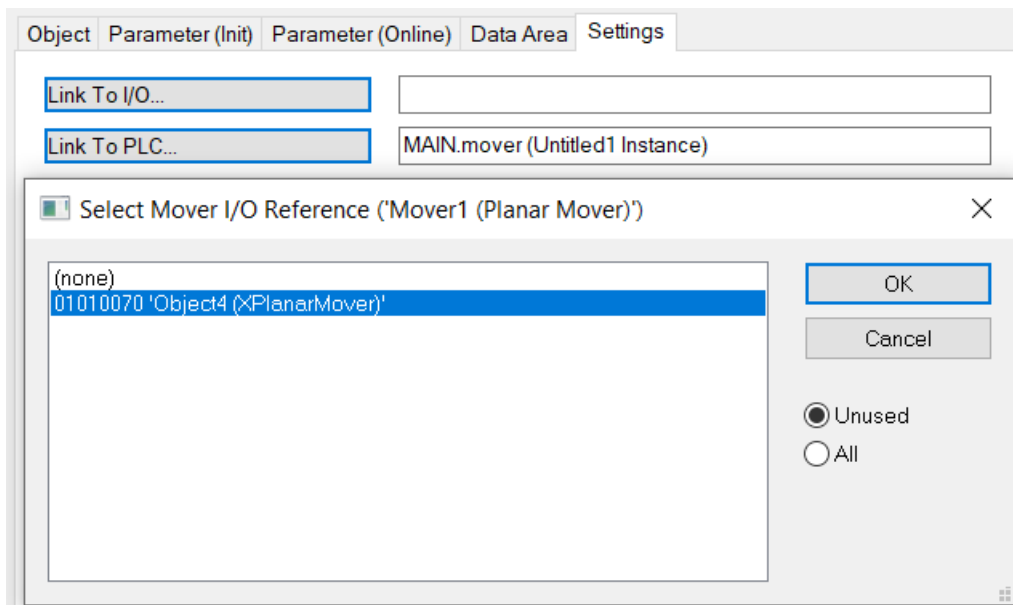
- 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



⇒ 随后，可以通过 Settings 选项卡上的 Link To PLC... 按钮链接“MC Project”（双击）中的 Planar 动子。



⇒ 此外，“MC Project”（双击）中的 Planar 动子可以通过 **Settings** 选项卡上的 **Link To I/O...** 按钮进行链接。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机结束时（状态=3），动子处于理想位置。该位置是在坐标系中指定的，而不是像 `target_position` 一样，在右侧零件（其对象 ID 为 `16#01010030`）的零件坐标系中指定的。2 个系统在 `x` 方向上都偏移了 240 毫米。还可以看到，`a`、`b` 和 `z` 轴出现了轻微的噪音。这是由 XPlanar Processing Unit 仿真生成的，对启动时最初接受的位置有影响。

Expression	Type	Value
[-] mover	MC_PlanarMover	
[-] PLCTOMC	CDT_PLCTOMC...	
[-] MCTOPLC	CDT_MCTOPLC...	
[-] STD	REFERENCE TO...	
[-] SET	REFERENCE TO...	
[-] SetPos	MoverVector	
x	LREAL	340
y	LREAL	100
z	LREAL	2.005329507...
a	LREAL	-0.00078074...
b	LREAL	-0.00097010...
c	LREAL	-3.05631426...
[-] SetVelo	MoverVector	
[-] SetAcc	MoverVector	
DcTimeStamp	ULINT	7562220860...
PhysicalAreaID	UDINT	16842848
[-] ACT	REFERENCE TO...	
[-] COORDMODE	REFERENCE TO...	
[-] SETONTRACK	REFERENCE TO...	
[-] Error	BOOL	FALSE
[-] ErrorId	UDINT	0
[-] state	UDINT	3
[-] target_position	PositionXYC	

6.1.5 示例“创建并移动带辅助轴的 Planar 动子”

通过本简短指南，您将创建 1 个包含 Planar 动子的 TwinCAT 项目，并以简单的方式移动它。

创建 Planar 动子

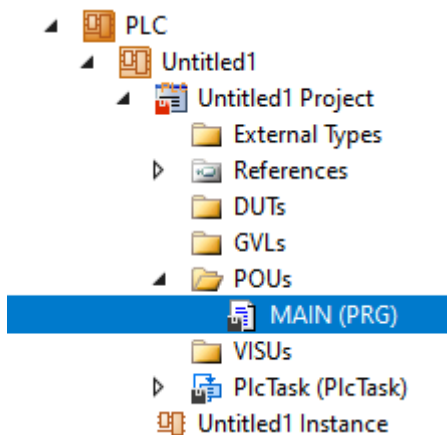
✓ 参阅 [配置 \[▶_16\]](#)。

1. 创建 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

创建 PLC

✓ 见初步步骤 [创建 PLC \[▶_19\]](#)。

1. 使用 MAIN 创建动子 (“MC_PlanarMover [▶_142]”) 如下。



⇒ 这/这些代表 MC Configuration 中的动子。

2. 创建 1 个 Planar 动子，为状态机创建 1 个任意状态变量，为动子的移动命令创建 1 个目标位置，如下图所示。

```
PROGRAM MAIN
VAR
    mover : MC_PlanarMover;
    state : UDINT;
    target_a : LREAL := 1.0;
    target_b : LREAL := -1.0;
    target_c : LREAL := 3.0;
    target_z : LREAL := 5.0;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 该程序代码可激活动子并移动 4 个辅助轴。

```
CASE state OF
0:
    mover.Enable(0);
    state := 1;
1:
    IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    mover.MoveA(0, target_a, 0);
    mover.MoveB(0, target_b, 0);
    // Since Version V3.1.10.11 MoveC has an options parameter,
    // details can be found in the CRotation example
    // and the options descriptions
    //mover.MoveC(0, target_c, 0); // until version V3.1.10.11
    mover.MoveC(0, target_c, 0, 0); // since version V3.1.10.30
    mover.MoveZ(0, target_z, 0);
    state := 3;
END_CASE
```

更多信息：

- [示例“在 CRotation 模式下移动 Planar 动子”](#) [► 36]
- [运动命令的限值和选项](#) [► 38]

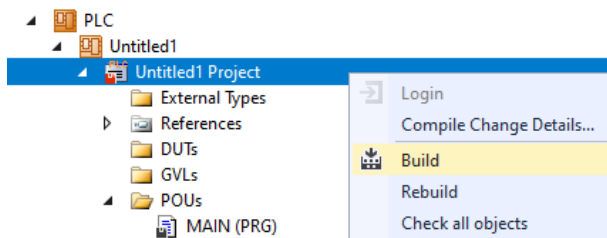
发送命令

4. 要发送命令，必须在 END_CASE 后循环调用动子的更新方法：

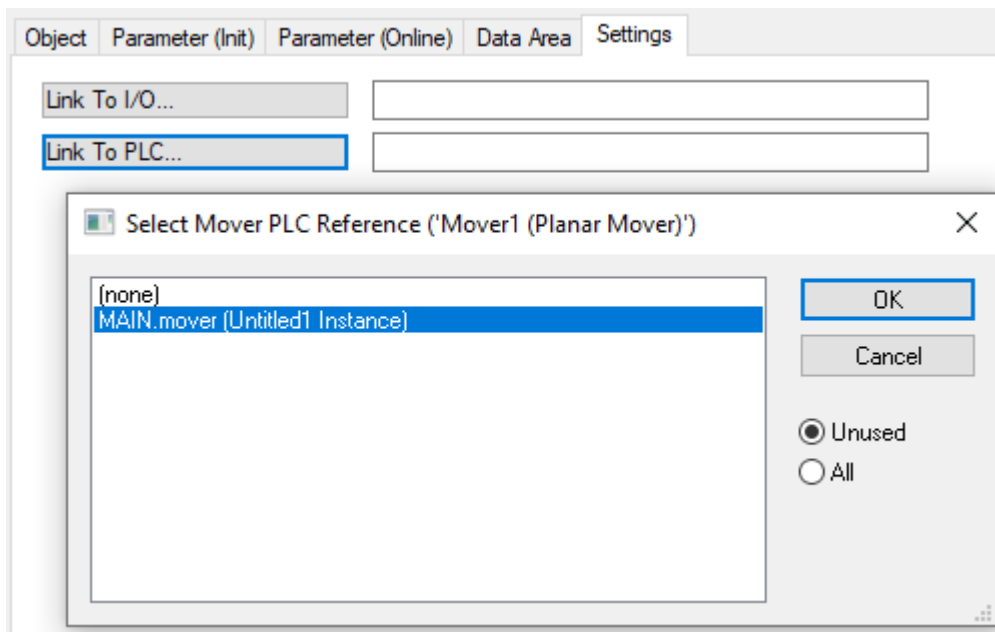
```
mover.Update();
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



⇒ 随后，可以通过 **Settings** 选项卡上的 **Link To PLC...** 按钮链接“MC Project”（双击）中的 Planar 动子。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机结束时（状态=3），动子处于理想位置。

Expression	Type	Value	Prepared value	Address	Comment
mover	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLA...			%Q*	Mover data that is tra...rred from the Plana...
MCTOPLC	CDT_MCTOPLC_PLA...			%I*	Mover data that is tra...rred from the Plana...
STD	REFERENCE TO CDT...			%IB*	Mover standard data t...is transferred from ...
SET	REFERENCE TO CDT...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	0			X coordinate.
y	LREAL	0			Y coordinate.
z	LREAL	5			Z coordinate.
a	LREAL	0.999999999999...			A coordinate.
b	LREAL	-0.999999999999...			B coordinate.
c	LREAL	2.999999999999...			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	6630642690730...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT...			%IB*	Mover busy informatio...at is transferred fro...
Error	BOOL	FALSE			Flag indicating a PlanarMover error.
ErrorId	UDINT	0			Error id indicating the PlanarMover error type.
state	UDINT	3			
target_a	LREAL	1			
target_b	LREAL	-1			
target_c	LREAL	3			
target_z	LREAL	5			

6.1.6 示例“利用外部设定点生成功能创建并移动 Planar 动子”

通过本简短指南，您将创建 1 个包含 Planar 动子的 TwinCAT 项目，并通过外部设定点生成功能以简单的方式移动 Planar 动子。

创建 Planar 动子

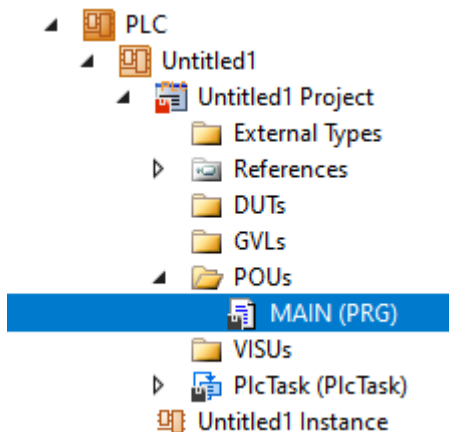
✓ 参阅 [配置 \[▶_16\]](#)。

1. 创建 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

创建 PLC

✓ 见初步步骤 [创建 PLC \[▶_19\]](#)。

1. 使用 MAIN 创建动子 (“MC_PlanarMover [▶_142]”) 如下。



⇒ 这/这些代表 MC Configuration 中的动子。

2. 创建 1 个 Planar 动子、1 个状态机的状态变量和外部设定点的变量，如下图所示。

```
PROGRAM MAIN
VAR
  mover : MC_PlanarMover;
  state : UDINT;
  p, v, a : MoverVector;
  deltat : LREAL := 0.001;
  velo, acc, jerk : LREAL;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 该程序代码激活动子，并开始生成外部设定点。然后动子会遵循一个以正速度结束的运动曲线。随后停止外部设定点的生成，确保动子将速度降为零，并在停止后处于 FreeMovement 状态（这是在动子的最大动态下完成的）。

```
CASE state OF
0:
  mover.Enable(0);
  state := 1;
1:
  IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
    state := 2;
  END_IF
2:
  p.x := 0.0; v.x := 0.0; a.x := 0.0;
  mover.StartExternalSetpointGeneration(0,0);
  mover.SetExternalSetpoint(p, v, a);
  state := 3;
3:
  velo := v.x;
  acc := a.x;
  p.x := p.x + deltat * velo + deltat * deltat / 2 * acc + deltat * deltat * deltat / 6 * jerk;
  v.x := v.x + deltat * acc + deltat * deltat / 2 * jerk;
  a.x := a.x + deltat * jerk;
  mover.SetExternalSetpoint(p, v, a);
  IF a.x >= 10.0 THEN
    jerk := -1;
```

```

END_IF;
IF a.x <= 0.0 THEN
    state := 4;
END_IF;
5:
mover.StopExternalSetpointGeneration(0);
state := 6;
END_CASE

```

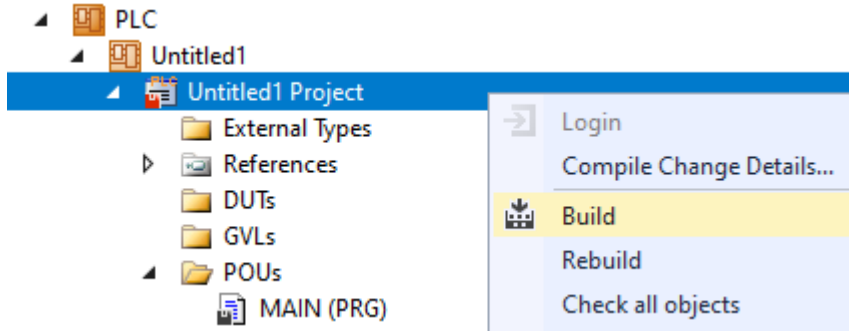
发送命令

- 要发送命令，需要在 END_CASE 之后触发动子的更新方法：

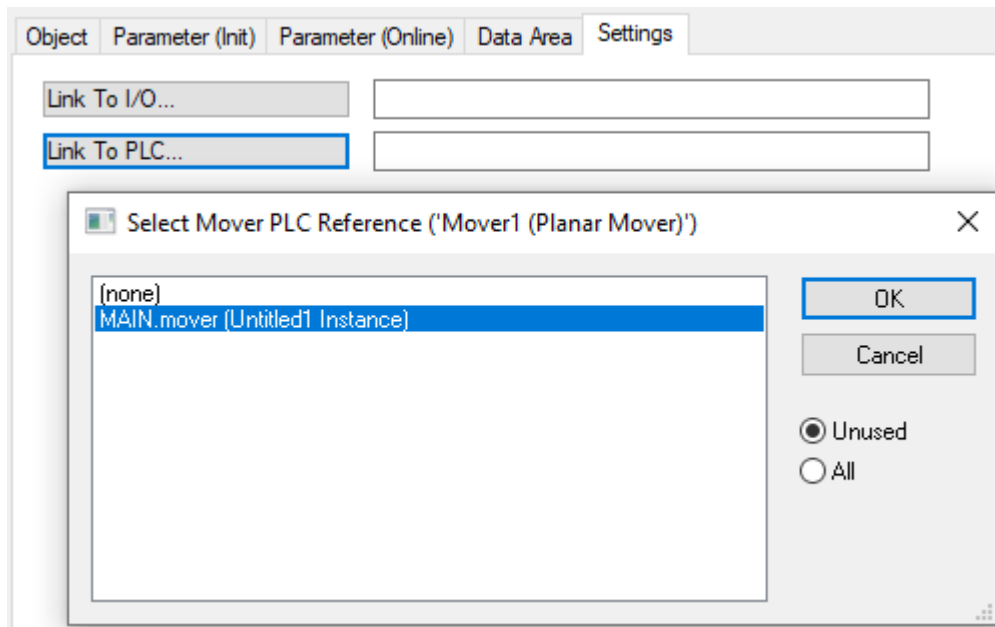
```
mover.Update();
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

- 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



⇒ 随后，可以通过 Settings 选项卡上的 Link To PLC... 按钮链接“MC Project”（双击）中的 Planar 动子。



激活并启动项目

- 通过菜单栏  中的按钮激活配置。
- 通过  按钮将 TwinCAT 系统设置为“Run”状态。
- 通过菜单栏  中的按钮登录 PLC。
- 通过菜单栏中的 Play 按钮启动 PLC。

状态机结束时（状态 = 6），动子处于所需的正 x 位置。

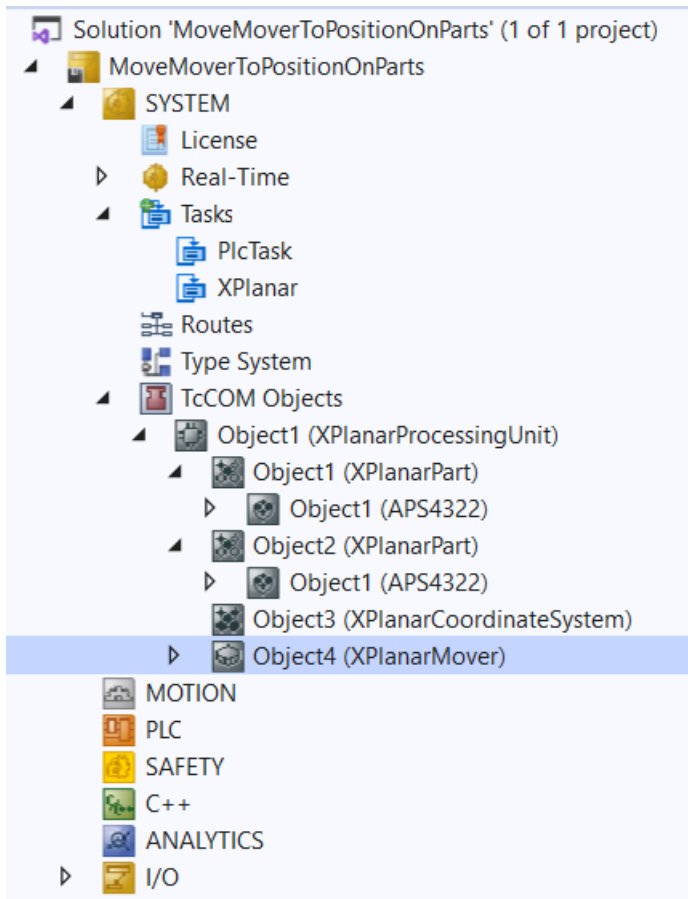
Expression	Type	Value	Prepared value	Address	Comment
[-] mover	MC_PlanarMover				
[-] PLCTOMC	CDT_PLCTOMC...			%Q*	Mover data that i...ansferred from ...
[-] MCTOPLC	CDT_MCTOPLC...			%I*	Mover data that i...ansferred from ...
[-] STD	REFERENCE TO...			%IB*	Mover standard d...that is transfer...
[-] SET	REFERENCE TO...			%IB*	Mover setpoint d...hat is transferr...
[-] SetPos	MoverVector				Current position.
[-] x	LREAL	1007.249774...			X coordinate.
[-] y	LREAL	0			Y coordinate.
[-] z	LREAL	0			Z coordinate.
[-] a	LREAL	0			A coordinate.
[-] b	LREAL	0			B coordinate.
[-] c	LREAL	0			C coordinate.
[-] SetVelo	MoverVector				Current velocity.
[-] SetAcc	MoverVector				Current acceleration.
[-] DcTimeStamp	ULINT	7238174810...			Current time stamp.
[-] PhysicalAreaID	UDINT	0			Current physical area id.
[-] ACT	REFERENCE TO...			%IB*	Mover actpoint d...hat is transferr...
[-] COORDMODE	REFERENCE TO...			%IB*	Mover coordinate...de information...
[-] SETONTRACK	REFERENCE TO...			%IB*	Mover busy information that is tran...
[-] Error	BOOL	FALSE			Flag indicating a Planar Mover error.
[-] ErrorId	UDINT	0			Error id indicating...e Planar Mover ...
[-] state	UDINT	6			
[-] p	MoverVector				

6.1.7 示例“将带有外部设定点生成功能的 Planar 动子移动到 Planar 部件”

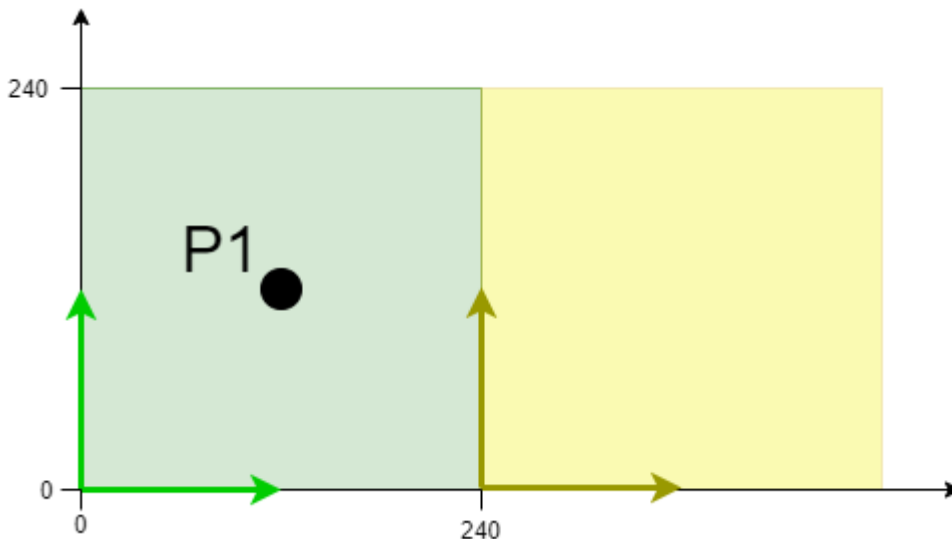
在此示例中，1 个 Planar 动子被移动到 2 个 Planar 部件上，并由外部生成设定点。

起点

您可以从包含 1 个完全配置好的 XPlanar Processing Unit 的解决方案开始。在 XPlanar Processing Unit 下创建了 2 个部分，1 个坐标系和 1 个动子。2 个部分下各创建 1 个 Planar 模块。



设定的几何情况如下：2 部分相邻，动子从左边部分的中间开始（位置 P1）。2 部分都不能移动，因此配置是静态的。



本示例就是在这种配置的基础上开发的。



XPlanar Processing Unit 的文档中介绍了初始化的步骤。

创建 Planar 动子和 Planar 环境

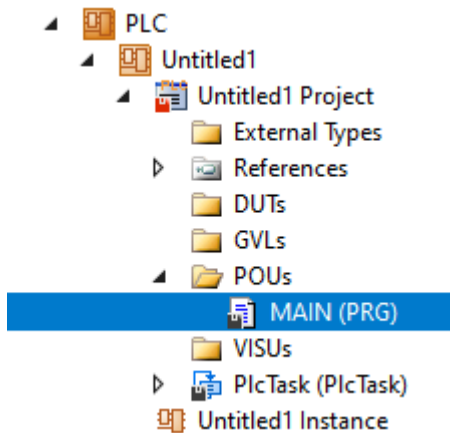
1. 为本示例创建 Planar 动子，请参阅配置 [▶_16]。
2. 创建 Planar 环境，请参阅配置 [▶_94]。

3. 将初始参数 XPlanar Processing Unit OID 设置为 XPlanar Processing Unit 的对象 ID。这将激活所有 MC Configuration 对象（尤其是已创建的 Planar 动子）Part 的功能。

创建 PLC

✓ 见初步步骤 [创建 PLC \[19\]](#)。

1. 使用 MAIN 创建动子（“MC PlanarMover [142]”）如下。



⇒ 这/这些代表 MC Configuration 中的动子。

2. 创建 1 个 Planar 动子、1 个状态机的状态变量和外部设定点的变量，如下图所示。

```
PROGRAM MAIN
VAR
    mover : MC_PlanarMover;
    state : UDINT;
    p, v, a : MoverVector;
    deltat : LREAL := 0.01;
    refsys : OTCID := 0;
    velo, acc, jerk : LREAL;
    feedback : MC_PlanarFeedback;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 该程序代码激活动子，并开始生成外部设定点。然后动子会遵循一个以正速度结束的运动曲线。随后停止外部设定点的生成，确保动子将速度降为零，并在停止后处于 FreeMovement 状态（这是在动子的最大动态下完成的）。

```
CASE state OF
0:
    mover.Enable(0);
    state := 1;
1:
    IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    p := mover.MCTOPLC.SET.SetPos;
    v.x := 0.0; a.x := 0.0;
    mover.StartExternalSetpointGeneration(0,0);
    mover.SetExternalSetpointReferenceId(feedback, p, v, a, refsys);
    jerk := 10;
    state := 3;
3:
    velo := v.x;
    acc := a.x;
    p.x := p.x + deltat * velo + deltat * deltat / 2 * acc + deltat * deltat * deltat / 6 * jerk;
    v.x := velo + deltat * acc + deltat * deltat / 2 * jerk;
    a.x := acc + deltat * jerk;
    mover.SetExternalSetpointReferenceId(feedback, p, v, a, refsys);
    IF a.x >= 100.0 THEN
        jerk := -10;
    END_IF;
    IF a.x <= 0.0 THEN
        state := 4;
    END_IF;
4:
    mover.StopExternalSetpointGeneration(0);
```

```
state := 5;
END_CASE
```

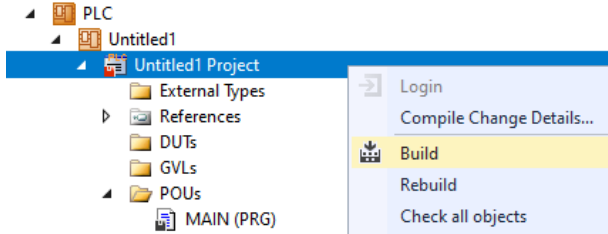
发送命令

4. 要发送命令，需要在 END_CASE 之后触发动子的更新方法：

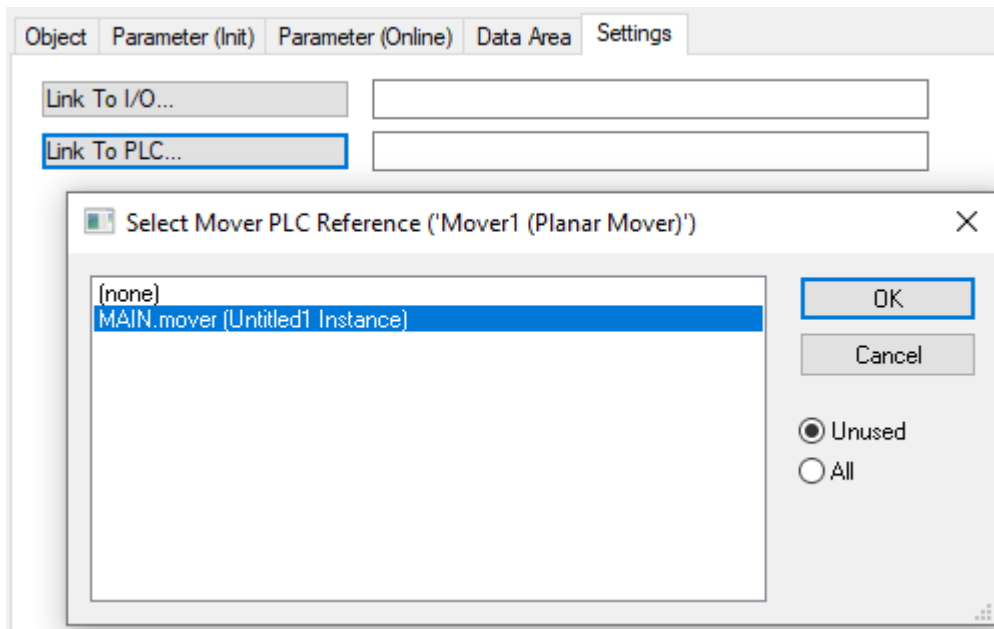
```
mover.Update();
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

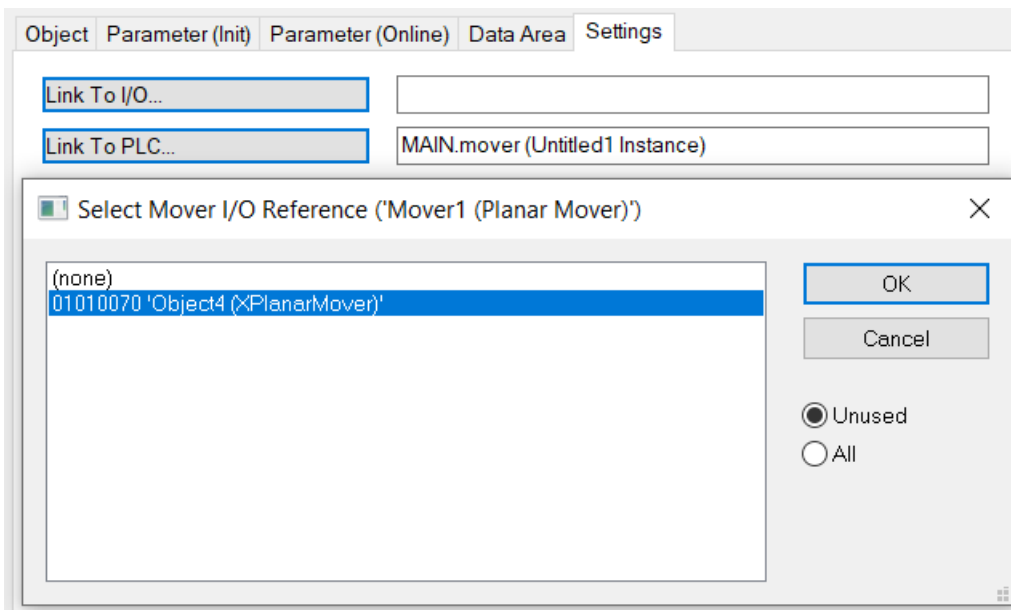
1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



- ⇒ 随后，可以通过 **Settings** 选项卡上的 **Link To PLC...** 按钮链接“MC Project”（双击）中的 Planar 动子。



- ⇒ 此外，“MC Project”（双击）中的 Planar 动子可以通过 **Settings** 选项卡上的 **Link To I/O...** 按钮进行链接。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

状态机结束时（状态 = 5），动子处于所需的正 x 位置。它还没有离开第 1 部分，因此在这种情况下，也可以指定第 1 部分的对象 ID 或坐标系为参考坐标。如果指定了第 1 部分的对象 ID，并且超出了到第 2 部分的限制，则必须使用第 2 部分的对象 ID 并将 x 坐标缩小 240（同时！）。无论您在哪个部分，全局坐标系的对象 ID 都会起作用。由于配置是静态的，全局坐标系 ID 可以选择 0。

Expression	Type	Value	Prepared value	Address	Comm
[-] mover	MC_PlanarMover				
[-] PLCTOMC	CDT_PLCTOMC...			%Q*	Mover d
[-] MCTOPLC	CDT_MCTOPLC...			%I*	Mover d
[-] STD	REFERENCE TO...			%IB*	Mover s
[-] SET	REFERENCE TO...			%IB*	Mover s
[-] SetPos	MoverVector				Current
[-] x	LREAL	223.2495384...			X coordi
[-] y	LREAL	120.0356967...			Y coordi
[-] z	LREAL	1.993983856...			Z coordi
[-] a	LREAL	-0.00807865...			A coordi
[-] b	LREAL	-0.00036782...			B coordi
[-] c	LREAL	0.010318374...			C coordi
[-] SetVelo	MoverVector				Current
[-] SetAcc	MoverVector				Current
[-] DcTimeStamp	ULINT	7238109826...			Current
[-] PhysicalAreaID	UDINT	16842848			Current
[-] ACT	REFERENCE TO...			%IB*	Mover a
[-] COORDMODE	REFERENCE TO...			%IB*	Mover c
[-] SETONTRACK	REFERENCE TO...			%IB*	Mover b
[-] Error	BOOL	FALSE			Flag ind
[-] ErrorId	UDINT	0			Error id
[-] state	UDINT	5			
[-] p	MoverVector				
[-] v	MoverVector				
[-] a	MoverVector				
[-] deltat	LREAL	0.01			

对反馈部分的评论:

在本例中，只创建了反馈，并在 SetExternalSetpointRefSys 调用中输入，而没有对其进行任何操作。由于本例中没有发生错误，因此反馈从状态 2 开始显示繁忙状态。否则，您应该始终使用此反馈执行错误处理，以便处理诸如位置的 RefSysId 不正确等错误。

这种反馈的特点是循环传递，每次调用都必须传递相同的反馈。如果外部设定点生成过程中出现错误，在下次调用带反馈的 SetExternalSetpointRefSys 后，这些错误将显示在反馈中。在此，没有必要周期性的更新反馈信息。此外，在调用 StopExternalSetpointGeneration 时不会进行反馈，在调用 Halt 时也不会中止反馈。

即使在调用 SetExternalSetpointRefSys 之后，反馈信息中出现了信息，错误也不一定直接来自这次调用，而也可能来自之前的调用。

6.1.8 示例“在 CRotationFreeMovement 模式下移动 Planar 动子”

通过本简短指南，您将创建 1 个包含 Planar 动子的 TwinCAT 项目，并以简单的方式移动它。

创建 Planar 动子

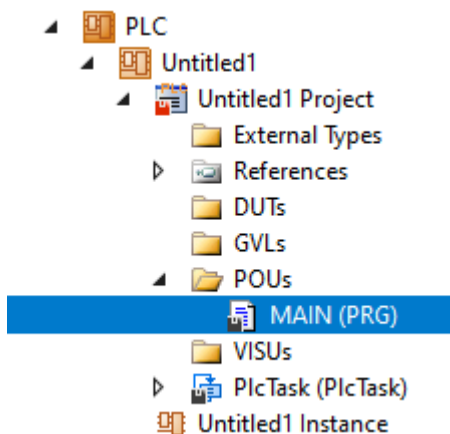
✓ 参阅 配置 [▶ 16]。

1. 创建 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

创建 PLC

✓ 见初步步骤 创建 PLC [▶ 19]。

1. 使用 MAIN 创建动子 (“MC PlanarMover [▶ 142]”) 如下。



⇒ 这/这些代表 MC Configuration 中的动子。

2. 创建 1 个 Planar 动子，为状态机创建 1 个任意状态变量，为动子的移动命令创建 1 个目标位置，如下图所示。

```
PROGRAM MAIN
VAR
  mover : MC_PlanarMover;
  state : UDINT;
  target_position_c : LREAL;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 这段程序代码激活了动子，并将其旋转到 c=20 的位置。

```
CASE state OF
  0:
    mover.Enable(0);
    state := 1;
  1:
    IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
      state := 2;
    END_IF
  2:
```

```
target_position_c := 20.0;
mover.MoveC(0, target_position_c, 0, 0);
state := 3;
END_CASE
```

发送命令

4. 要发送命令，必须在 END_CASE 后循环调用动子的更新方法：

```
mover.Update();
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。

⇒ 随后，可以通过 Settings 选项卡上的 Link To PLC... 按钮链接“MC Project”（双击）中的 Planar 动子。

激活并启动项目

1. 通过菜单栏  中的按钮激活配置。

2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。

3. 通过菜单栏  中的按钮登录 PLC。

4. 通过菜单栏中的 Play 按钮启动 PLC。

由于角度大于 15°，因此动子处于状态机末端（state=3）的理想（旋转）位置，并处于命令模式 CRotationFreeMovement 中。在完成命令后，如果 C 轴继续移动到最大，例如 90°，命令器模式将变回 Free Movement 模式。

ExampleCRotationFreeMovement.Untitled1.MAIN					
Expression	Type	Value	Prepared value	Address	Comment
mover	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC...			%Q*	Mover data that is transferred from the Planar Mover to this function block.
MCTOPLC	CDT_MCTOPLC...			%I*	Mover data that is transferred from the Planar Mover to this function block.
STD	REFERENCE TO...			%IB*	Mover standard data that is transferred from the Planar Mover to this function b
MoverOID	OTCID	16#05110010			Object id of the planar mover.
GroupOID	OTCID	16#00000000			Object id of the planar group the mover is in.
State	MC_PLANAR_S...	Enabled			State of the planar mover, e.g. enabled.
CommandMode	MC_PLANAR_M...	CRotationFre...			Command mode of the planar mover, e.g. onTrack.
Busy	MoverBusy				Busy state of the planar mover.
ErrorCode	HRESULT	16#00000000			Error code of the planar mover.
SET	REFERENCE TO...			%IB*	Mover setpoint data that is transferred from the Planar Mover to this function b
SetPos	MoverVector				Current position.
x	LREAL	0			X coordinate.
y	LREAL	0			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	19.99999999...			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	6915035679...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO...			%IB*	Mover actpoint data that is transferred from the Planar Mover to this function b
COORDMODE	REFERENCE TO...			%IB*	Mover coordinate mode information that is transferred from the Planar Mover to t
SETONTRACK	REFERENCE TO...			%IB*	Mover busy information that is transferred from the Planar Mover to this functio
Error	BOOL	FALSE			Flag indicating a Planar Mover error.
ErrorId	UDINT	0			Error id indicating the Planar Mover error type.
state	UDINT	3			
target_position_c	LREAL	20			

6.1.9 运动命令的限值 and 选项

Planar 动子可以执行不同类型的运动命令。除了外部设定点生成的特殊情况外，它们的结构相似。以下内容适用于其余的运动命令器。方法调用的第一个参数总是用户传输命令器的反馈。如果他传输的是“0”，这意味着他不想获得（或使用）反馈。接下来的一两个参数描述了运动的目的地，不能完全省略。接下来的参数是运动过程中应遵守的动态限值。如果用户在此输入“0”，则使用默认值（MC Project 中动子的 TCOM 参数）。最后一个参数是选项对象，根据命令而各不相同。

限值

每个运动命令都在理想时间内运行。要使得到的轨迹保持连续，位置的时间导数必须受到限制。限值包括速度、正负加速度和抖动的最大值。如果此处指定的值超过了动子的最大动态限值（MC 项目中动子的 TCOM 参数），则会相应减小，发出警告，并以减小的动态值执行命令。限制或约束对象只有一个。这可以理解为与动子运动方向相切的动力学限制。

对于外部设定点生成，唯一的参数是 Feedback 和 Options。

版本 V3.1.10.30 及以上： 限值应由 Constraints 取代，请参见 [Dynamics](#)。

选项

根据命令的不同，选项也有所不同：

MoveToPosition/JoinTrack/LeaveTrack 命令： 这些命令的唯一选项是“UseOrientation”标记。该标志表示是否同时使用 XYC 目标位置的 C 坐标。如果没有，可以通过“MoveC”单独移动 C 坐标。

MoveOnTrack： 第一个选项是“gap”。该数值表示在运动过程中（以及之后直到轨道上的下一个运动命令之前）与前方动子的距离。该距离是沿着轨道测量的（2 个动子的轨道位置之差）。因此，必须考虑到轨道的曲率，因为它们会减少实际的二维距离。间隙是从中心到中心计算的，因此必须考虑到动子的宽度。第二个选项是“Direction”，即在轨道上向目的地行进的方向。其值可以是“NonModulo”（= 绝对值）、“Positive”（= 正向）、“ShortestWay”（= 最短路径）和“Negative”（= 反向）。如果在适当的方向上可以到达目的地，则执行命令。

版本 V3.1.10.30 及以上： 第三个选项是“AdditionalTurns”：在“Direction”为“Positive”或“Negative”的“Closed Loop”赛道上行驶的额外圈数。对于其他“Direction”情况，“AdditionalTurns”必须为零。第四个选项是“ModuloTolerance”。当起始位置和目标位置非常相似时，该参数用于避免意外旋转。如果起始位置和目标位置之间的距离小于或等于“ModuloTolerance”，则以最短路径（如“Direction”=“Shortest Way”）接近目标位置，即与指定的“Direction”相反。对于“Direction”=“NonModulo”，ModuloTolerance 必须为零。详见 [Modulo positioning](#)。

版本 V3.3.39 及以上： 第五个选项是“GapMode”。该设置可设为“Standard1D”或“Fast1D”。

Standard1D： 这是默认值，与之前的行为一致。这种模式通常会导致两个动子之间的距离大于作为“Gap”选项通过的最小间隙。两个动子之间的距离通常会随着速度的增加而增加。这会导致动子链在行驶过程中分散开来。

注意

可能的碰撞造成的损坏

请注意，在下列情况下，在移动过程中更改参数可能会造成损坏。

在运动过程中减少一个或多个动态参数 A、D 或 J 可能会导致违反参数设置的最小“间隙”。不过，在这种模式下，增加一个或多个参数的数值都不会造成违规。同样，在这种模式下更改动态参数 V 也不会造成此类违规。

Fast1D： 这是一种新模式，目的是在不低于“间隙”选项所设置距离的情况下，尽可能晚地介入运动控制。与“Standard1D”模式相比，这通常会导致运动过程中相邻动子之间的距离变小。这样便能更快地移动动子链。

注意

可能的碰撞造成的损坏

请注意，在下列情况下，在移动过程中更改参数可能会造成损坏。

在运动过程中，一个或多个动态参数 A、D 或 J 的变化（即增加或减少）可能导致违反最小“间隙”。与“Standard1D”模式一样，在该模式下可根据需要更改动态参数 V，而不会造成此类违规。

版本 V3.1.10.30 及以上: MoveC: 第一个选项是“AdditionalTurns”:

与“Direction”“Positive”或“Negative”的动子的“C coordinate modulus”参数相关的附加完整 C 圈数。对于其他“Direction”情况，“AdditionalTurns”必须为零。第二个选项是“Direction”: C 坐标朝目标旋转的方向。其值可以是“NonModulo”(= 绝对值)、“Positive”(= 正向)、“ShortestWay”(= 最短路径)和“Negative”(= 反向)。详见 [Modulo positioning](#)。

版本 V3.1.10.51 及以上: AdoptTrackOrientation: 第一个选项是“AdditionalTurns”:

与“Direction”“Positive”或“Negative”动子的“C coordinate modulus”参数相关的附加完整 C 圈数。对于其他“Direction”情况，“AdditionalTurns”必须为零。第二个选项是“Direction”: C 坐标朝目标旋转的方向。其值可以是“NonModulo”(= 绝对值)、“Positive”(= 正向)、“ShortestWay”(= 最短路径)和“Negative”(= 反向)。详见 [Modulo positioning](#)。

版本 V3.1.10.44 及以上: GearInPosOnTrack: 第一个选项是“Gap”, 其解释与 MoveOnTrack 相同。第二个参数是“InSyncToleranceDistance”。它规定了在主动 Planar 动子失去同步性之前, 主动子和从动子可以相距多远。下面 2 个选项是“Direction”和“ModuloTolerance”, 它们都指向参数“SlaveSyncPosition”(函数调用时的输入)。只有当 Planar 动子执行同步运动的 Planar 轨道是闭合回路时, 这些选项才可用。在这种情况下, 这些选项的解释与 MoveC 类似, 这里的模量由 Planar 轨道的长度给出。详见 [Modulo positioning](#)。最后一个参数“AllowedSlaveSyncDirections”指定在同步阶段允许 Planar 动子向哪个方向移动, 即正向(默认)、反向或两者兼有。例如, 该参数可用于防止选项“Both”可能出现的反向振荡, 以实现最快的同步。如果 Planar 动子处于同步状态, 或曾处于同步状态, 且目前正试图恢复同步, 则该参数没有进一步的影响。

版本 V3.1.10.30 及以上: GearInPosOnTrackWithMasterMover: 前 4 个选

项“Gap”、“InSyncToleranceDistance”以及 SlaveSyncPosition 的 2 个模数选项在含义上与 GearInPosOnTrack 命令的前 4 个选项相同。随后是 MasterSyncPosition 的 2 个参数“Direction”和“ModuloTolerance”, 当主 Planar 动子处于闭环轨道上时, 这 2 个参数的使用方式与 SlaveSyncPosition 的公差模数参数相同。下面的选项“AllowedSlaveSyncDirections”与 GearInPosOnTrack 命令的功能完全相同。如果设置了最后一个选项“FollowMover”, 则可确保从 Planar 动子未必要获得 [Planar TrackTrail](#) [► 118] 才能知道它将在哪些 Planar 轨道上执行运动。从 Planar 动子只需跟随主 Planar 动子在网络中运动即可。如果接收到运动命令时主 Planar 动子和从 Planar 动子处于不同的 Planar 轨道上, 并且设置了“FollowMover”选项, 则从 Planar 动子系统将尝试以最短路径到达 MasterSyncPosition 命令的 Planar 轨道。除了“FollowMover”选项外, 还可以为从 Planar 动子指定 PlanarTrackTrail。在这种情况下, 它用于命令 MasterSyncPosition 所在 Planar 轨道的路径(例如, 如果要偏离最短路径)。如果它没有完全到达该 Planar 轨道, 剩余的路径将由最短路径填补。如果在设置“FollowMover”选项时指定了 PlanarTrackTrail, 则可以在与其初始轨道不同的 Planar 轨道上指定 SlaveSyncPosition。一般情况下, 适用以下规则: 如果设置了“FollowMover”选项, 则无论是否指定了 PlanarTrackTrail 对象, 从 Planar 动子都会从 MasterSyncPosition 所在的 Planar 轨道跟随主 Planar 动子。

新增版本 V3.1.10.30 - 该选项以前已有, 但类型不同: StartExternalSetpointGeneration: 用户可在“Absolute”和“Relative”模式之间进行选择。在绝对模式下, 动子只遵循用户指定的外部设定点, 并处于 ExternalSetpointGeneration 命令模式; 否则, 动子处于任何其他命令模式, 并将用户的外部设定点作为偏移量添加到当前设定点。

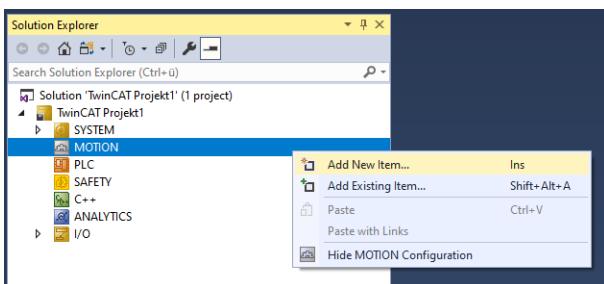
6.2 Planar 轨道

Planar 轨道是 1 个软件对象, 表示二维 XPlanar 定子表面上的(虚拟)一维路径。几个 Planar 动子可以排成一排, 在这条路径上移动。通过保持动子之间的预设值距离来防止碰撞。

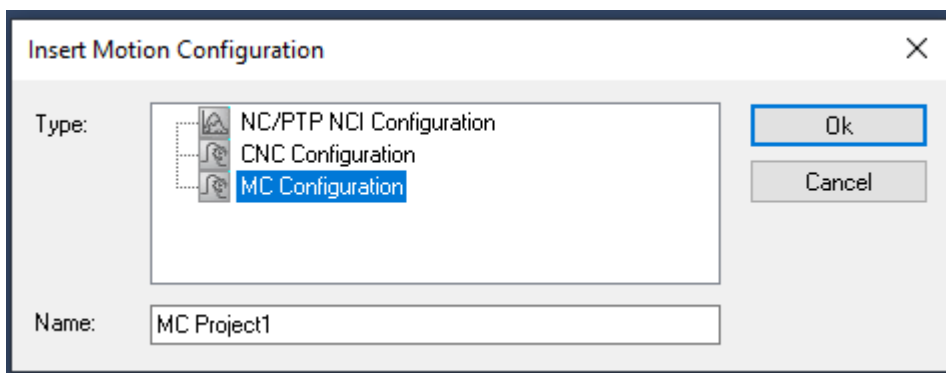
6.2.1 配置

✓ 要创建 Planar 轨道, 必须先创建 **MC Configuration**。

1. 选择 **MOTION > Add New Item...**。

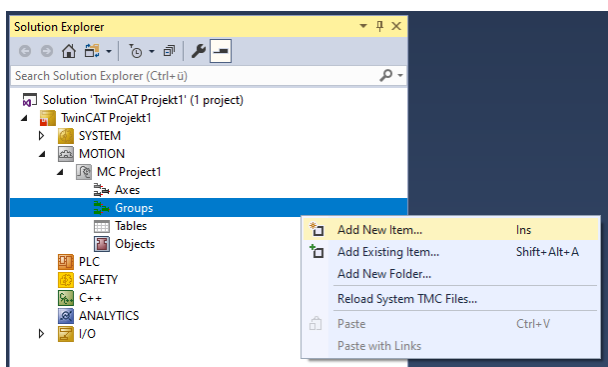


2. 在以下对话框中, 选择 **MC Configuration** 并按 **OK** 确认。

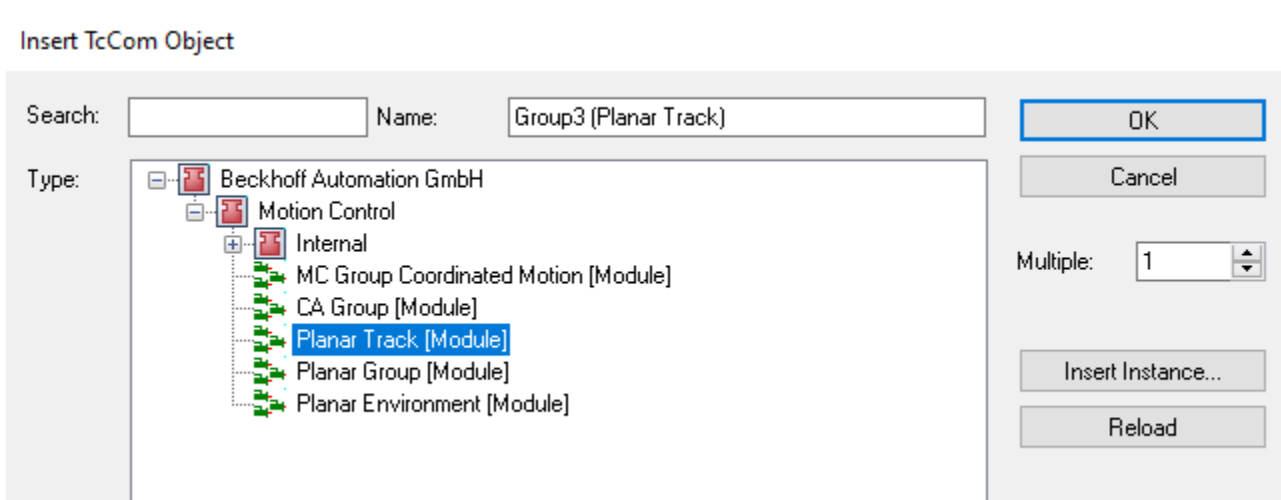


⇒ 您创建了 1 个 MC Project。

3. 选择 MC Project > Groups > Add New Item...



4. 在下面的对话框中，创建 1 个（或多个）Planar 轨道，并按 OK 确认。



⇒ Planar 轨道现已创建，可以进行参数设置。

打开详细说明

- 选择树中的 Planar 轨道并双击。

各标签的用途

对象： 一般信息（名称、类型、ID 等）可在此显示。

Object	Parameter (Init)	Parameter (Online)	Data Area
Object Id:	<input type="text" value="0x05120010"/>	<input type="checkbox"/>	Copy TMI to Target
Object Name:	<input type="text" value="Group 1 (Planar Track)"/>	<input type="checkbox"/>	Share TMC Description
Type Name:	<input type="text" value="Planar Track"/>	<input type="checkbox"/>	Keep Unrestored Link Info
GUID:	<input type="text" value="87201A49-0B3B-48DC-957C-3BC5C2E990F0"/>		
Class Id:	<input type="text" value="050300C8-0000-0000-F000-000000000064"/>		
Class Factory:	<input type="text" value="TcNc3"/>		
TMI/TMC	<input type="text" value="C:\TwinCAT\3.1\Config\Modules\TcNc3.tmc"/>		
Parent Id:	<input type="text" value="0x05100010"/>	<input type="checkbox"/>	Auto Reload TMI/TMC
Init Sequence:	<input type="text" value="PSO"/>		

参数 (Init)： 指定用户可以更改的初始参数，以影响轨道的行为。

Name	Value	CS	Unit	Type	PTCID	Comment
Maximal mover width	155.0	<input type="checkbox"/>		LREAL	0x050300B2	Maximal width for movers on the track, used for internal collision checks.
Maximal mover height	155.0	<input type="checkbox"/>		LREAL	0x050300B3	Maximal height for movers on the track, used for internal collision checks.
Check collisions against static objects	FALSE	<input type="checkbox"/>		BOOL	0x050300B4	If TRUE, collisions are also checked against static objects, i.e. tracks and environment.
Collision range at start	250.0	<input type="checkbox"/>		LREAL	0x050300B8	Distance of a mover to the start of the track, where it does not interfere anymore with other tr
Collision range at end	250.0	<input type="checkbox"/>		LREAL	0x050300BD	Distance of a mover to the end of the track, where it does not interfere anymore with other tr
Collision range mode	Automatic	<input type="checkbox"/>		MC.MC_PLANAR_TRACK_COLLISION_RANGE_MODE	0x050300D1	Either the collision range at start and end is calculated automatically when the track is enable
PartOID	00000000	<input type="checkbox"/>		OTCID	0x050300E0	Object id of the PlanarPart the track is in.
+ Geometric information		<input type="checkbox"/>	0 (Array Elements)		0x050300D7	Array containing the elements describing the geometric curve of the track.
Closed loop	FALSE	<input type="checkbox"/>		BOOL	0x05030066	Bool setting the tracks init configuration to closed loop.
+ Starts from tracks	[]	<input type="checkbox"/>	0 (Array Elements)		0x050300D8	Connection data of tracks where this track starts.
+ Ends at tracks	[]	<input type="checkbox"/>	0 (Array Elements)		0x050300D9	Connection data of tracks where this track ends.

首先设置初始参数，以便使用硬件穿越 Planar 轨道（已准备好链接）。如果轨道上的动子较大或较小，则应调整 2 个“Maximum mover width/height”参数。参数“Check collision against static objects”决定是否检查 Planar 组中的轨道与其他静态物体（轨道/定子表面边缘）的碰撞。“Collision range mode”参数决定了“Collision range at start/end”是由用户通过相应参数指定，还是由轨道内部自动计算。“Collision range”是指距轨道起点/终点的距离，从该距离考虑 Planar 动子，以避免其他轨道上的 Planar 动子发生碰撞。

自 V3.1.10.44 版起： 参数“Geometric information”“Closed loop”“Starts from tracks”和“Ends at tracks”可用于定义轨道的几何形状及其与其他轨道的互联互通。这些参数的作用与相应的 PLC 命令完全相同。

自 V3.2.60 版起： “PartOID”参数用于指定此轨道永久分配给哪个部分。如果存在唯一部件或未使用部件功能（环境未读取处理单元），则无需设置该参数。否则，必须设置该参数才能启动轨道。Init 中的所有位置和联机参数（几何[在线]信息）都在该部件系统中指定。“Starts from tracks”和“Ends at tracks”参数已经扩展，以体现与PLC关于Part部分相对应的新功能。

参数 (在线)： 显示运行时轨道的状态，例如 Planar 动子的数量或长度。

Name	Online	CS	Unit	Type	PTCID	Comment
Track length		<input type="checkbox"/>		LREAL	0x05030...	Length of the track (read only).
GroupOID		<input type="checkbox"/>		OTCID	0x05030...	Object id of the PlanarGroup the track is in, read only.
State		<input type="checkbox"/>		MC.MC_PLANAR_STATE	0x05030...	State, read only.
Operation mode		<input type="checkbox"/>		MC.MC_PLANAR_TRACK_OPERATION_MODE	0x05030...	Track state, read only.
Mover count on track		<input type="checkbox"/>		UDINT	0x05030...	Number of movers that are on this track, read only.
Moving mover count		<input type="checkbox"/>		UDINT	0x05030...	Number of movers that have requested a movement on the track, read only.
+ Geometric online information		<input type="checkbox"/>	0 (Array Elements)		0x05030...	Array containing the elements describing the geometric curve of the track.
+ Previous tracks		<input type="checkbox"/>	0 (Array Elements)		0x05030...	Array containing the object IDs of all tracks that end at this track's start vertex, read only.
+ Subsequent tracks		<input type="checkbox"/>	0 (Array Elements)		0x05030...	Array containing the object IDs of all tracks that start at this track's end vertex, read only.
Closed loop online information		<input type="checkbox"/>		BOOL	0x05030...	Bool indicating if the tracks online configuration is a closed loop.

自 V3.1.10.30 版起： 参数“Previous Tracks”和“Subsequent Tracks”是 1 个数组，包含该轨道之前或之后的所有轨道的 OID。

自 V3.1.10.44 版起： “Geometric online information”参数显示运行时可用的轨道几何图形。这源于相应的初始参数和/或使用的 PLC 命令。

从 V3.2.1 版开始：“Closed loop online information”参数用于指定轨道是否形成闭合回路（圆）。

数据区：显示轨道与 PLC 轨道通信的内存区域。

Object	Parameter (Init)	Parameter (Online)	Data Area			
	Area No	Name	Type	Size	CS	CD / Elements
+	1 (0)	McToPlc	OutputSrc	20	<input type="checkbox"/>	<input type="checkbox"/> 1 Symbols

6.2.2 轨道网络和防碰撞系统

轨道和轨道网络

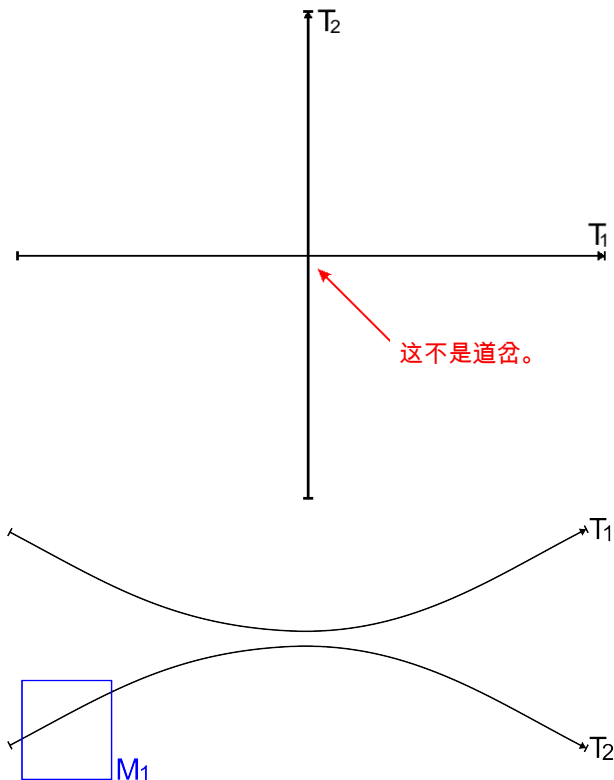
轨道是定子表面上用户指定的静态路径。多条轨道可在 1 点上连续连接（包括方向和弧度），以便动子从 1 条轨道切换到另 1 条轨道。如果有 2 条以上的轨道以这种方式在 1 点上连接，就会在那里创建 1 个道岔。这样就可以创建 1 个连续的轨道网络。

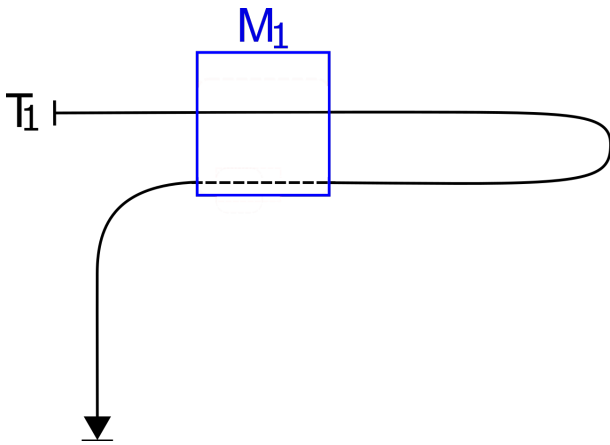
动子可以在 1 条轨道上前进和后退。只能从 1 条轨道终点移动到另 1 条轨道起点，而不能反过来。

轨道网络中的防碰撞系统

在轨道网络中移动的动子可避免与同一轨道网络中的其他动子发生碰撞。不包括无道岔的交叉轨道或彼此距离太近，或轨道自身折返的地方（见插图）。应避免这样的配置。

反面例子：





每个动子都有一个最小间隙，它必须与路径上排在它前面的动子保持最小间隙。这个间隙是在轨道上的动子位置之间测量的，可以通过每次移动命令进行重置。

在道岔附近，如有必要，动子还必须注意与位于与道岔连接的其他轨道上的动子可能发生的碰撞，即使这些轨道不属于动子的计划路径。在某个时间点，这种额外的防撞措施对动子是否有效取决于 4 个因素：

- 动子的当前位置，
- 动子可能的最早静止位置（由当前动态和动态限值产生），
- 动子的固定间隙，
- 当前轨道的相应 Collision Range 参数。

如果动子当前位置与最早可能的静止位置之间的距离小于设定的动子间隙+道岔防撞范围，则该动子的附加防撞功能处于激活状态。如果是这种情况，所有同时满足这一条件的其他动子都会被纳入动态规划。

碰撞范围的定义

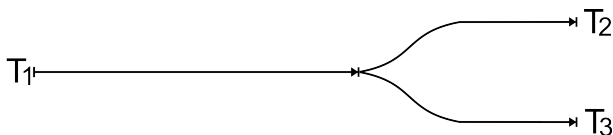
碰撞范围参数对避免碰撞的重要性已在上一节中说明。“Collision Range at start”指距轨道道岔起点的距离，“Collision Range at end”指距轨道道岔终点的距离。

对 Collision Range 参数更直观的理解源于以下建议：Collision Range 的设置应使在此道岔位置的动子不能与相连接轨道上的动子发生碰撞。

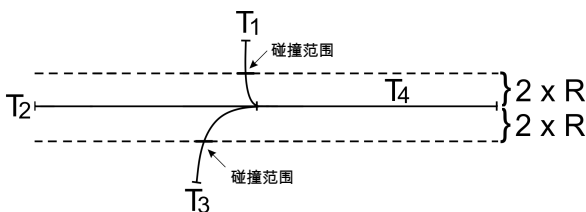
为了简化参数配置，当“Collision Range Mode”参数设置为“Automatic”时，Collision Range 的相应值会自动计算并应用。如果选择“Manual”而不是“Automatic”，则会使用用户输入的值。如果设置得过小，可能会导致碰撞。反之，如果将它们设置得太大，动子可能会在实际上相距甚远且不可能发生碰撞的不同轨道上相互阻挡。

如果起点（终点）处的轨道没有道岔，或者道岔处没有其他轨道起点（终点），则可将相应的 Collision Range 设置为 0。

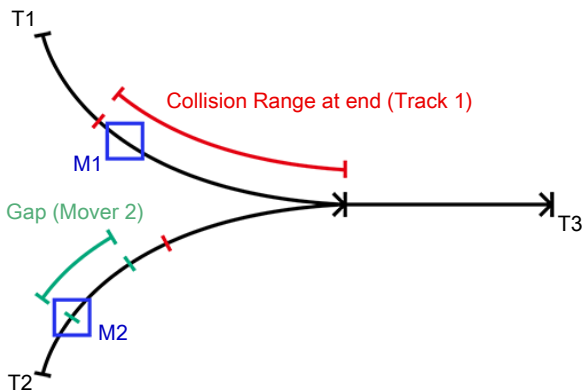
实例和插图：



在此示例中，轨道 1 的“Collision range at end”可以设置为零，因为虽然有 2 条轨道从道岔处开始，但没有其他轨道结束。轨道 2 和轨道 3 的“Collision range at start”参数应设置为：与道岔有此距离的动子不能与其他相应轨道上的动子发生碰撞。

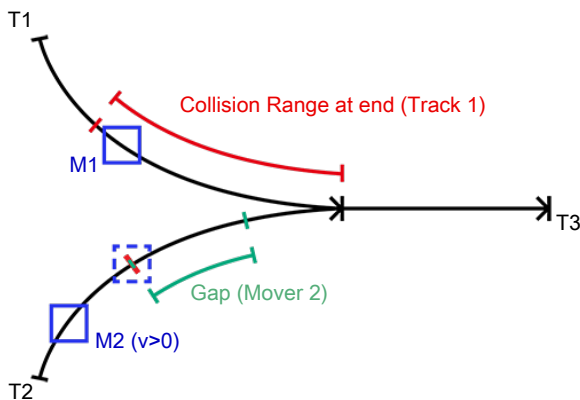


确定有意义的 Collision Range 参数示例 (T1、T2 和 T3 在 T4 开始时结束)：如果 R 是轨道上动子的最大移动半径，则可以在轨道周围 (本例中为轨道 2 周围) 放置半径为 $2 \cdot R$ 的“hose”，以确定其他轨道上 Collision Range 的最小值。在此示例中，轨道 1 的“Collision range at end”较小，因为它迅速远离其他轨道；轨道 3 和轨道 2 的“Collision range at end”较大，因为它们靠近运行的时间较长。

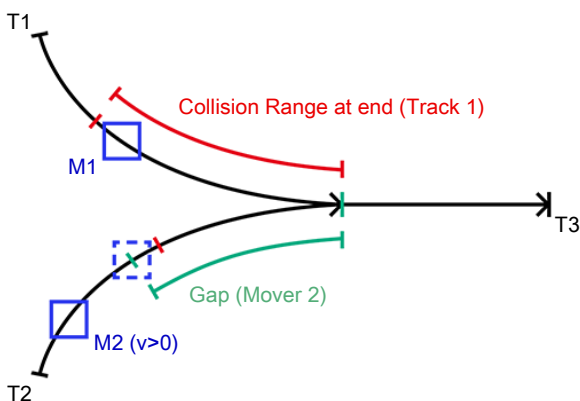


在此示例中，由于它与道岔的距离小于设定的碰撞范围，因此交换机的额外防碰撞功能对动子 1 起作用。

在此示例中，动子 2 处于静止状态，距离道岔比间隙或碰撞范围更远。因此，额外的防撞系统未启用，2 个动子此时不必考虑对方。

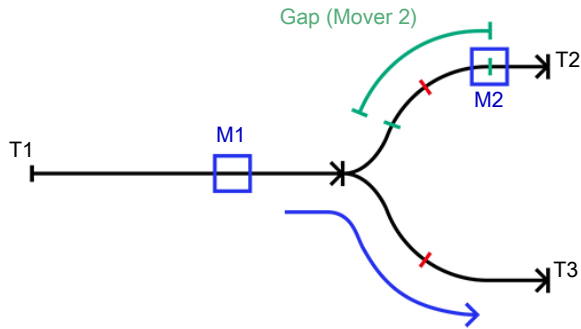


在此示例中，动子 1 位于 Collision Range 内的轨道 1 上，因此动子 1 阻止了动子 2 向轨道 3 的移动。动子 2 恰好停在轨道 2 Collision Range 的起点，因为这是最后一个安全停止点。如果动子 1 和动子 2 之间的间隙允许，动子 2 就会移动到相应较晚的停止点。

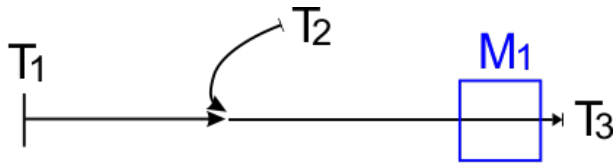


在此示例中，动子 1 位于 Collision Range 内的轨道 1 上，因此动子 1 阻止了动子 2 向轨道 3 的移动。动子 2 恰好停在其间隙到轨道 2 末端的距离上。如果动子 1 和动子 2 之间的间隙允许，动子 2 就会移动到相应较晚的停止点。

在最后 2 个例子中，当动子 1 向前移动了很远，以至于 2 个动子的最小距离都大于 2 个动子之间的间隙时，动子 2 继续前进。



在此示例中，动子 2 与道岔的距离大于 Gap 或 Collision Range，因此动子 1 可以不受阻碍地驶入轨道 3。若动子 2 向后移动，如果到道岔的距离小于 Gap 或 Collision Range，则可能发生堵塞。



这是应避免的设计示例，其中一条轨道（在本例中为 T2）的末端会影响另一条轨道（T3）start 的 Collision Range（反之亦然）。在 Automatic Collision Range Mode 下，不会检测到这种情况。但是，如果仍有需要，则需要手动调整 Collision Range。然而，由于动力学的强大限值（即使以低速驶过，大曲率轨迹也会产生巨大的离心力），像本例中的 T2 这样的大曲率轨道也是极不推荐的。

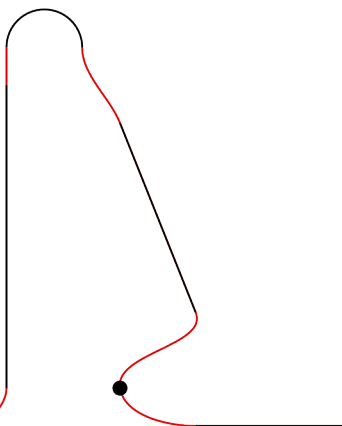
6.2.3 几何定义

Planar 轨迹利用相连直线、弧线和点逐步定义。通过 C2-连续插值（“混合”）自动完成这些元素之间的连接。

- 直线通过起点和终点定义，请参见 [AppendLine](#)。
- 弧线由起点、终点和中心点，以及指定顺时针或逆时针方向而定义，请参见 [AppendCircle](#)。
- 位置只能通过点来定义，请参见 [AppendPosition](#)。

示例

在下方示例中，自动生成的轨道部分显示为红色。



```
PROGRAM MAIN
VAR
  track : MC_PlanarTrack;
  pos1, pos2 : PositionXYC;
  center : PositionXY;
  state : UDINT;
END_VAR
```

```
CASE state OF
0:
  pos1.SetValuesXY(0, 0);
  pos2.SetValuesXY(400, 0);
  track.AppendLine(0, pos1, pos2);
  pos1.SetValuesXY(450, 50);
  pos2.SetValuesXY(450, 450);
  track.AppendLine(0, pos1, pos2);
  pos1.SetValuesXY(450, 500);
  pos2.SetValuesXY(550, 500);
  center.SetValuesXY(500, 500);
  track.AppendCircle(0, pos1, pos2, center, TRUE);
  pos1.SetValuesXY(600, 400);
  pos2.SetValuesXY(700, 150);
  track.AppendLine(0, pos1, pos2);
  pos1.SetValuesXY(600, 50);
  track.AppendPosition(0, pos1);
  pos1.SetValuesXY(700, 0);
  pos2.SetValuesXY(900, 0);
  track.AppendLine(0, pos1, pos2);
END_CASE
```

6.2.4 轨道和部件

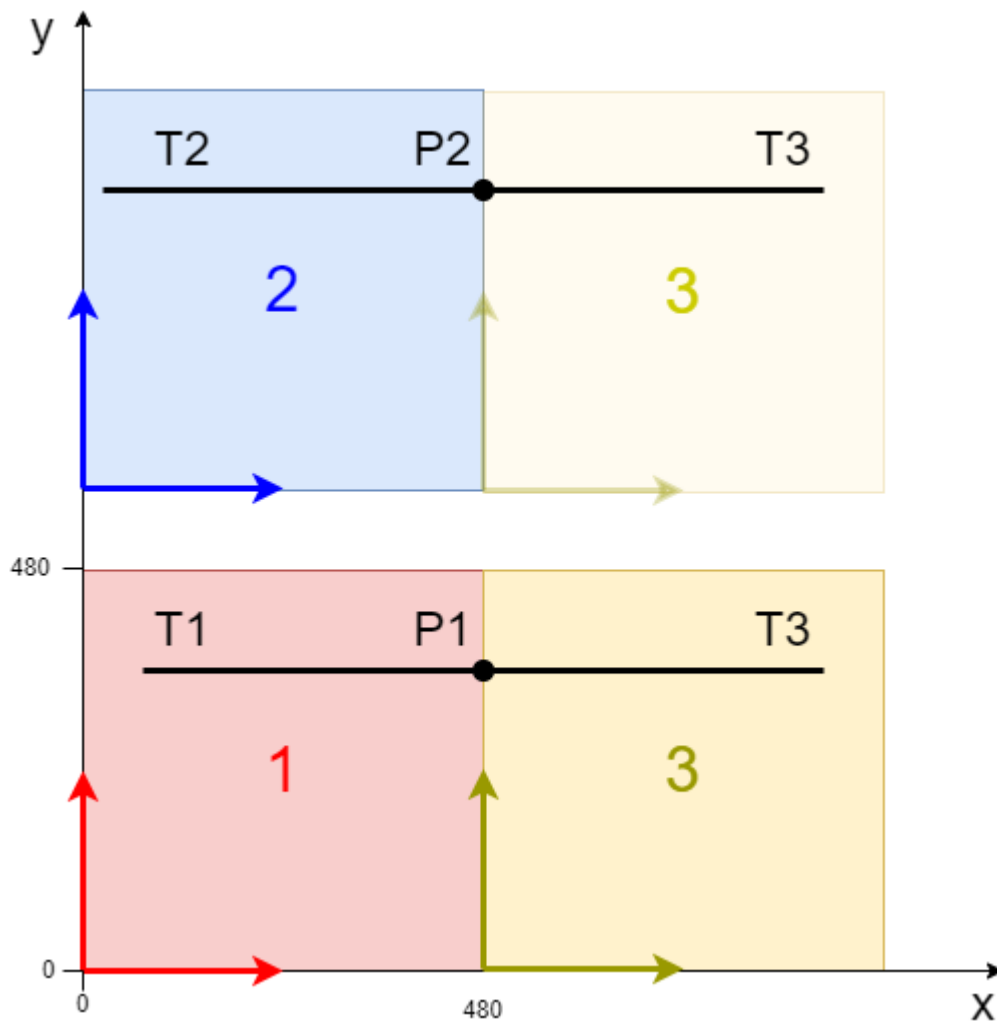
从 V3.2.60 版开始：可使用 Part 功能，这也是本节的主题。

轨道可与部件一起使用，但有一些特殊功能需要注意：

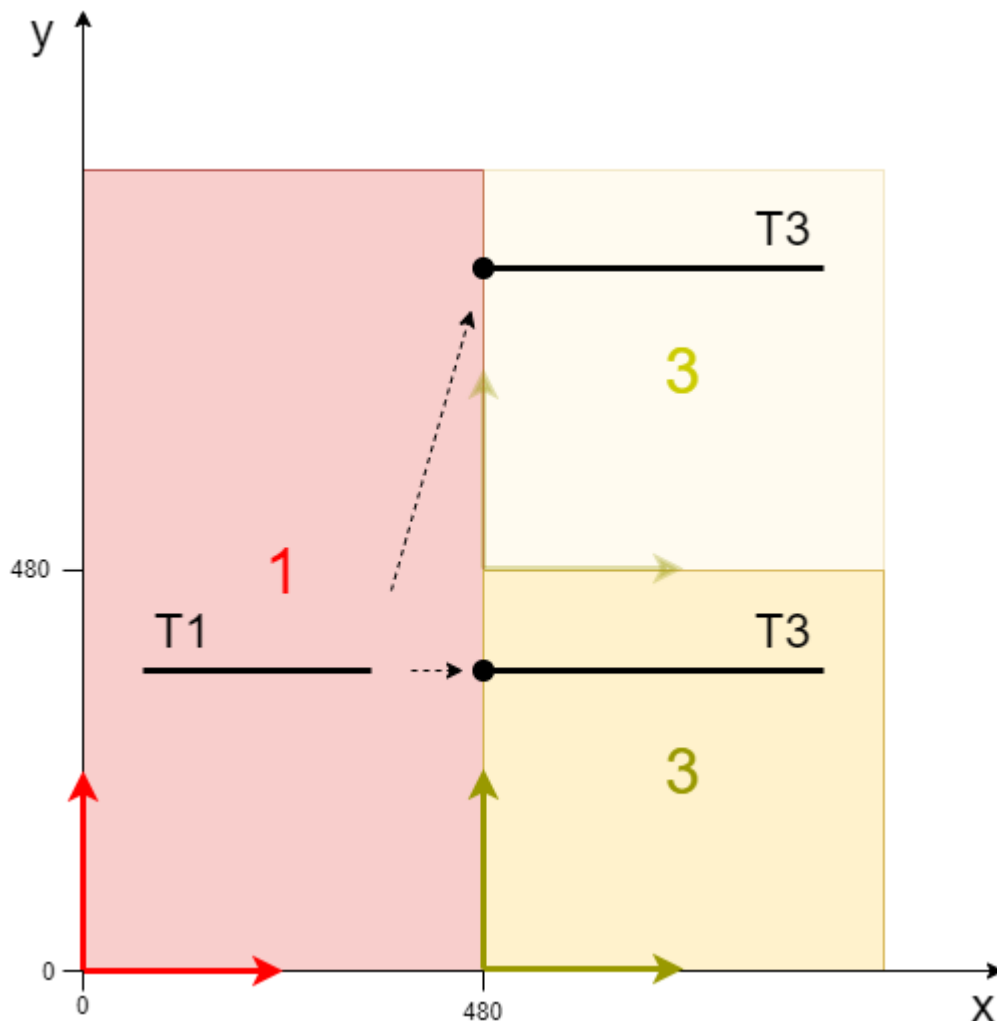
- 1 个轨道总是永久分配给 1 个部件。这可以通过初始参数“PartOID”来实现。
- 轨道必须在相应部件上几何形状完整。
- 轨道相对于其部件具有固定的静态几何形状。因此，如果他的部分位置发生变化，轨道的位置也会相应改变。
- 如果配置中存在多个部分，则必须指定初始参数“PartOID”，否则（每个）轨道将自动分配给唯一的部分。
- 同一部件上的轨道可轻松实现互联互通。每个轨道的起点和终点只能与另一个轨道互联互通 1 次。这意味着 [StartFromTrack \[▶_165\]](#) 和 [EndAtTrack \[▶_165\]](#) 在每个轨道上最多只能成功调用 1 次。
- 只有当不同部分的轨道之间的互联互通正好位于 2 个部分的边界上时，才能将它们连接起来。这意味着 2 条轨道中的 1 条轨道的起点或终点必须已经在部件边界上，这样另 1 条轨道才能互联互通。只有当 2 部分都在这一点上时，才能互联互通。如果将其中 1 个部件移动到不同的位置，则无法再进行互联互通（就像轨道的任何开放端口一样）。但在这种情况下，这 2 条轨道可以在其他位置与其他部件上的其他轨道互联互通。



为了关闭轨道之间所有这些不同的互联互通，可以为每个轨道多次调用 [StartFromTrack \[▶_165\]](#) 和 [EndAtTrack \[▶_165\]](#) 方法。



在本例中，第 1 和第 2 部分是静态的。第 3 部分有 2 个位置，因此它一次与第 1 部分互联互通，一次与第 2 部分互联互通。一旦在第 1 和第 2 部分上定义了轨道 1 和轨道 2，第 3 部分上的轨道 3 就可以同时与这 2 个部分互联互通。根据轨道的方向，轨道 3 必须通过 2 次 `StartFromTrack` [► 165] 调用或 2 次 `EndAtTrack` [► 165] 调用与位置 1 和 2（或轨道 1 和 2）互联互通。第 1 次调用定义了轨道 3 起点或终点的几何形状和逻辑互联互通，而第 2 次调用只定义了逻辑互联互通，并且只假定（并检查）几何形状的互联互通是合适的。



在这个例子中，部件 1 是静态的，而部件 3 有 2 个位置，因此它有与部件 1 不同的接触点。定义轨道 3 后，如何解释从轨道 1 到轨道 3 的 `EndAtTrack` [▶_165] 尚不清楚。轨道 1 与轨道 3 的几何互联互通应该是上部配置还是下部配置？这可以通过新方法 `EndAtTrackAdvanced` [▶_167] 和 `StartFromTrackAdvanced` [▶_166] 来实现。为了以这种配置互联互通轨道，需要指定所涉及轨道 2 部分的确切位置。

6.2.5 示例“Planar 动子加入并在轨道上移动”

通过本指南，您将创建 1 个包含 2 个 Planar 动子和 1 个 Planar 轨道的 TwinCAT 项目。2 个动子加入并在轨道上移动。

创建 Planar 动子

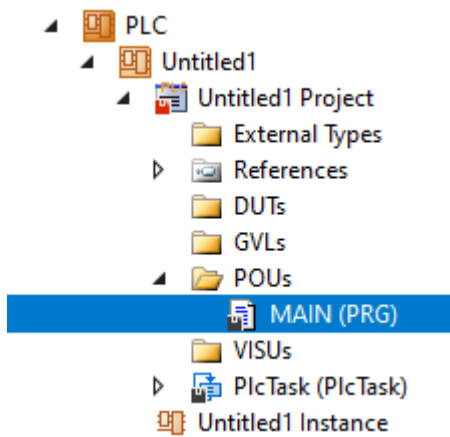
- ✓ 参阅 [配置 \[▶_16\]](#)。
- 1. 创建 2 个 Planar 动子。
- 2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。
- 3. 将第 2 个动子的起始位置改为 $x = 240$ 。

创建 Planar 轨道

- 4. 通过 `Groups > Add New Item...` 添加 Planar 轨道，请参阅 [配置 \[▶_39\]](#)。

创建 PLC

- ✓ 请参阅 [创建 PLC \[▶_19\]](#) 下的初步步骤。
- 1. 通过 `MAIN` 创建所需数量的动子 (“MC_PlanarMover”) 和轨道 (“MC_PlanarTrack”)。



⇒ 这些代表 MC Configuration 中的动子和轨道。

2. 如下图所示，创建 2 个 Planar 动子、1 个 Planar 轨道、1 个状态机的任意状态变量以及轨道的 2 个辅助位置。

```
PROGRAM MAIN
VAR
    mover_one, mover_two : MC_PlanarMover;
    track : MC_PlanarTrack;
    state : UDINT;
    pos1, pos2 : PositionXYC;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 该程序代码创建并激活 1 个轨道和 2 个动子。之后，2 个动子加入并在轨道上移动。

```
CASE state OF
0:
    pos1.SetValuesXY(0, 0);
    pos2.SetValuesXY(400, 0);
    track.AppendLine(0, pos1, pos2);
    track.Enable(0);
    state := 1;
1:
    IF track.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    mover_one.Enable(0);
    mover_two.Enable(0);
    state := 3;
3:
    IF mover_one.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled
    AND mover_two.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 4;
    END_IF
4:
    mover_one.JoinTrack(0, track, 0, 0);
    mover_two.JoinTrack(0, track, 0, 0);
    state := 5;
5:
    IF mover_one.MCTOPLC.STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack
    AND mover_two.MCTOPLC.STD.CommandMode=MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
        state := 6;
    END_IF
6:
    mover_one.MoveOnTrack(0, 0, 150.0, 0, 0);
    mover_two.MoveOnTrack(0, 0, 350.0, 0, 0);
    state := 7;
7:
    IF mover_one.MCTOPLC.SETONTRACK.SetPos >= 149.9
    AND mover_two.MCTOPLC.SETONTRACK.SetPos >= 349.9 THEN
        state := 8;
    END_IF
END_CASE
```

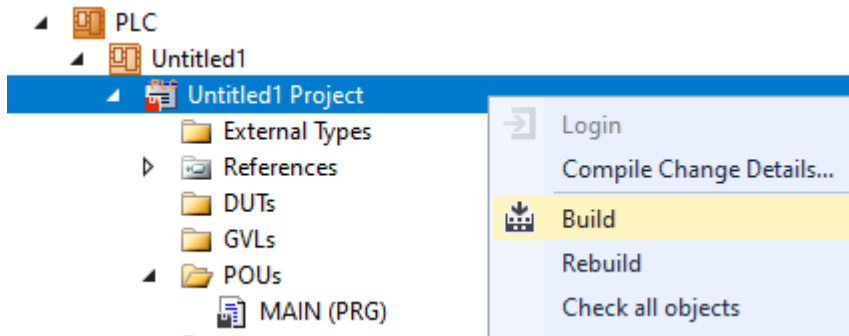
发送命令

4. 要发送命令，必须在 END_CASE 之后循环调用动子和轨道的更新方法：

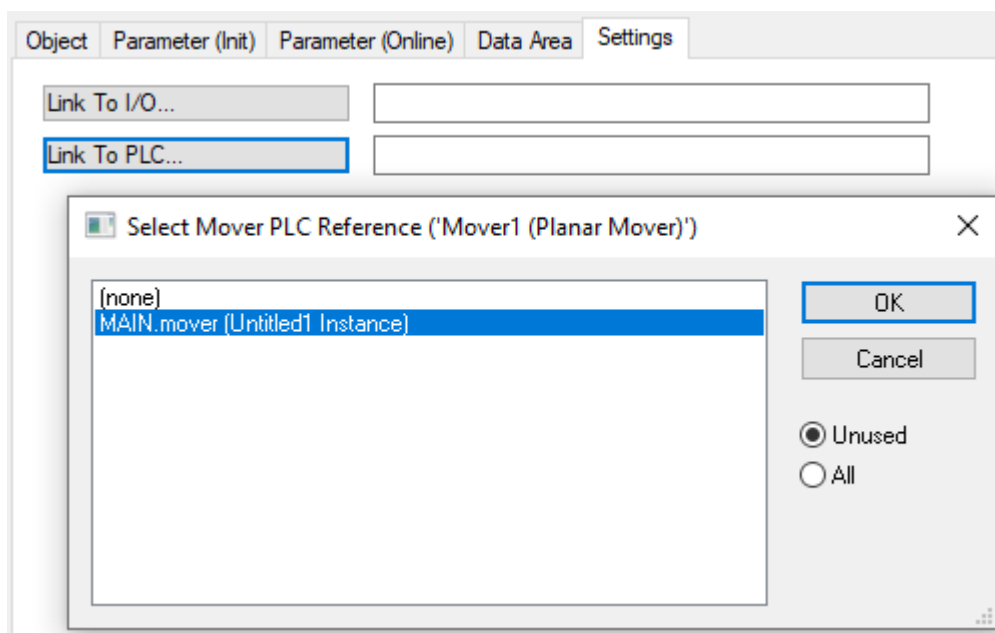
```
mover_one.Update();
mover_two.Update();
track.Update();
```

构建 PLC 时会创建“PLC mover”和“Track”的符号，然后将其链接到 MC Project 中的轴和轨道实例。

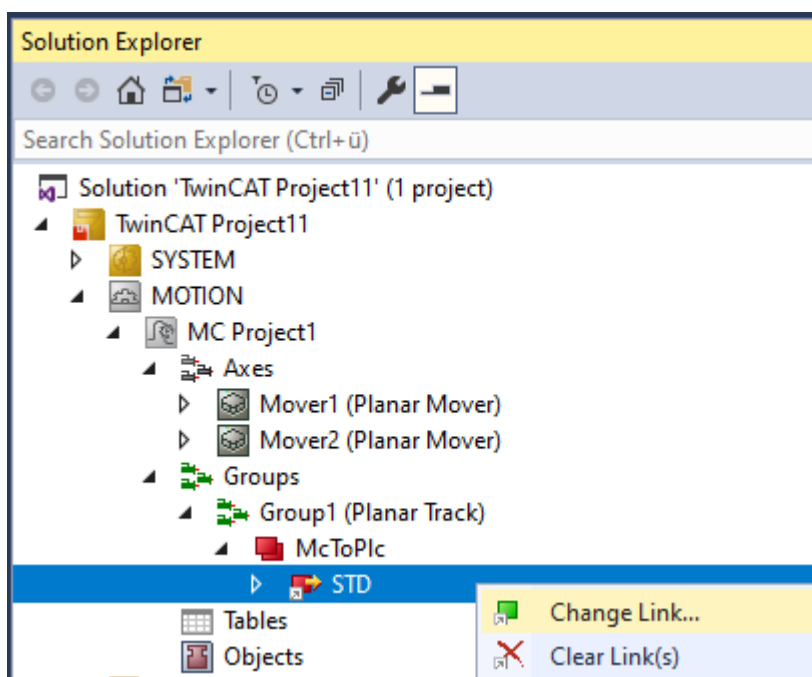
1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。

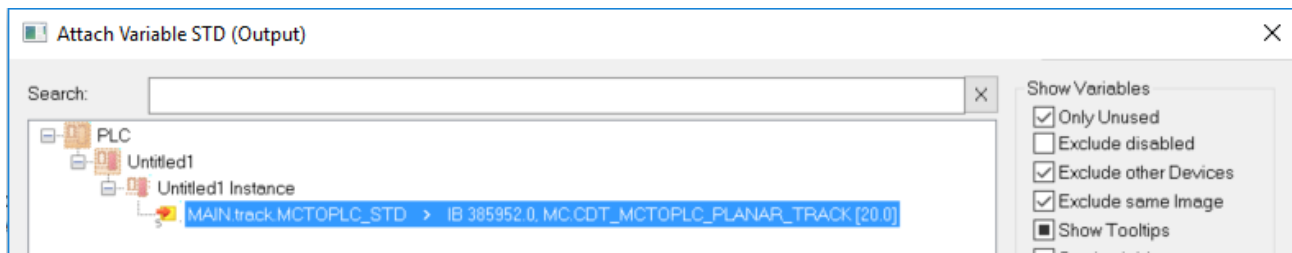


⇒ 随后，“MC Project”中的 Planar 轴子可通过 **Setting** 选项卡上的 **Link To PLC...** 按钮进行链接。



⇒ 必须通过以下对话框单独链接轨道。





激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机结束时（状态=8），动子处于所需的位置。

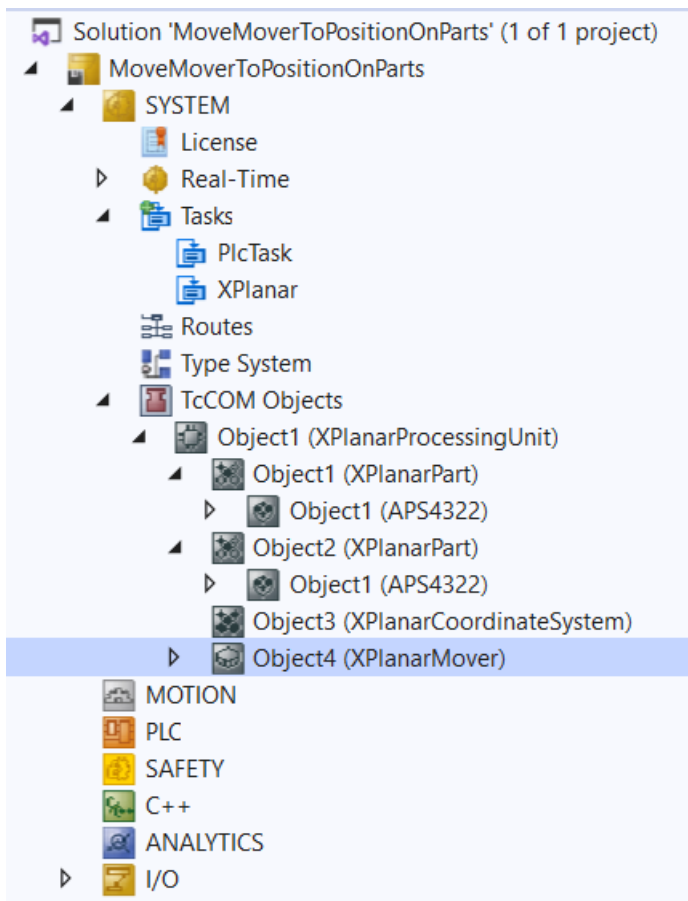
Expression	Type	Value	Prepared value	Address	Comment
mover_one	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLANAR_M...			%Q*	Mover data that is tra...rred from the Plana...
MCTOPLC	CDT_MCTOPLC_PLANAR_M...			%I*	Mover data that is tra...rred from the Plana...
STD	REFERENCE TO CDT_MCTO...			%IB*	Mover standard data t...is transferred from ...
SET	REFERENCE TO CDT_MCTO...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	149.9999999...			X coordinate.
y	LREAL	0			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	0			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	66246393761...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT_MCTO...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT_MCTO...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT_MCTO...			%IB*	Mover busy informatio...at is transferred fro...
Error	BOOL	FALSE			Flag indicating a PlanarMover error.
ErrorId	UDINT	0			Error id indicating the PlanarMover error type.
mover_two	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLANAR_M...			%Q*	Mover data that is tra...rred from the Plana...
MCTOPLC	CDT_MCTOPLC_PLANAR_M...			%I*	Mover data that is tra...rred from the Plana...
STD	REFERENCE TO CDT_MCTO...			%IB*	Mover standard data t...is transferred from ...
SET	REFERENCE TO CDT_MCTO...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	349.9999999...			X coordinate.
y	LREAL	0			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	0			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	66246393761...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT_MCTO...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT_MCTO...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT_MCTO...			%IB*	Mover busy informatio...at is transferred fro...
Error	BOOL	FALSE			Flag indicating a PlanarMover error.

6.2.6 示例“在多 Planar 部件的轨道上移动 Planar 动子”

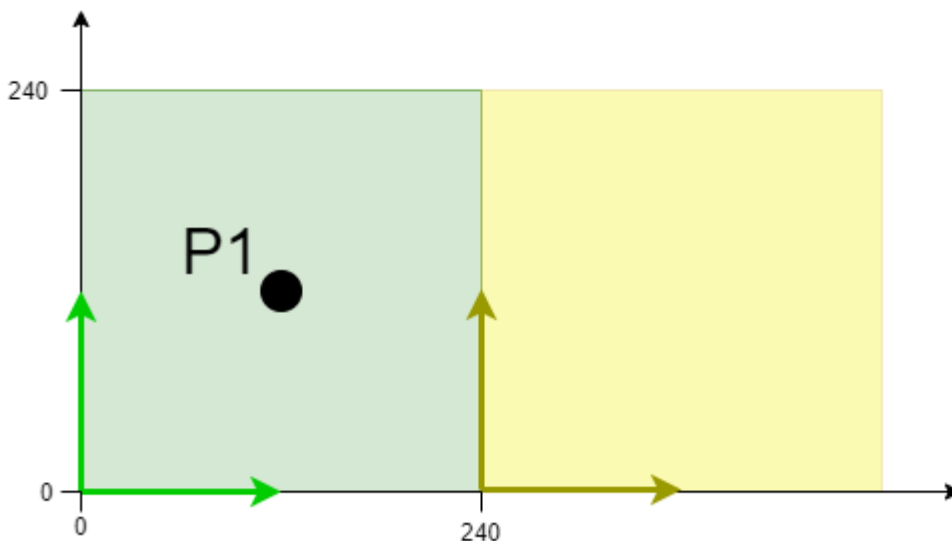
在这个例子中，1 个 Planar 动子在 2 个 Planar 部件上的 2 个 Planar 轨道上移动。

起点

您可以从包含 1 个完全配置好的 XPlanar Processing Unit 的解决方案开始。在 XPlanar Processing Unit 下创建了 2 个部件，1 个坐标系和 1 个动子。2 个部件下各创建 1 个平面模块。



设定的几何情况如下：2 个部件相邻，动子从左侧部件的中间位置开始（位置 P1）。这两个部件都是不可移动的，因此配置是静态的。



本示例就是在这种配置的基础上开发的。



XPlanar Processing Unit 的文档中介绍了初始化的步骤。

创建 Planar 动子、Planar 轨道和 Planar 环境

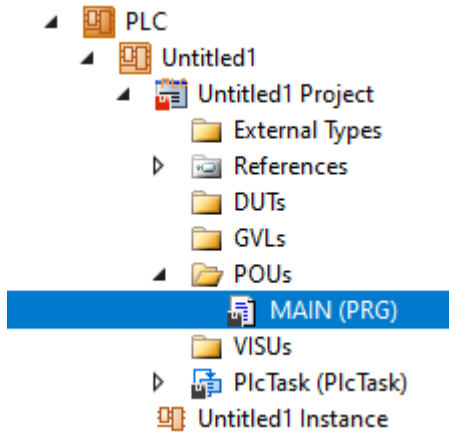
1. 要为本示例创建 Planar 动子，请参见 [配置 \[► 16\]](#)。
2. 要创建 Planar 环境，请参见 [配置 \[► 94\]](#)。

3. 将初始参数 XPlanar Processing Unit OID 设置为 XPlanar Processing Unit 的对象 ID。这样可以激活所有 MC Configuration 对象（尤其是已创建的 Planar 动子）的 Part 功能。
4. 通过 Groups > Add New Item... 添加 2 条 Planar 轨道，请参见 配置 [▶ 39]。
5. 将 2 个轨道的初始参数“PartOID”设置为相应的部件；在本例中，第一个轨道设置为部件 1，第二个轨道设置为部件 2。

创建 PLC

✓ 请参阅 创建 PLC [▶ 19] 下的初步步骤。

1. 通过 MAIN 创建所需数量的动子 (“MC_PlanarMover”) 和轨道 (“MC_PlanarTrack”)。



⇒ 这些代表 MC Configuration 中的动子和轨道。

2. 如下图所示，创建 1 个 Planar 动子、2 条 Planar 轨道、1 个状态机的任意状态变量以及轨道的 2 个辅助位置。

```
PROGRAM MAIN
VAR
    mover : MC_PlanarMover;
    track_one, track_two : MC_PlanarTrack;
    state : UDINT;
    pos1, pos2 : PositionXYC;
END_VAR
```

3. 然后在 MAIN 中设计 1 个序列。

⇒ 该程序代码创建并激活 2 条轨道和动子。然后，动子与第一条轨道耦合，驶入第二条轨道，跨越部件 1 和部件 2 之间的边界。

```
CASE state OF
0:
    pos1.SetValuesXYCReferenceId(40, 120, 0, 16#01010060);
    pos2.SetValuesXYCReferenceId(240, 120, 0, 16#01010060);
    track_one.AppendLine(0, pos1, pos2);
    track_two.StartFromTrack(0, track_one);
    pos1.SetValuesXYCReferenceId(260, 120, 0, 16#01010060);
    pos2.SetValuesXYCReferenceId(440, 120, 0, 16#01010060);
    track_two.AppendLine(0, pos1, pos2);
    track_one.Enable(0);
    track_two.Enable(0);
    state := 1;
1:
    IF track_one.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled AND
    track_two.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    mover.Enable(0);
    state := 3;
3:
    IF mover.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 4;
    END_IF
4:
    mover.JoinTrack(0, track_one, 0, 0);
    state := 5;
5:
    IF mover.MCTOPLC_STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
        state := 6;
```

```

        END_IF
    6:
        mover.MoveOnTrack(0, track_two, 150.0, 0, 0);
        state := 7;
END_CASE

```

发送命令

4. 要发送命令，必须在 END_CASE 之后循环调用动子和轨道的更新方法：

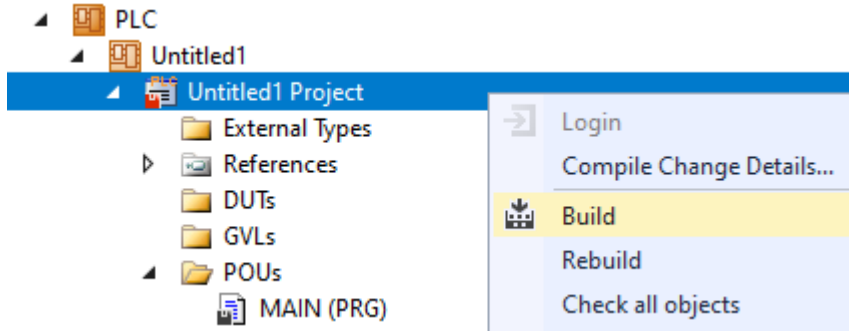
```

mover.Update();
track_one.Update();
track_two.Update();

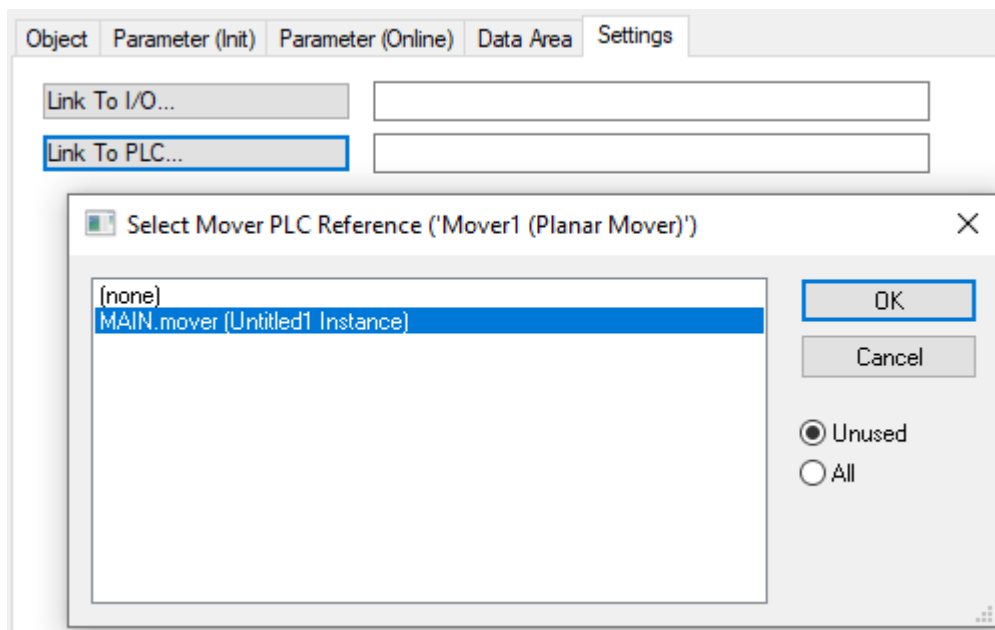
```

构建 PLC 时会创建“PLC mover”和“Track”的符号，然后将其链接到 MC Project 中的动子和轨道实例。

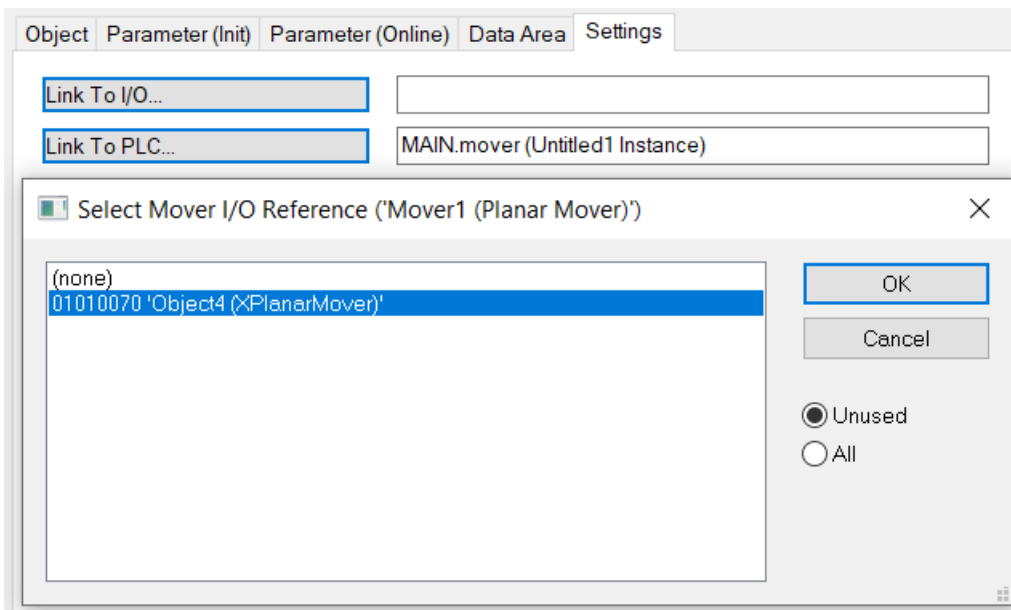
1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



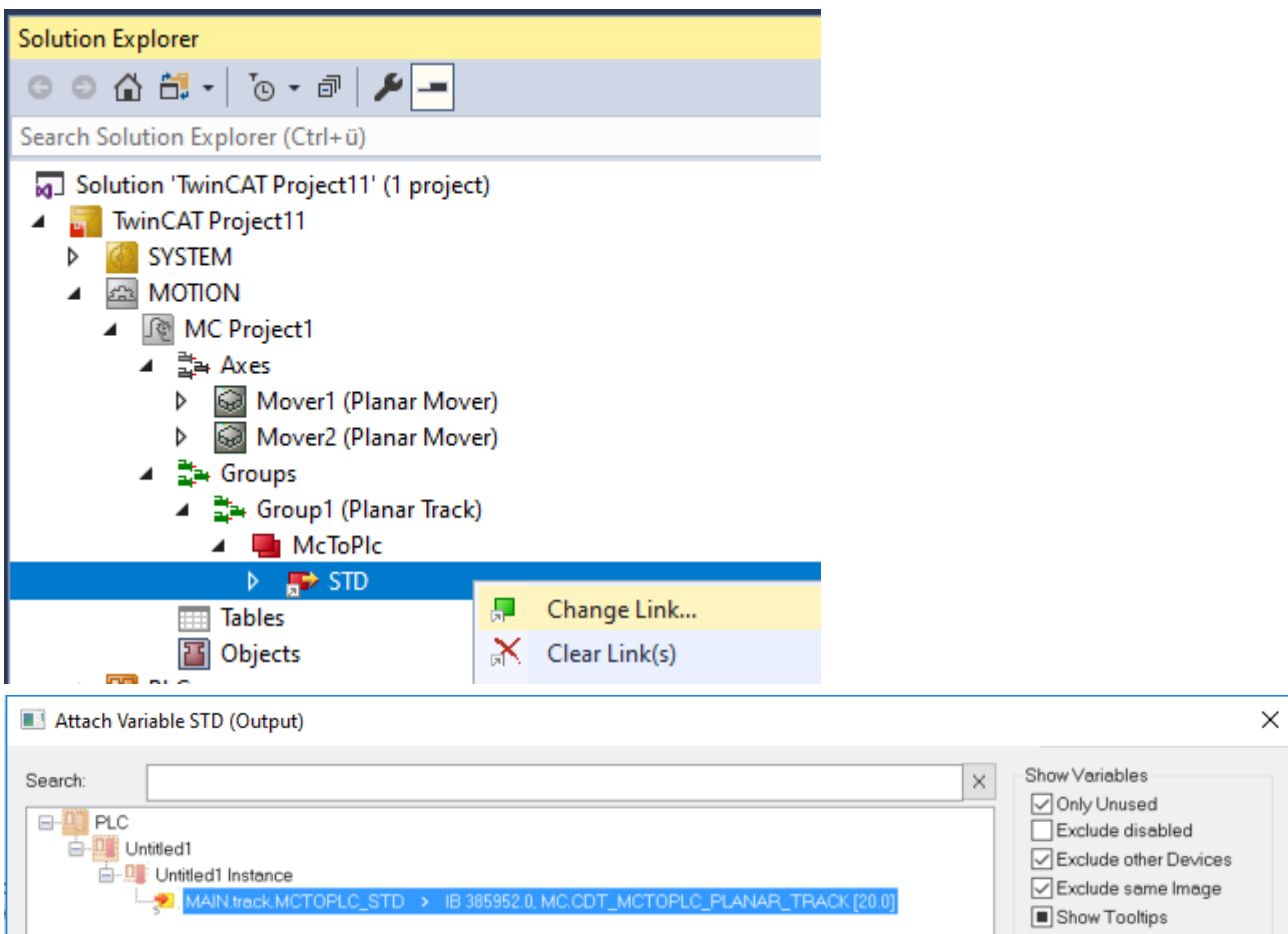
⇒ 随后，“MC Project”中的 Planar 动子可通过 **Setting** 选项卡上的 **Link To PLC...** 按钮进行链接。



⇒ 此外，“MC Project”（双击）中的 Planar 动子可以通过 **Settings** 选项卡上的 **Link To I/O...** 按钮进行链接。



必须通过以下对话框分别链接对应轨道。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。

4. 通过菜单栏中的 Play 按钮启动 PLC。

动子在状态机走到最后一步时 (state=7) 位于部件 2 的第二条轨道上。AppendLine 命令的位置是在全局坐标系中指定的，动子的终点位置也是如此。

Expression	Type	Value
mover	MC_PlanarMover	
PLCTOMC	CDT_PLCTOMC...	
MCTOPLC	CDT_MCTOPLC...	
STD	REFERENCE TO...	
SET	REFERENCE TO...	
SetPos	MoverVector	
x	LREAL	390.0000000...
y	LREAL	120
z	LREAL	2.013173942...
a	LREAL	-0.01372402...
b	LREAL	0.010024976...
c	LREAL	0
SetVelo	MoverVector	
SetAcc	MoverVector	
DcTimeStamp	ULINT	7562236335...
PhysicalAreaID	UDINT	16842848
ACT	REFERENCE TO...	
COORDMODE	REFERENCE TO...	
SETONTRACK	REFERENCE TO...	
Error	BOOL	FALSE
ErrorId	UDINT	0
track_one	MC_PlanarTrack	
track_two	MC_PlanarTrack	
state	UDINT	7
pos1	PositionXYC	
pos2	PositionXYC	

6.2.7 示例“将 Planar 动子耦合到轨道上，并在 CRotationOnTrack 模式下移动”

通过本指南，您将创建 1 个包含 2 个 Planar 动子和 1 个 Planar 轨道的 TwinCAT 项目。2 个动子加入并在轨道上移动。

创建 Planar 动子

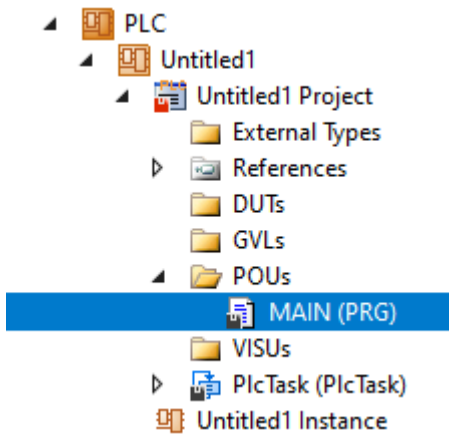
- ✓ 请参见 配置 [▶ 16]。
- 1. 创建 2 个 Planar 动子。
- 2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。
- 3. 将第二个动子的起始位置改为 x=240。

创建 Planar 轨道

- 4. 通过 Groups > Add New Item... 添加 Planar 轨道，请参见 配置 [▶ 39]。

创建 PLC

- ✓ 请参阅 创建 PLC [▶ 19] 下的初步步骤。
- 1. 通过 MAIN 创建所需数量的动子 (“MC_PlanarMover”) 和轨道 (“MC_PlanarTrack”)。



⇒ 这些代表 MC Configuration 中的动子和轨道。

2. 如下图所示，创建 2 个 Planar 动子、1 个 Planar 轨道、1 个状态机的任意状态变量以及轨道的 2 个辅助位置。

```
PROGRAM MAIN
VAR
    mover_one, mover_two : MC_PlanarMover;
    track : MC_PlanarTrack;
    state : UDINT;
    pos1, pos2 : PositionXYC;
    join_track_options : ST_JoinTrackOptions;
END_VAR
```

3. 然后在 MAIN 中设计 1 个序列。

⇒ 该程序代码创建并激活 1 个轨道和 2 个动子。然后，2 个动子与轨道耦合并旋转。

```
CASE state OF
0:
    pos1.SetValuesXY(0, 0);
    pos2.SetValuesXY(400, 0);
    track.AppendLine(0, pos1, pos2);
    track.Enable(0);
    state := 1;
1:
    IF track.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    mover_one.Enable(0);
    mover_two.Enable(0);
    state := 3;
3:
    IF mover_one.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled
    AND mover_two.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 4;
    END_IF
4:
    join_track_options.useOrientation := FALSE;
    mover_one.JoinTrack(0, track, 0, join_track_options);
    mover_two.JoinTrack(0, track, 0, join_track_options);
    state := 5;
5:
    IF mover_one.MCTOPLC.STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack
    AND mover_two.MCTOPLC.STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
        state := 6;
    END_IF
6:
    mover_one.MoveC(0, 20.0, 0, 0);
    mover_two.MoveC(0, 90.0, 0, 0);
    state := 7;
7:
    IF mover_one.MCTOPLC.SET.SetPos.c >= 19.9
    AND mover_two.MCTOPLC.SET.SetPos.c >= 89.9 THEN
        state := 8;
    END_IF
END_CASE
```

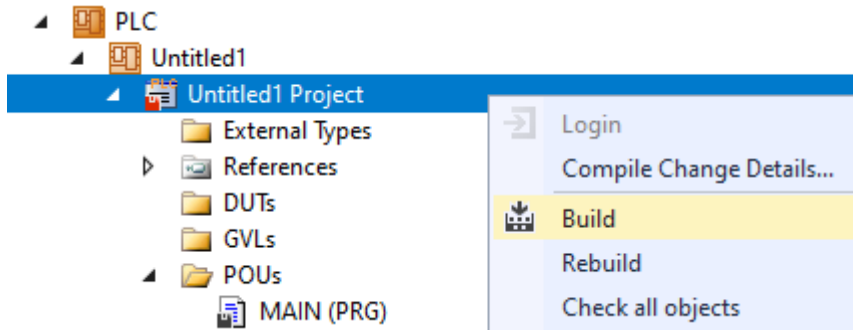
发送命令

4. 要发送命令，必须在 END_CASE 之后循环调用动子和轨道的更新方法：

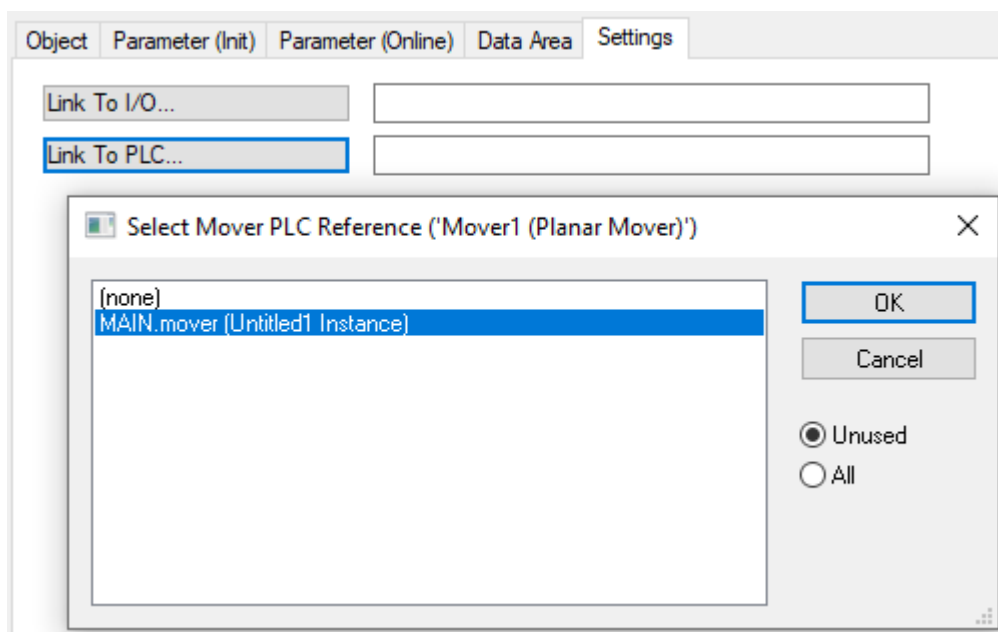
```
mover_one.Update();
mover_two.Update();
track.Update();
```

构建 PLC 时会创建“PLC mover”和“Track”的符号，然后将其链接到 MC Project 中的动子和轨道实例。

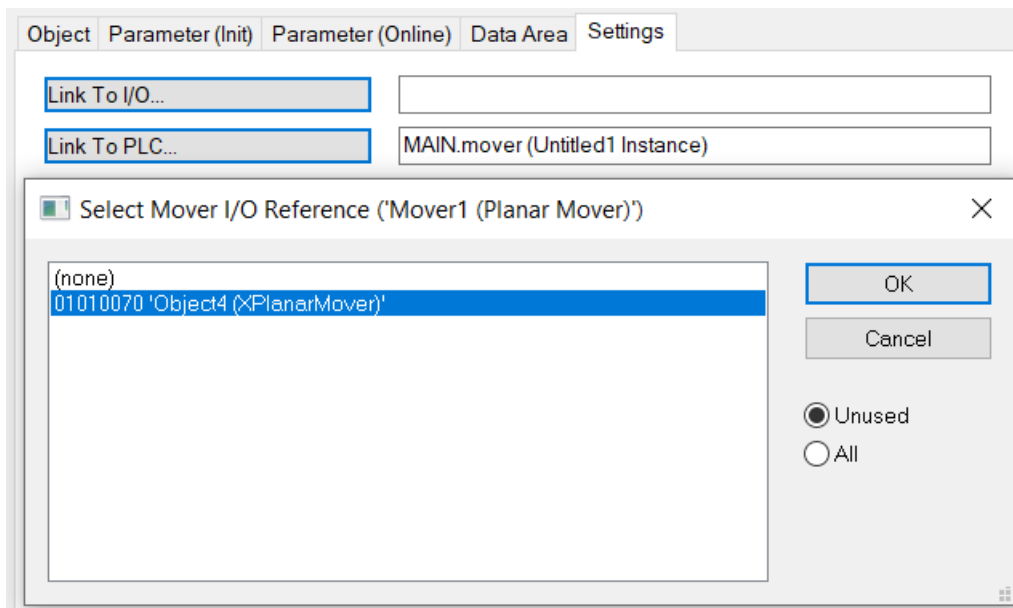
1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



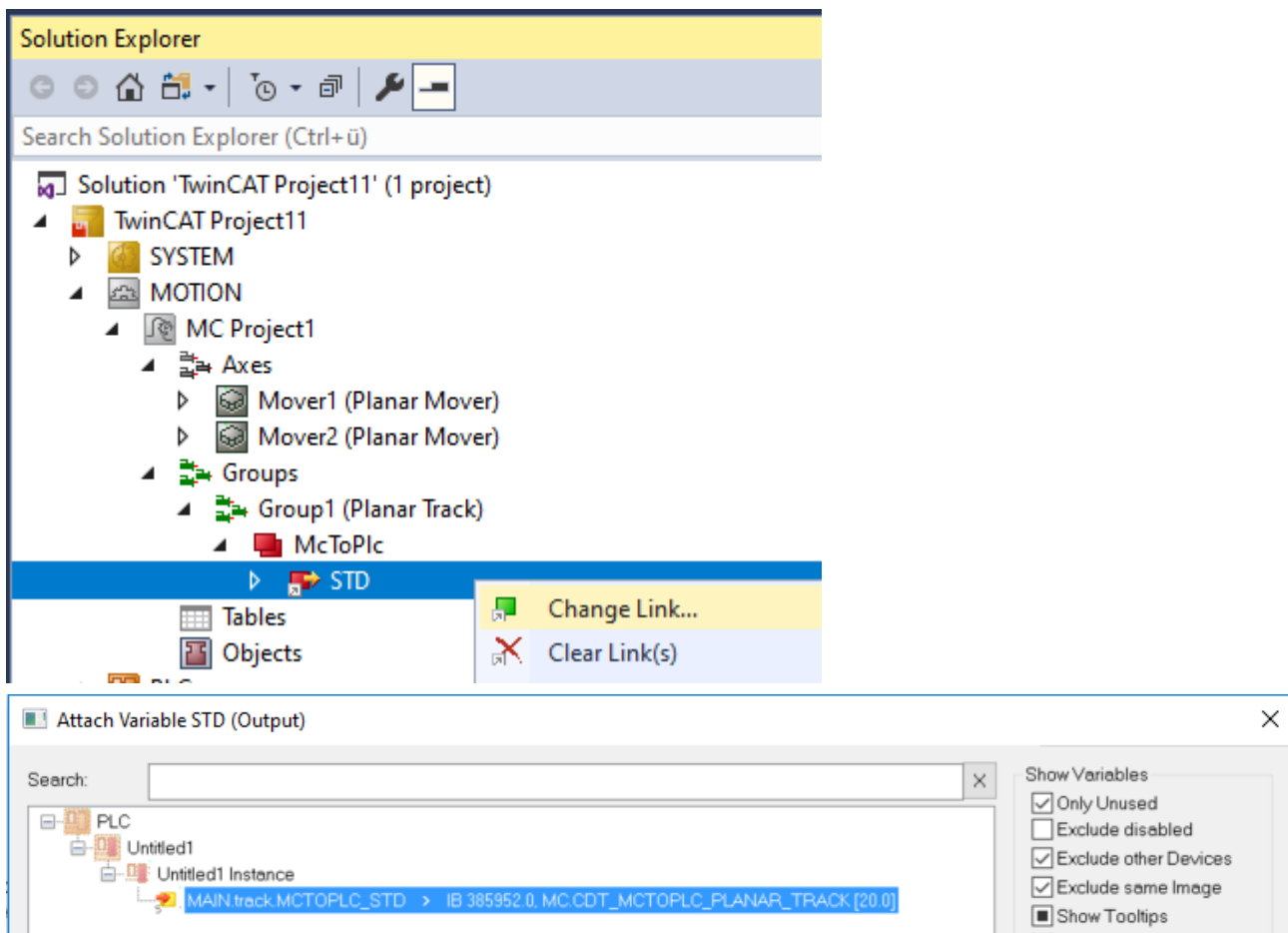
⇒ 随后，“MC Project”中的 Planar 动子可通过 **Setting** 选项卡上的 **Link To PLC...** 按钮进行链接。



⇒ 此外，“MC Project”（双击）中的 Planar 动子可以通过 **Settings** 选项卡上的 **Link To I/O...** 按钮进行链接。



必须通过以下对话框单独链接轨道。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。

4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机结束时 (state=8)，动子处于所需的位置。动子 2 (再次) 处于 OnTrack 状态，动子 1 处于 CRotationOnTrack 状态，而在运动过程中两者都处于 CRotationOnTrack 状态。现在，动子 1 只能继续旋转，而动子 2 可以继续继续在轨道上移动，甚至离开轨道。

ExampleCRotationOnTrack.Untitled1.MAIN					
Expression	Type	Value	Prepared value	Address	Comment
[-] mover_one	MC_PlanarMover				
[-] PLCTOMC	CDT_PLCTOMC...			%Q*	Mover data that is transferred from the Planar Mover to this function block.
[-] MCTOPLC	CDT_MCTOPLC...			%I*	Mover data that is transferred from the Planar Mover to this function block.
[-] STD	REFERENCE TO...			%IB*	Mover standard data that is transferred from the Planar Mover to this function b
[-] MoverOID	OTCID	16#05110010			Object id of the planar mover.
[-] GroupOID	OTCID	16#00000000			Object id of the planar group the mover is in.
[-] State	MC_PLANAR_S...	Enabled			State of the planar mover, e.g. enabled.
[-] CommandMode	MC_PLANAR_M...	CRotationOn...			Command mode of the planar mover, e.g. onTrack.
[-] Busy	MoverBusy				Busy state of the planar mover.
[-] ErrorCode	HRESULT	16#00000000			Error code of the planar mover.
[-] SET	REFERENCE TO...			%IB*	Mover setpoint data that is transferred from the Planar Mover to this function b
[-] ACT	REFERENCE TO...			%IB*	Mover actpoint data that is transferred from the Planar Mover to this function b
[-] COORDMODE	REFERENCE TO...			%IB*	Mover coordinate mode information that is transferred from the Planar Mover to t
[-] SETONTRACK	REFERENCE TO...			%IB*	Mover busy information that is transferred from the Planar Mover to this functio
[-] Error	BOOL	FALSE			Flag indicating a Planar Mover error.
[-] ErrorId	UDINT	0			Error id indicating the Planar Mover error type.
[-] mover_two	MC_PlanarMover				
[-] PLCTOMC	CDT_PLCTOMC...			%Q*	Mover data that is transferred from the Planar Mover to this function block.
[-] MCTOPLC	CDT_MCTOPLC...			%I*	Mover data that is transferred from the Planar Mover to this function block.
[-] STD	REFERENCE TO...			%IB*	Mover standard data that is transferred from the Planar Mover to this function b
[-] MoverOID	OTCID	16#05110020			Object id of the planar mover.
[-] GroupOID	OTCID	16#00000000			Object id of the planar group the mover is in.
[-] State	MC_PLANAR_S...	Enabled			State of the planar mover, e.g. enabled.
[-] CommandMode	MC_PLANAR_M...	OnTrack			Command mode of the planar mover, e.g. onTrack.
[-] Busy	MoverBusy				Busy state of the planar mover.
[-] ErrorCode	HRESULT	16#00000000			Error code of the planar mover.
[-] SET	REFERENCE TO...			%IB*	Mover setpoint data that is transferred from the Planar Mover to this function b
[-] ACT	REFERENCE TO...			%IB*	Mover actpoint data that is transferred from the Planar Mover to this function b
[-] COORDMODE	REFERENCE TO...			%IB*	Mover coordinate mode information that is transferred from the Planar Mover to t
[-] SETONTRACK	REFERENCE TO...			%IB*	Mover busy information that is transferred from the Planar Mover to this functio
[-] Error	BOOL	FALSE			Flag indicating a Planar Mover error.

6.2.8 示例“将 Planar 动子耦合到轨道，并使用 AdoptTrackOrientation 移动它”

通过本指南，您将创建 1 个包含 2 个 Planar 动子和 1 个 Planar 轨道的 TwinCAT 项目。2 个动子加入并在轨道上移动。

创建 Planar 动子

✓ 参阅 [配置 \[▶ 16\]](#)。

1. 创建 2 个 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。
3. 将第 2 个动子的起始位置改为 $x = 240$ 。

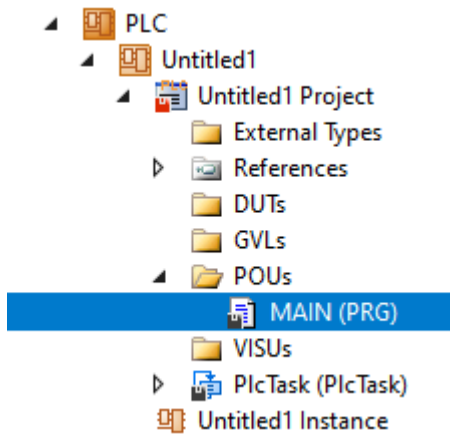
创建 Planar 轨道

4. 通过 **Groups > Add New Item...** 添加 Planar 轨道，请参阅 [配置 \[▶ 39\]](#)。

创建 PLC

✓ 请参阅 [创建 PLC \[▶ 19\]](#) 下的初步步骤。

1. 通过 MAIN 创建所需数量的动子 (“MC_PlanarMover”) 和轨道 (“MC_PlanarTrack”)。



⇒ 这些代表 MC Configuration 中的动子和轨道。

2. 如下图所示，创建 2 个 Planar 动子、1 个 Planar 轨道、1 个状态机的任意状态变量以及轨道的 2 个辅助位置。

```
PROGRAM MAIN
VAR
    mover_one, mover_two : MC_PlanarMover;
    track : MC_PlanarTrack;
    state : UDINT;
    pos1, pos2 : PositionXYC;
    join_track_options : ST_JoinTrackOptions;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 该程序代码创建并激活 1 个轨道和 2 个动子。然后，2 个动子与轨道耦合并旋转。

```
CASE state OF
0:
    pos1.SetValuesXY(0, 0);
    pos2.SetValuesXY(400, 0);
    track.AppendLine(0, pos1, pos2);
    track.Enable(0);
    state := 1;
1:
    IF track.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    mover_one.Enable(0);
    mover_two.Enable(0);
    state := 3;
3:
    IF mover_one.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled
    AND mover_two.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 4;
    END_IF
4:
    join_track_options.useOrientation := TRUE;
    mover_one.JoinTrack(0, track, 0, join_track_options);
    mover_two.JoinTrack(0, track, 0, join_track_options);
    state := 5;
5:
    IF mover_one.MCTOPLC.STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack
    AND mover_two.MCTOPLC.STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
        state := 6;
    END_IF
6:
    mover_one.MoveC(0, 20.0, 0, 0);
    mover_two.MoveC(0, 190.0, 0, 0);
    state := 7;
7:
    IF mover_one.MCTOPLC.SET.SetPos.c >= 19.9
    AND NOT mover_one.MCTOPLC.STD.Busy.busyMover
    AND mover_two.MCTOPLC.SET.SetPos.c >= 189.9
    AND NOT mover_two.MCTOPLC.STD.Busy.busyMover THEN
        state := 8;
    END_IF
8:
    mover_one.AdoptTrackOrientation(0, 0, 0);
    mover_two.AdoptTrackOrientation(0, 0, 0);
```

```
state := 9;
END_CASE
```

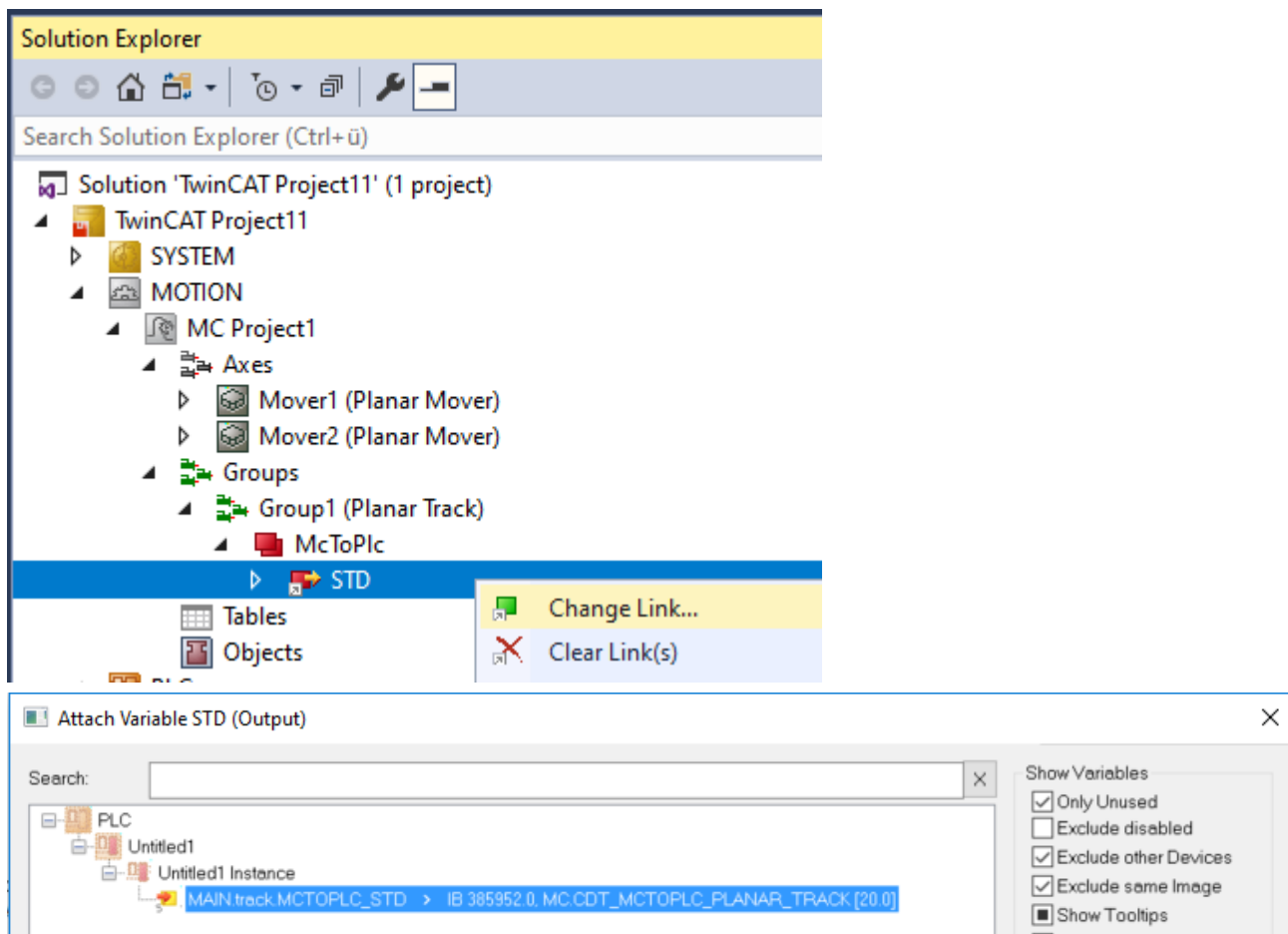
发送命令

- 要发送命令，必须在 END_CASE 之后循环调用动子和轨道的更新方法：

```
mover_one.Update();
mover_two.Update();
track.Update();
```

构建 PLC 时会创建“PLC mover”和“Track”的符号，然后将其链接到 MC Project 中的动子和轨道实例。

- 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。
 - ⇒ 随后，“MC Project”中的 Planar 动子可通过 **Setting** 选项卡上的 **Link To PLC...** 按钮进行链接。
 - ⇒ 必须通过以下对话框单独链接轨道。



激活并启动项目

- 通过菜单栏  中的按钮激活配置。
- 通过  按钮将 TwinCAT 系统设置为“Run”状态。
- 通过菜单栏  中的按钮登录 PLC。
- 通过菜单栏中的 Play 按钮启动 PLC。

2 个动子都加入轨道，方向与轨道耦合。之后，通过 MoveC 将方向与轨道解耦。在状态机结束时（状态=9），动子处于所需的位置。2 个动子（再次）处于 OnTrack 状态，并通过 AdoptTrackOrientation 命令再次将方向与轨道耦合。该命令有 3 个参数：第一，1 个可选的反馈对象；第二，1 个可选的约束对象；第三，1 个可选的选项对象。

ExampleAdoptTrackOrientation.Untitled1.MAIN					
Expression	Type	Value	Prepared value	Address	Comment
[-] mover_one	MC_PlanarMover				
[-] PLCTOMC	CDT_PLCTOMC...			%Q*	Mover data that is transferred from the Planar Mover to this function block.
[-] MCTOPLC	CDT_MCTOPLC...			%I*	Mover data that is transferred from the Planar Mover to this function block.
[-] STD	REFERENCE TO...			%IB*	Mover standard data that is transferred from the Planar Mover to this function b
[-] MoverOID	OTCID	16#05110010			Object id of the planar mover.
[-] GroupOID	OTCID	16#00000000			Object id of the planar group the mover is in.
[-] State	MC_PLANAR_S...	Enabled			State of the planar mover, e.g. enabled.
[-] CommandMode	MC_PLANAR_M...	OnTrack			Command mode of the planar mover, e.g. onTrack.
[-] Busy	MoverBusy				Busy state of the planar mover.
[-] ErrorCode	HRESULT	16#00000000			Error code of the planar mover.
[-] SET	REFERENCE TO...			%IB*	Mover setpoint data that is transferred from the Planar Mover to this function b
[-] ACT	REFERENCE TO...			%IB*	Mover actpoint data that is transferred from the Planar Mover to this function b
[-] COORDMODE	REFERENCE TO...			%IB*	Mover coordinate mode information that is transferred from the Planar Mover to t
[-] XYCoordinateMode	MC_PLANAR_C...	Dependent			X and Y coordinate.
[-] ZCoordinateMode	MC_PLANAR_C...	Independent			Z coordinate.
[-] ACoordinateMode	MC_PLANAR_C...	Independent			A coordinate.
[-] BCoordinateMode	MC_PLANAR_C...	Independent			B coordinate.
[-] CCoordinateMode	MC_PLANAR_C...	Dependent			C coordinate.
[-] SETONTRACK	REFERENCE TO...			%IB*	Mover busy information that is transferred from the Planar Mover to this functio
[-] Error	BOOL	FALSE			Flag indicating a Planar Mover error.
[-] ErrorId	UDINT	0			Error id indicating the Planar Mover error type.
[-] mover_two	MC_PlanarMover				
[-] PLCTOMC	CDT_PLCTOMC...			%Q*	Mover data that is transferred from the Planar Mover to this function block.
[-] MCTOPLC	CDT_MCTOPLC...			%I*	Mover data that is transferred from the Planar Mover to this function block.
[-] STD	REFERENCE TO...			%IB*	Mover standard data that is transferred from the Planar Mover to this function b
[-] MoverOID	OTCID	16#05110020			Object id of the planar mover.
[-] GroupOID	OTCID	16#00000000			Object id of the planar group the mover is in.
[-] State	MC_PLANAR_S...	Enabled			State of the planar mover, e.g. enabled.
[-] CommandMode	MC_PLANAR_M...	OnTrack			Command mode of the planar mover, e.g. onTrack.
[-] Busy	MoverBusy				Busy state of the planar mover.
[-] ErrorCode	HRESULT	16#00000000			Error code of the planar mover.
[-] SET	REFERENCE TO...			%IB*	Mover setpoint data that is transferred from the Planar Mover to this function b
[-] ACT	REFERENCE TO...			%IB*	Mover actpoint data that is transferred from the Planar Mover to this function b
[-] COORDMODE	REFERENCE TO...			%IB*	Mover coordinate mode information that is transferred from the Planar Mover to t
[-] XYCoordinateMode	MC_PLANAR_C...	Dependent			X and Y coordinate.
[-] ZCoordinateMode	MC_PLANAR_C...	Independent			Z coordinate.
[-] ACoordinateMode	MC_PLANAR_C...	Independent			A coordinate.
[-] BCoordinateMode	MC_PLANAR_C...	Independent			B coordinate.
[-] CCoordinateMode	MC_PLANAR_C...	Dependent			C coordinate.
[-] SETONTRACK	REFERENCE TO...			%IB*	Mover busy information that is transferred from the Planar Mover to this functio
[-] Error	BOOL	FALSE			Flag indicating a Planar Mover error.
[-] ErrorId	UDINT	0			Error id indicating the Planar Mover error type.
[-] track	MC_PlanarTrack				
[-] state	UDINT	9			
[-] pos1	PositionXYC				
[-] pos2	PositionXYC				
[-] join_track_options	ST_JoinTrackO...				

6.2.9 示例“Planar 动子在轨道上与外部轴同步”

通过这些说明，您将创建 1 个 TwinCAT 项目，在该项目中，位于轨道上的 Planar 动子与 1 个轴耦合，然后按照轴的设定值运行。

在这种情况下，Planar 动子控制器不直接受 MoveOnTrack 命令控制，在 MoveOnTrack 命令中，接近指定的目标位置，随后停止，见 示例“Planar 动子加入并在轨道上移动” [▶ 48]。相反，Planar 动子一直与轴耦合，直到后续命令终止耦合或发生错误。

发送启动与轴耦合的 GearInPosOnTrack [▶ 147] 命令后，如果与之耦合的轴位于 masterSyncPosition，则 Planar 动子将尝试移动至指定的 slaveSyncPosition，并同时采用主轴的动态性能设置。如果能更早达到同步（即 Planar 动子在从同步位置 - x 处已经具有与主轴相同的动态性能，此时主轴在 masterSyncPosition - x 处），那么 Planar 动子将激活此配置，更早实现同步。如果在指定时间内无法达到同步，则 Planar 动子将尝试与主轴同步，直到收到后续命令或发生错误。

如果 Planar 动子失去同步状态，例如由于主轴的动态快速变化，它会尽快尝试再次同步。可随时通过相应的反馈对象从 PLC 访问同步状态。如果与前方的 Planar 动子保持指定距离，需要同步 Planar 动子减速，也会失去同步。同样，一旦障碍物被移除，系统会尝试尽快恢复同步。

导致命令中止的错误示例是，主轴行为会迫使 Planar 动子以负速移动，超出 Planar 轨道的起点。即使使用 `MoveOnTrack` [▶ 146] 命令，也不允许进行这样的移动。在这种情况下，Planar 动子将保持同步（或尝试同步，如果还没有同步的话），直到被迫停止，从而停在 Planar 轨道的起点。此外，还会报告 1 个错误。Planar 动子停止的确切位置取决于当前的动态限值。

如果 `GearInPosOnTrack` [▶ 147] 命令给出的动态限值的速度极限低于主轴的当前速度，则 Planar 动子仍将尝试同步，因为不能排除主轴会在稍后的时间点减速，从而再次达到同步。特别是在这种情况下，不会返回错误信息。

创建 Planar 动子

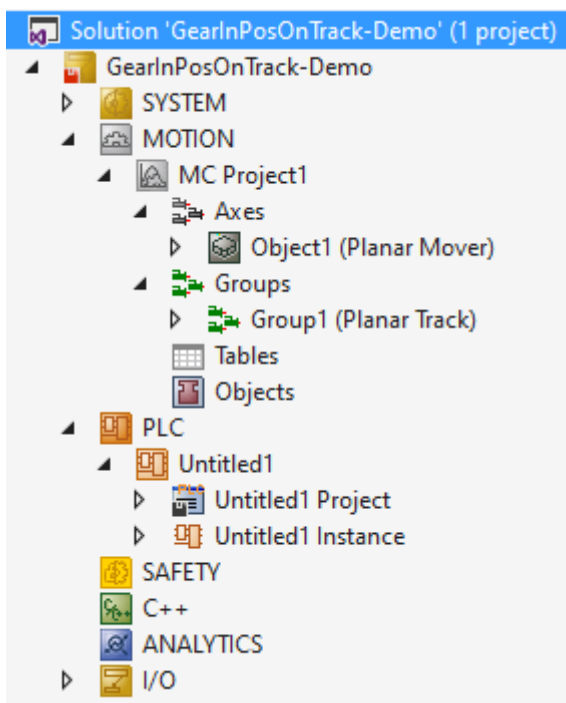
✓ 参阅 [配置 \[▶ 16\]](#)。

1. 创建 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

创建 Planar 轨道

3. 通过 `Groups > Add New Item...` 添加 Planar 轨道，请参阅 [配置 \[▶ 39\]](#)。

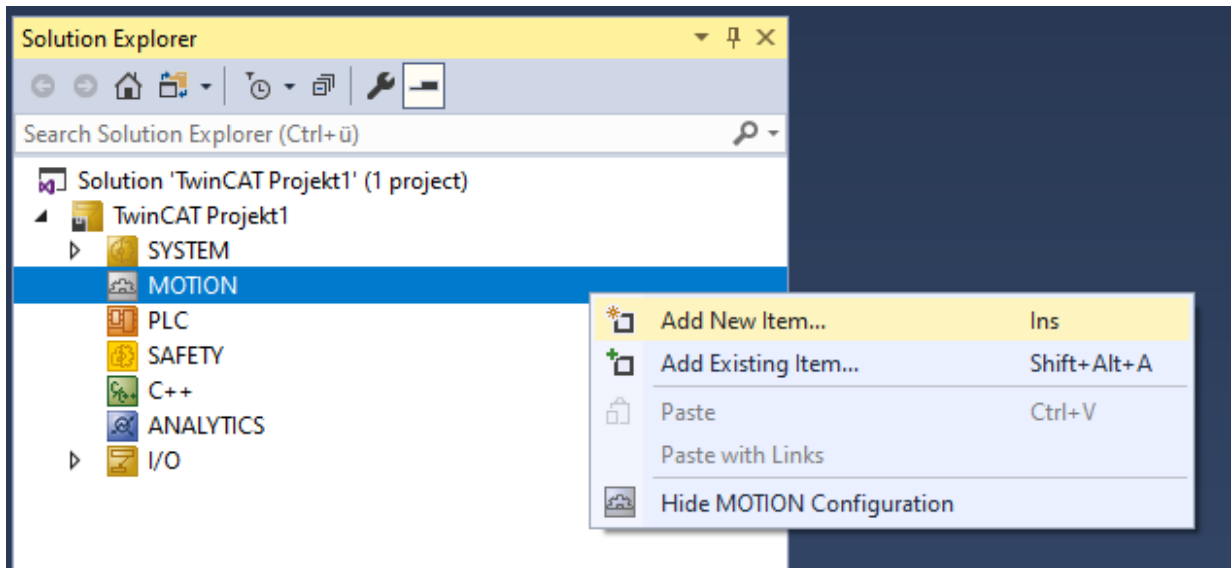
⇒ 解决方案资源管理器中有以下条目：



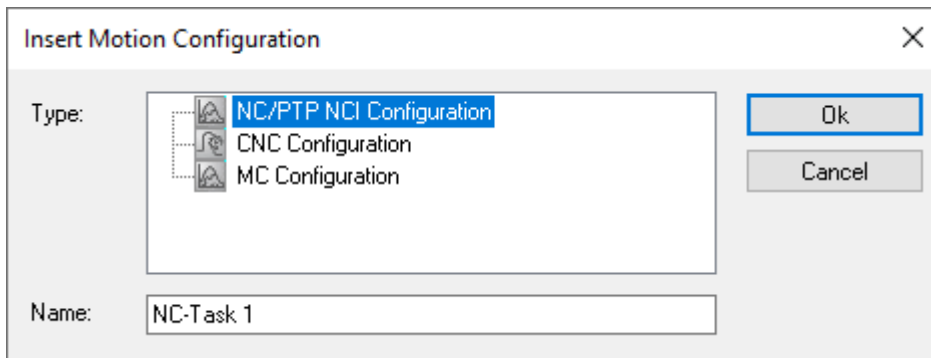
创建主轴

✓ 要创建主轴，必须首先创建 `NC/PTP NCI Configuration`。

1. 选择 MOTION > Add New Item...

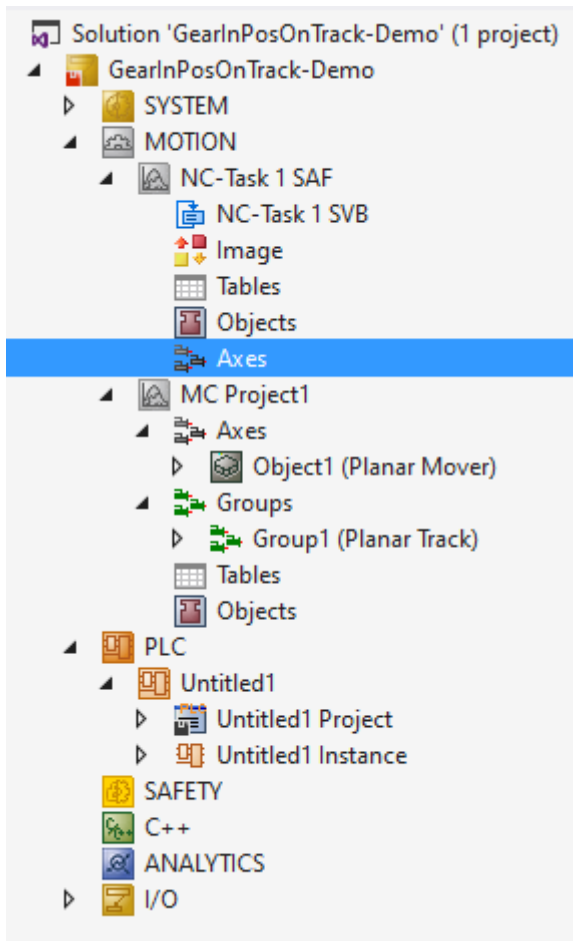


2. 在以下对话框中，选择 NC/PTP NCI Configuration，并按 OK 确认。



⇒ 您创建了 1 个 NC/PTP NCI Project。

3. 右键单击创建的 NC 项目 **Axes** > **Add New Item...**。



4. 在以下对话框中，创建 1 个（或多个）坐标轴，并按 **OK**

确认

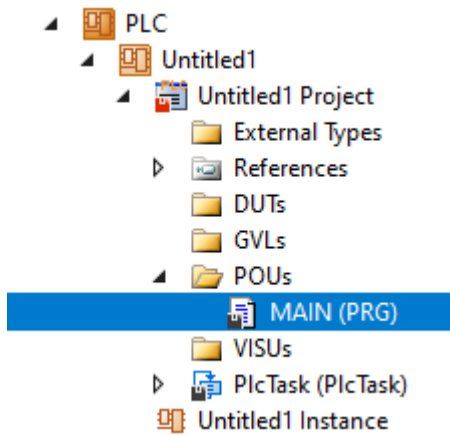
创建 PLC



对于该 PLC 项目，还必须添加“Tc2_MC2”来控制主轴，请参阅 [插入功能库 \[► 123\]](#)。

✓ 请参阅 [创建 PLC \[► 19\]](#) 下的初步步骤。

1. 通过 **MAIN** 创建所需数量的动子（“MC_PlanarMover”）和轨道（“MC_PlanarTrack”）。

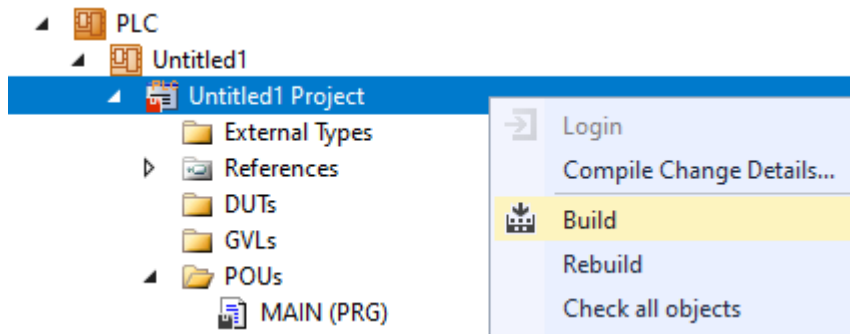


⇒ 这些代表 MC Configuration 中的动子和轨道。

2. 创建以下任意变量。

```
PROGRAM MAIN
VAR
  mover          : MC_PlanarMover;
  track          : MC_PlanarTrack;
  axis           : AXIS_REF;
  power_axis    : MC_Power;
  move_axis     : MC_MoveAbsolute;
  state         : UDINT;
  pos1, pos2    : PositionXYC;
END_VAR
```

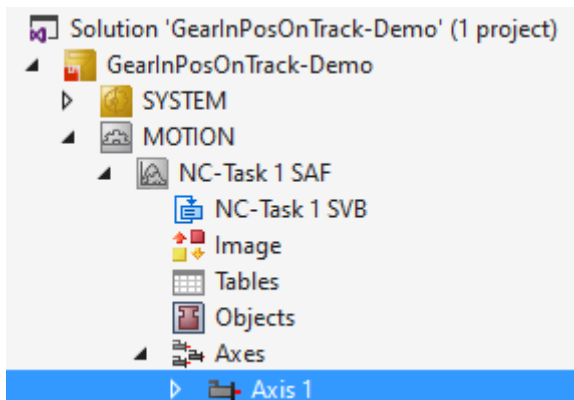
3. 构建 PLC，创建“PLC mover”“PLC track”和“PLC axis”的符号。



4. 按照下一节所述，将 Planar 动子、Planar 轨道（见 [示例“Planar 动子加入并在轨道上移动”](#) [P. 48]）和轴连接起来。

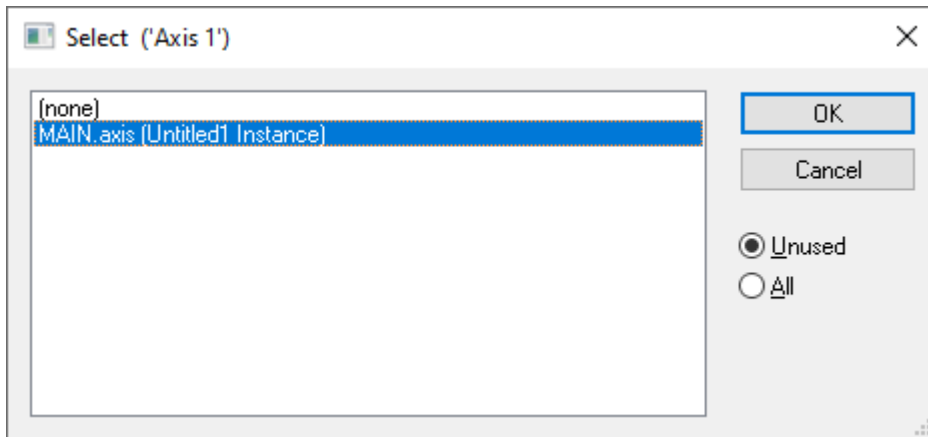
连接轴

5. 双击解决方案资源管理器中的 Axis 1



6. 切换到 Setting 选项卡。

7. 单击 **Link to PLC...**，在随后的对话框中选择 **MAIN.axis** 项，并按 **OK** 确认。



状态机编程

通过 MAIN 编程的以下状态机，Planar 轨道被几何定义并激活（状态 0），Planar 动子被激活并与 Planar 轨道耦合（状态 2 或 4），主轴被激活（状态 6）并移动（状态 7）。

最后，向 Planar 动子（状态 8）发送开始与主轴同步的命令（“`GearInPosOnTrack [▸_147]`”）。在这里，也是循环更新 Planar 对象或调用轴 FB（在 `END_CASE` 语句之后）：

```

CASE state OF
0:
  pos1.SetValuesXYC(100, 100, 0);
  pos2.SetValuesXYC(860, 100, 0);
  track.AppendLine(0, pos1, pos2);
  track.Enable(0);
  state := state + 1;
1:
  IF track.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
2:
  mover.Enable(0);
  state := state + 1;
3:
  IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
4:
  mover.JoinTrack(0, track, 0, 0);
  state := state + 1;
5:
  IF mover.MCTOPLC.STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
    state := state + 1;
  END_IF
6:
  power_axis(Axis := axis,
  Enable := TRUE,
  Enable_Positive := TRUE);
  IF power_axis.Status THEN
    move_axis(Axis := axis, Execute := FALSE);
    state := state + 1;
  END_IF
7:
  move_axis(Axis := axis,
  Position := 600,
  Velocity := 30,
  Acceleration := 100,
  Deceleration := 100,
  Jerk := 100,
  Execute := TRUE);
  state := state + 1;
8:
  mover.GearInPosOnTrack(0, axis.DriveAddress.TcAxisObjectId, 0, 100, 100, track, 0, 0);
  state := state + 1;
END_CASE

mover.Update();
track.Update();
















```

```
power_axis(Axis := axis);
move_axis(Axis := axis);
axis.ReadStatus();
```

激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

主轴将移动到给定的目标位置（本例中为 600），Planar 动子将跟随其移动。Planar 动子的位置可以在实时视图中跟踪（点击按钮）。

Expression	Type	Value
 mover	MC_PlanarMover	
 PLCTOMC	CDT_PLCTOMC_PLA...	
 MCTOPLC	CDT_MCTOPLC_PLA...	
 STD	REFERENCE TO CDT...	
 SET	REFERENCE TO CDT...	
 ACT	REFERENCE TO CDT...	
 COORDMODE	REFERENCE TO CDT...	
 SETONTRACK	REFERENCE TO CDT...	
 SetPos	LREAL	274.1483232748 ...
 SetVelo	LREAL	29.999999999999 ...
 SetAcc	LREAL	0
 SetJerk	LREAL	0
 TrackOID	OTCID	16#05120010
 Error	BOOL	FALSE
 ErrorId	UDINT	0

动子在位置 600 处停止，因为主轴也在此处达到零动力。如果在状态 7 中为主轴的目标位置编程了 1 个大于轨道长度（本例中为 760）的值，则 Planar 动子在 Planar 轨道的末端停止，以避免脱轨，并且不再跟随主轴前进。MC 可能会将这种场景中的错误返回给 PLC，但在这种情况下，上述 PLC 代码并未编译接收这种错误。为此需要 1 个 [Feedback \[► 117\]](#) 对象，用于监控同步状态。

在状态 8 的函数调用中，主轴（第 3 个参数）或从动子（第 4 个参数）的同步位置被传递给 Planar 动子。这些是从动子与主轴同步的相应位置，即达到动态值的位置。函数调用中的第 5 个参数指定了前 1 个参数中的位置所指向的 Planar 轨道。事实上，从动子有可能更快地与主轴同步。

通过使用 [Planar TrackTrail \[► 118\]](#) 对象，可以在一连串连续轨道上进行同步运动。在这种情况下，在进入同步阶段或同步阶段，可以从 1 个 Planar 轨道过渡到下 1 个 Planar 轨道。如果主轴的运动需要动子超出 Planar 轨道，那么与上例中只有 1 条 Planar 轨道的例子类似，Planar 动子只会在整条 Planar 轨道的末端进行减速。

6.2.10 示例：“使轨道上的 Planar 动子与另一个 Planar 动子同步”

在这些说明的指导下，您将创建 1 个 TwinCAT 项目，其中位于 Planar 轨道上的 Planar 动子与平行 Planar 轨道上的另 1 个 Planar 动子耦合，然后遵循其设定点。

将 1 个 Planar 动子耦合到另 1 个 Planar 动子上，与将 1 个 Planar 动子耦合到 1 个轴上基本类似；见 [示例“Planar 动子在轨道上与外部轴同步” \[► 64\]](#)。这个示例很短，是在上述示例的基础上编写的。

创建 Planar 动子

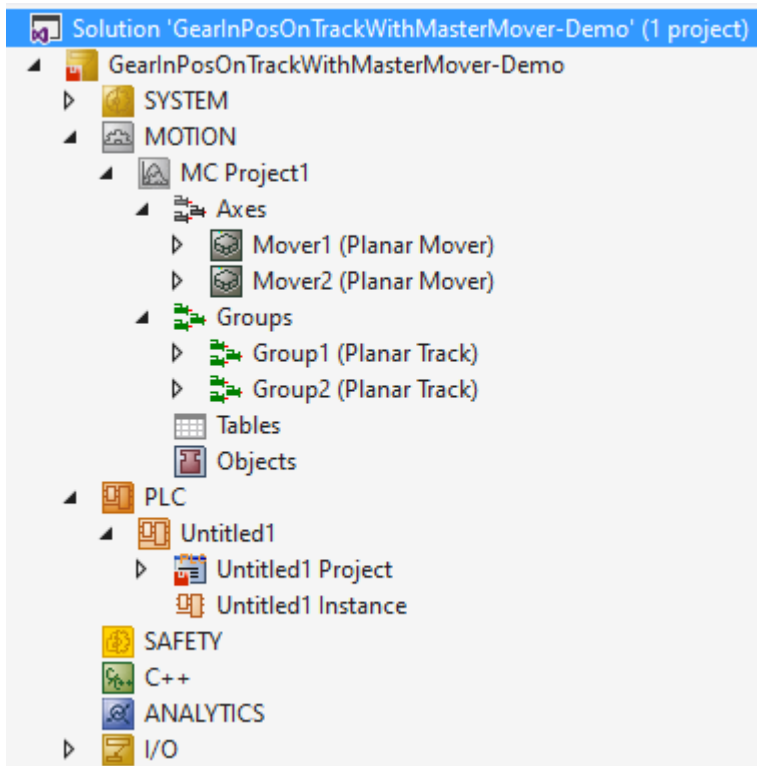
✓ 参阅 [配置 \[▶ 16\]](#)。

1. 创建 2 个 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

创建 Planar 轨道

3. 通过 **Groups > Add New Item...** 添加 2 条 Planar 轨道，请参阅 [配置 \[▶ 39\]](#)。

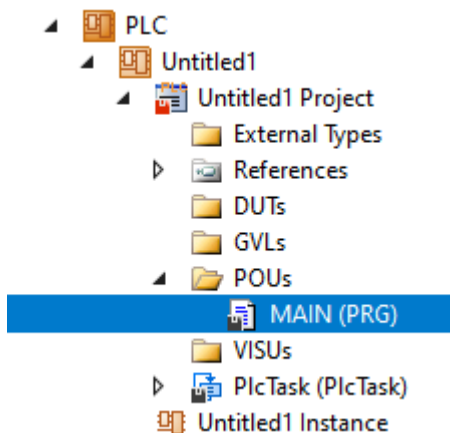
⇒ 解决方案资源管理器中有以下条目：



创建 PLC

✓ 请参阅 [创建 PLC \[▶ 19\]](#) 下的初步步骤。

1. 通过 **MAIN** 创建所需数量的动子 (“MC_PlanarMover”) 和轨道 (“MC_PlanarTrack”)。



⇒ 这些代表 MC Configuration 中的动子和轨道。

2. 创建以下任意变量。

```
PROGRAM MAIN
VAR
  master_mover : MC_PlanarMover;
  slave_mover  : MC_PlanarMover;
```

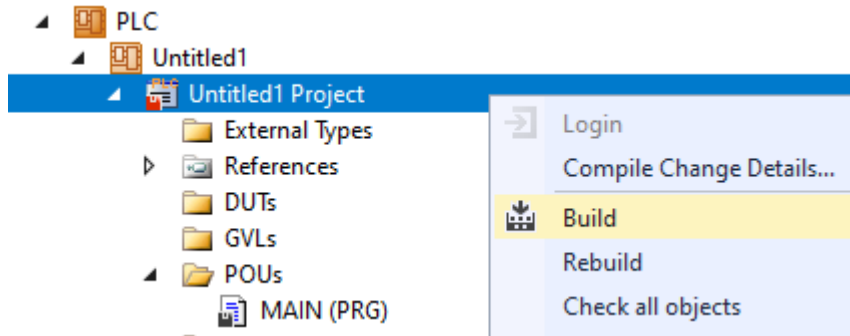
```

master_track : MC_PlanarTrack;
slave_track  : MC_PlanarTrack;
state        : UDINT;
pos1, pos2   : PositionXYC;

```

```
END_VAR
```

3. 构建 PLC，创建“PLC mover”和“PLC track”的符号。



4. 连接 Planar 动子和 Planar 轨道（见 示例“Planar 动子加入并在轨道上移动” [▶ 48]）。

状态机编程

通过 MAIN 编程的以下状态机，Planar 轨道被几何定义并激活（状态 0 至 3），Planar 动子被激活并耦合到各自的 Planar 轨道上（状态 4 至 11），作为主控的 Planar 动子在其轨道上移动（状态 12）。

最后，向 Slave Planar Mover 发送开始与 Master Planar Mover 同步的命令（`GearInPosOnTrackWithMasterMover` [▶ 148]）（状态 13）。在 `END_CASE` 语句之后，将循环更新 Planar 对象。

```

CASE state OF
0:
  pos1.SetValuesXYC(100, 620, 0);
  pos2.SetValuesXYC(860, 620, 0);
  master_track.AppendLine(0, pos1, pos2);
  master_track.Enable(0);
  state := state + 1;
1:
  IF master_track.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
2:
  pos1.SetValuesXYC(100, 100, 0);
  pos2.SetValuesXYC(860, 100, 0);
  slave_track.AppendLine(0, pos1, pos2);
  slave_track.Enable(0);
  state := state + 1;
3:
  IF slave_track.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
4:
  master_mover.Enable(0);
  state := state + 1;
5:
  IF master_mover.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
6:
  master_mover.JoinTrack(0, master_track, 0, 0);
  state := state + 1;
7:
  IF master_mover.MCTOPLC_STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
    state := state + 1;
  END_IF
8:
  slave_mover.Enable(0);
  state := state + 1;
9:
  IF slave_mover.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
10:
  slave_mover.JoinTrack(0, slave_track, 0, 0);
  state := state + 1;

```



```

11:
  IF slave_mover.MCTOPLC.STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
    state := state + 1;
  END_IF
12:
  master_mover.MoveOnTrack(0, 0, 500.0, 0, 0);
  state := state + 1;
13:
  slave_mover.GearInPosOnTrackWithMasterMover(0, master_mover, 0, 100.0, master_track, 100.0, slave_track, 0, 0);
  state := state + 1;
END_CASE

master_mover.Update();
slave_mover.Update();
master_track.Update();
slave_track.Update();

```

激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

Master Planar Mover 将移动到指定 Planar 轨道上的给定目标位置（本例中为 500），Slave Planar Mover 将跟随其移动。Planar 动子的位置可以在实时视图中跟踪（点击按钮）。

Slave Planar Mover 停止在 500 位置，因为 Master Planar Mover 也在此处达到零动态。

在状态 13 的函数调用中，主动子（参数 4 和 5）或从动子（参数 6 和 7）的同步位置被传递给 Slave Planar Mover。这些是从动子与主轴同步的相应位置，即达到动态值的位置。事实上，在这里以及在 [示例“Planar 动子在轨道上与外部轴同步” \[► 64\]](#) 中，从动子有可能更快地与主动子同步。与轴同步一样，需要 1 个特殊的 [专门的反馈类型 \[► 117\]](#) 对象来监控同步状态和可能出现的错误。

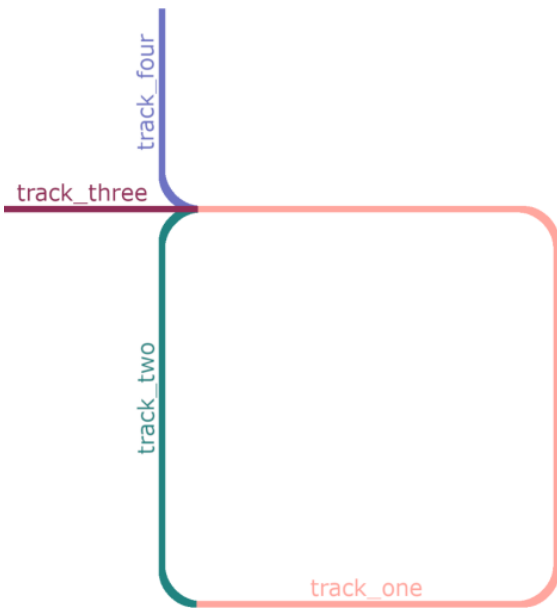
与主动轴同步一样，从动轴的同步运动也可以通过指定 [Planar TrackTrail \[► 118\]](#) 对象在多个轨道上进行编程。

如果 Master Planar Mover 在主动同步命令期间移动过轨道边界，那么它传递给从动子的位置会在轨道边界上简单相加。

如果将来要在 Master Planar Mover 通过的 Planar 轨道上指定主同步位置，请确保在发送 [GearInPosOnTrackWithMasterMover \[► 148\]](#) 命令时，Master Planar Mover 已经发出了涉及该 Planar 轨道的移动指令。

6.2.11 示例“将 Planar 轨道与网络互联互通”

通过本指南，您可创建 1 个将 4 个 Planar 轨道与网络连接的 TwinCAT 项目。



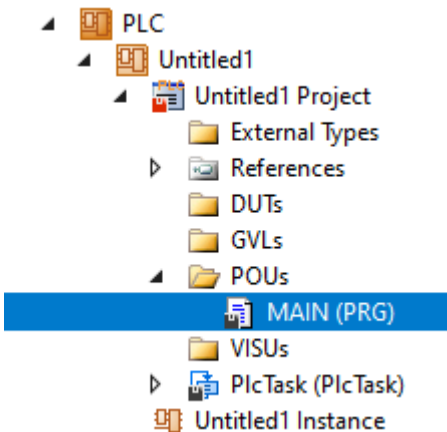
创建 Planar 轨道

1. 通过 **Groups > Add New Item...** 添加 4 个 Planar 轨道，请参见 [配置 \[▶ 39\]](#)。

创建 PLC

- ✓ 请参阅 [创建 PLC \[▶ 19\]](#) 下的初步步骤。

1. 通过 **MAIN** 创建所需数量的动子 (“MC_PlanarMover”) 和轨道 (“MC_PlanarTrack”)。



⇒ 这些代表 MC Configuration 中的动子和轨道。

2. 如下图所示，创建 4 个轨道，外加 1 个状态机的任意状态变量和 2 个轨道的辅助位置。

```
PROGRAM MAIN
VAR
    track_one, track_two, track_three, track_four : MC_PlanarTrack;
    state : UDINT;
    pos1, pos2 : PositionXYC;
END_VAR
```

3. 然后在 MAIN 中设计 1 个序列。

⇒ 如上图所示，该程序代码可创建并激活与网络连接的 4 个轨道。所谓的“混合”（请参见 [几何定义 \[▶ 45\]](#)），即本示例中轨道的非线性部分，便是在此处自动生成。您只需指定直线部分。

```
CASE state OF
0:
    pos1.SetValuesXY(250, 120);
    pos2.SetValuesXY(650, 120);
    track_one.AppendLine(0, pos1, pos2);
    pos1.SetValuesXY(700, 170);
    pos2.SetValuesXY(800, 450);
    track_one.AppendLine(0, pos1, pos2);
    pos1.SetValuesXY(650, 500);
    pos2.SetValuesXY(250, 500);
```

```

    track_one.AppendLine(0, pos1, pos2);
    state := 1;
1:
    pos1.SetValuesXY(200, 450);
    pos2.SetValuesXY(200, 170);
    track_two.StartFromTrack(0, track_one);
    track_two.AppendLine(0, pos1, pos2);
    track_two.EndAtTrack(0, track_one);
    state := 2;
2:
    pos1.SetValuesXY(200, 500);
    pos2.SetValuesXY(120, 500);
    track_three.StartFromTrack(0, track_one);
    track_three.AppendLine(0, pos1, pos2);
    state := 3;
3:
    pos1.SetValuesXY(200, 550);
    pos2.SetValuesXY(200, 750);
    track_four.StartFromTrack(0, track_one);
    track_four.AppendLine(0, pos1, pos2);
    state := 4;
4:
    track_one.Enable(0);
    track_two.Enable(0);
    track_three.Enable(0);
    track_four.Enable(0);
    state := 5;
5:
    IF track_one.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled AND
    track_two.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled AND
    track_three.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled AND
    track_four.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 6;
    END_IF
END_CASE

```



轨道必须在所有点上保持 C2 连续性。这意味着它们的位置、方向和曲率必须无缝融合。自动生成的混合效果考虑到了这一要求，请参见 [几何定义 \[► 45\]](#)。即使转弯段看起来像四分之一圆，它们也不是，因为圆的每 1 点都有正曲率（恒定曲率），而直线的曲率为零。

发送命令

4. 要发送命令，必须在 END_CASE 之后循环调用轨道部分更新的方法：

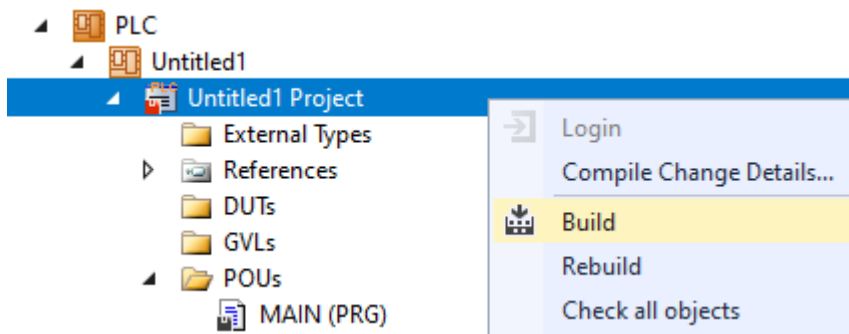
```

track_one.Update();
track_two.Update();
track_three.Update();
track_four.Update();

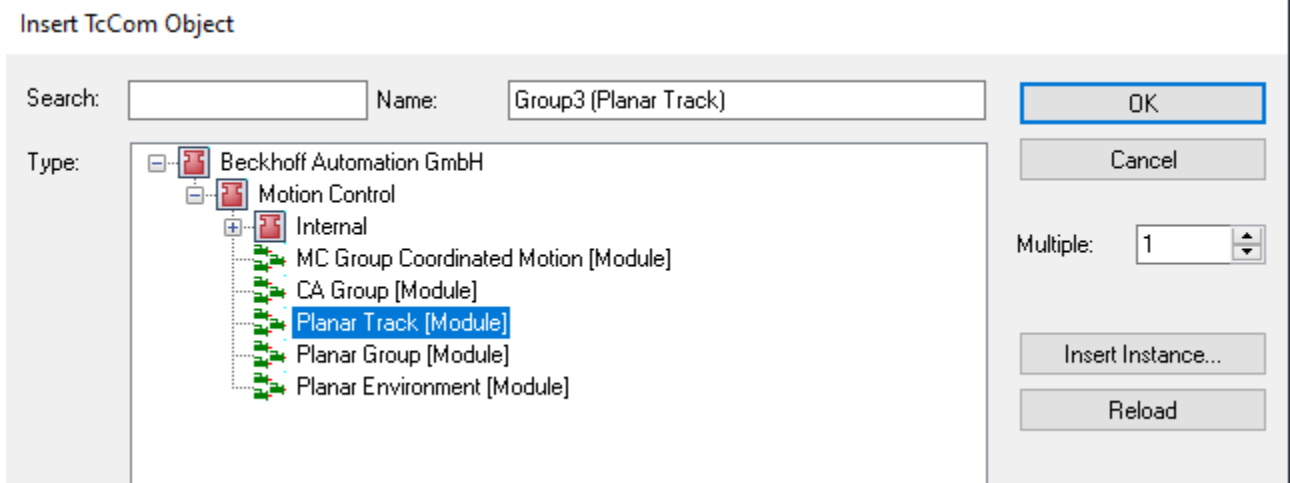
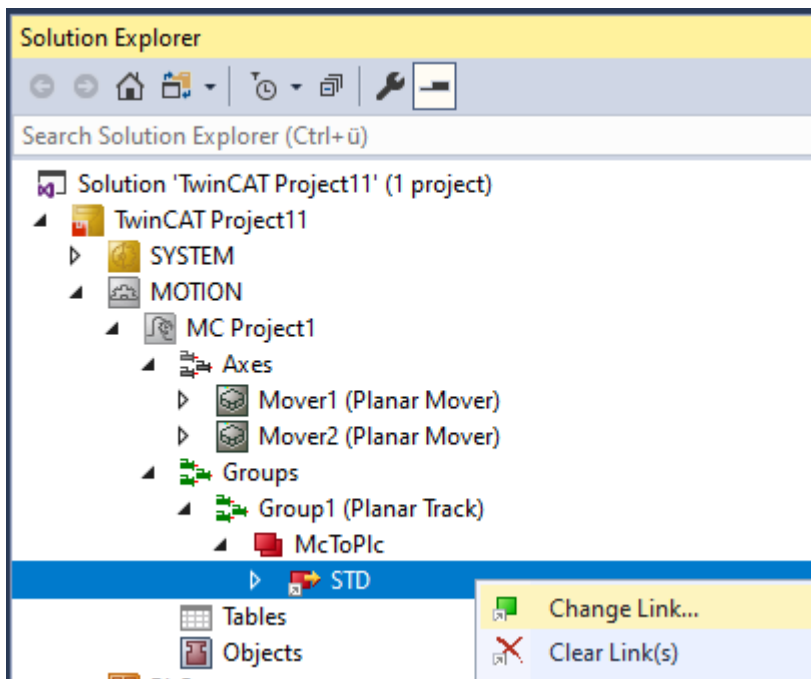
```

构建 PLC 时会创建“PLC mover”和“Track”的符号，然后将其链接到 MC Project 中的动子和轨道实例。

1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



⇒ 必须通过以下对话框分别链接各轨道。



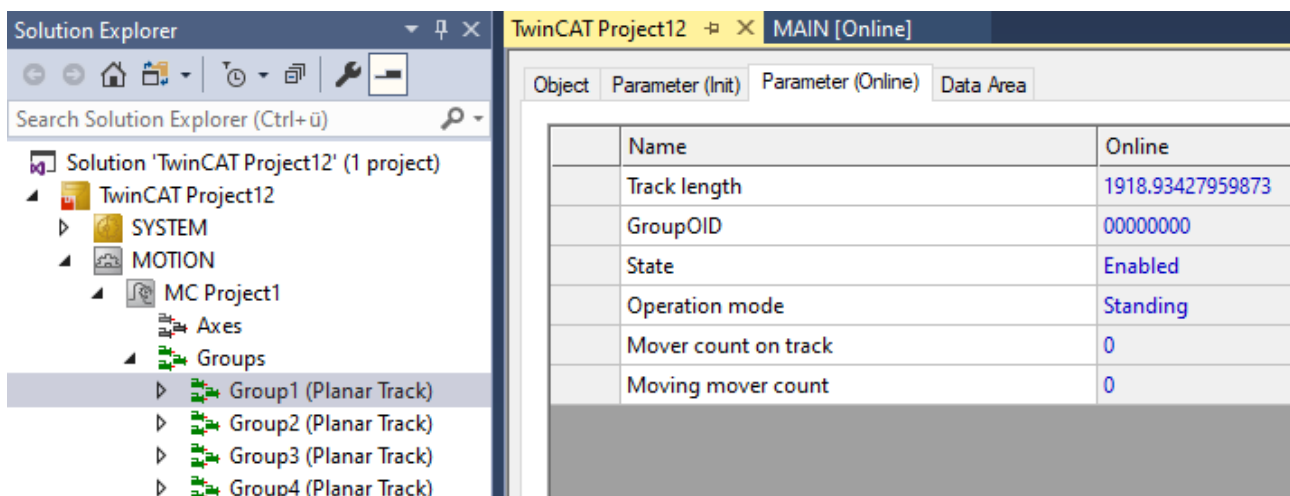
激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

轨道网络的创建在状态机 (state=6) 结束时完成。

Expression	Type	Value	Prepared value	Address	Comment
track_one	MC_PlanarTrack				
MCTOPLC_STD	CDT_MCTOPLC_PLA...			%I*	Track data that is tran...rred from the Planar...
TrackOID	OTCID	16#05120010		%IB*	Object id of the planar track.
GroupOID	OTCID	16#00000000		%IB*	Object id of the planar group the track is in.
State	MC_PLANAR_STATE	Enabled		%IB*	State of the planar track, e.g. disabled.
OperationMode	MC_PLANAR_TRACK...	Standing		%IB*	Operation mode of the...nar track, e.g. movi...
MoverCountOnTrack	UDINT	0		%IB*	Number of movers coupled to this track.
MovingMoverCount	UDINT	0		%IB*	Number of movers ha... a movement planne...
Error	BOOL	FALSE			Flag indicating a PlanarTrack error.
ErrorId	UDINT	0			Error id indicating the PlanarTrack error type.
track_two	MC_PlanarTrack				
MCTOPLC_STD	CDT_MCTOPLC_PLA...			%I*	Track data that is tran...rred from the Planar...
TrackOID	OTCID	16#05120010		%IB*	Object id of the planar track.
GroupOID	OTCID	16#00000000		%IB*	Object id of the planar group the track is in.
State	MC_PLANAR_STATE	Enabled		%IB*	State of the planar track, e.g. disabled.
OperationMode	MC_PLANAR_TRACK...	Standing		%IB*	Operation mode of the...nar track, e.g. movi...
MoverCountOnTrack	UDINT	0		%IB*	Number of movers coupled to this track.
MovingMoverCount	UDINT	0		%IB*	Number of movers ha... a movement planne...
Error	BOOL	FALSE			Flag indicating a PlanarTrack error.
ErrorId	UDINT	0			Error id indicating the PlanarTrack error type.
track_three	MC_PlanarTrack				
MCTOPLC_STD	CDT_MCTOPLC_PLA...			%I*	Track data that is tran...rred from the Planar...
TrackOID	OTCID	16#05120010		%IB*	Object id of the planar track.
GroupOID	OTCID	16#00000000		%IB*	Object id of the planar group the track is in.
State	MC_PLANAR_STATE	Enabled		%IB*	State of the planar track, e.g. disabled.
OperationMode	MC_PLANAR_TRACK...	Standing		%IB*	Operation mode of the...nar track, e.g. movi...
MoverCountOnTrack	UDINT	0		%IB*	Number of movers coupled to this track.
MovingMoverCount	UDINT	0		%IB*	Number of movers ha... a movement planne...
Error	BOOL	FALSE			Flag indicating a PlanarTrack error.
ErrorId	UDINT	0			Error id indicating the PlanarTrack error type.
track_four	MC_PlanarTrack				
MCTOPLC_STD	CDT_MCTOPLC_PLA...			%I*	Track data that is tran...rred from the Planar...
TrackOID	OTCID	16#05120010		%IB*	Object id of the planar track.
GroupOID	OTCID	16#00000000		%IB*	Object id of the planar group the track is in.
State	MC_PLANAR_STATE	Enabled		%IB*	State of the planar track, e.g. disabled.
OperationMode	MC_PLANAR_TRACK...	Standing		%IB*	Operation mode of the...nar track, e.g. movi...
MoverCountOnTrack	UDINT	0		%IB*	Number of movers coupled to this track.
MovingMoverCount	UDINT	0		%IB*	Number of movers ha... a movement planne...
Error	BOOL	FALSE			Flag indicating a PlanarTrack error.
ErrorId	UDINT	0			Error id indicating the PlanarTrack error type.
state	UDINT	6			

每条轨道的长度都在 MC 项目中 TCOM 对象的在线参数中。

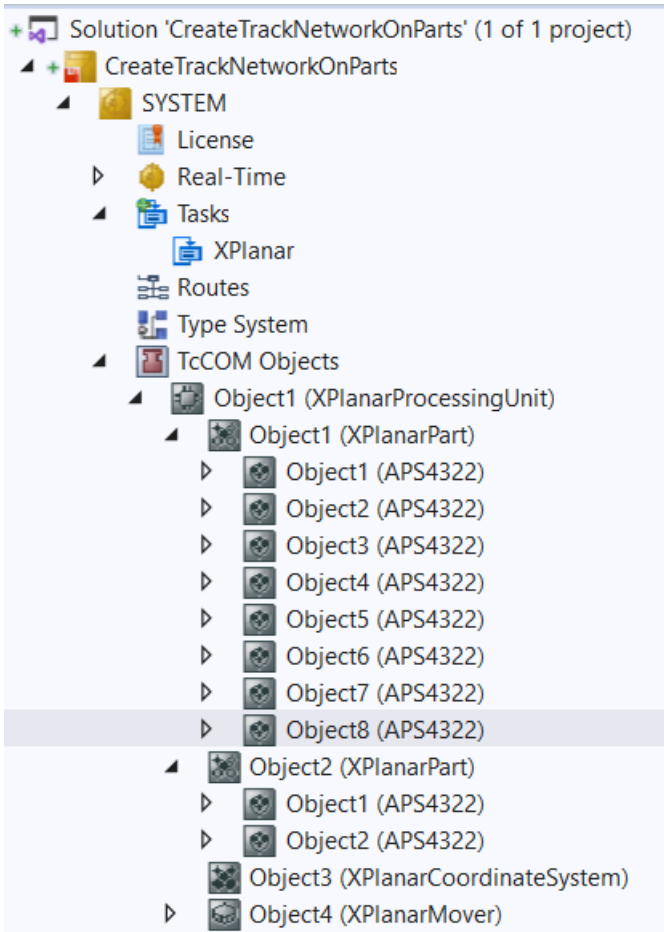


6.2.12 示例“将 Planar 轨道与 Planar 部件上的网络互联互通”

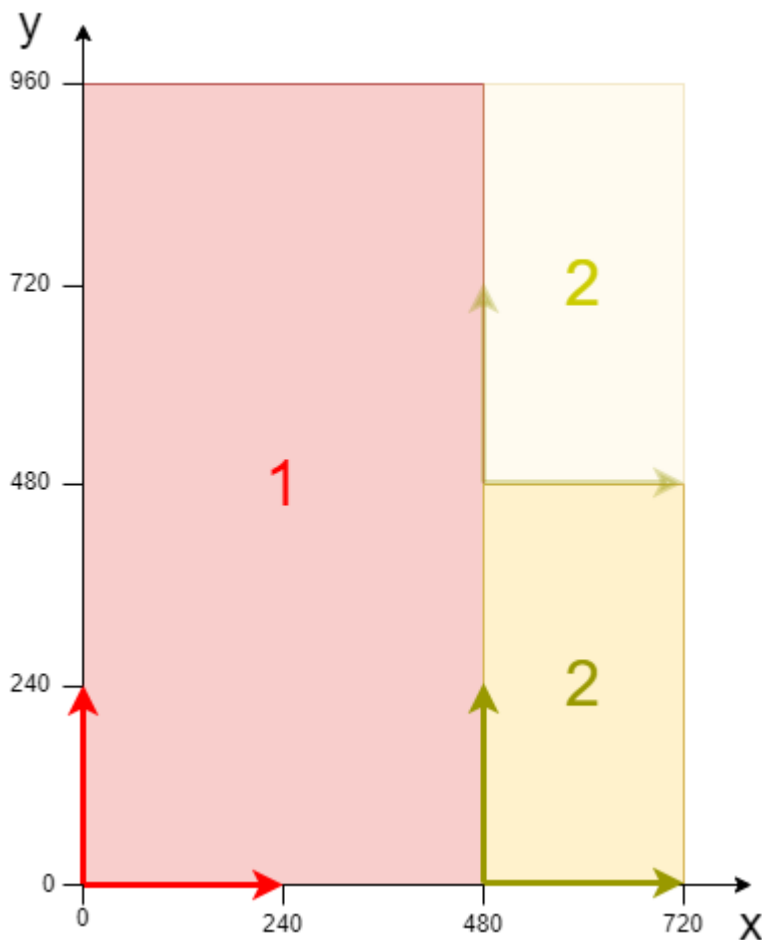
在这个例子中，在移动的 Planar 部件上创建了 1 个 Planar 轨道网络。

起点

我们从包含完全配置的 XPlanar Processing Unit 的解决方案开始。在 XPlanar Processing Unit 下创建了 2 个部分，1 个坐标系和 1 个动子。第 1 部分下面有 8 个 Planar 模块，第 2 部分下面有 2 个 Planar 模块。



几何情况如下：2 个部件相邻，部件 2 可以占据 2 个位置。这意味着，根据部件位置的不同，部件之间有 2 种不同的互联互通。



本示例就是在这种配置的基础上开发的。XPlanar Processing Unit 的文档中介绍了初始化的步骤。

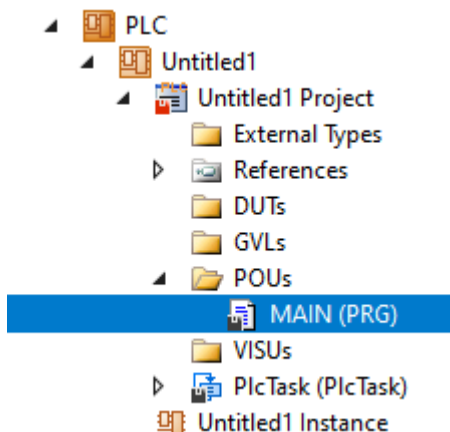
创建 Planar 轨道和 Planar 环境

1. 创建 Planar 环境，请参阅 [配置 \[► 94\]](#)。
2. 将初始参数 XPlanar Processing Unit OID 设置为 XPlanar Processing Unit 的对象 ID。这将激活所有 MC Configuration 对象（尤其是已创建的 Planar 动子）Part 的功能。
3. 通过 **Groups > Add New Item...** 添加 3 个 Planar 轨道，请参阅 [配置 \[► 39\]](#)。
4. 将 3 个轨道的初始参数“PartOID”设置为相应的部分。在这个例子中，第 1 和第 2 轨道位于第 1 部分，第 3 轨道位于第 2 部分。

创建 PLC

✓ 请参阅 [创建 PLC \[► 19\]](#) 下的初步步骤。

1. 通过 MAIN 创建所需数量的动子（“MC_PlanarMover”）和轨道（“MC_PlanarTrack”）。



⇒ 这些代表 MC Configuration 中的动子和轨道。

2. 创建如下图所示的 3 个 Planar 轨道、1 个状态机的任意状态变量和 2 个轨道的辅助位置。

```
PROGRAM MAIN
VAR
    track_one, track_two, track_three : MC_PlanarTrack;
    state : UDINT;
    pos1, pos2 : PositionXYC;
    part_one_oid : OTCID := 16#01010020;
    part_two_oid : OTCID := 16#01010030;
    start_options : ST_StartFromTrackAdvancedOptions;
    end_options : ST_EndAtTrackAdvancedOptions;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 该程序代码在部件上创建了 3 个轨道。根据活动部件的位置，轨道会在不同位置互联互通。首先，在第 2 部分创建轨道 3（状态=0）。然后，轨道 2 从轨道 3 开始，第 2 部分位于上部位置（索引=2）。轨道 2 也在轨道 3 处结束，但第 2 部分处于较低位置（索引=1）。最后，轨道 1 在轨道 3 处开始，第 2 部分在较低位置（索引=1），完成后，第 2 部分在较高位置（索引=2）。

```
CASE state OF
0:
    pos1.SetValuesXYCReferenceId(0, 360, 0, part_two_oid);
    pos2.SetValuesXYCReferenceId(80, 360, 0, part_two_oid);
    track_three.AppendLine(0, pos1, pos2);
    pos1.SetValuesXYCReferenceId(120, 320, 0, part_two_oid);
    pos2.SetValuesXYCReferenceId(120, 160, 0, part_two_oid);
    track_three.AppendLine(0, pos1, pos2);
    pos1.SetValuesXYCReferenceId(80, 120, 0, part_two_oid);
    pos2.SetValuesXYCReferenceId(0, 120, 0, part_two_oid);
    track_three.AppendLine(0, pos1, pos2);
    state := 1;
1:
    start_options.thisTrackPartPositionIndex := 1;
    start_options.otherTrackPartPositionIndex := 2;
    track_two.StartFromTrackAdvanced(0, track_three, start_options);
    pos1.SetValuesXYCReferenceId(440, 600, 0, part_one_oid);
    pos2.SetValuesXYCReferenceId(400, 600, 0, part_one_oid);
    track_two.AppendLine(0, pos1, pos2);
    pos1.SetValuesXYCReferenceId(360, 560, 0, part_one_oid);
    pos2.SetValuesXYCReferenceId(360, 400, 0, part_one_oid);
    track_two.AppendLine(0, pos1, pos2);
    pos1.SetValuesXYCReferenceId(400, 360, 0, part_one_oid);
    pos2.SetValuesXYCReferenceId(440, 360, 0, part_one_oid);
    track_two.AppendLine(0, pos1, pos2);
    end_options.thisTrackPartPositionIndex := 1;
    end_options.otherTrackPartPositionIndex := 1;
    track_two.EndAtTrackAdvanced(0, track_three, end_options);
    state := 2;
2:
    start_options.thisTrackPartPositionIndex := 1;
    start_options.otherTrackPartPositionIndex := 1;
    track_one.StartFromTrackAdvanced(0, track_three, start_options);
    pos1.SetValuesXYCReferenceId(440, 120, 0, part_one_oid);
    pos2.SetValuesXYCReferenceId(160, 120, 0, part_one_oid);
    track_one.AppendLine(0, pos1, pos2);
    pos1.SetValuesXYCReferenceId(120, 160, 0, part_one_oid);
    pos2.SetValuesXYCReferenceId(120, 800, 0, part_one_oid);
    track_one.AppendLine(0, pos1, pos2);
    pos1.SetValuesXYCReferenceId(160, 840, 0, part_one_oid);
    pos2.SetValuesXYCReferenceId(440, 840, 0, part_one_oid);
    track_one.AppendLine(0, pos1, pos2);
    end_options.thisTrackPartPositionIndex := 1;
    end_options.otherTrackPartPositionIndex := 2;
    track_one.EndAtTrackAdvanced(0, track_three, end_options);
    state := 3;
END_CASE
```

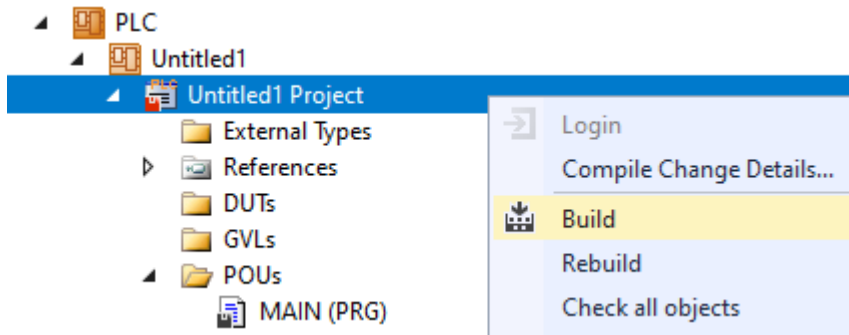
发送命令

4. 要发送命令，必须在 END_CASE 之后循环调用轨道的更新方法：

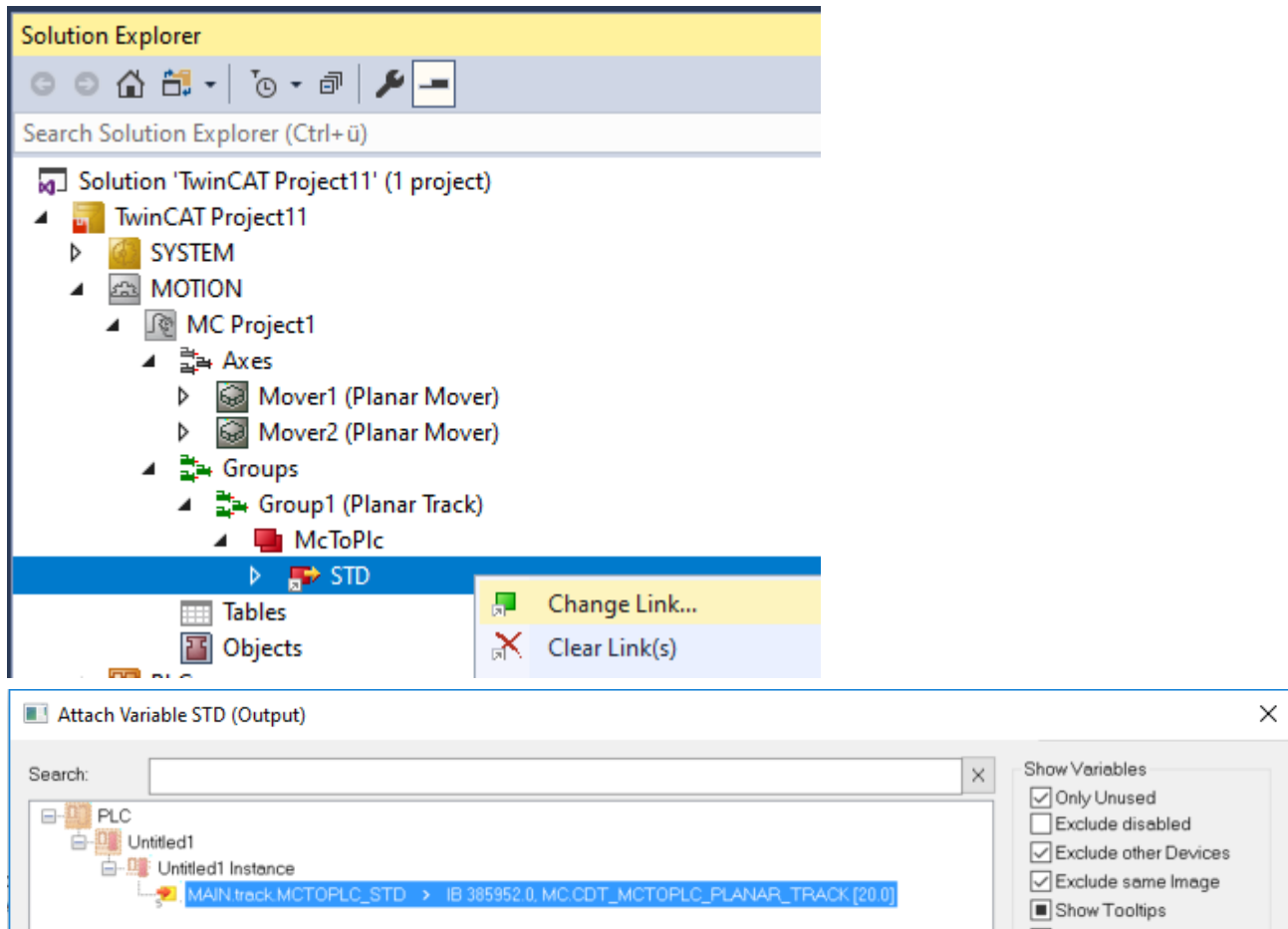
```
track_one.Update();
track_two.Update();
track_three.Update();
```

构建 PLC 时会创建“PLC mover”和“Track”的符号，然后将其链接到 MC Project 中的动子和轨道实例。

1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



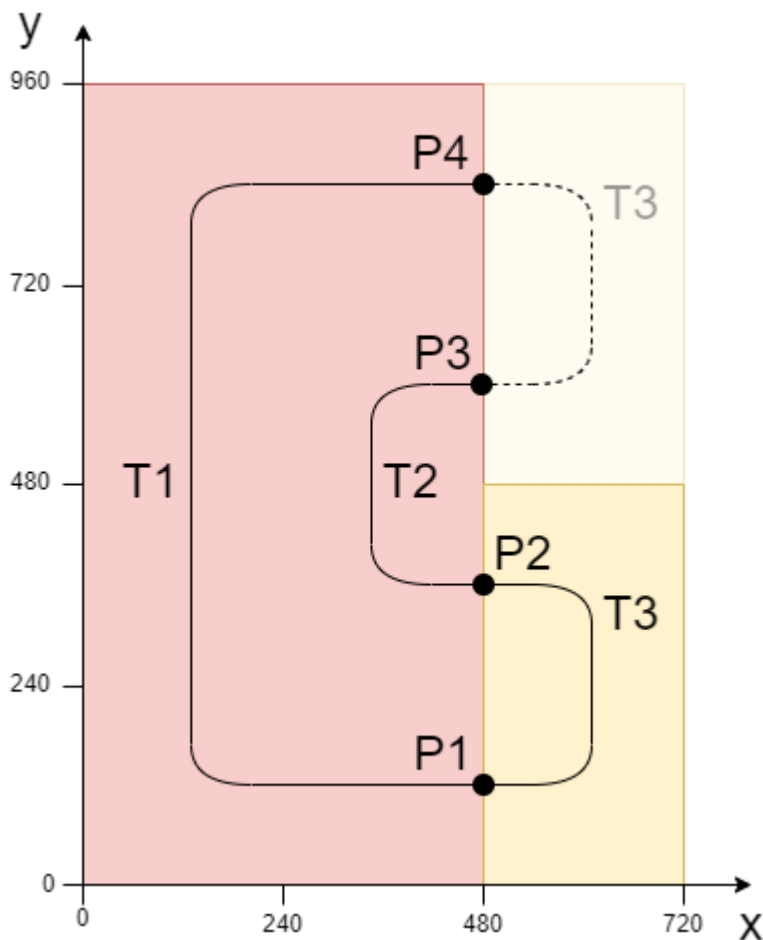
⇒ 现在可以通过以下对话框连接音轨。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机结束时 (state=3)，3 条轨道的配置如图所示。轨道 3 的起点与位于位置 2 的轨道 2 的终点互联互通，当部件 2 位于较低位置时，轨道 3 的终点与轨道 1 的起点和位置 2 互联互通。在上部位置，轨道 3 的起点与位于位置 4 的轨道 1 的终点互联互通，轨道 3 的终点与轨道 2 的起点和位置 1 互联互通。轨道 2 和轨道 3 的长度相同。



6.2.13 示例：“跟随 Planar 动子穿过 Track Network”

在这些说明的指导下，您将创建 1 个 TwinCAT 项目，在该项目中，1 个 Planar 轨道上的 Planar 动子将跟随同一 Planar 轨道上的前 1 个 Planar 动子在轨道网络中运行。

通过 `GearInPosOnTrackWithMasterMover` [► 148] 命令实现在轨道网络中的跟随，详见 [示例：“使轨道上的 Planar 动子与另一个 Planar 动子同步”](#) [► 70]。创建和构建 Planar 轨道网络的更多详情，请参阅 [示例“将 Planar 轨道与网络互联互通”](#) [► 73]。这个示例很短，是在上述示例的基础上编写的。

创建 Planar 动子

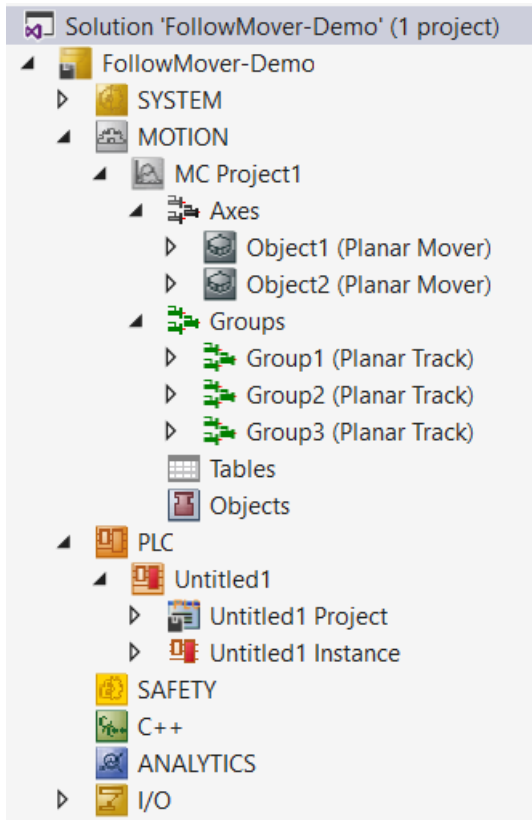
✓ 参阅 [配置](#) [► 16]。

1. 创建 2 个 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

创建 Planar 轨道

3. 通过 `Groups > Add New Item...` 添加 3 个 Planar 轨道，请参阅 [配置](#) [► 39]。

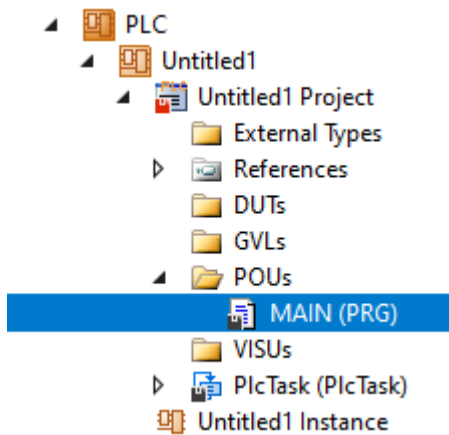
⇒ 解决方案资源管理器中有以下条目：



创建 PLC

✓ 请参阅 [创建 PLC \[▶_19\]](#) 下的初步步骤。

1. 通过 **MAIN** 创建所需数量的动子 (“MC_PlanarMover”) 和轨道 (“MC_PlanarTrack”)。

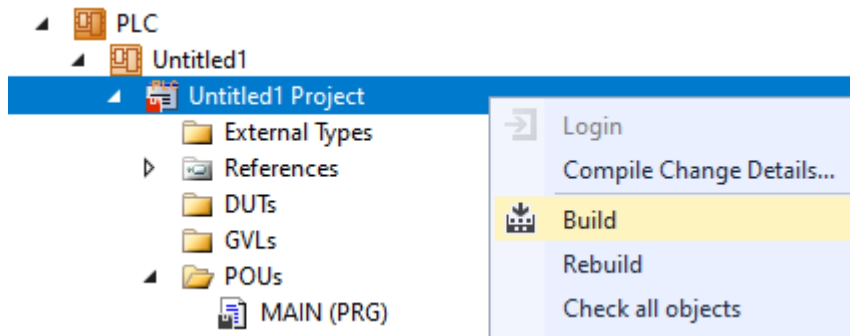


⇒ 这些代表 MC Configuration 中的动子和轨道。

2. 创建以下任意变量。

```
PROGRAM MAIN
VAR
  master_mover      : MC_PlanarMover;
  slave_mover       : MC_PlanarMover;
  track_in          : MC_PlanarTrack;
  track_out1        : MC_PlanarTrack;
  track_out2        : MC_PlanarTrack;
  move_feedback     : MC_PlanarFeedback;
  options           : ST_GearInPosOnTrackWithMasterMoverOptions;
  state             : UDINT;
  pos1, pos2        : PositionXYC;
END_VAR
```

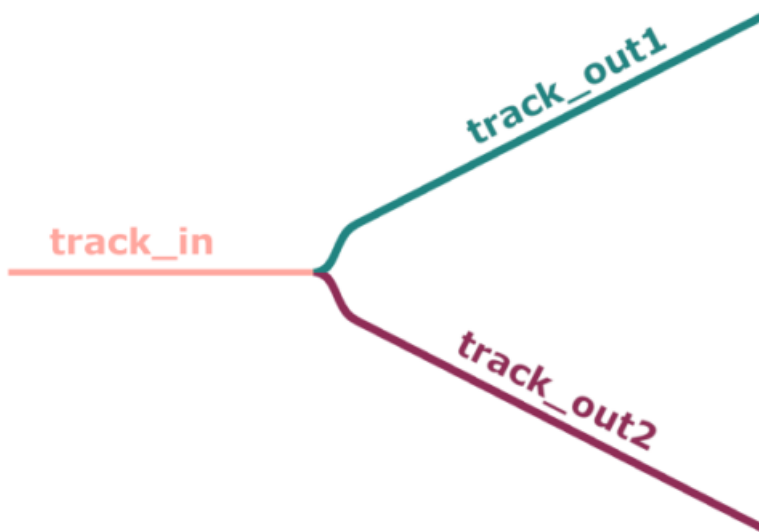
3. 构建 PLC，创建 “PLC mover”和“PLC track”的符号。



4. 连接 Planar 动子和 Planar 轨道（见 示例 “ Planar 动子加入并在轨道上移动” [► 48]）。

状态机编程

通过 MAIN 编程的以下状态机，首先对 Planar 轨道进行几何定义并激活（状态 0 至 7），使其代表以下开关状态配置：



在状态 8 至 19 中，2 个 Planar 动子被激活，耦合到岔道(track_in) 前面的 Planar 轨道上，并移动到位置 200 (master_mover) 或 0 (slave_mover)。然后命令 Master Planar Mover 将 500 定位在 2 条分支 Planar 轨道的上端 (track_out1)（状态 20）。最后，在状态 21 中，向 Slave Planar Mover 发送 `GearInPosOnTrackWithMasterMover` [► 148] 命令。与往常一样，Planar 对象会在 `END_CASE` 语句后循环更新。

```



CASE state OF
0:
  pos1.SetValuesXYC(100, 360, 0);
  pos2.SetValuesXYC(400, 360, 0);
  track_in.AppendLine(0, pos1, pos2);
  track_in.Enable(0);
  state := state + 1;
1:
  IF track_in.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
2:
  track_out1.StartFromTrack(0, track_in);
  state := state + 1;
3:
  pos1.SetValuesXYC(450, 410, 0);
  pos2.SetValuesXYC(860, 620, 0);
  track_out1.AppendLine(0, pos1, pos2);
  track_out1.Enable(0);
  state := state + 1;
4:
  IF track_out1.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
5:
  track_out2.StartFromTrack(0, track_in);

```


```
state := state + 1;
6:
  pos1.SetValuesXYC(450, 310, 0);
  pos2.SetValuesXYC(860, 100, 0);
  track_out2.AppendLine(0, pos1, pos2);
  track_out2.Enable(0);
  state := state + 1;
7:
  IF track_out2.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
8:
  master_mover.Enable(0);
  state := state + 1;
9:
  IF master_mover.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
10:
  master_mover.JoinTrack(0, track_in, 0, 0);
  state := state + 1;
11:
  IF master_mover.MCTOPLC_STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
    state := state + 1;
  END_IF
12:
  master_mover.MoveOnTrack(move_feedback, track_in, 200, 0, 0);
  state := state + 1;
13:
  IF move_feedback.Done THEN
    state := state + 1;
  END_IF
14:
  slave_mover.Enable(0);
  state := state + 1;
15:
  IF slave_mover.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
16:
  slave_mover.JoinTrack(0, track_in, 0, 0);
  state := state + 1;
17:
  IF slave_mover.MCTOPLC_STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
    state := state + 1;
  END_IF
18:
  slave_mover.MoveOnTrack(move_feedback, track_in, 0, 0, 0);
  state := state + 1;
19:
  IF move_feedback.Done THEN
    state := state + 1;
  END_IF
20:
  master_mover.MoveOnTrack(0, track_out1, 500, 0, 0);
  state := state + 1;
21:
  options.followMover := TRUE;
  slave_mover.GearInPosOnTrackWithMasterMover(0, master_mover, 0, 210, track_in, 10, track_in, 0,
options);
  state := state + 1;
END_CASE

master_mover.Update();
slave_mover.Update();
track_in.Update();
track_out1.Update();
track_out2.Update();
move_feedback.Update();
```

激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。

3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

Master Planar Mover 将移动到指定 Planar 轨道上的给定目标位置（本例中为 500），Slave Planar Mover 将跟随其移动。Planar 动子的位置可以在实时视图中跟踪（点击按钮 ）。

由于在状态 21 的函数调用中指定了主动子的位置 210 和从动子的位置 10 作为 2 个 Planar 动子的同步位置，因此 Slave Planar Mover 将以 200 的距离跟随其主动子通过网络。它停在 2 条分支 Planar 轨道（主 Planar 动子也位于其上）上部的 300 号位置，可在在线视图中查看：

Expression	Type	Value
[-] master_mover	MC_PlanarMover	
+ PLCTOMC	CDT_PLCTOMC...	
[-] MCTOPLC	CDT_MCTOPLC...	
+ STD	REFERENCE TO...	
+ SET	REFERENCE TO...	
+ ACT	REFERENCE TO...	
+ COORDMODE	REFERENCE TO...	
[-] SETONTRACK	REFERENCE TO...	
+ SetPos	LREAL	499.9999999...
+ SetVelo	LREAL	0
+ SetAcc	LREAL	0
+ SetJerk	LREAL	0
+ TrackOID	OTCID	16#05120020
+ Error	BOOL	FALSE
+ ErrorId	UDINT	0
[-] slave_mover	MC_PlanarMover	
+ PLCTOMC	CDT_PLCTOMC...	
[-] MCTOPLC	CDT_MCTOPLC...	
+ STD	REFERENCE TO...	
+ SET	REFERENCE TO...	
+ ACT	REFERENCE TO...	
+ COORDMODE	REFERENCE TO...	
[-] SETONTRACK	REFERENCE TO...	
+ SetPos	LREAL	299.9999999...
+ SetVelo	LREAL	0
+ SetAcc	LREAL	0
+ SetJerk	LREAL	0
+ TrackOID	OTCID	16#05120020
+ Error	BOOL	FALSE
+ ErrorId	UDINT	0

请注意，在 Options 对象中设置“FollowMover”选项并在状态 21 的函数调用中传递该选项，可以避免指定 [PlanarTrackTrail \[► 118\]](#) 对象。Slave Planar Mover 在网络中应走的路径无需明确确定，因为它会自动跟随 Master Planar Mover，并在开关处转向正确的 Planar 轨道。通过设置选项，这种行为也会在更大的网络中重现，Master Planar Mover 会在多个轨道边界之间移动。

6.2.14 StartFromTrackAdvanced”和”EndAtTrackAdvanced”命令的选项

正如导言和示例中所述，[StartFromTrack \[► 165\]](#) 或 [EndAtTrack \[► 165\]](#) 命令的含义并不总是很明确。因此，这 2 个命令都提供了“Advanced”变体（[StartFromTrackAdvanced \[► 166\]](#) 和 [EndAtTrackAdvanced \[► 167\]](#)），并通过选项参数进行了扩展。该扩展名也映射在 Init 参数中。

选项参数由 3 个部分组

成：“thisTrackPartPositionIndex”“otherTrackPartPositionIndex”和“linkOnlyInSpecifiedPartPositions”。第 1 个组件“thisTrackPartPositionIndex”指定了被调用轨道所在部件的位置。必须指定位置的唯一索引（索引）。第 2 个分量对于作为参数传递的轨道具有相同的意义。第 3 个分量指定 2 条轨道是只在指定位置互联互通，还是也在所有其他几何上兼容的位置互联互通。

如果 2 个索引都为零，且标志 false，则行为与前面的命令器相同。只需将“0”作为“Advanced”命令的选项参数，也可以实现这一功能。

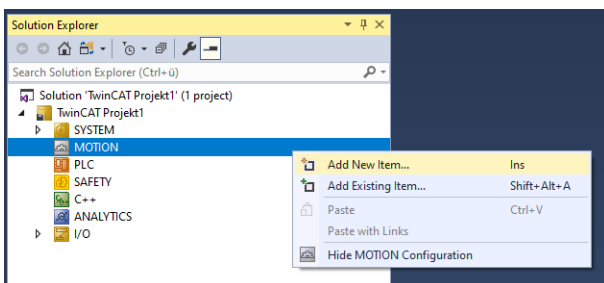
6.3 Planar 组

Planar 组是 1 个软件对象，可防止 Planar 动子之间的碰撞以及 Planar 动子与二维 XPlanar 定子区域边界的碰撞。为此，该组中所有对象的 2D 区域都被封锁。当向动子传输运动命令时，会向 Planar 组申请所需的区域，如果该区域会与已预留的区域相撞，则运动命令会被拒绝。如果运动命令可以执行，则该区域将被添加到预留区域集，并被相应地封锁。

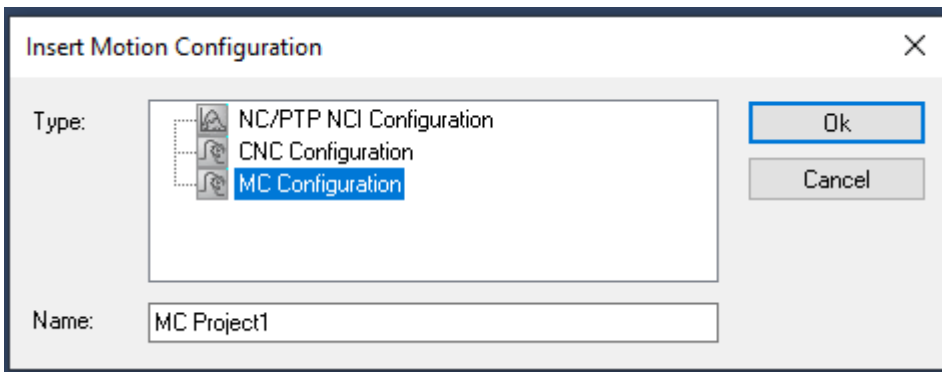
6.3.1 配置

✓ 要创建 Planar 组，必须先创建 MC Configuration。

1. 选择 MOTION > Add New Item...

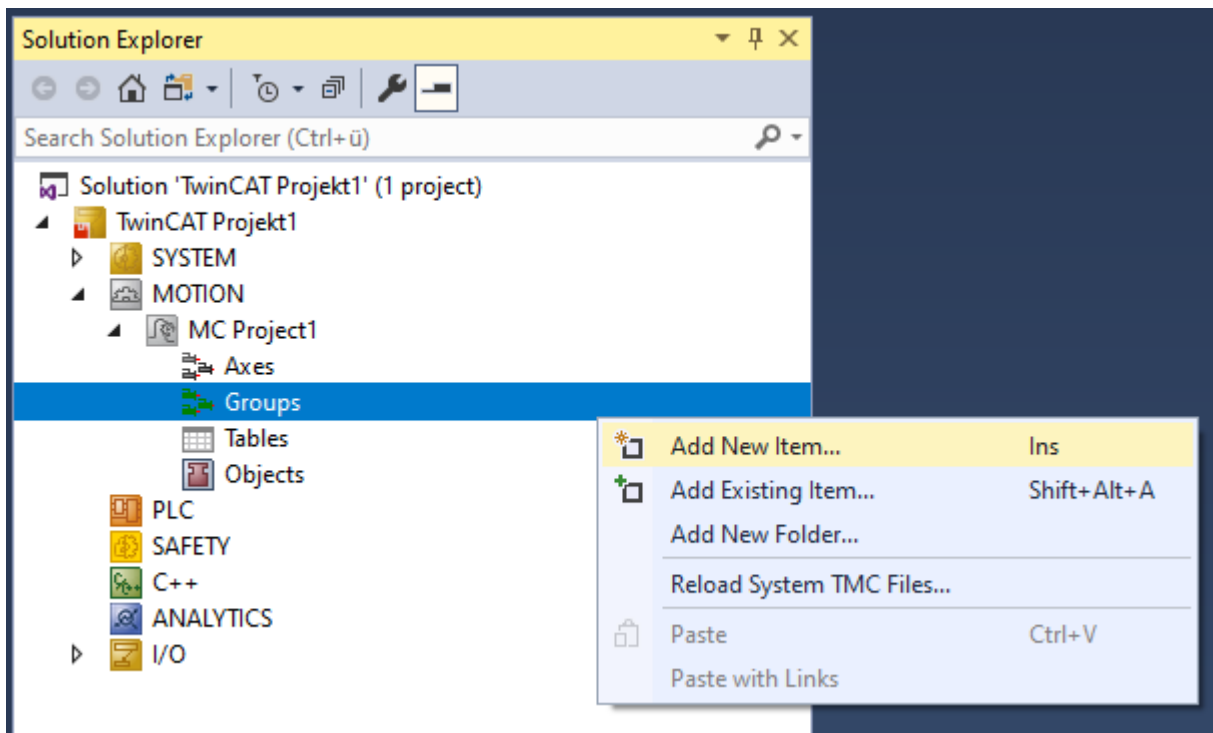


2. 在以下对话框中，选择 MC Configuration 并按 OK 确认。



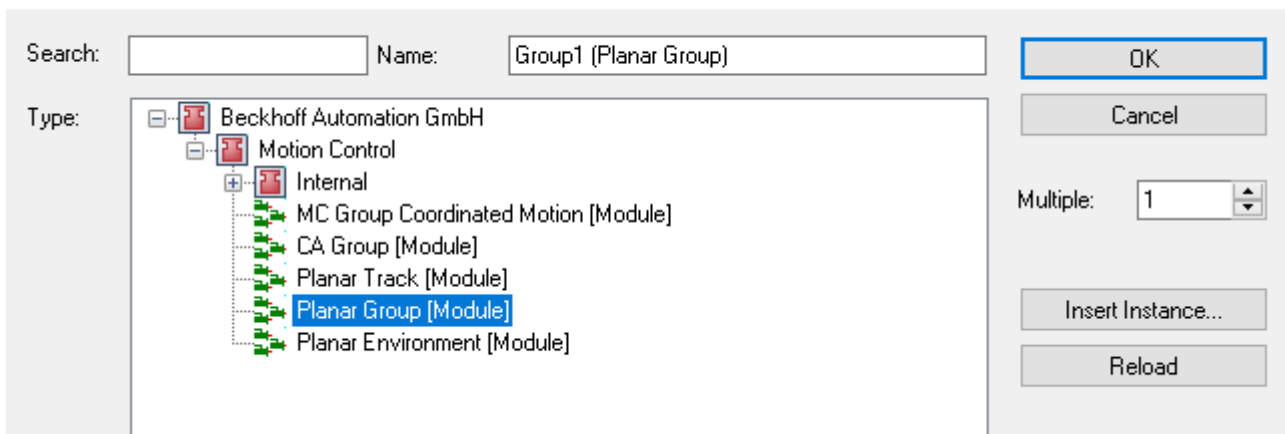
⇒ 您创建了 1 个 MC Project。

3. 选择 MC Project > Groups > Add New Item...



4. 在下面的对话框中，创建 1 个（或多个）Planar 组，并按 **OK** 确认。

Insert TcCom Object



⇒ Planar 组现已创建，可以进行参数设置。

打开详细说明

- 在树中选择 Planar 组并双击。

各标签的含义

对象： 一般信息（名称、类型、ID 等）可在此显示。

Object	Parameter (Init)	Parameter (Online)	Data Area
Object Id:	<input type="text" value="0x05120020"/>	<input type="checkbox"/>	Copy TMI to Target
Object Name:	<input type="text" value="Group2 (Planar Group)"/>	<input type="checkbox"/>	Share TMC Description
Type Name:	<input type="text" value="Planar Group"/>	<input type="checkbox"/>	Keep Unrestored Link Info
GUID:	<input type="text" value="C2D951C9-E4FA-409C-8B5E-58B3BE9EB676"/>		
Class Id:	<input type="text" value="050300C9-0000-0000-F000-000000000064"/>		
Class Factory:	<input type="text" value="TcNc3"/>		
TMI/TMC	<input type="text" value="C:\TwinCAT\3.1\Config\Modules\TcNc3.tmc"/>		
Parent Id:	<input type="text" value="0x05100010"/>	<input type="checkbox"/>	Auto Reload TMI/TMC
Init Sequence:	<input type="text" value="PSO"/>		

参数 (Init)： 组没有初始参数。

参数 (联机)： 此处显示组 (动子、轨道、环境) 中管理的对象数量。组的状态也会显示出来。

Object	Parameter (Init)	Parameter (Online)	Data Area
	Name	Online	CS Type PTC... Comment
	Group object count	0	<input type="checkbox"/> UDINT 0x0... Number of objects in the group, read only.
	State	Disabled	<input type="checkbox"/> MC.MC_PL... 0x0... State, read only.

数据区： 显示组与 PLC 轨道通信的内存区域。

Object	Parameter (Init)	Parameter (Online)	Data Area
	Area No	Name	Type Size CS CD / Elements
+	1 (0)	McToPlc	OutputSrc 12 <input type="checkbox"/> <input type="checkbox"/> 1 Symbols

6.3.2 示例“创建和移动具有防撞组功能的 Planar 动子”

通过本指南，您将创建 1 个包含 2 个 Planar 动子和 1 个 Planar 组的 TwinCAT 项目。2 个动子都被添加到组中并被移动。

创建 Planar 动子

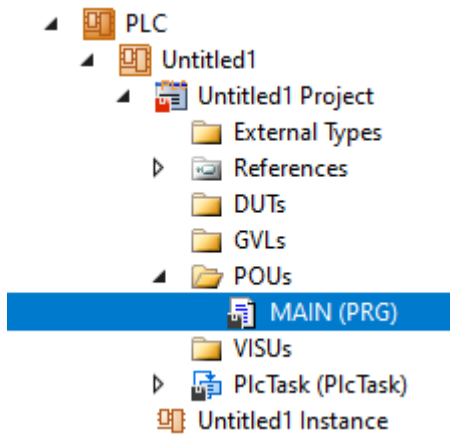
- ✓ 参阅 [配置 \[▶ 16\]](#)。
- 1. 创建 2 个 Planar 动子。
- 2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。
- 3. 将第 2 个动子的起始位置改为 $x = 240$ 。

创建 1 个 Planar 组

- 4. 通过 **Groups > Add New Item...** 添加 Planar 组，请参阅[配置 \[▶ 87\]](#)。

创建 PLC

- ✓ 请参阅 [创建 PLC \[▶ 19\]](#) 下的初步步骤。
- 1. 通过 **MAIN** 创建 2 个动子 (“MC_PlanarMover”) 和 1 个 Planar 组 “MC_PlanarGroup”。



⇒ 这些代表着 MC Configuration 中的动子和群体。

2. 如下图所示，为状态机创建 1 个状态变量，外加 2 个辅助位置，用于执行动子的 `MoveToPosition [D_144]` 命令。

```
PROGRAM MAIN
VAR
    mover_one, mover_two : MC_PlanarMover;
    group : MC_PlanarGroup;
    state : UDINT;
    pos1, pos2 : PositionXYC;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 该程序代码可激活组和 2 个动子。然后，2 个动子都被添加到组中。

```
CASE state OF
0:
    mover_one.Enable(0);
    mover_two.Enable(0);
    state := 1;
1:
    IF mover_one.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled
    AND mover_two.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    group.Enable(0);
    state := 3;
3:
    IF group.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 4;
    END_IF
4:
    mover_one.AddToGroup(0, group);
    mover_two.AddToGroup(0, group);
    state := 5;
5:
    IF mover_one.MCTOPLC.STD.GroupOID = group.MCTOPLC.STD.GroupOID
    AND mover_two.MCTOPLC.STD.GroupOID = group.MCTOPLC.STD.GroupOID THEN
        state := 6;
    END_IF
6:
    pos1.SetValuesXY(0, 240);
    pos2.SetValuesXY(0, 0);
    mover_one.MoveToPosition(0, pos1, 0, 0);
    mover_two.MoveToPosition(0, pos2, 0, 0);
    state := 7;
END_CASE
```

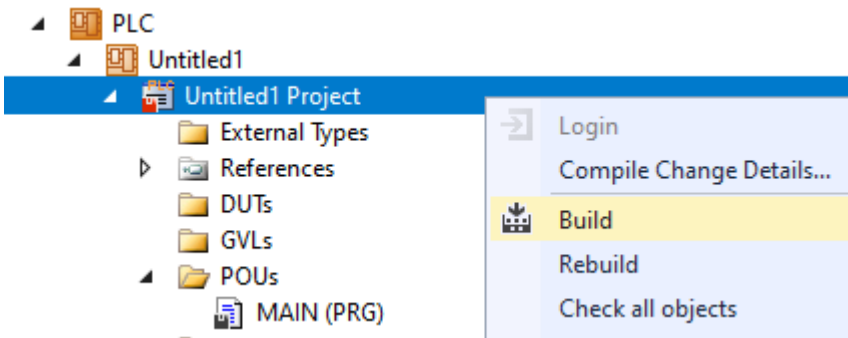
发送命令

4. 要发送命令，必须使用更新方法循环触发动子和组：

```
mover_one.Update();
mover_two.Update();
group.Update();
```

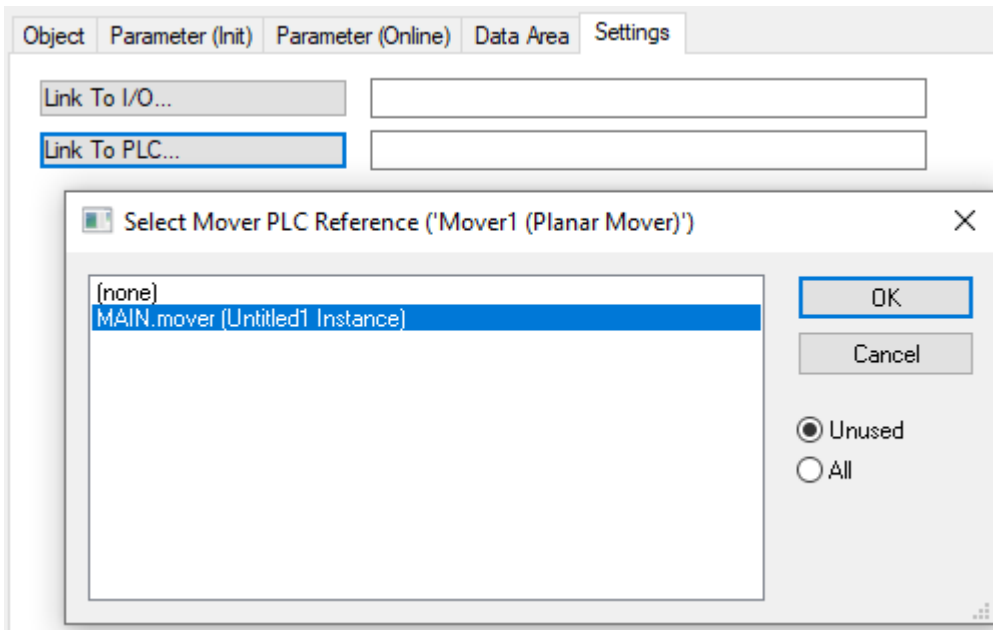
构建 PLC 时会创建 “PLC Mover”和“PLC Group”的符号，然后将其链接到 MC 项目中的动子或组实例。

5. 如要创建，请使用 `PLC > Untitled1 > Untitled1 Project > Build` 路径。

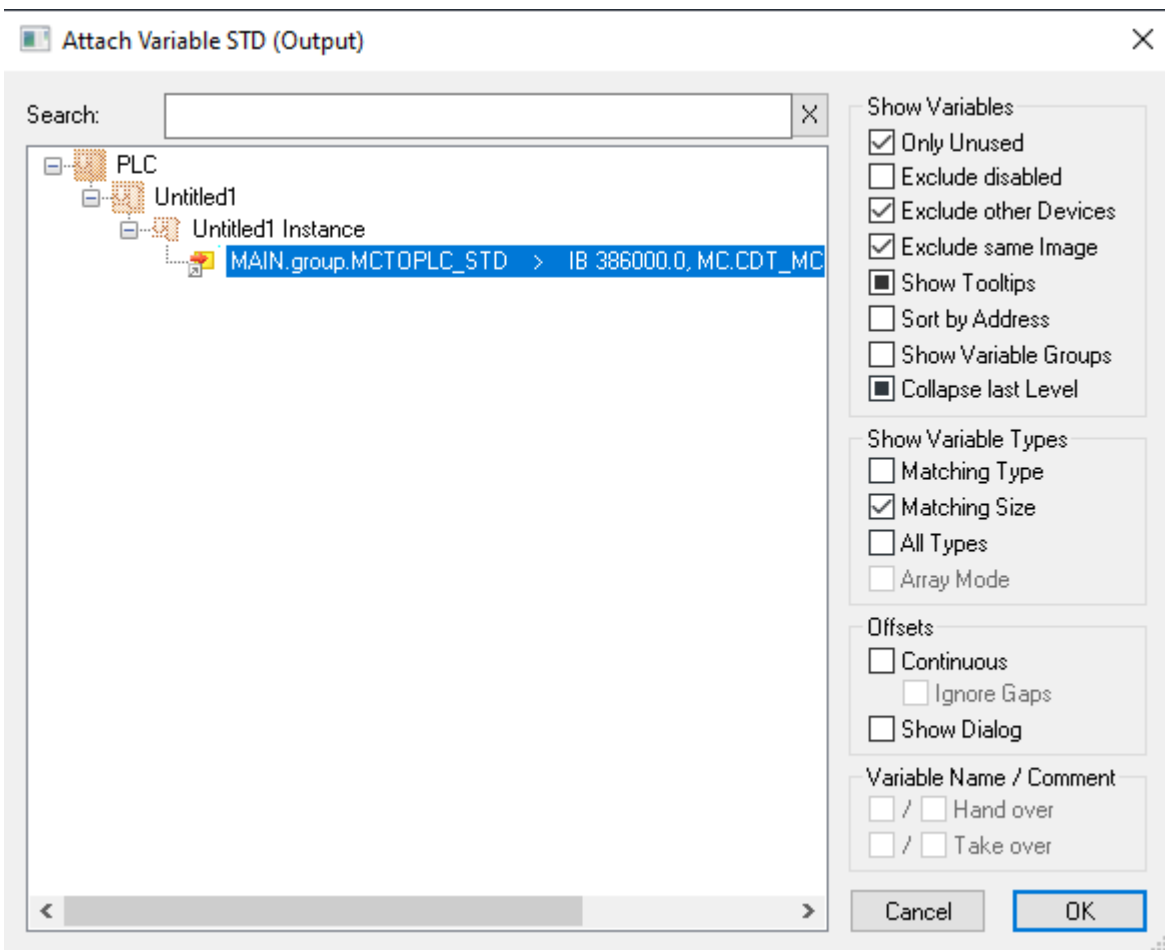
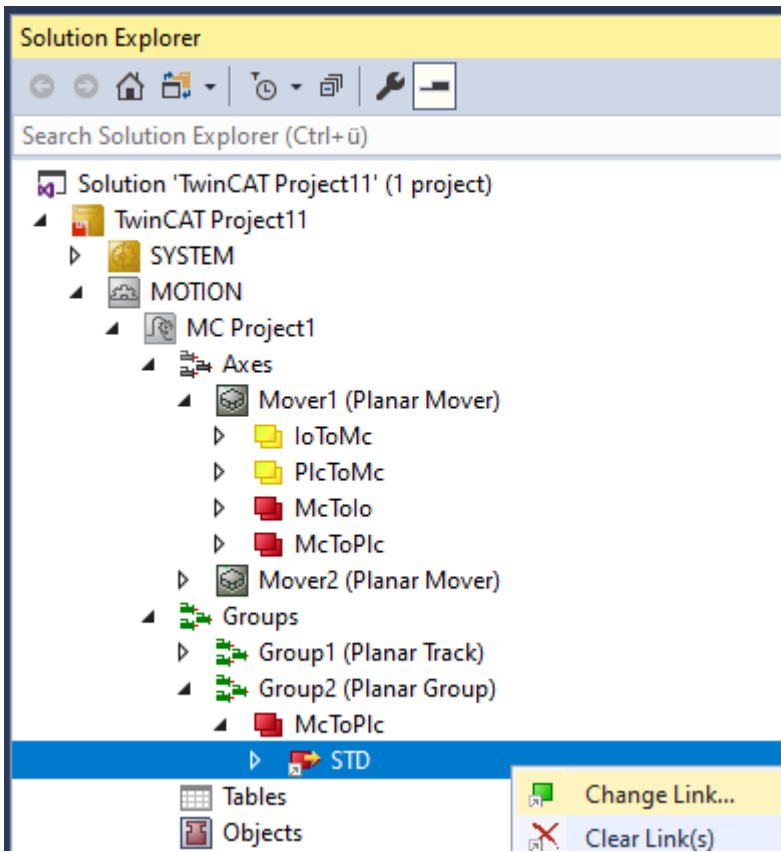


⇒ 随后，“MC Project”中的 Planar 动子可通过 **Setting** 选项卡上的 **Link To PLC...** 按钮进行链接。

6. 先双击动子一，再双击动子二。



⇒ 必须通过以下对话框分别链接组。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

日志记录并启动 PLC 后，您会发现在状态机结束时 (state=7)，动子并没有同时位于目标位置。动子 1 移动到了 x = 0 和 y = 240 处。动子二号没有移动到原点，因为动子一号仍然站在那里，命令因此被拒绝，因为两者都在 1 个共同的组中。

由于默认情况下，动子的动态限值相当高，因此日志记录后的位置变化可能难以用肉眼跟踪。有关动态限值，请参阅 Planar 动子 [16]。

Expression	Type	Value	Prepared value	Address	Comment
mover_one	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLA...			%Q*	Mover data that is tra...red from the Plana...
MCTOPLC	CDT_MCTOPLC_PLA...			%I*	Mover data that is tra...red from the Plana...
STD	REFERENCE TO CDT...			%IB*	Mover standard data t...is transferred from ...
SET	REFERENCE TO CDT...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	0			X coordinate.
y	LREAL	240			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	0			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	6630674363340 ...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT...			%IB*	Mover busy informatio...at is transferred fro...
Error	BOOL	FALSE			Flag indicating a PlanarMover error.
ErrorId	UDINT	0			Error id indicating the PlanarMover error type.
mover_two	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLA...			%Q*	Mover data that is tra...red from the Plana...
MCTOPLC	CDT_MCTOPLC_PLA...			%I*	Mover data that is tra...red from the Plana...
STD	REFERENCE TO CDT...			%IB*	Mover standard data t...is transferred from ...
SET	REFERENCE TO CDT...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	240			X coordinate.
y	LREAL	0			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	0			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	6630674363340 ...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT...			%IB*	Mover busy informatio...at is transferred fro...

6.4 Planar 环境

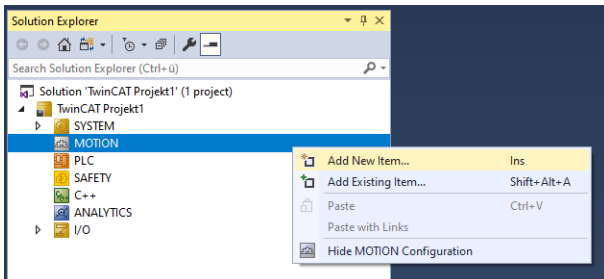
Planar 环境是表示二维 XPlanar 定子表面的软件对象。与 Planar 动子组成 1 组，可防止动子与表面边缘发生碰撞。

版本 V3.3.39 及以上：Planar 环境还支持边长为 160 或 320 mm 的平面模块（正方形或长方形）。此外，平面模块现在可以在 20 mm 的网格内相对移动。因此，边界元素现在更短（20 mm），外部边界中的边界元素也相应更多。

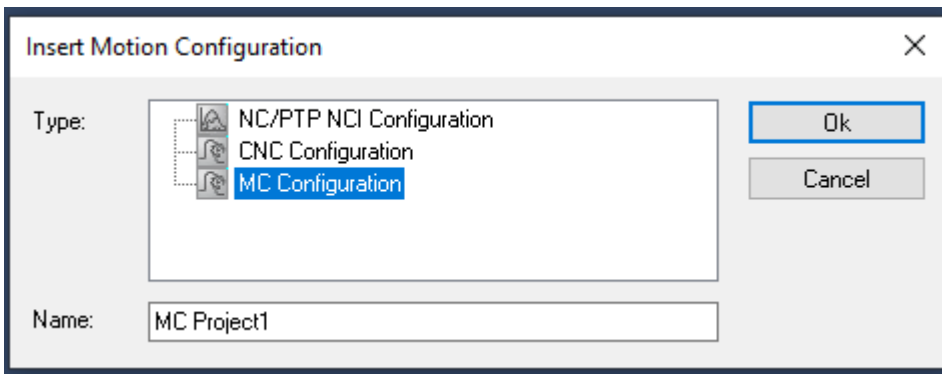
6.4.1 配置

✓ 要创建 Planar 环境，首先必须创建 MC Configuration。

1. 选择 MOTION > Add New Item...。

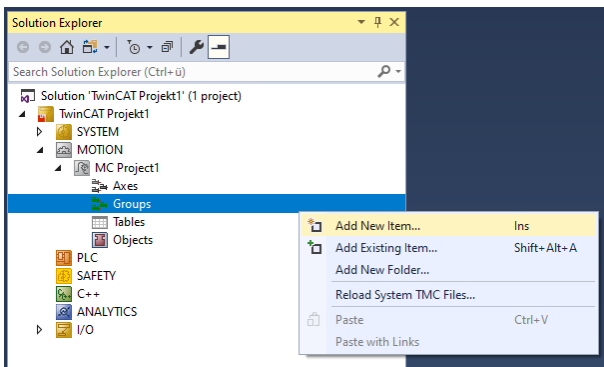


2. 在以下对话框中，选择 MC Configuration 并按 OK 确认。

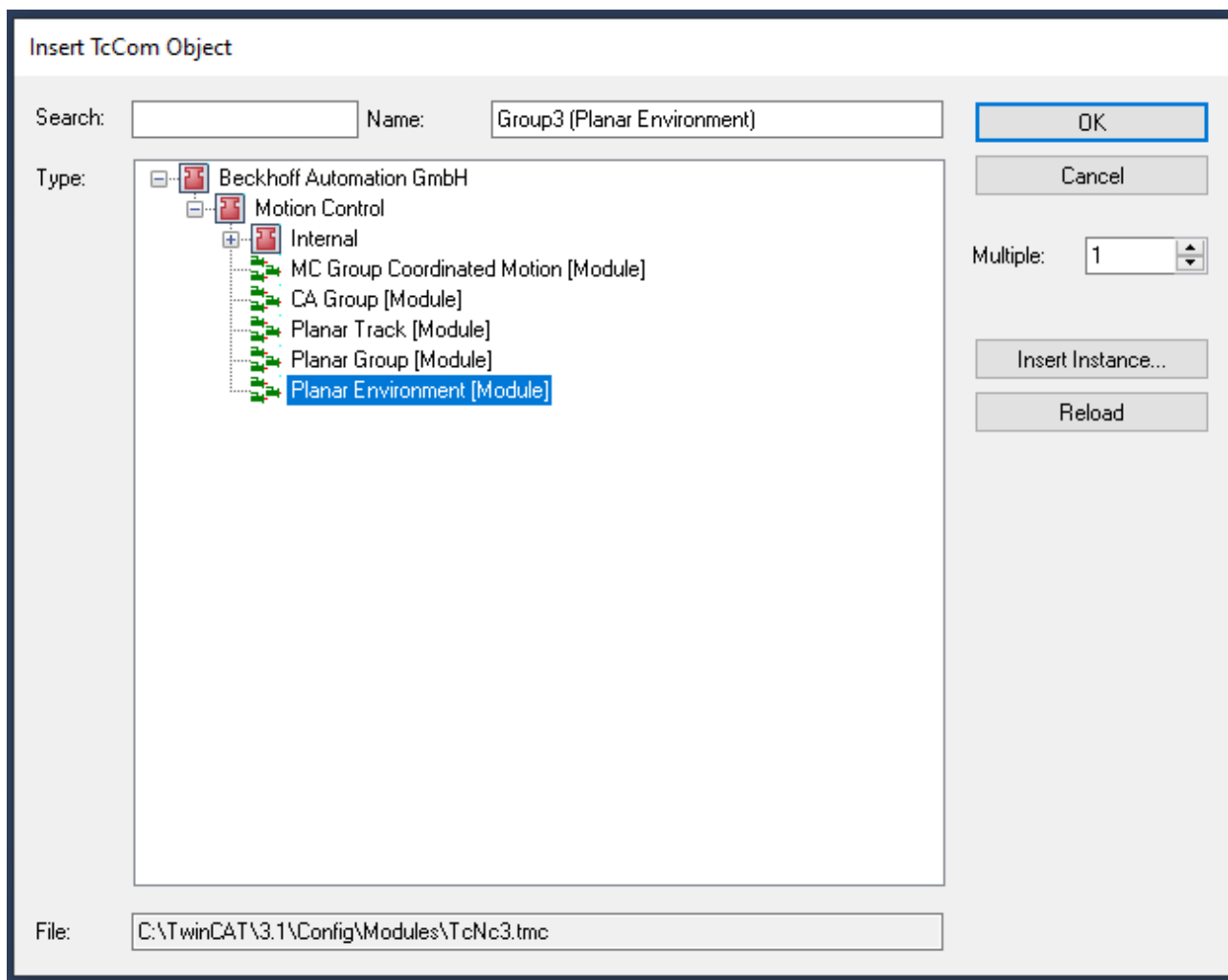


⇒ 您创建了 1 个 MC Project。

3. 选择 MC Project > Groups > Add New Item...。



4. 在以下对话框中，创建 1 个（或多个）Planar 环境，并按 OK 确认。



⇒ Planar 环境现已创建，可以进行参数设置。

打开详细描述

- 在树中选择 Planar 环境并双击。

各标签的用途

对象：一般信息（名称、类型、ID 等）可在此显示。

Object	Parameter (Init)	Parameter (Online)	Data Area
Object Id:	<input type="text" value="0x05120010"/>	<input type="checkbox"/> Copy TMI to Target	
Object Name:	<input type="text" value="Group1 (Planar Environmer)"/>	<input type="checkbox"/> Share TMC Description	
Type Name:	<input type="text" value="Planar Environment"/>	<input type="checkbox"/> Keep Unrestored Link Info	
GUID:	<input type="text" value="9A40A34B-DD2F-4271-8FEE-3ADE2271DB79"/>		
Class Id:	<input type="text" value="050300CB-0000-0000-F000-000000000064"/>		
Class Factory:	<input type="text" value="TcNc3"/>		
TMI/TMC	<input type="text" value="C:\TwinCAT\3.1\Config\Modules\TcNc3.tmc"/>		
Parent Id:	<input type="text" value="0x05100010"/>	<input type="checkbox"/> Auto Reload TMI/TMC	
Init Sequence:	<input type="text" value="PSO"/>		

参数 (Init)：指定用户可以更改的初始参数，以影响环境行为。

Object							
Parameter (Init)		Parameter (Online)		Data Area			
Name	Value	CS	Unit	Type	PTCID	Comment	
XPlanar processing unit OID	00000000	<input type="checkbox"/>		OTCID	0x050...	Object ID of the...	
Reset timeout for the parts of this environment	2.0	<input type="checkbox"/>	s	LREAL	0x050...		

Show Online Values Show Hidden Parameter

该环境的初始参数为“XPlanar processing unit OID”。当此项 (>0) 设置为 XPlanar 处理单元的目标检测 ID 时，环境会自动从 XPlanar 处理单元读取定子配置，并根据这些信息生成用于碰撞检测的边界元素。只要用户在 PLC 中调用 CreateBoundary() 命令，就会发生这种情况。

版本 V3.2.60 及以上：如果将“XPlanar Processing Unit OID”参数设置为 XPlanar Processing Unit OID，环境也会从 XPlanar Processing Unit 读取部件配置，并生成所有部件的内部表示。这既用于在环境处于 Planar 组时与部件边缘进行碰撞检查，也用于为所有部件（定子、轨道、组）提供完整的系统描述。

版本 V3.3.19 及以上：“状态机超时”参数规定了 Planar 部件保持 Enabling 或 Resetting 状态的最长时间，请参见 [Planar 对象状态图 \[► 9\]](#) 中的“过渡状态超时”。

参数（联机）：显示对象运行时的环境状态。

Object							
Parameter (Init)		Parameter (Online)		Data Area			
Name	Online	CS	Unit	Type	PTCID	Comment	
GroupOID	'object ID is invalid'	<input type="checkbox"/>		OTCID	0x0503...	Object id of the PlanarGroup the environment is in, read only.	
StatorCount	'object ID is invalid'	<input type="checkbox"/>		UDINT	0x0503...	Number of stators in the environment, read only.	
BoundaryElementCount	'object ID is invalid'	<input type="checkbox"/>		UDINT	0x0503...	Number of boundary elements, i.e. number of outer sides of all st	
PartCount	'object ID is invalid'	<input type="checkbox"/>		UDINT	0x0503...	Number of parts, i.e. collection of stators with fixed position relat	
- PlanarPartsInfo		<input type="checkbox"/>	0 (Array Elements)		0x0503...	Array containing the planar states, active position index, and activ	

这里显示了插入环境中的定子数量以及根据这些定子计算出的边界元素。

版本 V3.2.60 及以上：“PartCount”参数指定内部读出和创建的部件数。“PlanarPartsInfo”参数显示所有部件的信息。这些信息包括部件的对象 ID、部件的 Planar 状态、活动位置索引、由坐标系对象 ID 和 x/y 坐标组成的部件位置以及部件的“disableForced”标志。

数据区：显示组与 PLC 环境通信的内存区域。

Object						
Parameter (Init)		Parameter (Online)		Data Area		
Area No	Name	Type	Size	CS	CD / Elements	
+ 1 (0)	McToPlc	OutputSrc	16	<input type="checkbox"/>	<input type="checkbox"/> 1 Symbols	

6.4.2 示例“配置定子区域和边界”

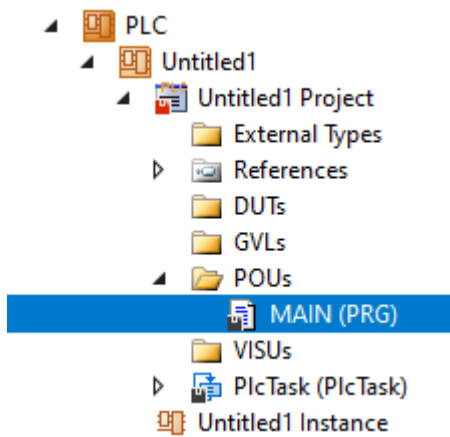
使用本指南，您将能够创建一个包含 Planar 环境的 TwinCAT 项目，并配置其定子表面和边界。

创建 Planar 环境

1. 创建 Planar 环境，请参阅 [配置 \[► 94\]](#)。

创建 PLC

- ✓ 见初步步骤 [创建 PLC \[► 19\]](#)。
1. 通过 MAIN 创建“MC_PlanarEnvironment”。



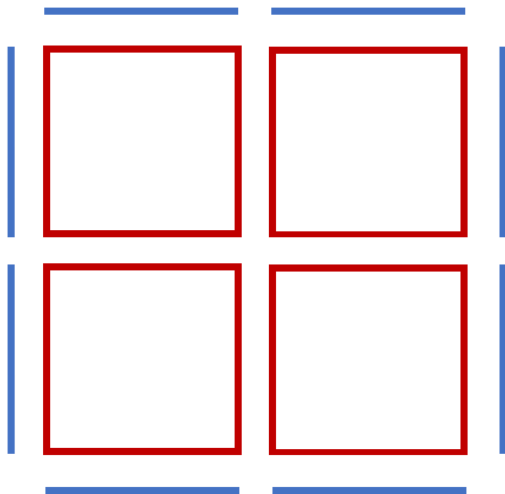
⇒ 这代表 MC Configuration 中的环境。

2. 如下图所示，为状态机创建 1 个状态任意变量。

```
PROGRAM MAIN
VAR
    environment : MC_PlanarEnvironment;
    state : UDINT;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 这段程序代码为环境添加了 4 个定子。每种情况都指定了方形定子的左下角（边长 240 mm）。然后，CreateBoundary() 会计算定子表面的外部边界。定子（红色）和边界元素（蓝色）的示意图如下。



```
CASE state OF
0:
    environment.AddStator(0,0.0,0.0);
    environment.AddStator(0,240.0,0.0);
    environment.AddStator(0,0.0,240.0);
    environment.AddStator(0,240.0,240.0);
    environment.CreateBoundary(0);
    state := 1;
END_CASE
```

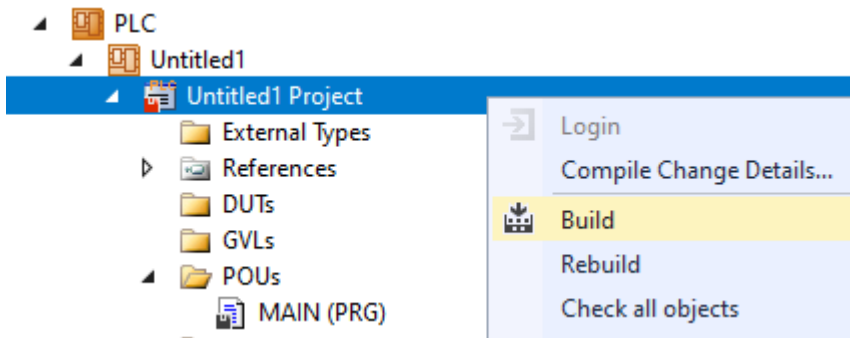
发送命令

4. 要发送命令，必须在 END_CASE 之后循环调用环境的更新方法：

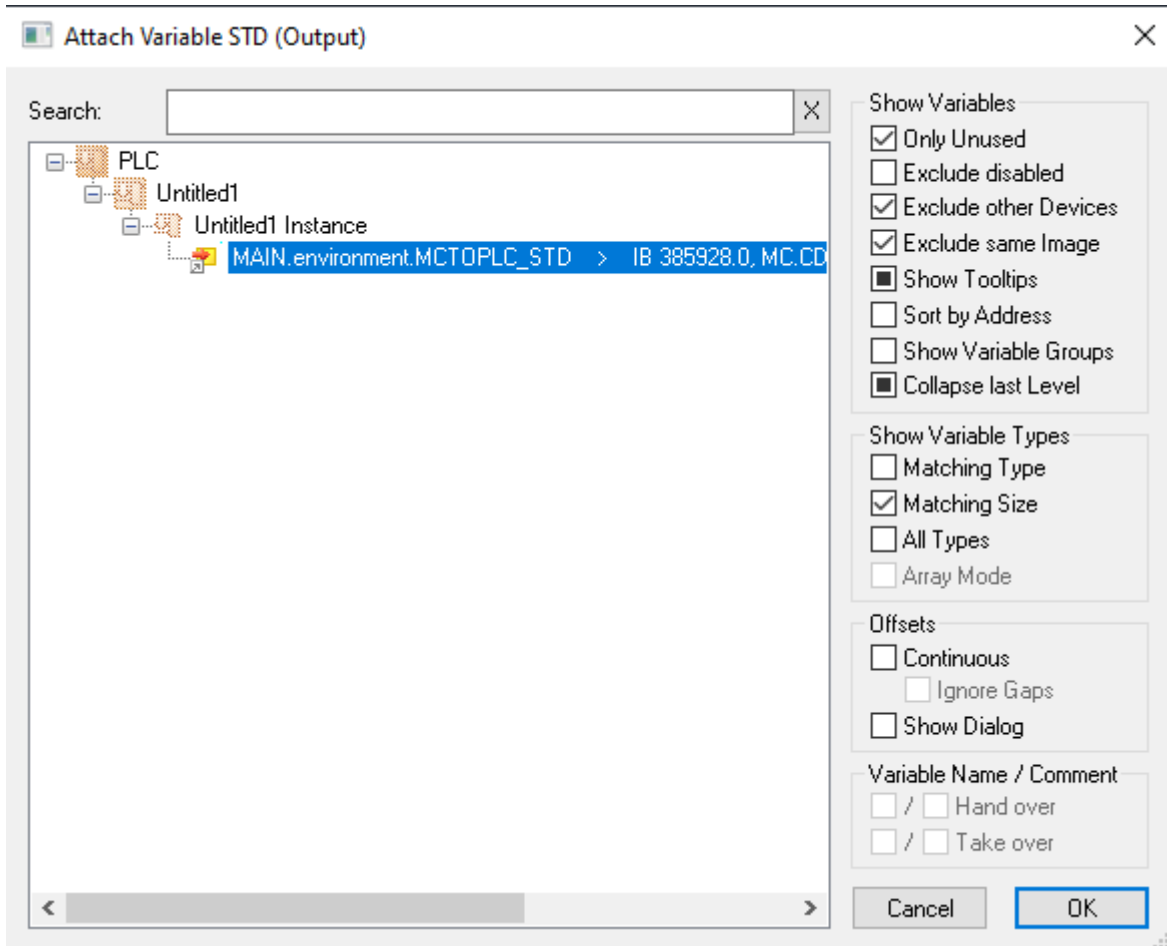
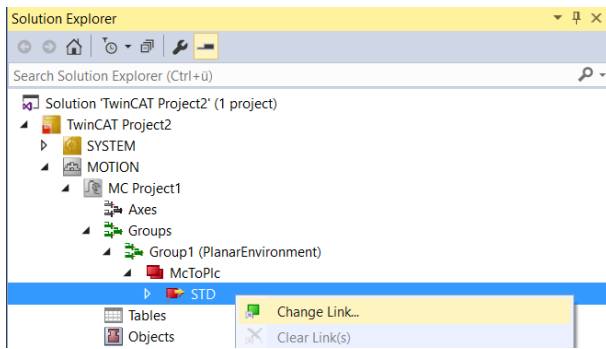
```
environment.Update();
```

在创建 PLC 时，会创建 1 个“PLC environment”符号，然后将其链接到 MC 项目中的 Planar 环境。



5. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



⇒ 然后就可以在“MC Project”中链接 Planar 环境。

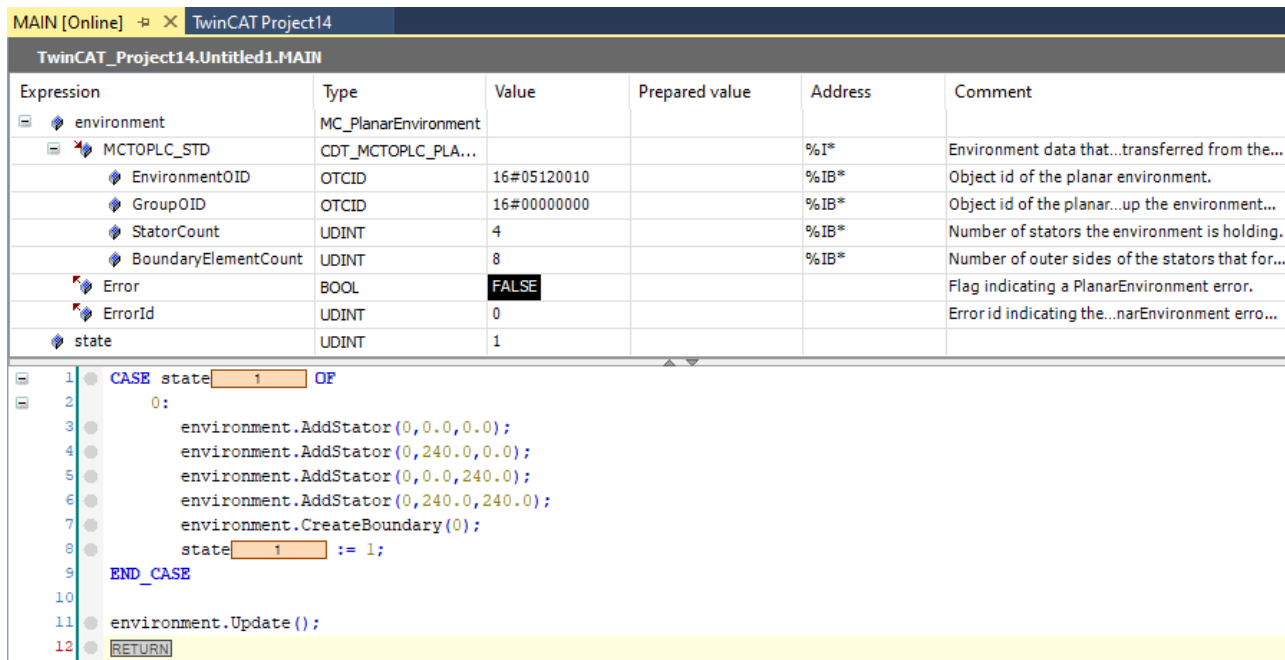


激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。

3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

状态机结束时，环境处于理想状态（状态 = 1）。



Expression	Type	Value	Prepared value	Address	Comment
environment	MC_PlanarEnvironment				
MCTOPLC_STD	CDT_MCTOPLC_PLA...			%I*	Environment data that...transferred from the...
EnvironmentOID	OTCID	16#05120010		%IB*	Object id of the planar environment.
GroupOID	OTCID	16#00000000		%IB*	Object id of the planar...up the environment...
StatorCount	UDINT	4		%IB*	Number of stators the environment is holding.
BoundaryElementCount	UDINT	8		%IB*	Number of outer sides of the stators that for...
Error	BOOL	FALSE			Flag indicating a PlanarEnvironment error.
ErrorId	UDINT	0			Error id indicating the...narEnvironment erro...
state	UDINT	1			

```

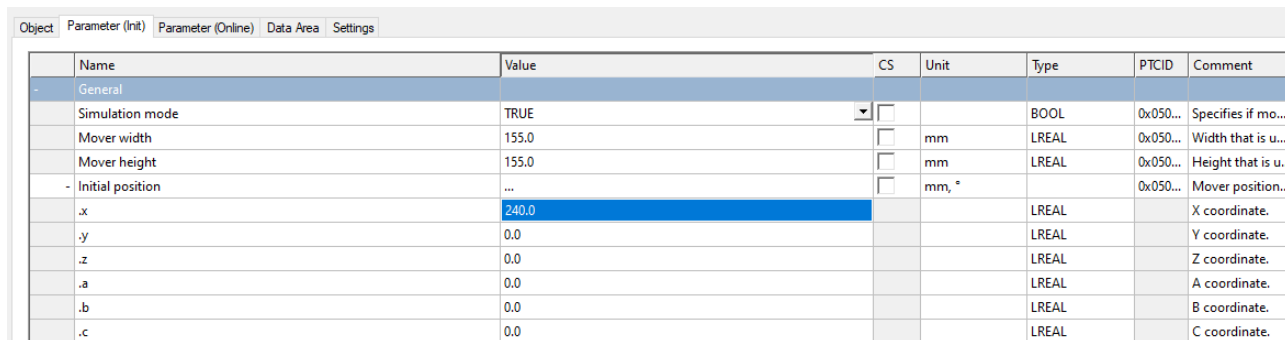
1 CASE state = 1 OF
2   0:
3     environment.AddStator(0,0.0,0.0);
4     environment.AddStator(0,240.0,0.0);
5     environment.AddStator(0,0.0,240.0);
6     environment.AddStator(0,240.0,240.0);
7     environment.CreateBoundary(0);
8     state := 1;
9 END_CASE
10
11 environment.Update();
12 RETURN
    
```

6.5 示例：“使用轨道和组创建并移动 Planar 动子”

通过本指南，您将创建 1 个 TwinCAT 项目，其中包括 2 个 Planar 动子、1 个 Planar 轨道和 1 个 Planar 组，并在轨道上和轨道旁移动动子。

创建 Planar 动子

- ✓ 请参见 [配置 \[▶ 16\]](#)。
1. 创建 2 个 Planar 动子。
 2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。
 3. 将第二个动子的起始位置改为 x=240。



Name	Value	CS	Unit	Type	PTCID	Comment
Simulation mode	TRUE	<input type="checkbox"/>		BOOL	0x050...	Specifies if mo...
Mover width	155.0	<input type="checkbox"/>	mm	LREAL	0x050...	Width that is u...
Mover height	155.0	<input type="checkbox"/>	mm	LREAL	0x050...	Height that is u...
Initial position	...	<input type="checkbox"/>	mm, °		0x050...	Mover position...
.x	240.0			LREAL		X coordinate.
.y	0.0			LREAL		Y coordinate.
.z	0.0			LREAL		Z coordinate.
.a	0.0			LREAL		A coordinate.
.b	0.0			LREAL		B coordinate.
.c	0.0			LREAL		C coordinate.

创建 Planar 轨道和 Planar 组

4. 通过 Groups > Add New Item... 添加 Planar 轨道，请参见 [配置 \[▶ 39\]](#)。
5. 用同样的方法处理 Planar 组。

创建 PLC

- ✓ 要控制动子、轨道和组，必须创建 1 个 PLC，用户可以从 PLC 向动子发出命令，请参见 [创建 PLC \[▶ 19\]](#)。

6. 通过 MAIN 创建 2 个动子 (MC_PlanarMovers [▶_142])、1 个 MC_PlanarTrack [▶_161] 和 1 个 MC_PlanarGroup [▶_141]。
 - ⇒ 它们分别代表 MC Configuration 中的动子、轨道和组。
7. 如下图所示，为状态机创建 1 个状态任意变量，并为轨道创建 2 个辅助位置。
8. 同时创建反馈，请参见 Planar_Feedback [▶_110]。
 - ⇒ 反馈可以与任何命令相关联。可提供有关命令执行情况和执行时间的详细信息。

```
PROGRAM MAIN
VAR
    mover_one, mover_two : MC_PlanarMover;
    track : MC_PlanarTrack;
    group : MC_PlanarGroup;
    state : UDINT;
    pos1, pos2 : PositionXYC;
    feedback : MC_PlanarFeedback;
END_VAR
```

9. 然后在 MAIN 中设计 1 个序列。

⇒ 该程序代码可创建并激活 1 个轨道、1 个组和 2 个动子。动子和轨道都被添加到组中。之后，动子 1 加入并在轨道上移动。移动时，我们会收到反馈信息，将拒绝执行命令的情况视为错误。由于动子 2 挡住了轨道（碰撞错误），命令被拒绝。

```
CASE state OF
0:
    pos1.SetValuesXY(0, 0);
    pos2.SetValuesXY(400, 0);
    track.AppendLine(0, pos1, pos2);
    track.Enable(0);
    group.Enable(0);
    state := 1;
1:
    IF track.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled
    AND group.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 2;
    END_IF
2:
    mover_one.Enable(0);
    mover_two.Enable(0);
    state := 3;
3:
    IF mover_one.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled
    AND mover_two.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 4;
    END_IF
4:
    mover_one.AddToGroup(0, group);
    mover_two.AddToGroup(0, group);
    track.AddToGroup(0, group);
    state := 5;
5:
    IF mover_one.MCTOPLC_STD.GroupOID > 0
    AND mover_two.MCTOPLC_STD.GroupOID > 0
    AND track.MCTOPLC_STD.GroupOID > 0 THEN
        state := 6;
    END_IF
6:
    mover_one.JoinTrack(0, track, 0, 0);
    state := 7;
7:
    IF mover_one.MCTOPLC_STD.CommandMode = MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
        state := 8;
    END_IF
8:
    mover_one.MoveOnTrack(feedback, 0, 150.0, 0, 0);
    pos2.SetValuesXY(240, 320);
    mover_two.MoveToPosition(0, pos2, 0, 0);
    state := 9;
9:
    IF mover_two.MCTOPLC.SET.SetPos.x >= 240 AND mover_two.MCTOPLC.SET.SetPos.y >= 320 THEN
        state := 10;
    END_IF
10:
    mover_one.MoveOnTrack(0, 0, 150.0, 0, 0);
    state := 11;
11:
    IF mover_one.MCTOPLC.SETONTRACK.SetPos >= 150.0 THEN
```

```

state := 12;
END_IF
END_CASE

```

发送命令

10. 要发送命令，必须在 END_CASE 之后循环调用动子、轨道和组的更新方法：

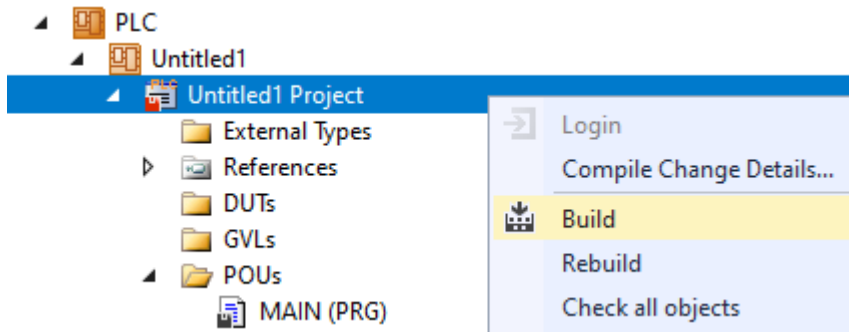
```

mover_one.Update();
mover_two.Update();
track.Update();
group.Update();
feedback.Update();

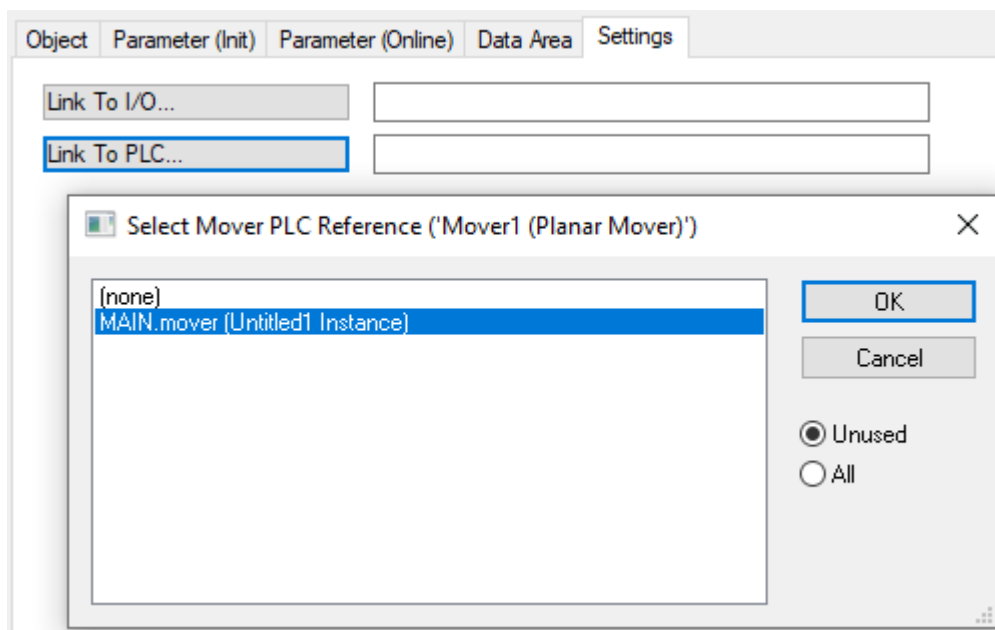
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

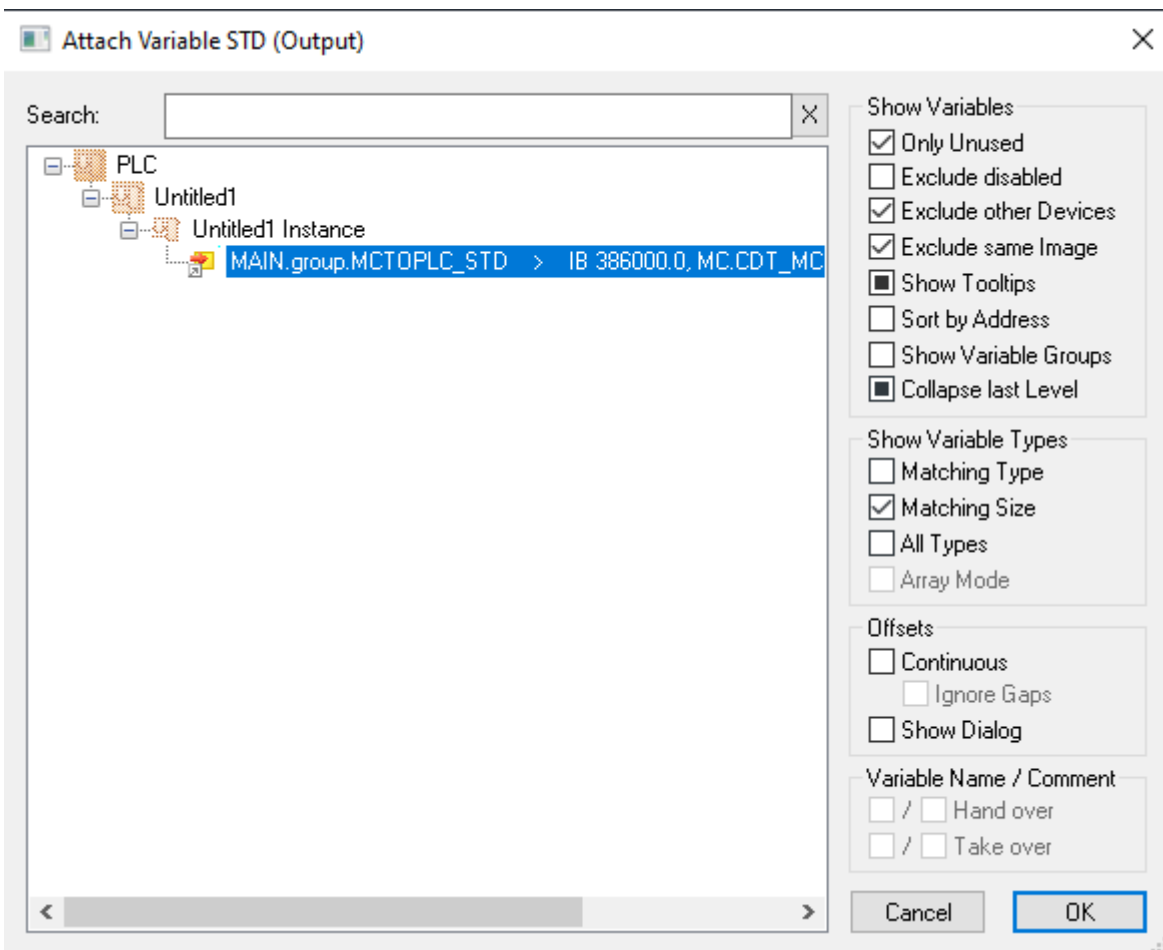
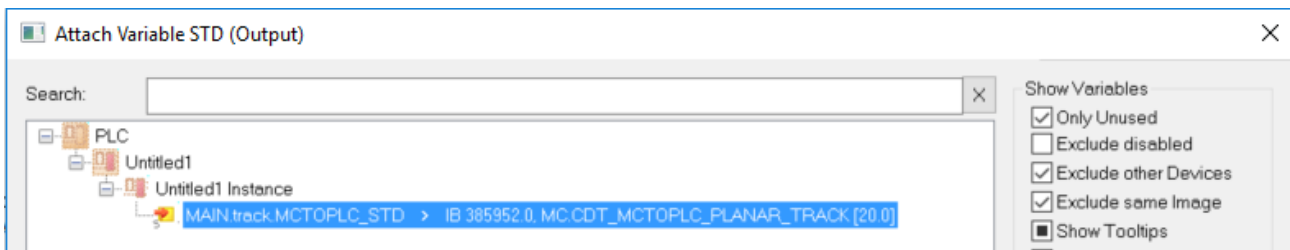
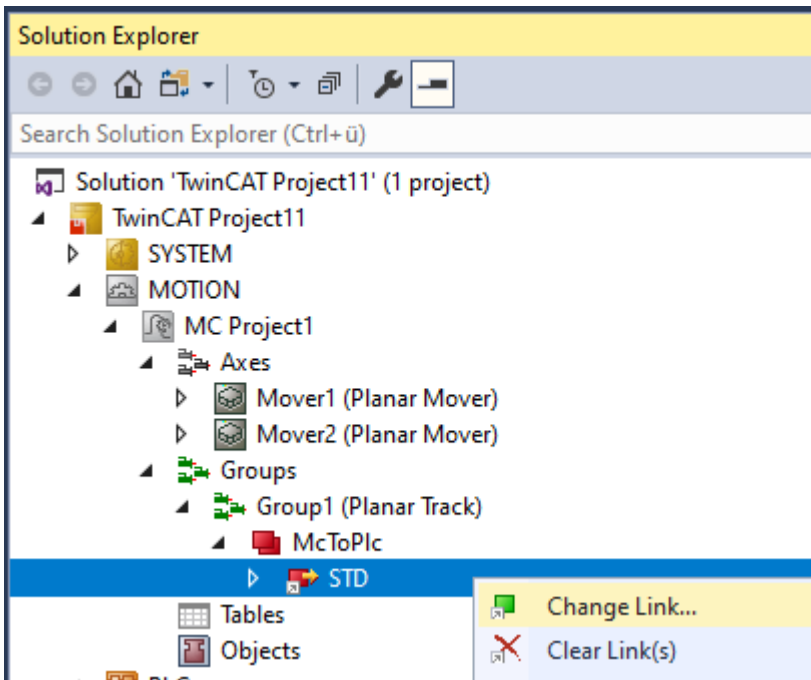
1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



⇒ 随后，可以通过 Settings 选项卡上的 Link To PLC... 按钮链接“MC Project”（双击）中的 Planar 动子。



轨道和组必须分别链接。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机结束时 (state=12)，动子处于所需的位置。

反馈显示碰撞错误。此外，如果反馈中出现碰撞错误，则会显示阻塞对象及其 OID。现在，在动子 2 移开后，便可在轨道上移动动子 1。

Expression	Type	Value	Prepared value	Address	Comment
mover_one	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLA...			%Q*	Mover data that is tra...red from the Plana...
MCTOPLC	CDT_MCTOPLC_PLA...			%I*	Mover data that is tra...red from the Plana...
STD	REFERENCE TO CDT...			%IB*	Mover standard data t...is transferred from ...
SET	REFERENCE TO CDT...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	0			X coordinate.
y	LREAL	240			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	0			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	6630674363340...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT...			%IB*	Mover busy informatio...at is transferred fro...
Error	BOOL	FALSE			Flag indicating a PlanarMover error.
ErrorId	UDINT	0			Error id indicating the PlanarMover error type.
mover_two	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLA...			%Q*	Mover data that is tra...red from the Plana...
MCTOPLC	CDT_MCTOPLC_PLA...			%I*	Mover data that is tra...red from the Plana...
STD	REFERENCE TO CDT...			%IB*	Mover standard data t...is transferred from ...
SET	REFERENCE TO CDT...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	240			X coordinate.
y	LREAL	0			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	0			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	6630674363340...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT...			%IB*	Mover busy informatio...at is transferred fro...

6.6 Planar 部件

版本 V3.2.60 及以上：可使用 Part 功能，这也是本节的主题。

MC_PlanarPart [▶_158] 是 PLC 软件对象，代表 PLC 中的部件。它可以显示部件的状态，并提供更改状态的方法。

MC PlanarPart [▶ 158] 的连接是通过 MC PlanarEnvironment [▶ 133] 间接建立的。为此，必须在启动 PLC 后，以正确的部件对象 ID 调用 MC PlanarPart [▶ 158] 的初始化方法。MC PlanarPart [▶ 158] 的状态可以通过方法调用来读取和改变。

MC PlanarPart [▶ 158] 可能使用的命令器有：ActivatePosition [▶ 159] 方法以及 AllowEnable [▶ 160]、ForceDisable [▶ 160] 和 Reset [▶ 160] 方法。ActivatePosition [▶ 159] 方法会将部件移动到其中 1 个可能的位置。AllowEnable [▶ 160]、ForceDisable [▶ 160] 和 Reset [▶ 160] 方法可更改 MC PlanarPart [▶ 158] 的 PlanarState。

版本 V3.3.19 及以上：Planar 部件的 ForceEnable 方法可在必要时取消之前的 ForceDisable 命令并激活 Planar 部件。如果超过时间，ForceEnable 和 Reset 现在都会以超时错误终止，请参见 配置 [▶ 94] 中的参数“状态机超时”。

调用 AllowEnable [▶ 160] 方法后，允许 PlanarPart 启动 PlanarState Machine（上至 CoE 状态 OperationEnabled）。只要用 Enable() 命令激活坐标系中的第一个动子，该系统中的所有部件都会被激活。激活动子定位的部件是激活动子的必要条件，也是首先要做的。激活坐标系中的其余部分是可选的，可能会失败，但不会阻止或取消激活。反之，对坐标系中最后一个激活的动子调用 Disable() 命令会关闭所有部件。关闭动子时不一定要关闭部件，这是在关闭动子后进行的。

同样，当第一个/最后一个被激活的动子进入或离开坐标系时，通过调用其部件的 ActivatePosition [▶ 159] 方法来激活或停用部件。如果最后一个激活动子出现错误，并伴有“abort”错误反应，各部件也会停用。如果动子出现“quick stop”错误反应错误，则各部件会保持激活状态。这适用于出错和随后的重置，直到部件重新激活。如果在出错期间动子发出 Disable() 命令或 Reset [▶ 160] 命令，将出错反应改为“abort”，部件就会关闭。

如果通过调用 ForceDisable [▶ 160] 方法强制某个部件进入禁用状态，该部件上的所有动子都会被设置为错误状态，如果坐标系中没有其他活动动子，坐标系中的其他部件也可能被关闭。至少在下一次调用 AllowEnable [▶ 160] 之前，部件一直处于禁用状态。

部件处于错误状态时，该部件上所有动子都处于错误状态。动子报错先于部件报错达到。坐标系中的其他动子或部件不受这些报错的影响。

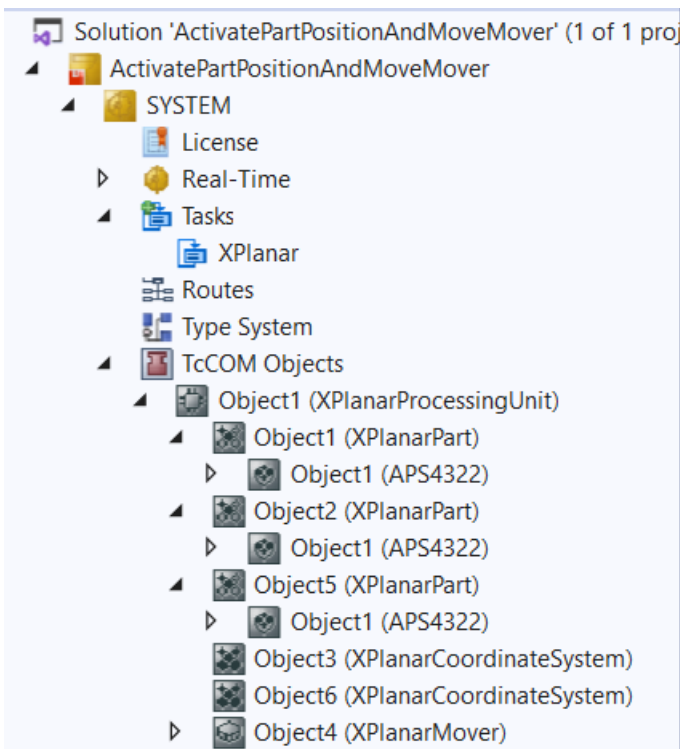
如果出现部件错误，可以触发 Reset [▶ 160] 命令来纠正错误。任何动子错误都不会受此影响。反之，动子发出的 Reset [▶ 160] 命令会触发坐标系中所有出错部件的 Reset [▶ 160] 命令。动子的 Reset [▶ 160] 命令以其自身部件无误为必要条件；因此，部件会在动子之前复位。

6.6.1 示例“激活 Planar 部件位置并移动 Planar 动子”

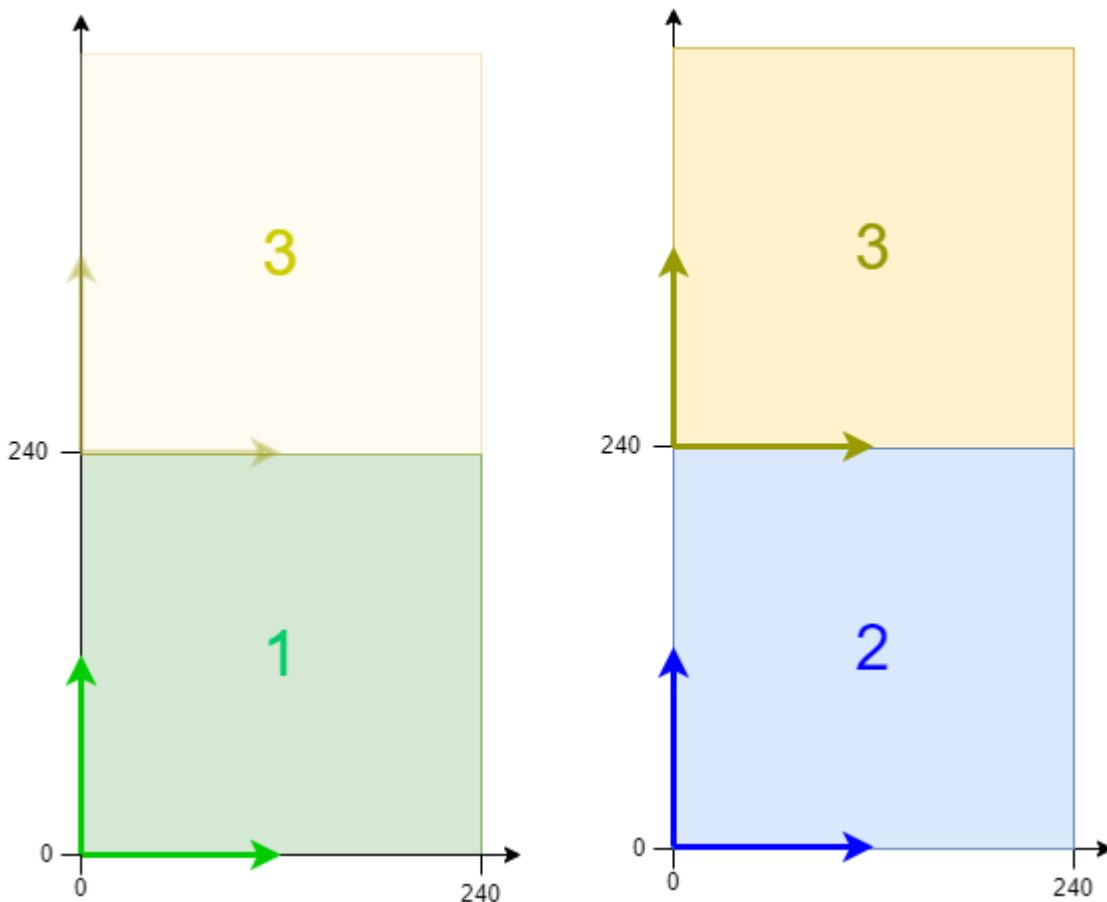
在这个例子中，1 个 Planar 动子被移动到 3 个 Planar 部件上，同时移动其中可移动的 1 个部件。

起点

您可以从包含 1 个完全配置好的 XPlanar Processing Unit 的解决方案开始。在 XPlanar Processing Unit 下创建了 3 个部件、2 个坐标系和 1 个动子。3 个部件下各创建 1 个 Planar 模块。



几何情况设置如下：前 2 个部分固定在原点的 2 个坐标系中，第 3 个部分可以在 2 个坐标系之间改变位置。例如，可以设想 2 个坐标系在 2 个 Planar 上一上一下排列，而第 3 部分是 2 个坐标系之间的电梯。动子从坐标系 1 的第 1 部分中间开始，而第 3 部分则从坐标系 2 开始。



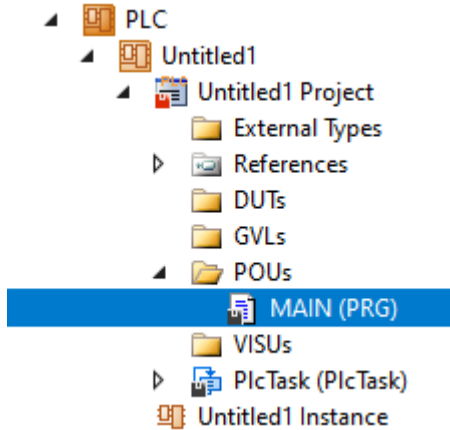
创建 Planar 动子和 Planar 环境

1. 为本示例创建 Planar 动子，请参阅配置 [▶_16]。

2. 创建 Planar 环境，请参阅 配置 [▸_94]。
3. 将初始参数 XPlanar Processing Unit OID 设置为 XPlanar Processing Unit 的对象 ID。这将激活所有 **MC Configuration** 对象（尤其是已创建的 Planar 动子）Part 的功能。

创建 PLC

- ✓ 见初步步骤 创建 PLC [▸_19]。
1. 使用 **MAIN** 创建动子（“MC PlanarMover [▸_142]”）如下。



⇒ 这/这些代表 MC Configuration 中的动子。

2. 如下图所示，创建 1 个 Planar 动子、1 个 Planar 环境、1 个 Planar 部件、1 个状态机的任意状态变量和 1 个目标位置，用于动子的移动命令。

```
PROGRAM MAIN
VAR
    mover : MC_PlanarMover;
    environment : MC_PlanarEnvironment;
    part_three : MC_PlanarPart;
    state : UDINT;
    target_position : PositionXYC;
END_VAR
```

3. 然后在 MAIN 中编程 1 个序列。

⇒ 该程序代码初始化第 3 部分，激活动子，将第 3 部分移至坐标系 1，将动子移至第 3 部分，将第 3 部分移回坐标系 2，最后将动子移至坐标系 2 中的第 2 部分。

```
CASE state OF
0:
    part_three.Initialize(0, 16#01010080, environment);
    state := 1;
1:
    IF part_three.IsInitialized THEN
        state := 2;
    END_IF
2:
    mover.Enable(0);
    state := 3;
3:
    IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := 4;
    END_IF
4:
    part_three.ActivatePosition(0, 1);
    state := 5;
5:
    IF part_three.PositionIndex = 1 THEN
        state := 6;
    END_IF
6:
    target_position.SetValuesXYCReferenceId(120, 120, 0, part_three.PartOID);
    mover.MoveToPosition(0, target_position, 0, 0);
    state := 7;
7:
    IF mover.MCTOPLC.SET.SetPos.y > 300 AND NOT mover.MCTOPLC.STD.Busy.busyXYC THEN
        state := 8;
    END_IF
8:
    part_three.ActivatePosition(0, 2);
```

```

state := 9;
9:
IF part_three.PositionIndex = 2 THEN
state := 10;
END_IF
10:
target_position.SetValuesXYCReferenceId(120, 120, 0, 16#01010030); // Position on part two
mover.MoveToPosition(0, target_position, 0, 0);
state := 11;
END_CASE

```

发送命令

- 要发送运动命令，必须在 END_CASE 后循环调用动子的更新方法；要发送 Planar 部件的命令，必须循环调用环境的更新方法：

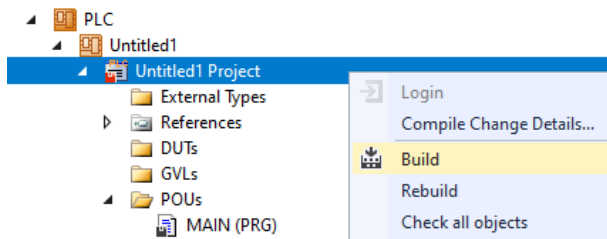
```

mover.Update();
environment.Update();

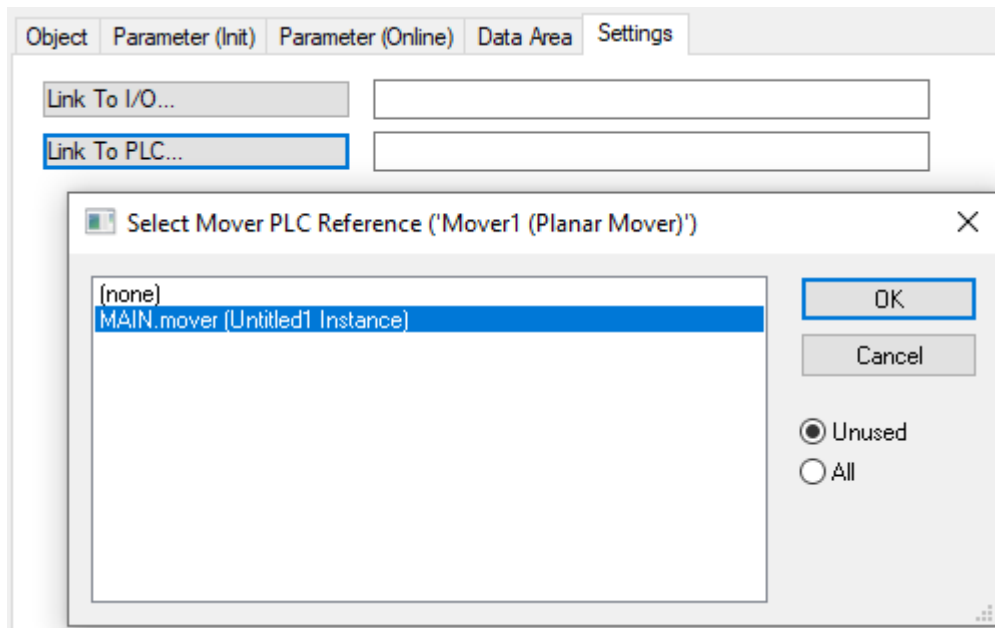
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

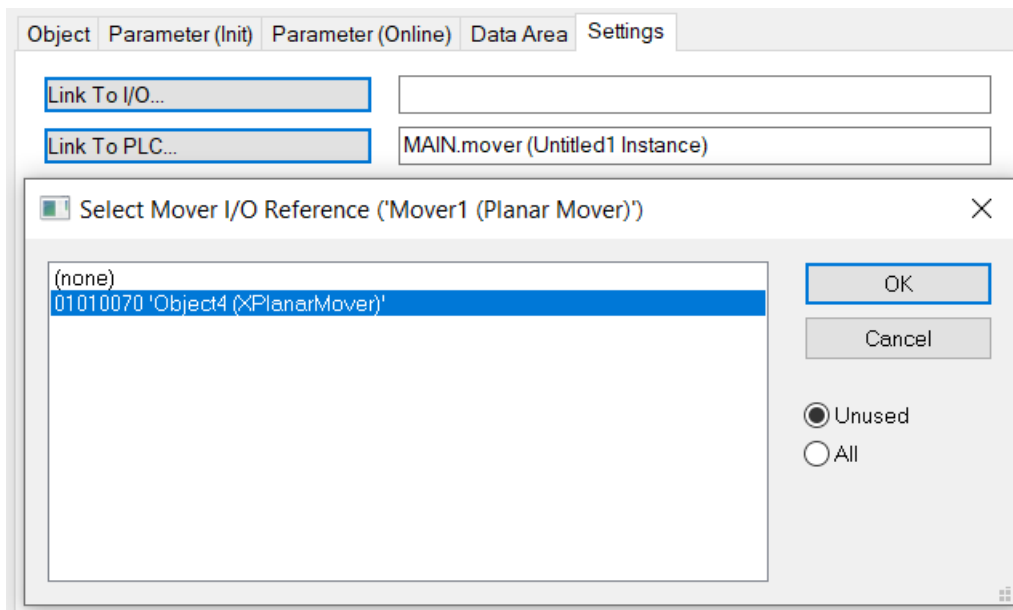
- 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



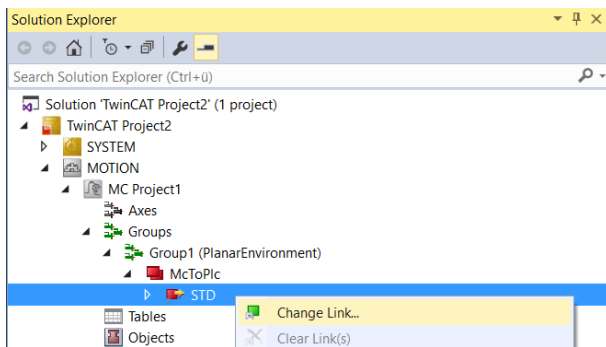
⇒ 随后，可以通过 Settings 选项卡上的 Link To PLC... 按钮链接“MC Project”（双击）中的 Planar 动子。

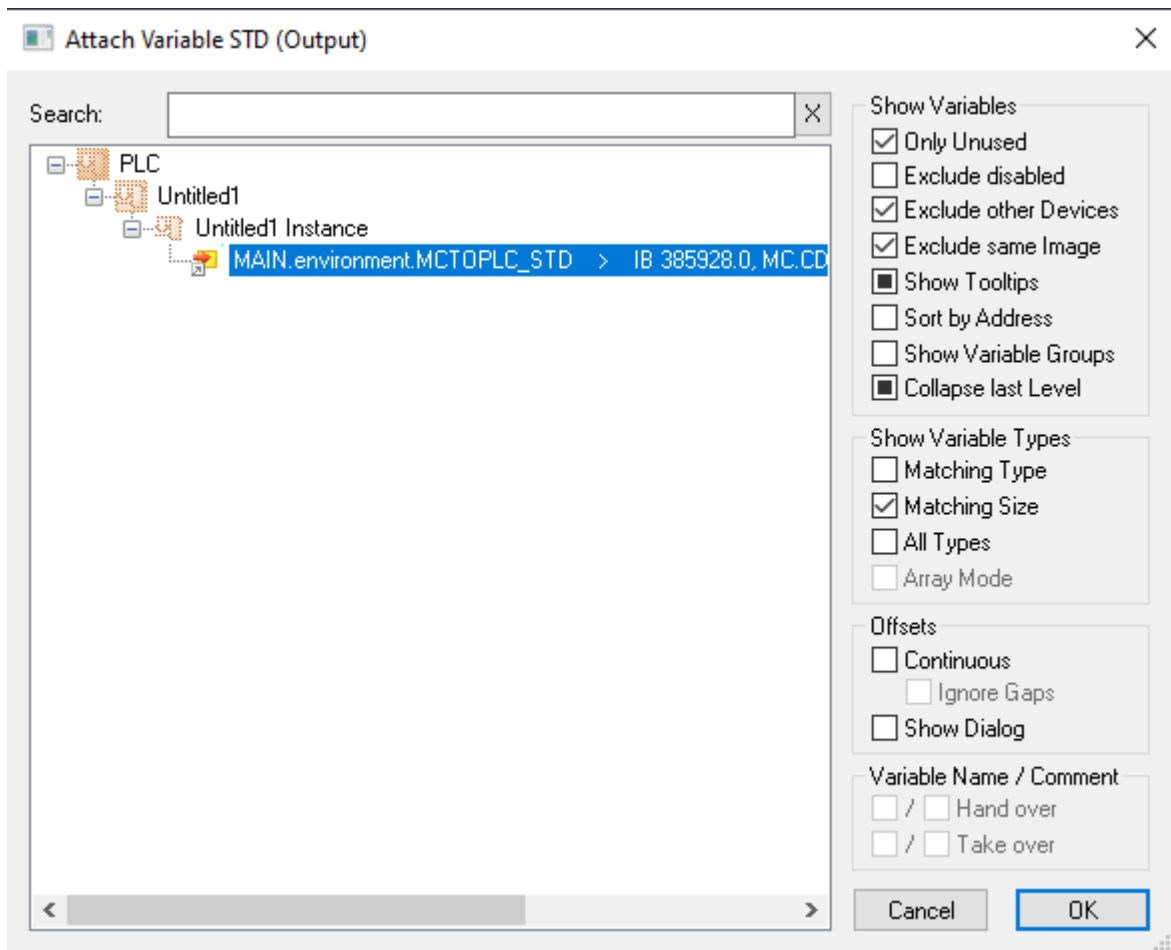


⇒ 此外，“MC Project”（双击）中的 Planar 动子可以通过 Settings 选项卡上的 Link To I/O... 按钮进行链接。



⇒ 然后就可以在“MC Project”中链接 Planar 环境。





激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机结束时（状态 = 11，下图中十六进制表示为 B），动子处于所需的位置。位置在坐标系二中指定（对象标识 16#010100A0）。动子与第 3 部分一起改变了坐标系，或者与电梯一起移动到了不同的楼层。总之，这个例子清楚地表明，Planar-Part PLC 对象只是轻量级的 Planar Environment，必须用 Planar Environment 进行初始化，并通过 Planar Environment 循环发送命令。

Expression	Type	Value
[-] mover	MC_PlanarMover	
[-] PLCTOMC	CDT_PLCTOMC_PLANAR_MOVER	
[-] MCTOPLC	CDT_MCTOPLC_PLANAR_MOVER	
[-] STD	REFERENCE TO CDT_MCTOPLC_PLANAR_MOVER_STD	
[-] SET	REFERENCE TO CDT_MCTOPLC_PLANAR_MOVER_SET	
[-] SetPos	MoverVector	
x	LREAL	120
y	LREAL	120
z	LREAL	1.9944148439562
a	LREAL	-0.00494750319896669...
b	LREAL	-0.00251608611460904...
c	LREAL	0
[-] SetVelo	MoverVector	
[-] SetAcc	MoverVector	
DcTimeStamp	ULINT	16#0A7EA6CA5EFC0000
PhysicalAreaID	UDINT	16#010100A0
[-] ACT	REFERENCE TO CDT_MCTOPLC_PLANAR_MOVER_ACT	
[-] COORDMODE	REFERENCE TO CDT_MCTOPLC_PLANAR_MOVER_COORD...	
[-] SETONTRACK	REFERENCE TO CDT_MCTOPLC_PLANAR_MOVER_TRACK	
[-] Error	BOOL	FALSE
[-] ErrorId	UDINT	16#00000000
[-] environment	MC_PlanarEnvironment	
[-] part_three	MC_PlanarPart	
state	UDINT	16#0000000B
[-] target_position	PositionXYC	

6.7 Planar Feedback

MC Planar Feedback [►_136] 是 1 个 PLC 对象，它捆绑了用户向动子、轨道、编组或其他 Planar 组件发出的命令的所有反馈状态信息。

这包括从用户发送命令到各组件处理命令，以及从随后接受（或可能拒绝）命令到执行和终止命令。如果用户愿意，可以使用反馈对象来跟踪所有这一切。

为此，必须在调用不同命令时，在 PLC 中传输 1 个反馈对象作为第 1 个参数。随后，每当用户触发反馈对象（或调用其更新方法）时，就能检索到当前的命令状态。

要使用 Planar Feedback，必须在 PLC 中声明。Planar Feedback 在运动控制侧的 TCOM 对象中没有相应的对象。它直接从在 PLC 中调用它的相应 TCOM 对象（如 Planar 动子）那里接收信息，并执行相应的命令。因此，无需在 MC 项目中单独创建反馈、设置参数或进行链接。

6.7.1 示例“创建 Planar 动子和 Planar Feedback”

通过本简短指南，您将创建 1 个包含 Planar 动子和 Planar Feedback 的 TwinCAT 项目。

创建 Planar 动子

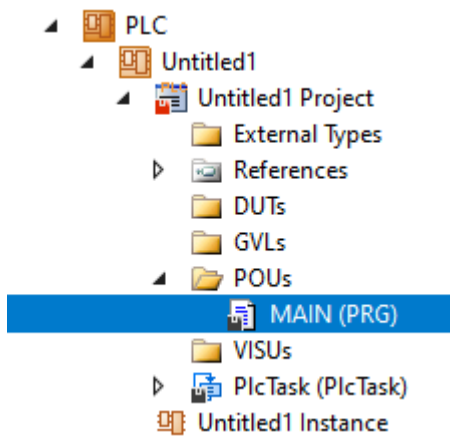
✓ 参阅 配置 [►_16]。

1. 创建 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

创建 PLC

✓ 请参阅 创建 PLC [►_19] 下的初步步骤。

1. 通过 MAIN 创建 1 个动子 (“MC_PlanarMover”) 和 1 个 Planar Feedback (“MC_PlanarFeedback”)，如下所示。



⇒ 它们代表了 MC Configuration 中的动子和 Planar Feedback。

```
PROGRAM MAIN
VAR
  mover : MC_PlanarMover;
  feedback : MC_PlanarFeedback;
  state : UDINT;
  target_position : PositionXYC;
END_VAR
```

在这个简单的示例中，您为状态机创建了 1 个状态变量，并为 Mover 的行走命令创建了 1 个目标位置。为了监控命令的运行过程，还声明了 1 个反馈信息，以便随后在 MAIN 中对序列进行编程：

```
CASE state OF
  0:
    mover.Enable(feedback);
    state := 1;
  1:
    IF feedback.Done THEN
      state := 2;
    END_IF
  2:
    target_position.SetValuesXY(1000, 1000);
    mover.MoveToPosition(feedback, target_position, 0, 0);
    state := 3;
END_CASE
```

这段程序代码激活了动子，并将其移动到 $x = 1000$ 和 $y = 1000$ 的位置。

请注意，只有当反馈通过其“Done”标志发出成功终止命令的信号时，状态机才会向前推进。

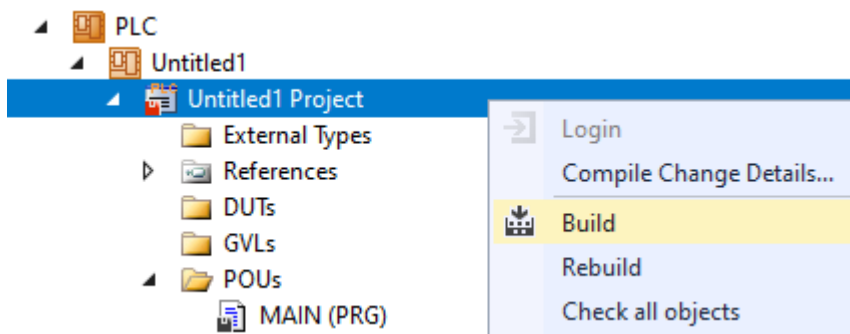
发送命令

2. 要发出命令并监控反馈，必须在 END_CASE 之后循环调用动子和反馈的更新方法：

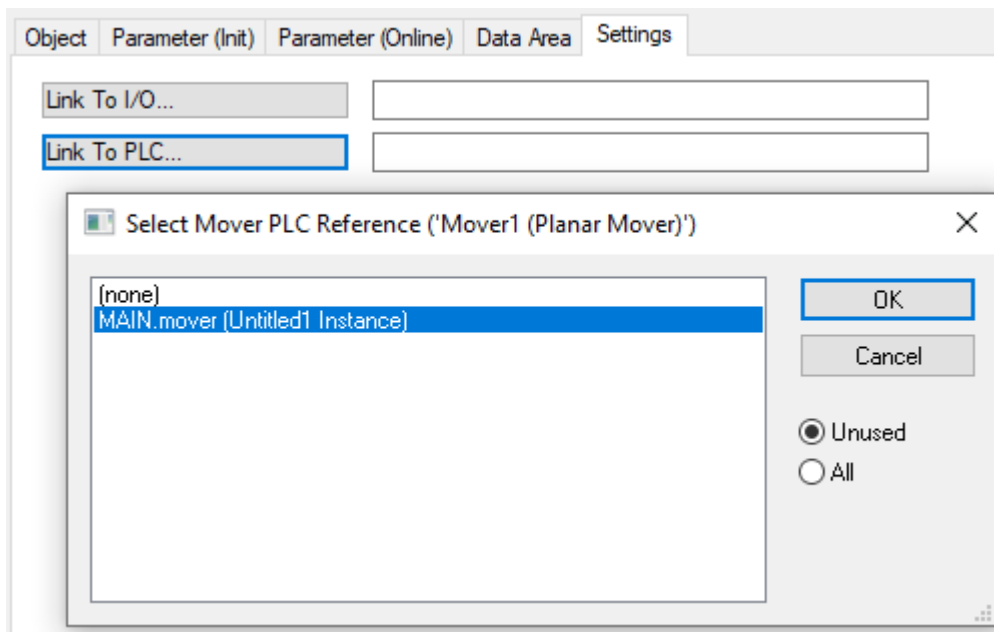
```
mover.Update();
feedback.Update();
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



⇒ 随后，可以通过 Settings 选项卡上的 Link To PLC... 按钮链接“MC Project”（双击）中的 Planar 动子。



激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机末端端子模块（状态 = 3），动子处于所需的位置，反馈以“Done”标志发出命令终止信号。

Expression	Type	Value	Prepared value	Address	Comment
mover	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLA...			%Q*	Mover data that is tra...red from the Plana...
MCTOPLC	CDT_MCTOPLC_PLA...			%I*	Mover data that is tra...red from the Plana...
STD	REFERENCE TO CDT...			%IB*	Mover standard data...is transferred from ...
SET	REFERENCE TO CDT...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	1000			X coordinate.
y	LREAL	1000			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	0			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	6630698779040 ...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT...			%IB*	Mover busy informati...at is transferred fro...
Error	BOOL	FALSE			Flag indicating a PlanarMover error.
ErrorId	UDINT	0			Error id indicating the PlanarMover error type.
feedback	MC_PlanarFeedback				
objectInfo	PlanarObjectInfo				Indicates which object one would collide with.
Active	BOOL	FALSE			Indicates an active co...nd, i.e. command w...
Busy	BOOL	FALSE			Indicates a busy com..., i.e. command is b...
Done	BOOL	TRUE			Indicates the comman...done, i.e. executio...
Aborted	BOOL	FALSE			Indicates the comman...aborted, i.e. execut...
Error	BOOL	FALSE			Indicates the command has an error.
ErrorId	UDINT	0			Indicates the error id of the command error.
state	UDINT	3			
target_position	PositionXYC				

6.7.2 示例 “Planar 运动组件：避免碰撞”

通过本简要指南，您将创建 1 个 TwinCAT 项目，该项目包含 1 个 Planar 动子，由于与 Planar 环境发生碰撞，该动子的移动命令被拒绝。

创建 Planar 动子

✓ 请参见 配置 [▶ 16]。

1. 创建 Planar 动子。
2. 将“Parameter (Init)”设置为仿真模式 (TRUE)。参数是隐藏的，只有激活“Show Hidden Parameters”复选框后才会显示。

创建 Planar 环境

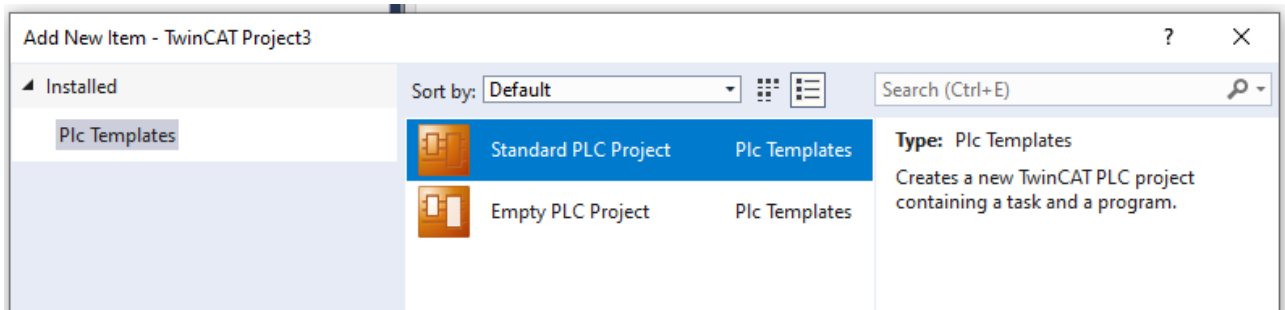
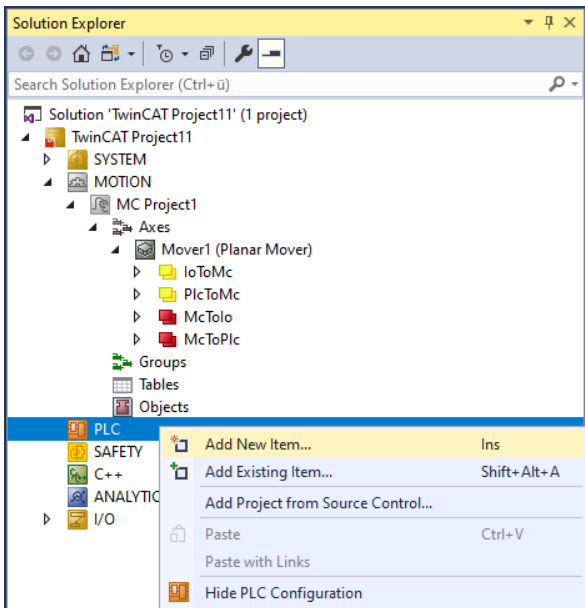
3. 要创建 Planar 环境，请参见 配置 [▶ 94]。

创建 1 个 Planar 组

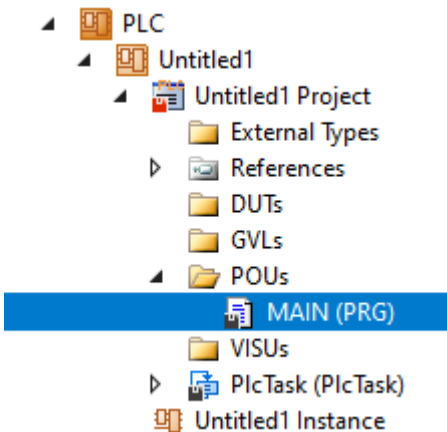
4. 创建 Planar 组，请参阅 配置 [▶ 87]。

创建 PLC

✓ 为了创建环境的几何形状并控制动子，必须创建 1 个 PLC，用户可以从 PLC 向两者发送命令。



5. 将 `Tc3_Physics` 和 `Tc3_Mc3PlanarMotion` 库添加到 PLC 项目中，请参见插入功能库 [▶_123]。
6. 通过 `MAIN` 创建 `MC PlanarMover` [▶_142] 和 `MC PlanarEnvironment` [▶_133]。



⇒ 它们代表了 MC Configuration 中的动子和环境。

```
PROGRAM MAIN
VAR
  mover           : MC_PlanarMover;
  environment     : MC_PlanarEnvironment;
  group          : MC_PlanarGroup;
  feedback       : MC_PlanarFeedback;
  state          : UDINT;
  target_position : PositionXYC;
END_VAR
```

在本示例中，您为简单状态机创建了 1 个任意状态变量，并为动子的移动命令创建了 1 个目标位置，随后就可以在 `MAIN` 中对该位置序列进行编程：

```
CASE state OF
  0:
    environment.AddStator(0, -120.0, -120.0);
    environment.CreateBoundary(0);
```

```

state := 1;
1:
mover.Enable(0);
group.Enable(0);
state := 2;
2:
IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled AND
group.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
state := 3;
END_IF
3:
mover.AddToGroup(0,group);
environment.AddToGroup(0,group);
state := 4;
4:
IF mover.MCTOPLC.STD.GroupOID > 0 AND
environment.MCTOPLC_STD.GroupOID > 0 THEN
state := 5;
END_IF
5:
target_position.SetValuesXY(100, 100);
mover.MoveToPosition(feedback, target_position, 0, 0);
state := 6;
END_CASE

```

该程序代码可激活动子，并从动子所定位的 Planar 模块中创建 1 个环境。然后尝试将动子移动到 x=100 和 y=100 的位置。

发送命令

7. 为了发出命令并监控反馈，必须在 END_CASE 之后循环调用对象的更新方法：

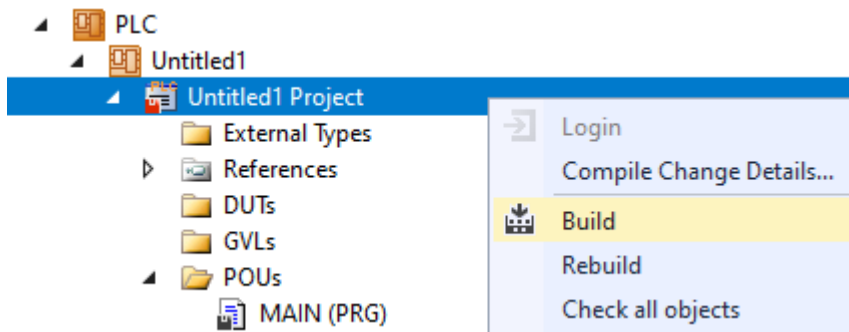
```

mover.Update();
environment.Update();
group.Update();
feedback.Update();

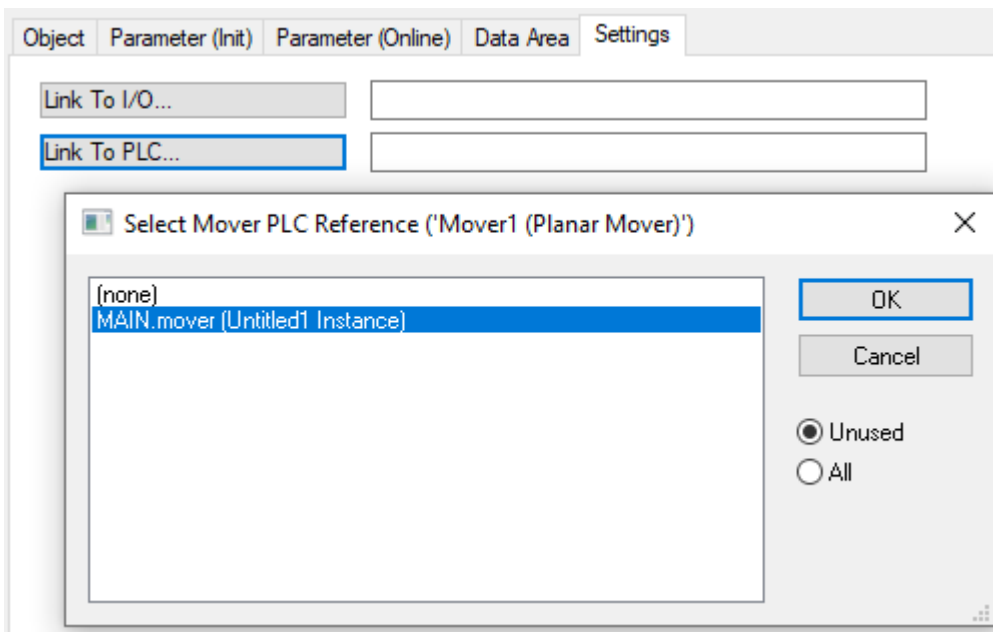
```

创建 PLC 时，会创建 1 个“PLC Mover”符号，然后将其链接到 MC 项目中的动子实例。

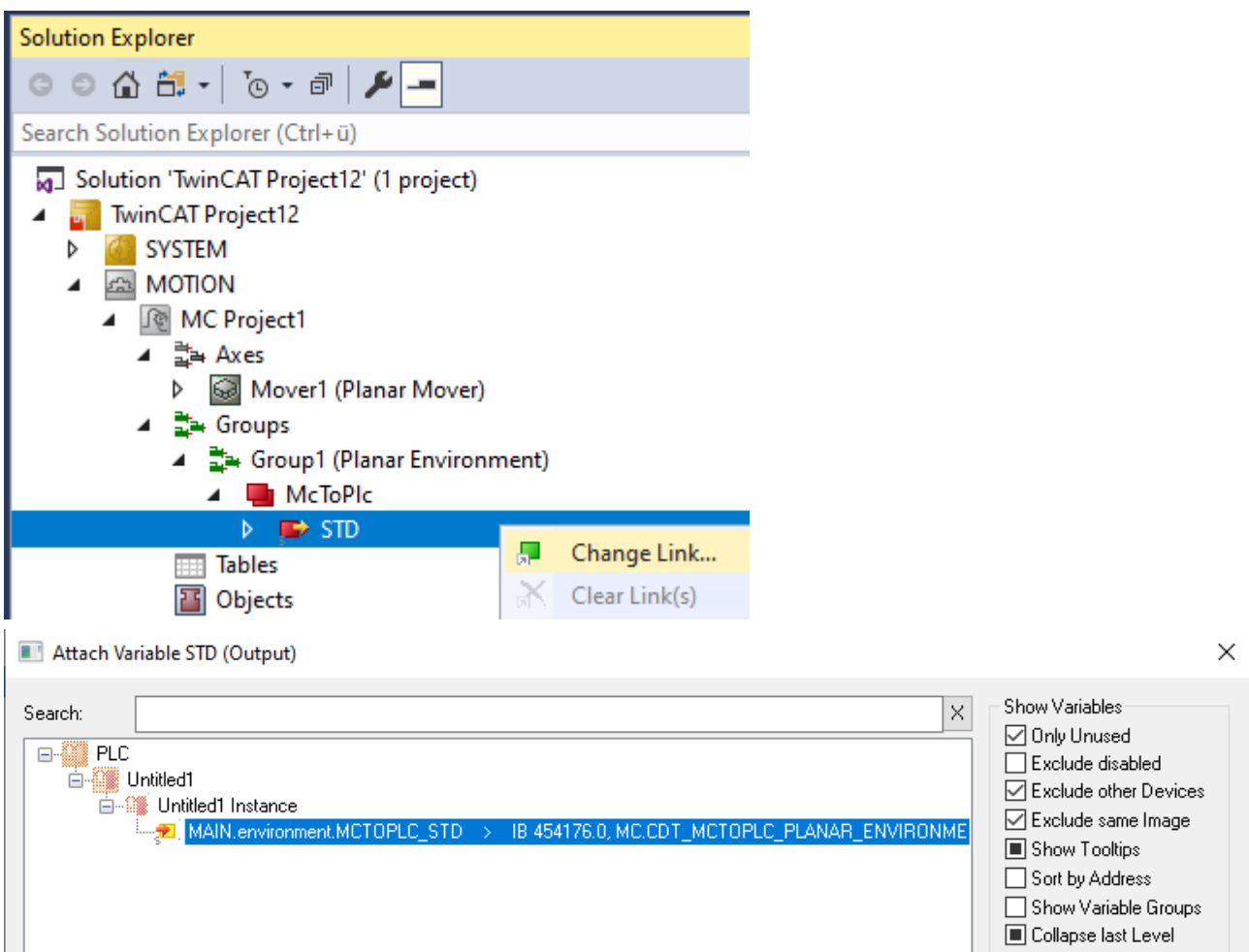
1. 如要创建，请使用 PLC > Untitled1 > Untitled1 Project > Build 路径。



- ⇒ 随后，可以通过 Settings 选项卡上的 Link To PLC... 按钮链接“MC Project”（双击）中的 Planar 动子。





⇒ 随后，可通过“MC Project”中的以下对话框链接 Planar 环境。



⇒ 组与组之间的联系也是如此。

激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。

3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

在状态机结束时 (state=6)，动子处于理想位置。动子没有动，因为命令被拒绝了。反馈会显示碰撞错误，而环境在 ObjectInfo 中被指定为碰撞对象。

Expression	Type	Value	Prepared value	Address	Comment
mover	MC_PlanarMover				
PLCTOMC	CDT_PLCTOMC_PLA...			%Q*	Mover data that is tra...red from the Plana...
MCTOPLC	CDT_MCTOPLC_PLA...			%I*	Mover data that is tra...red from the Plana...
STD	REFERENCE TO CDT...			%IB*	Mover standard data t...is transferred from ...
SET	REFERENCE TO CDT...			%IB*	Mover setpoint data th...is transferred from t...
SetPos	MoverVector				Current position.
x	LREAL	0			X coordinate.
y	LREAL	0			Y coordinate.
z	LREAL	0			Z coordinate.
a	LREAL	0			A coordinate.
b	LREAL	0			B coordinate.
c	LREAL	0			C coordinate.
SetVelo	MoverVector				Current velocity.
SetAcc	MoverVector				Current acceleration.
DcTimeStamp	ULINT	6631508341750...			Current time stamp.
PhysicalAreaID	UDINT	0			Current physical area id.
ACT	REFERENCE TO CDT...			%IB*	Mover actpoint data th...is transferred from t...
COORDMODE	REFERENCE TO CDT...			%IB*	Mover coordinate mod...ormation that is tra...
SETONTRACK	REFERENCE TO CDT...			%IB*	Mover busy informatio...at is transferred fro...
Error	BOOL	FALSE			Flag indicating a PlanarMover error.
ErrorId	UDINT	0			Error id indicating the PlanarMover error type.
environment	MC_PlanarEnvironment				
group	MC_PlanarGroup				
feedback	MC_PlanarFeedback				
objectInfo	PlanarObjectInfo				Indicates which object one would collide with.
ObjectType	EPLANAROBJECTTYPE	Environment			Object type.
Id	UDINT	85065744			Object id.
Active	BOOL	FALSE			Indicates an active co...nd, i.e. command w...
Busy	BOOL	FALSE			Indicates a busy com..., i.e. command is b...
Done	BOOL	FALSE			Indicates the comman...done, i.e. executio...
Aborted	BOOL	FALSE			Indicates the comman...aborted, i.e. execut...
Error	BOOL	TRUE			Indicates the command has an error.
ErrorId	UDINT	33158			Indicates the error id of the command error.
state	UDINT	6			
target_position	PositionXYC				

6.7.3 专门的反馈类型

除了大多数命令都接受的通用 MC_PlanarFeedback 类型外，某些命令可能还需要专门的反馈类型。Planar Feedback [▶ 110] 适用于一般反馈的原则也适用于这些类型。

专门的反馈可以有一般反馈输出的子集，这取决于它们的类型。其中包括命令是否处于活动状态或是否导致错误等信息。此外，专门的反馈类型还可能与其应用范围相对应的其他输出或无效功能。

MC_PlanarFeedbackGearInPosOnTrack

示例“Planar 动子在轨道上与外部轴同步” [▶ 64] 接受这种反馈类型。它还有 1 个附加输出 inSync，用于指示执行动子是否与主轴同步。

MC_PlanarFeedbackGearInPosOnTrackWithMasterMover

示例：“使轨道上的 Planar 动子与另一个 Planar 动子同步” [▶ 70] 接受这种反馈类型。它还有 1 个附加输出 inSync，用于指示执行动子是否与 Master Planar Mover 同步。

6.8 Planar TrackTrail

MC PlanarTrackTrail [▶_171] 是 1 个定义网络中连续 Planar 轨道路径的对象。与建立 Planar 轨道网络的单个 Planar 轨道不同，Planar 轨道网络在 MC 侧的 TCOM 对象中没有相应的对象，而是仅在 PLC 中声明，类似于 Planar Feedback [▶_110]。

Planar 轨道网络可用于定义 用于同步的 Planar 轨道网络，Slave Planar Mover 与 [▶_64] Master Planar Mover [▶_70] 的同步运动可通过该路径网络进行（如果该路径由 Slave Planar Mover 的多个当前 Planar 轨道组成）。

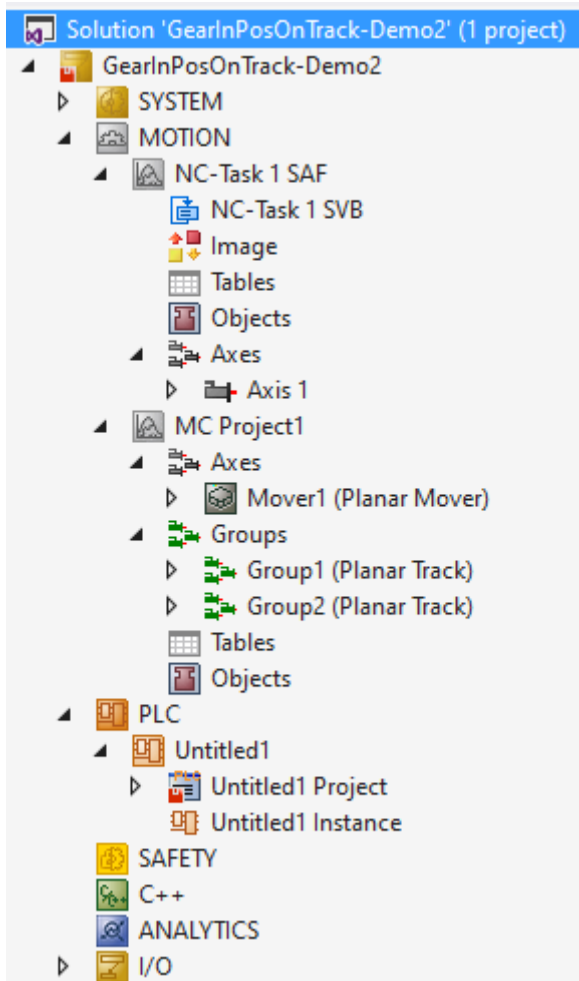
Planar-TrackTrail 提供了在其末端添加 Planar 轨道和清空其配置的方法。这些方法只修改 Planar-TrackTrail，而不触及底层的 Planar 轨道和网络。

添加 Planar 轨道时，确保该轨道与 Planar-TrackTrail 中当前最后 1 条 Planar 轨道的末端相链接。此外，也无法多次添加 Planar 轨道。

版本 V3.3.39 及以上：可通过调用 MC_PlanarMover 的“GetTrackTrailInformation [▶_157]”方法对 Planar TrackTrail 进行初始化。方法调用后，Planar TrackTrail 表达动子通过轨道网络到达目标的规划路线。例如，在轨道上静止时，轨迹只包含动子所定位的当前轨道。当动子主动移动到另一条轨道时，至少当前轨道和目标轨道都会被包含在轨迹中；作为第一条和最后一条轨道。

6.8.1 示例“2 个 Planar 轨道上的同步运动”

本示例是“在带轴轨道上同步 Planar 动子 [▶_64]”示例的延伸，其中 Planar 动子的同步运动是通过两条 Planar 轨道进行的。修改上述示例，以便在 MC Configuration 中创建 2 个 Planar 轨道，这样解决方案资源管理器中就有了以下条目：

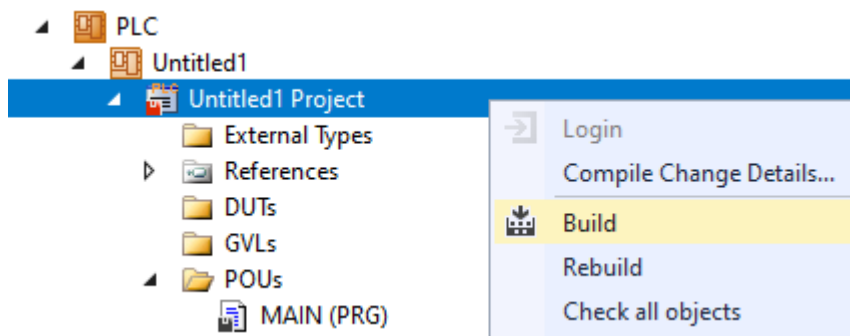


调整 PLC 项目

1. 将 Tc3_Mc3PlanarMotion、Tc3_Physics 和 Tc2_MC2 库添加到 PLC 项目中，请参见 [插入功能库 \[► 123\]](#)。
2. 在 MAIN 中声明以下任意变量：

```
PROGRAM MAIN
VAR
  mover          : MC_PlanarMover;
  track1         : MC_PlanarTrack;
  track2         : MC_PlanarTrack;
  trail          : MC_PlanarTrackTrail;
  axis           : AXIS_REF;
  power_axis    : MC_Power;
  move_axis     : MC_MoveAbsolute;
  state          : UDINT;
  pos1, pos2    : PositionXYC;
END_VAR
```

3. 构建 PLC，创建“PLC mover”“PLC track”和“PLC axis”的符号。



4. 连接 Planar 动子、Planar 轨道（见示例“[将 Planar 动子耦合到轨道并移动 \[► 19\]](#)”）和轴（见示例“[使轨道上的 Planar 动子与轴同步 \[► 64\]](#)”）。



除了将 Planar 轨道的数量增加一倍以及对代码稍作修改外，所有先前步骤都与示例“[使轨道上的 Planar 动子与轴同步 \[► 64\]](#)”中的步骤相同。

状态机编程

下一步是修改程序代码，以便将 Planar TrackTrail 传递给 GearInPosOnTrack 命令。后者预先由 2 条 Planar 轨道组成，在本示例中，这 2 条 Planar 轨道构成了一个简单的网络 [\[► 73\]](#)，该网络由一个 L 配置和一个混合件组成（请参见 [几何定义 \[► 45\]](#)）：

```
CASE state OF
0:
  pos1.SetValuesXYC(100, 100, 0);
  pos2.SetValuesXYC(400, 100, 0);
  track1.AppendLine(0, pos1, pos2);
  track1.Enable(0);
  state := state + 1;
1:
  IF track1.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
2:
  track2.StartFromTrack(0, track1);
  state := state + 1;
3:
  pos1.SetValuesXYC(500, 100, 0);
  pos2.SetValuesXYC(860, 100, 0);
  track2.AppendLine(0, pos1, pos2);
  track2.Enable(0);
  state := state + 1;
4:
  IF track2.MCTOPLC_STD.State = MC_PLANAR_STATE.Enabled THEN
    state := state + 1;
  END_IF
5:
  mover.Enable(0);
  state := state + 1;
6:
```

```

    IF mover.MCTOPLC.STD.State = MC_PLANAR_STATE.Enabled THEN
        state := state + 1;
    END_IF
7:
    mover.JoinTrack(0, track1, 0, 0);
    state := state + 1;
8:
    IF mover.MCTOPLC.STD.CommandMode =
    MC_PLANAR_MOVER_COMMAND_MODE.OnTrack THEN
        state := state + 1;
    END_IF
9:
    power_axis(Axis := axis,
    Enable := TRUE,
    Enable_Positive := TRUE);
    IF power_axis.Status THEN
        move_axis(Axis := axis, Execute := FALSE);
        state := state + 1;
    END_IF
10:
    move_axis(Axis := axis,
        Position := 700,
        Velocity := 30,
        Acceleration := 100,
        Deceleration := 100,
        Jerk := 100,
        Execute := TRUE);
    state := state + 1;
11:
    trail.AddTrack(track1);
    trail.AddTrack(track2);
    mover.GearInPosOnTrack(0, axis.DriveAddress.TcAxisObjectId, trail, 100, 100, track1, 0, 0);
    state := state + 1;
END_CASE

mover.Update();
track1.Update();
track2.Update();
power_axis(Axis := axis);
move_axis(Axis := axis);
axis.ReadStatus();

```

在状态 11 中，2 条 Planar 轨道被添加到 Planar TrackTrail 中。这里的顺序很关键，因为 track2 紧跟 track1，反之则不然。Planar TrackTrail 作为第三个参数传递到 GearInPosOnTrack 命令。

激活并启动项目

1. 通过菜单栏  中的按钮激活配置。
2. 通过  按钮将 TwinCAT 系统设置为“Run”状态。
3. 通过菜单栏  中的按钮登录 PLC。
4. 通过菜单栏中的 Play 按钮启动 PLC。

通过在线视图中观察流程

5. 请注意在线视图中的 Planar 动子最初是如何沿着第一条 Planar 轨道向终点移动的：

Expression	Type	Value
[-] mover	MC_PlanarMover	
[+] PLCTOMC	CDT_PLCTOMC_PLA...	
[-] MCTOPLC	CDT_MCTOPLC_PLA...	
[+] STD	REFERENCE TO CDT...	
[+] SET	REFERENCE TO CDT...	
[+] ACT	REFERENCE TO CDT...	
[+] COORDMODE	REFERENCE TO CDT...	
[-] SETONTRACK	REFERENCE TO CDT...	
SetPos	LREAL	190.8083232748 ...
SetVelo	LREAL	29.999999999999...
SetAcc	LREAL	0
SetJerk	LREAL	0
TrackOID	OTCID	16#05120010

6. 然后您会看到它切换到随后的 Planar 轨道（请参见 TrackOID）：

Expression	Type	Value
[-] mover	MC_PlanarMover	
[+] PLCTOMC	CDT_PLCTOMC_PLA...	
[-] MCTOPLC	CDT_MCTOPLC_PLA...	
[+] STD	REFERENCE TO CDT...	
[+] SET	REFERENCE TO CDT...	
[+] ACT	REFERENCE TO CDT...	
[+] COORDMODE	REFERENCE TO CDT...	
[-] SETONTRACK	REFERENCE TO CDT...	
SetPos	LREAL	33.90832327486...
SetVelo	LREAL	29.999999999999...
SetAcc	LREAL	0
SetJerk	LREAL	0
TrackOID	OTCID	16#05120020

7. 最后，您可以看到它是如何在第二个 Planar 轨道上停滞的：

Expression	Type	Value
[-] mover	MC_PlanarMover	
[+] PLCTOMC	CDT_PLCTOMC_PLA...	
[-] MCTOPLC	CDT_MCTOPLC_PLA...	
[+] STD	REFERENCE TO CDT...	
[+] SET	REFERENCE TO CDT...	
[+] ACT	REFERENCE TO CDT...	
[+] COORDMODE	REFERENCE TO CDT...	
[-] SETONTRACK	REFERENCE TO CDT...	
SetPos	LREAL	400.0000000000...
SetVelo	LREAL	0
SetAcc	LREAL	0
SetJerk	LREAL	0
TrackOID	OTCID	16#05120020

⇒ 在本例中，如果主轴的行为要求它越过第二个 Planar 轨道的末端（例如，选择使主轴的目标位置大于 2 个 Planar 轨道的长度之和），则 Planar 动子将放弃同步运动。在这种情况下，Planar 动子在第二条轨道的末端停止，失去潜在的同步状态并报错。

如果在第一条轨道的末端添加另外一条 Planar 轨道，从而在其末端创建 1 个岔道，则 Planar 动子通过 Planar TrackTrail 明确“知道”它应该转向 2 条 Planar 轨道中的哪一条，从而继续其同步运动

（毕竟，主轴产生的设定点与 Planar 轨道无关）。这样，就可以使用 Planar TrackTrail 在任意长度的唯一路径上通过任意复杂程度的轨道网络执行同步运动。

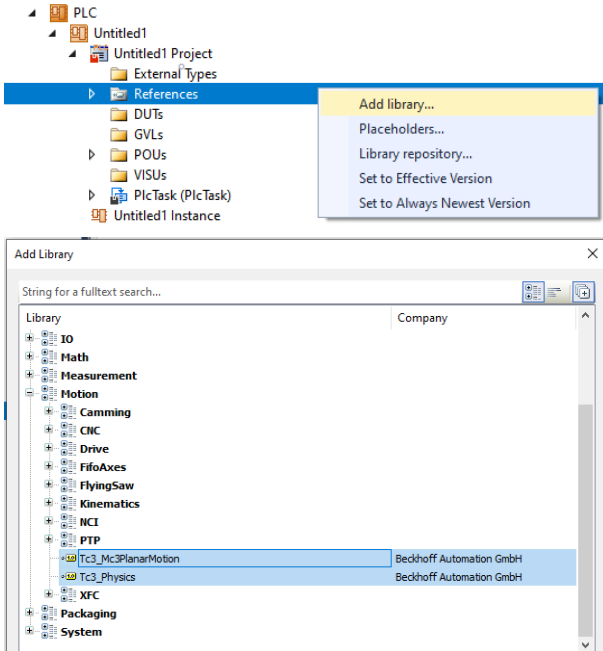
由于 Planar TrackTrail 是一个纯 PLC 对象，不通过 TCOM 通信，只充当容器，因此不需要循环更新，例如对于 Planar 动子、Planar 轨道，甚至是 Planar 回流 [▶ 110]（本示例中未出现），也没有相应的循环调用方法。

7 PLC 功能库

7.1 插入功能库

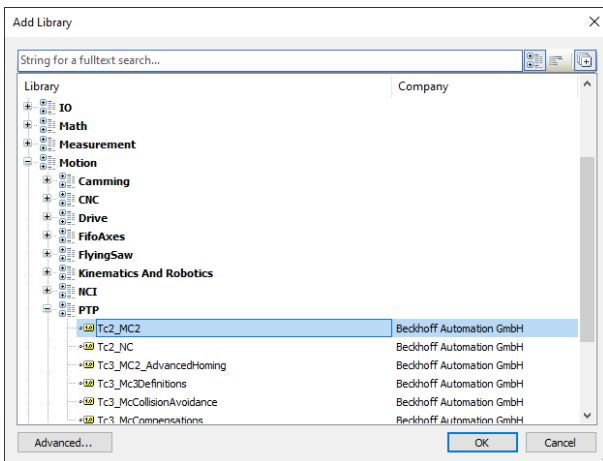
✓ 要控制 XPlanar 组件，必须集成“Tc3_Physics”和“Tc3_Mc3PlanarMotion”库。

1. 通过References > Add library...将所需功能库逐个添加到项目中



⇒ 一旦集成了程序库，就可以在 PLC 中对其余流程进行编程。

要控制主轴，还必须包含“Tc2_MC2”库。



7.2 Tc3_Mc3PlanarMotion API

7.2.1 Data Types

7.2.1.1 Enums

7.2.1.1.1 EPlanarObjectType

Identifies a planar object type.

Syntax

Definition:

```

TYPE EPlanarObjectType :
(
  Invalid      := 0,
  None        := 301,
  Mover       := 302,
  Track       := 303,
  Environment := 304
) UINT;
END_TYPE

```

Values

Name	Description
Invalid	Indicates invalid information, e.g. no connection or component not yet ready.
None	No planar object.
Mover	Planar Mover.
Track	Planar Track.
Environment	Planar Environment.

7.2.1.1.2 MC_DIRECTION

Indicates the movement direction of the Planar Mover on a Planar Track.

Syntax

Definition:

```

TYPE MC_DIRECTION :
(
  mcDirectionNonModulo := 0,
  mcDirectionPositive  := 1,
  mcDirectionShortestWay := 2,
  mcDirectionNegative  := 3
) UINT;
END_TYPE

```

Values

Name	Description
mcDirectionNonModulo	The Planar Mover moves to the absolute value of the target position. Depending on the current position, this may induce forward or backward movement. On looped tracks, multiple passes are possible.
mcDirectionPositive	The Planar Mover moves to the target position in a forward direction. No backward movement is allowed.
mcDirectionShortestWay	The Planar Mover takes the shortest way to the target position. May induce forward or backward movement.
mcDirectionNegative	The Planar Mover moves to the target position in a backward direction. No forward movement is allowed.



结合 Tc2_MC2 库，数据类型可能无法做唯一解析（“MC_Direction”名称使用不明确）。在这种情况下，使用数据类型时必须指定命名空间（Tc3_Mc3PlanarMotion.MC_DIRECTION, Tc3_Mc3Definitions.MC_DIRECTION 或 Tc2_MC2.MC_DIRECTION）。

7.2.1.1.3 MC_GAP_MODE_ON_TRACK

Indicates the way in which gap control is done on track.

Syntax

Definition:

```
TYPE MC_GAP_MODE_ON_TRACK :
(
  Standard1D := 1,
  Fast1D     := 2
)UINT;
END_TYPE
```

Values

Name	Description
Standard1D	The Planar Mover keeps a distance to its neighbours on track that is at least the sum of gap and stop distance.
Fast1D	The Planar Mover keeps a distance to its neighbours on track that is at least the gap.

7.2.1.1.4 MC_SYNC_DIRECTIONS

Directions in which a slave is allowed to move during synchronizing phase.

Syntax

Definition:

```
TYPE MC_SYNC_DIRECTIONS :
(
  Positive := 1,
  Negative := 2,
  Both     := 3
)UINT;
END_TYPE
```

Values

Name	Description
Positive	Movement is allowed only in positive direction while synchronizing.
Negative	Movement is allowed only in negative direction while synchronizing.
Both	Movement is allowed in any direction while synchronizing.

7.2.1.2 Structs

7.2.1.2.1 CDT_MCTOPLC_PLANAR_MOVER

Contains the information of the Planar Mover passed from MC to PLC.

Syntax

Definition:

```
TYPE CDT_MCTOPLC_PLANAR_MOVER :
STRUCT
  STD      : Reference To CDT_MCTOPLC_PLANAR_MOVER_STD;
  SET      : Reference To CDT_MCTOPLC_PLANAR_MOVER_SET;
  ACT      : Reference To CDT_MCTOPLC_PLANAR_MOVER_ACT;
```

```

COORDMODE : Reference To CDT_MCTOPLC_PLANAR_MOVER_COORDMODE;
SETONTRACK : Reference To CDT_MCTOPLC_PLANAR_MOVER_TRACK;
END_STRUCT
END_TYPE
    
```

Parameters

Name	Type	Description
STD	Reference To CDT_MCTOPLC_PLANAR_MOVER_STD	Mover standard data that is transferred from the Planar Mover to this function block.
SET	Reference To CDT_MCTOPLC_PLANAR_MOVER_SET	Mover setpoint data that is transferred from the Planar Mover to this function block.
ACT	Reference To CDT_MCTOPLC_PLANAR_MOVER_ACT	Mover actpoint data that is transferred from the Planar Mover to this function block.
COORDMODE	Reference To CDT_MCTOPLC_PLANAR_MOVER_COORDMODE	Mover coordinate mode information that is transferred from the Planar Mover to this function block.
SETONTRACK	Reference To CDT_MCTOPLC_PLANAR_MOVER_TRACK	Mover busy information that is transferred from the Planar Mover to this function block.

7.2.1.2.2 CDT_PLCTOMC_PLANAR_MOVER

Contains the information of the Planar Mover passed from PLC to MC.

Syntax

Definition:

```

TYPE CDT_PLCTOMC_PLANAR_MOVER :
STRUCT
    STD : Reference To CDT_PLCTOMC_PLANAR_MOVER_STD;
END_STRUCT
END_TYPE
    
```

Parameters

Name	Type	Description
STD	Reference To CDT_PLCTOMC_PLANAR_MOVER_STD	Mover standard data that is transferred from this function block to the Planar Mover.

7.2.1.2.3 PlanarObjectInfo

Identifies a planar object uniquely by object id and type.

Syntax

Definition:

```

TYPE PlanarObjectInfo :
STRUCT
    ObjectType : EPlanarObjectType;
    Id : UDINT;
END_STRUCT
END_TYPE
    
```

Parameters

Name	Type	Description
ObjectType	EPlanarObjectType [▶ 123]	Object type.
Id	UDINT	Object id.

7.2.1.2.4 ST_AdoptTrackOrientationOptions

Options for the "AdoptTrackOrientation" command of the Planar Mover.

Syntax

Definition:

```

TYPE ST_AdoptTrackOrientationOptions :
STRUCT
    additionalTurns : UDINT;
    direction       : MC_DIRECTION;
END_STRUCT
END_TYPE
    
```

Parameters

Name	Type	Default	Description
additionalTurns	UDINT	0	Addition turns to move in modulo movement (positive or negative).
direction	MC_DIRECTION [▶ 124]	MC_DIRECTION.mcDirectionShortestWay	Direction in which the target is approached.

7.2.1.2.5 ST_EndAtTrackAdvancedOptions

Options for the "EndAtTrackAdvanced" command of the Planar Track.

Syntax

Definition:

```

TYPE ST_EndAtTrackAdvancedOptions :
STRUCT
    thisTrackPartPositionIndex : UDINT;
    otherTrackPartPositionIndex : UDINT;
    linkOnlyInSpecifiedPartPositions : BOOL;
END_STRUCT
END_TYPE
    
```

Parameters

Name	Type	Default	Description
thisTrackPartPositionIndex	UDINT	0	The index of the position in which the part of this track is for track connection.
otherTrackPartPositionIndex	UDINT	0	The index of the position in which the part of the other track is for track connection.
linkOnlyInSpecifiedPartPositions	BOOL	FALSE	If false the tracks are connected not only in the given positions configuration of their parts but also in all other (geometrically

Name	Type	Default	Description
			compatible) locations, otherwise only the specified location is connected.

7.2.1.2.6 ST_ExternalSetpointGenerationOptions

Options for the "ExternalSetpointGeneration" command of the Planar Mover.

Syntax

Definition:

```

TYPE ST_ExternalSetpointGenerationOptions :
STRUCT
  mode : MC_EXTERNAL_SET_POSITION_MODE;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
mode	MC_EXTERNAL_SET_POSITION_MODE	MC_EXTERNAL_SET_POSITION_MODE.Absolute	Mode can be relative or absolute, relative can be used parallel to all other commands, absolute only alone.

7.2.1.2.7 ST_GearInPosOnTrackOptions

Options for the "GearInPosOnTrack" command of the Planar Mover.

Syntax

Definition:

```

TYPE ST_GearInPosOnTrackOptions :
STRUCT
  gap : LREAL;
  inSyncToleranceDistance : LREAL;
  directionSlaveSyncPosition : MC_DIRECTION;
  moduloToleranceSlaveSyncPosition : LREAL;
  allowedSlaveSyncDirections : MC_SYNC_DIRECTIONS;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
gap	LREAL	200.0	Minimal distance [mm] to next Planar Mover on track.
inSyncToleranceDistance	LREAL	0.0	Tolerance in absolute value [mm] of position difference to master axis for inSync flag.
directionSlaveSyncPosition	MC_DIRECTION [► 124]	MC_DIRECTION.mcDirectionNonModulo	Direction in which the slave sync position is approached. This parameter is interpreted only in case the slave mover is on a closed-loop track.

Name	Type	Default	Description
			Otherwise, it must be set to its default value.
moduloToleranceSlaveSyncPosition	LREAL	0.0	Tolerance "window" for slave sync position. This parameter is interpreted only in case the slave mover is on a closed-loop track. Otherwise, it must be set to its default value.
allowedSlaveSyncDirections	MC_SYNC DIRECTIONS [▶ 125]	MC_SYNC DIRECTIONS. Positive	Directions in which the slave is allowed to move while in synchronizing phase.

7.2.1.2.8 ST_GearInPosOnTrackWithMasterMoverOptions

Options for the "GearInPosOnTrackWithMasterMover" command of the Planar Mover.

Syntax

Definition:

```

TYPE ST_GearInPosOnTrackWithMasterMoverOptions :
STRUCT
    gap : LREAL;
    inSyncToleranceDistance : LREAL;
    directionSlaveSyncPosition : MC_DIRECTION;
    moduloToleranceSlaveSyncPosition : LREAL;
    directionMasterSyncPosition : MC_DIRECTION;
    moduloToleranceMasterSyncPosition : LREAL;
    allowedSlaveSyncDirections : MC_SYNC DIRECTIONS;
    followMover : BOOL;
END_STRUCT
END_TYPE
    
```

Parameters

Name	Type	Default	Description
gap	LREAL	200.0	Minimal distance [mm] to next Planar Mover on track.
inSyncToleranceDistance	LREAL	0.0	Tolerance in absolute value [mm] of position difference to master axis for inSync flag.
directionSlaveSyncPosition	MC_DIRECTION [▶ 124]	MC_DIRECTION.mcDirectionNonModulo	Direction in which the slave sync position is approached. This parameter is interpreted only in case the slave mover is on a closed-loop track. Otherwise, it must be set to its default value.
moduloToleranceSlaveSyncPosition	LREAL	0.0	Tolerance "window" [mm] for slave sync position. This parameter is

Name	Type	Default	Description
			interpreted only in case the slave mover is on a closed-loop track. Otherwise, it must be set to its default value.
directionMasterSyncPosition	MC_DIRECTION [▶ 124]	MC_DIRECTION.mcDirectionNonModulo	Direction in which the master sync position is approached. This parameter is interpreted only in case the master mover is on a closed-loop track. Otherwise, it must be set to its default value.
moduloToleranceMasterSyncPosition	LREAL	0.0	Tolerance "window" [mm] for master sync position. This parameter is interpreted only in case the master mover is on a closed-loop track. Otherwise, it must be set to its default value.
allowedSlaveSyncDirections	MC_SYNC DIRECTIONS [▶ 125]	MC_SYNC DIRECTIONS.Positionive	Directions in which the slave is allowed to move while in synchronizing phase.
followMover	BOOL	FALSE	If true, the slave PlanarMover will proceed to follow the master PlanarMover after the latter has traversed the masterSyncPosition. In this case the PlanarTrackTrail describes the slave's path towards the masterSyncPosition. If false, the slave moves exclusively on the PlanarTrackTrail specified.

7.2.1.2.9 ST_JoinTrackOptions

Options for the "JoinTrack" command of the Planar Mover.

Syntax

Definition:

```

TYPE ST_JoinTrackOptions :
STRUCT
    useOrientation : BOOL;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
useOrientation	BOOL	TRUE	If true, the target orientation is also reached at the end of the movement.

7.2.1.2.10 ST_LeaveTrackOptions

Options for the "LeaveTrack" command of the Planar Mover.

Syntax

Definition:

```
TYPE ST_LeaveTrackOptions :
STRUCT
    useOrientation : BOOL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
useOrientation	BOOL	TRUE	If true, the target orientation is also reached at the end of the movement.

7.2.1.2.11 ST_MoveCOptions

Options for the "MoveC" command of the Planar Mover.

Syntax

Definition:

```
TYPE ST_MoveCOptions :
STRUCT
    additionalTurns : UDINT;
    direction : MC_DIRECTION;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
additionalTurns	UDINT	0	Addition turns to move in modulo movement (positive or negative).
direction	MC_DIRECTION [▶ 124]	MC_DIRECTION.mcDirectionNonModulo	Direction in which the target is approached.

7.2.1.2.12 ST_MoveOnTrackOptions

Options for the "MoveOnTrack" command of the Planar Mover.

Syntax

Definition:

```

TYPE ST_MoveOnTrackOptions :
STRUCT
    gap          : LREAL;
    direction    : MC_DIRECTION;
    additionalTurns : UDINT;
    moduloTolerance : LREAL;
    gapMode      : MC_GAP_MODE_ON_TRACK;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
gap	LREAL	200.0	Minimal distance [mm] to next Planar Mover on track.
direction	MC_DIRECTION	MC_DIRECTION.mcDirectionNonModulo	Direction in which the target is approached.
additionalTurns	UDINT	0	Addition turns to move in modulo movement (positive or negative).
moduloTolerance	LREAL	0.0	Tolerance "window" [mm] in modulo movement.
gapMode	MC_GAP_MODE_ON_TRACK [▶ 125]	MC_GAP_MODE_ON_TRACK.StandardID	The way in which gap control is done with all neighbours on track.

7.2.1.2.13 ST_MoveToPositionOptions

Options for the "MoveToPosition" command of the Planar Mover.

Syntax

Definition:

```

TYPE ST_MoveToPositionOptions :
STRUCT
    useOrientation : BOOL;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
useOrientation	BOOL	TRUE	If true, the target orientation is also reached at the end of the movement.

7.2.1.2.14 ST_StartFromTrackAdvancedOptions

Options for the "StartFromTrackAdvanced" command of the Planar Track.

Syntax

Definition:

```

TYPE ST_StartFromTrackAdvancedOptions :
STRUCT
    thisTrackPartPositionIndex : UDINT;
    otherTrackPartPositionIndex : UDINT;
    linkOnlyInSpecifiedPartPositions : BOOL;
END_STRUCT
END_TYPE

```

Parameters

Name	Type	Default	Description
thisTrackPartPositionIndex	UDINT	0	The index of the position in which the part of this track is for track connection.
otherTrackPartPositionIndex	UDINT	0	The index of the position in which the part of the other track is for track connection.
linkOnlyInSpecifiedPartPositions	BOOL	FALSE	If false the tracks are connected not only in the given positions configuration of their parts but also in all other (geometrically compatible) locations, otherwise only the specified location is connected.

7.2.2 Function Blocks

7.2.2.1 MC_PlanarEnvironment

A Planar Environment object specifies the environment that Planar Movers can move in. It contains information about the stator objects and boundaries of the movement area.

Do not call the main FB directly. Only use the available methods.

 **Methods**

Name	Description
Clear [▶ 134]	Clears the Planar Environment (stators and boundary).
AddStator [▶ 134]	Adds a stator to the Planar Environment.
CreateBoundary [▶ 134]	Creates a boundary for the Planar Environment based on the previously added stator information or hardware information.
Update [▶ 135]	Updates internal state of the object, must be triggered each cycle.
AddToGroup [▶ 135]	Adds the Planar Environment to the given Planar Group.
RemoveFromGroup [▶ 136]	Removes the Planar Environment from its current Planar Group, i.e. disables collision checks.
GetPlanarObjectInfo [▶ 136]	Returns environment object info (type: environment, id: OID of nc environment).

Required License

TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3.1.4024.12 Advanced Motion Pack V3.1.10.11	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

7.2.2.1.1 Clear

Clear

— **commandFeedback** *Reference To MC_PlanarFeedback*

Clears the Planar Environment (stators and boundary).

Syntax

Definition:

```
METHOD Clear
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.

7.2.2.1.2 AddStator

AddStator

— **commandFeedback** *Reference To MC_PlanarFeedback*

— **lowerX** *LREAL*

— **lowerY** *LREAL*

Adds a stator to the Planar Environment.

Syntax

Definition:

```
METHOD AddStator
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    lowerX           : LREAL;
    lowerY           : LREAL;
END_VAR
```

Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
lowerX	LREAL	The lower x position [mm] of the stator.
lowerY	LREAL	The lower y position [mm] of the stator.

7.2.2.1.3 CreateBoundary

CreateBoundary

— **commandFeedback** *Reference To MC_PlanarFeedback*

Creates a boundary for the Planar Environment based on the previously added stator information or hardware information.

Syntax

Definition:

```
METHOD CreateBoundary
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.

7.2.2.1.4 Update



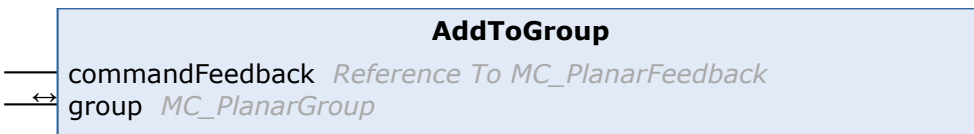
Updates internal state of the object, must be triggered each cycle.

Syntax

Definition:

```
METHOD Update
```

7.2.2.1.5 AddToGroup



Adds the Planar Environment to the given Planar Group.


Syntax

Definition:

```
METHOD AddToGroup
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    group : MC_PlanarGroup;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.

 **In/Outputs**

Name	Type	Description
group	<u>MC_PlanarGroup</u> [▶ 141]	The Planar Group that the mover joins.

7.2.2.1.6 RemoveFromGroup

RemoveFromGroup

— `commandFeedback` *Reference To MC_PlanarFeedback*

Removes the Planar Environment from its current Planar Group, i.e. disables collision checks.

Syntax

Definition:

```
METHOD RemoveFromGroup
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

7.2.2.1.7 GetPlanarObjectInfo

GetPlanarObjectInfo

PlanarObjectInfo GetPlanarObjectInfo —

Returns environment object info (type: environment, id: OID of nc environment).

Syntax

Definition:

```
METHOD GetPlanarObjectInfo : PlanarObjectInfo
```

Return value

PlanarObjectInfo [▶_126]

7.2.2.2 MC_PlanarFeedback

MC_PlanarFeedback

BOOL Active

BOOL Busy

BOOL Aborted

BOOL Error

UDINT ErrorId

PlanarObjectInfo objectInfo

BOOL Done

Displays specific command status information for an associated command, given back by the MC Project.

Syntax

Definition:

```
FUNCTION_BLOCK MC_PlanarFeedback
VAR_OUTPUT
    Active : BOOL;
```



```

Busy      : BOOL;
Aborted   : BOOL;
Error     : BOOL;
ErrorId   : UDINT;
objectInfo : PlanarObjectInfo;
Done      : BOOL;
END_VAR
    
```

 **Outputs**

Name	Type	Description
Active	BOOL	Indicates an active command, i.e. command was accepted and is being executed.
Busy	BOOL	Indicates a busy command, i.e. command is being processed, waiting for execution, or already executing (= also active).
Aborted	BOOL	Indicates the command is aborted, i.e. execution of the command finished due the start of other commands.
Error	BOOL	Indicates the command has an error.
ErrorId	UDINT	Indicates the error id of the command error.
objectInfo	PlanarObjectInfo [▶ 126]	Indicates which object one would collide with.
Done	BOOL	Indicates the command is done, i.e. execution of the command finished successfully.

 **Methods**

Name	Description
Update [▶ 137]	Updates internal state of the object.

Required License

TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3. 1.4024.12 Advanced Motion Pack V3. 1.10.11	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

7.2.2.2.1 Update



Updates internal state of the object.

Syntax

Definition:

```
METHOD Update
```

7.2.2.3 MC_PlanarFeedbackBase

Displays general command status information for an associated command, given back by the MC Project.

Do not call the main FB directly. Only use the available methods.

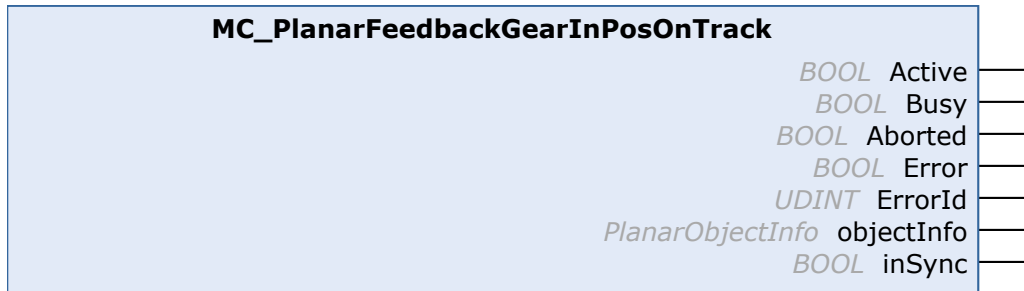
Required License

TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3.1.4024.12 Advanced Motion Pack V3.1.10.11	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

7.2.2.4 MC_PlanarFeedbackGearInPosOnTrack



Displays specific command status information for an associated GearInPosOnTrack command, given back by the MC Project.

Syntax

Definition:

```
FUNCTION_BLOCK MC_PlanarFeedbackGearInPosOnTrack
VAR_OUTPUT
    Active      : BOOL;
    Busy        : BOOL;
    Aborted     : BOOL;
    Error       : BOOL;
    ErrorId     : UDINT;
    objectInfo  : PlanarObjectInfo;
    inSync      : BOOL;
END_VAR
```

 **Outputs**

Name	Type	Description
Active	BOOL	Indicates an active command, i.e. command was accepted and is being executed.
Busy	BOOL	Indicates a busy command, i.e. command is being processed, waiting for execution, or already executing (= also active).
Aborted	BOOL	Indicates the command is aborted, i.e. execution of the command finished due the start of other commands.
Error	BOOL	Indicates the command has an error.
ErrorId	UDINT	Indicates the error id of the command error.
objectInfo	PlanarObjectInfo [▶ 126]	Indicates which object one would collide with.
inSync	BOOL	Indicates whether the mover is currently in sync with the master (within tolerance specified in the command options).

 Methods

Name	Description
Update [ 139]	Updates internal state of the object.

Required License

TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3.1.4024.12 Advanced Motion Pack V3.1.10.11	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

7.2.2.4.1 Update



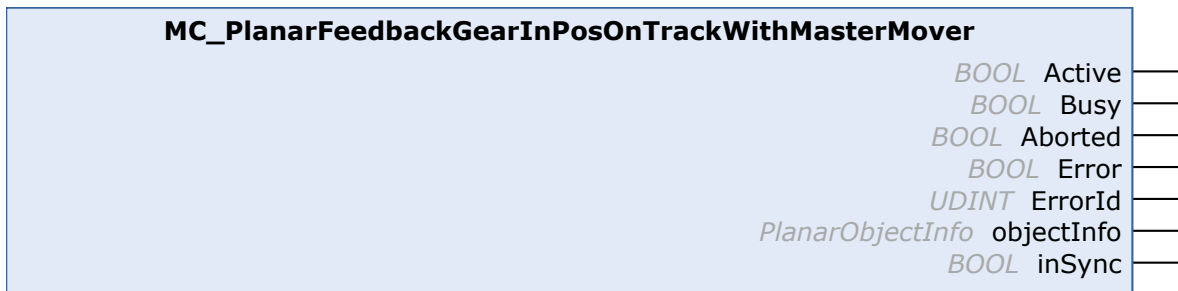
Updates internal state of the object.

Syntax

Definition:

METHOD Update

7.2.2.5 MC_PlanarFeedbackGearInPosOnTrackWithMasterMover



Displays specific command status information for an associated GearInPosOnTrack command, given back by the MC Project.

Syntax

Definition:

```
FUNCTION_BLOCK MC_PlanarFeedbackGearInPosOnTrackWithMasterMover
VAR_OUTPUT
    Active      : BOOL;
    Busy        : BOOL;
    Aborted     : BOOL;
    Error       : BOOL;
    ErrorId     : UDINT;
    objectInfo  : PlanarObjectInfo;
    inSync     : BOOL;
END_VAR
```

 Outputs

Name	Type	Description
Active	BOOL	Indicates an active command, i.e. command was accepted and is being executed.

Name	Type	Description
Busy	BOOL	Indicates a busy command, i.e. command is being processed, waiting for execution, or already executing (= also active).
Aborted	BOOL	Indicates the command is aborted, i.e. execution of the command finished due the start of other commands.
Error	BOOL	Indicates the command has an error.
ErrorId	UDINT	Indicates the error id of the command error.
objectInfo	PlanarObjectInfo [▶ 126]	Indicates which object one would collide with.
inSync	BOOL	Indicates whether the mover is currently in sync with the master (within tolerance specified in the command options).

Methods

Name	Description
Update [▶ 140]	Updates internal state of the object.

Required License

TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3.1.4024.12 Advanced Motion Pack V3.1.10.11	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

7.2.2.5.1 Update

Update

Updates internal state of the object.

Syntax

Definition:

METHOD Update

7.2.2.6 MC_PlanarFeedbackInSync

Base class for all specialized feedbacks featuring an inSync output.

Do not call the main FB directly. Only use the available methods.

Required License

TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3.1.4024.12 Advanced Motion Pack V3.1.10.11	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

7.2.2.7 MC_PlanarGroup

A Planar Group object. Planar Movers and other objects added to the group perform collision checks against each other.

Do not call the main FB directly. Only use the available methods.

Methods

Name	Description
Enable [▶ 141]	Starts enabling the Planar Group.
Disable [▶ 141]	Starts disabling the Planar Group.
Reset [▶ 142]	Starts resetting the Planar Group.
Update [▶ 142]	Updates internal state of the object, must be triggered each cycle.

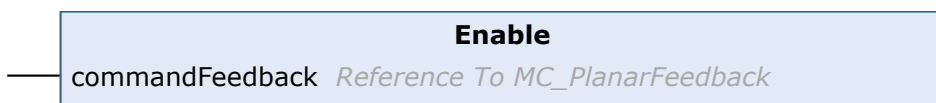
Required License

TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3.1.4024.12 Advanced Motion Pack V3.1.10.11	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

7.2.2.7.1 Enable



Starts enabling the Planar Group.

Syntax

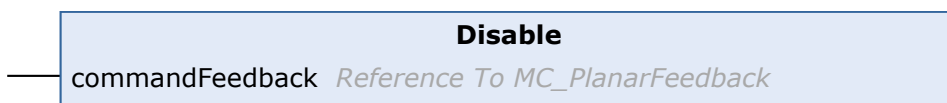
Definition:

```
METHOD Enable
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.

7.2.2.7.2 Disable



Starts disabling the Planar Group.

Syntax

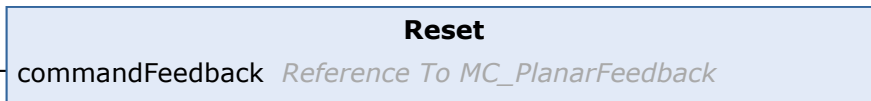
Definition:

```
METHOD Disable
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedba ck	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.

7.2.2.7.3 Reset



Starts resetting the Planar Group.

Syntax

Definition:

```
METHOD Reset
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedba ck	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.

7.2.2.7.4 Update



Updates internal state of the object, must be triggered each cycle.

Syntax

Definition:

```
METHOD Update
```

7.2.2.8 MC_PlanarMover

A Planar Mover object capable of moving within a plane. Limited movement vertical to the plane is available.

Do not call the main FB directly. Only use the available methods.

Methods

Name	Description
MoveToPosition [▶ 144]	Initiates a direct movement to the specified position.
JoinTrack [▶ 144]	Initiates a direct movement to the specified track. At the end of the movement the mover joins the track.
LeaveTrack [▶ 145]	Initiates a direct movement to the specified position. At the beginning of the movement the track is left.
MoveOnTrack [▶ 146]	Initiates a movement on the track to the specified position.
GearInPosOnTrack [▶ 147]	Initiates a GearInPos movement along a specified trail.
GearInPosOnTrackWithMasterMover [▶ 148]	Initiates a GearInPos movement along a specified trail, in which the master setpoints are provided by another PlanarMover.
MoveZ [▶ 149]	Initiates a movement for the z component.
MoveA [▶ 149]	Initiates a movement for the a component.
MoveB [▶ 150]	Initiates a movement for the b component.
MoveC [▶ 150]	Initiates a movement for the c component.
AdoptTrackOrientation [▶ 151]	Initiates a movement for the c component.
Halt [▶ 151]	Initiates a halt.
Enable [▶ 152]	Starts enabling the Planar Mover.
Disable [▶ 152]	Starts disabling the Planar Mover.
Reset [▶ 153]	Starts resetting the Planar Mover.
Update [▶ 153]	Updates internal state of the object, must be triggered each cycle.
SetPosition [▶ 153]	Sets the position of the Planar Mover. Only possible if the Planar Mover is disabled.
StartExternalSetpointGeneration [▶ 154]	Starts the external setpoint generation, the user must supply setpoints from this PLC cycle on in every PLC cycle.
StopExternalSetpointGeneration [▶ 154]	Ends the external setpoint generation, called after last SetExternalSetpoint (in the same PLC cycle).
SetExternalSetpoint [▶ 155]	Sets the external setpoint for the Planar Mover without ref sys id (for relative mode), must be called each PLC cycle during external setpoint generation.
SetExternalSetpointReferenceId [▶ 155]	Sets the external setpoint for the Planar Mover with corresponding reference system id, must be called each PLC cycle during external setpoint generation.
AddToGroup [▶ 156]	Adds the Planar Mover to the given Planar Group.
RemoveFromGroup [▶ 156]	Removes the Planar Mover from its current Planar Group, i.e. disables collision checks.
GetPositionOnCurrentPart [▶ 157]	Sets the values of the given position to the movers position values on the current part.
GetTrackTrailInformation [▶ 157]	Sets the values of the given track trail to the movers internally planned track trail.
GetPlanarObjectInfo [▶ 157]	Returns mover object info (type: mover, id: OID of nc mover).

Required License

TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3.1.4024.12 Advanced Motion Pack V3.1.10.11	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

7.2.2.8.1 MoveToPosition



Initiates a direct movement to the specified position.

Syntax

Definition:


```

METHOD MoveToPosition
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    targetPosition : PositionXYC;
END_VAR
VAR_INPUT
    constraint      : Reference To IPlcDynamicConstraint;
    options         : Reference To ST_MoveToPositionOptions;
END_VAR

```

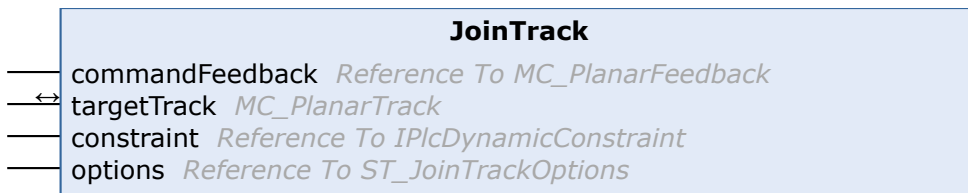
 Inputs

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.
constraint	Reference To <u>IPlcDynamicConstraint</u>	Dynamic constraints for this movement.
options	Reference To <u>ST_MoveToPositionOptions</u> [▶ 132]	Options for the movement.

 In/Outputs

Name	Type	Description
targetPosition	PositionXYC	Target position for the movement.

7.2.2.8.2 JoinTrack



Initiates a direct movement to the specified track. At the end of the movement the mover joins the track.

Syntax

Definition:

```

METHOD JoinTrack
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;

```




```

END_VAR
VAR_IN_OUT
    targetTrack      : MC_PlanarTrack;
END_VAR
VAR_INPUT
    constraint       : Reference To IPlcDynamicConstraint;
    options          : Reference To ST_JoinTrackOptions;
END_VAR
    
```

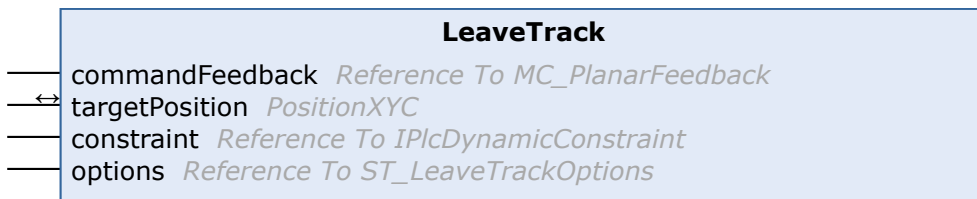
 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
constraint	Reference To IPlcDynamicConstraint	Dynamic constraints for this movement.
options	Reference To ST_JoinTrackOptions [▶ 130]	Options for the movement.

 **In/Outputs**

Name	Type	Description
targetTrack	MC_PlanarTrack [▶ 161]	Target track for the movement.

7.2.2.8.3 LeaveTrack



Initiates a direct movement to the specified position. At the beginning of the movement the track is left.

Syntax

Definition:


```

METHOD LeaveTrack
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    targetPosition  : PositionXYC;
END_VAR
VAR_INPUT
    constraint       : Reference To IPlcDynamicConstraint;
    options          : Reference To ST_LeaveTrackOptions;
END_VAR
    
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
constraint	Reference To IPlcDynamicConstraint	Dynamic constraints for this movement.

Name	Type	Description
options	Reference To <u>ST_LeaveTrackOptions</u> [▶_131]	Options for the movement.

 In/Outputs

Name	Type	Description
targetPosition	PositionXYC	Target position for the movement.

7.2.2.8.4 MoveOnTrack

MoveOnTrack

- **commandFeedback** *Reference To MC_PlanarFeedback*
- **targetTrack** *Reference To MC_PlanarTrack*
- **targetPositionOnTrack** *LREAL*
- **constraint** *Reference To DynamicConstraint_PathXY*
- **options** *Reference To ST_MoveOnTrackOptions*

Initiates a movement on the track to the specified position.

Syntax

Definition:

```
METHOD MoveOnTrack
VAR_INPUT
    commandFeedback      : Reference To MC_PlanarFeedback;
    targetTrack          : Reference To MC_PlanarTrack;
    targetPositionOnTrack : LREAL;
    constraint           : Reference To DynamicConstraint_PathXY;
    options              : Reference To ST_MoveOnTrackOptions;
END_VAR
```

 Inputs

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶_136]	The feedback object for the command.
targetTrack	Reference To <u>MC_PlanarTrack</u> [▶_161]	Target track for the movement. If none is specified, this defaults to the current track.
targetPositionOnTrack	LREAL	Target position [mm] on the target track.
constraint	Reference To <u>DynamicConstraint_PathXY</u>	Constraint on maximal dynamics during the movement (V, A, D, J).
options	Reference To <u>ST_MoveOnTrackOptions</u>	Options for the movement.

7.2.2.8.5 GearInPosOnTrack

GearInPosOnTrack	
—	commandFeedback <i>Reference To MC_PlanarFeedbackGearInPosOnTrack</i>
—	masterAxis <i>OTCID</i>
—	trackTrail <i>Reference To MC_PlanarTrackTrail</i>
—	masterSyncPosition <i>LREAL</i>
—	slaveSyncPosition <i>LREAL</i>
↔	slaveSyncPositionTrack <i>MC_PlanarTrack</i>
—	constraint <i>Reference To DynamicConstraint_PathXY</i>
—	options <i>Reference To ST_GearInPosOnTrackOptions</i>

Initiates a GearInPos movement along a specified trail.

Syntax


Definition:

```

METHOD GearInPosOnTrack
VAR_INPUT
  commandFeedback      : Reference To MC_PlanarFeedbackGearInPosOnTrack;
  masterAxis           : OTCID;
  trackTrail           : Reference To MC_PlanarTrackTrail;
  masterSyncPosition   : LREAL;
  slaveSyncPosition    : LREAL;
END_VAR
VAR_IN_OUT
  slaveSyncPositionTrack : MC_PlanarTrack;
END_VAR
VAR_INPUT
  constraint           : Reference To DynamicConstraint_PathXY;
  options              : Reference To ST_GearInPosOnTrackOptions;
END_VAR
    
```

 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedbackGearInPosOnTrack [▶_138]	The command specific feedback object for the command.
masterAxis	OTCID	Master axis being followed.
trackTrail	Reference To MC_PlanarTrackTrail [▶_171]	Track trail determining along which tracks the GearInPos movement is allowed to proceed.
masterSyncPosition	LREAL	Position of the master axis (interpreted as [mm]) at which the slave is inSync.
slaveSyncPosition	LREAL	Arc length [mm] on track given by slaveSyncPositionTrackOID at which the slave is inSync. Possibly interpreted in modulo fashion, depending on options.
constraint	Reference To DynamicConstraint_PathXY	Constraint on maximal dynamics during the movement (V, A, D, J).
options	Reference To ST_GearInPosOnTrackOptions [▶_128]	Options for the movement.

 In/Outputs

Name	Type	Description
slaveSyncPositionTrack	MC_PlanarTrack [▶_161]	Track on which the slave is inSync.

7.2.2.8.6 GearInPosOnTrackWithMasterMover

GearInPosOnTrackWithMasterMover	
commandFeedback	Reference To MC_PlanarFeedbackGearInPosOnTrackWithMasterMover
masterMover	MC_PlanarMover
trackTrail	Reference To MC_PlanarTrackTrail
masterSyncPosition	LREAL
masterSyncPositionTrack	MC_PlanarTrack
slaveSyncPosition	LREAL
slaveSyncPositionTrack	MC_PlanarTrack
constraint	Reference To DynamicConstraint_PathXY
options	Reference To ST_GearInPosOnTrackWithMasterMoverOptions

Initiates a GearInPos movement along a specified trail, in which the master setpoints are provided by another PlanarMover.

Syntax


Definition:

```

METHOD GearInPosOnTrackWithMasterMover
VAR_INPUT
    commandFeedback      : Reference To MC_PlanarFeedbackGearInPosOnTrackWithMasterMover;
END_VAR
VAR_IN_OUT
    masterMover          : MC_PlanarMover;
END_VAR
VAR_INPUT
    trackTrail           : Reference To MC_PlanarTrackTrail;
    masterSyncPosition   : LREAL;
END_VAR
VAR_IN_OUT
    masterSyncPositionTrack : MC_PlanarTrack;
END_VAR
VAR_INPUT
    slaveSyncPosition    : LREAL;
END_VAR
VAR_IN_OUT
    slaveSyncPositionTrack : MC_PlanarTrack;
END_VAR
VAR_INPUT
    constraint           : Reference To DynamicConstraint_PathXY;
    options              : Reference To ST_GearInPosOnTrackWithMasterMoverOptions;
END_VAR
    
```

Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedbackGearInPosOnTrackWithMasterMover [▶_139]	The command specific feedback object for the command.
trackTrail	Reference To MC_PlanarTrackTrail [▶_171]	Track trail determining along which tracks the GearInPos movement is allowed to proceed.
masterSyncPosition	LREAL	Position of the master axis (interpreted as [mm]) at which the slave is inSync.
slaveSyncPosition	LREAL	Arc length on track [mm] given by slaveSyncPositionTrackOID at which the slave is inSync. Possibly interpreted in modulo fashion, depending on options.
constraint	Reference To DynamicConstraint_PathXY	Constraint on maximal dynamics during the movement (V, A, D, J).
options	Reference To ST_GearInPosOnTrackWithMasterMoverOptions [▶_129]	Options for the movement.

 In/Outputs

Name	Type	Description
masterMover	MC_PlanarMover [▶ 142]	Master mover being followed.
masterSyncPositionTrack	MC_PlanarTrack [▶ 161]	Track on which the master is inSync.
slaveSyncPositionTrack	MC_PlanarTrack [▶ 161]	Track on which the slave is inSync.

7.2.2.8.7 MoveZ

MoveZ

— **commandFeedback** *Reference To MC_PlanarFeedback*

— **targetPosition** *LREAL*

— **constraint** *Reference To IPlcDynamicConstraint*

Initiates a movement for the z component.

Syntax

Definition:

```

METHOD MoveZ
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    targetPosition  : LREAL;
    constraint       : Reference To IPlcDynamicConstraint;
END_VAR
    
```

 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
targetPosition	LREAL	Target position [mm] for the movement.
constraint	Reference To IPlcDynamicConstraint	Dynamic constraints for this movement.

7.2.2.8.8 MoveA

MoveA

— **commandFeedback** *Reference To MC_PlanarFeedback*

— **targetPosition** *LREAL*

— **constraint** *Reference To IPlcDynamicConstraint*

Initiates a movement for the a component.

Syntax

Definition:

```

METHOD MoveA
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    targetPosition  : LREAL;
    constraint       : Reference To IPlcDynamicConstraint;
END_VAR
    
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
targetPosition	LREAL	Target position [deg] for the movement.
constraint	Reference To IPlcDynamicConstraint	Dynamic constraints for this movement.

7.2.2.8.9 MoveB

MoveB

— **commandFeedback** *Reference To MC_PlanarFeedback*

— **targetPosition** *LREAL*

— **constraint** *Reference To IPlcDynamicConstraint*

Initiates a movement for the b component.

Syntax

Definition:

```

METHOD MoveB
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    targetPosition  : LREAL;
    constraint       : Reference To IPlcDynamicConstraint;
END_VAR
    
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
targetPosition	LREAL	Target position [deg] for the movement.
constraint	Reference To IPlcDynamicConstraint	Dynamic constraints for this movement.

7.2.2.8.10 MoveC

MoveC

— **commandFeedback** *Reference To MC_PlanarFeedback*

— **targetPosition** *LREAL*

— **constraint** *Reference To IPlcDynamicConstraint*

— **options** *Reference To ST_MoveCOptions*

Initiates a movement for the c component.

Syntax


Definition:

```

METHOD MoveC
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    targetPosition  : LREAL;
    
```

```

constraint      : Reference To IPlcDynamicConstraint;
options        : Reference To ST_MoveCOptions;
END_VAR
    
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.
targetPosition	LREAL	Target position [deg] for the movement.
constraint	Reference To IPlcDynamicConstraint	Dynamic constraints for this movement.
options	Reference To <u>ST_MoveCOptions</u> [▶ 131]	Options for the rotation.

7.2.2.8.11 **AdoptTrackOrientation**

AdoptTrackOrientation

— **commandFeedback** *Reference To MC_PlanarFeedback*

— **constraint** *Reference To IPlcDynamicConstraint*

— **options** *Reference To ST_AdoptTrackOrientationOptions*

Initiates a movement for the c component.

Syntax

Definition:

```

METHOD AdoptTrackOrientation
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    constraint      : Reference To IPlcDynamicConstraint;
    options        : Reference To ST_AdoptTrackOrientationOptions;
END_VAR
    
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.
constraint	Reference To IPlcDynamicConstraint	Dynamic constraints for this movement.
options	Reference To <u>ST_AdoptTrackOrientationOptions</u> [▶ 127]	Options for the rotation.

7.2.2.8.12 **Halt**

Halt

— **commandFeedback** *Reference To MC_PlanarFeedback*

— **constraint** *Reference To IPlcDynamicConstraint*

Initiates a halt.

Syntax

Definition:

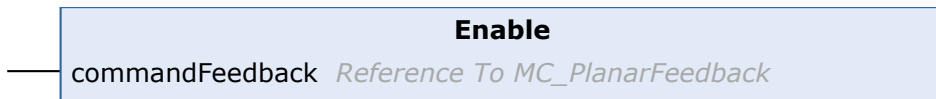
```

METHOD Halt
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    constraint       : Reference To IPlcDynamicConstraint;
END_VAR

```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
constraint	Reference To IPlcDynamicConstraint	Dynamic constraints for this movement.

7.2.2.8.13 Enable

Starts enabling the Planar Mover.

Syntax

Definition:

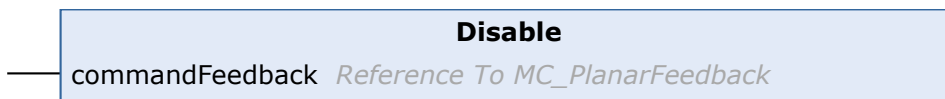
```

METHOD Enable
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR

```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.

7.2.2.8.14 Disable

Starts disabling the Planar Mover.

Syntax

Definition:

```

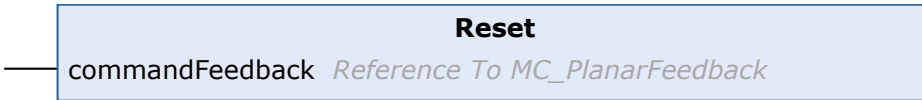
METHOD Disable
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR

```


 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ __136]	The feedback object for the command.

7.2.2.8.15 Reset



Starts resetting the Planar Mover.

Syntax

Definition:

```
METHOD Reset
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ __136]	The feedback object for the command.

7.2.2.8.16 Update



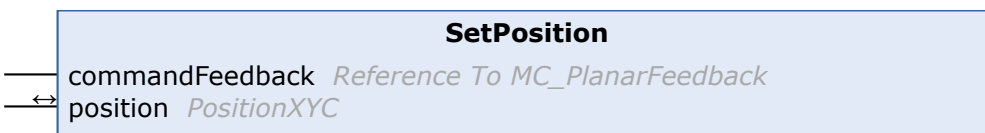
Updates internal state of the object, must be triggered each cycle.

Syntax

Definition:

```
METHOD Update
```

7.2.2.8.17 SetPosition



Sets the position of the Planar Mover. Only possible if the Planar Mover is disabled.

Syntax

Definition:

```
METHOD SetPosition
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    position : PositionXYC;
END_VAR
```

✚ Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.

✚ In/Outputs

Name	Type	Description
position	PositionXYC	New position of the Planar Mover.

7.2.2.8.18 StartExternalSetpointGeneration

StartExternalSetpointGeneration

commandFeedback	Reference To MC_PlanarFeedback
options	Reference To ST_ExternalSetpointGenerationOptions

Starts the external setpoint generation, the user must supply setpoints from this PLC cycle on in every PLC cycle.

Syntax

Definition:

```
METHOD StartExternalSetpointGeneration
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    options          : Reference To ST_ExternalSetpointGenerationOptions;
END_VAR
```

✚ Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
options	Reference To ST_ExternalSetpointGene- rationOptions [▶ 128]	Options for the movement.

7.2.2.8.19 StopExternalSetpointGeneration

StopExternalSetpointGeneration

commandFeedback	Reference To MC_PlanarFeedback
-----------------	--------------------------------

Ends the external setpoint generation, called after last SetExternalSetpoint (in the same PLC cycle).

Syntax

Definition:

```
METHOD StopExternalSetpointGeneration
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

7.2.2.8.20 SetExternalSetpoint

SetExternalSetpoint

- setPosition *MoverVector*
- setVelocity *MoverVector*
- setAcceleration *MoverVector*

Sets the external setpoint for the Planar Mover without ref sys id (for relative mode), must be called each PLC cycle during external setpoint generation.

Syntax

Definition:

```
METHOD SetExternalSetpoint
VAR_INPUT
    setPosition      : MoverVector;
    setVelocity      : MoverVector;
    setAcceleration  : MoverVector;
END_VAR
```

 Inputs

Name	Type	Description
setPosition	MoverVector	Position that is send to the Planar Mover.
setVelocity	MoverVector	Velocity that is send to the Planar Mover.
setAcceleration	MoverVector	Acceleration that is send to the Planar Mover.

7.2.2.8.21 SetExternalSetpointReferenceId

SetExternalSetpointReferenceId

- commandFeedback *Reference To MC_PlanarFeedback*
- setPosition *MoverVector*
- setVelocity *MoverVector*
- setAcceleration *MoverVector*
- referenceSystemOid *OTCID*

Sets the external setpoint for the Planar Mover with corresponing reference system id, must be called each PLC cycle during external setpoint generation.

Syntax

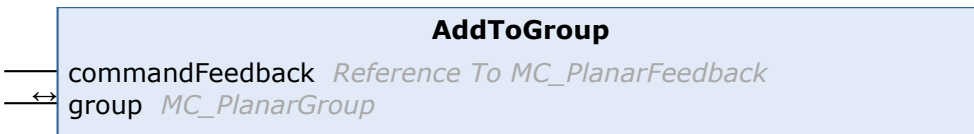
Definition:

```
METHOD SetExternalSetpointReferenceId
VAR_INPUT
    commandFeedback  : Reference To MC_PlanarFeedback;
    setPosition      : MoverVector;
    setVelocity      : MoverVector;
    setAcceleration  : MoverVector;
    referenceSystemOid : OTCID;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
setPosition	MoverVector	Position that is send to the Planar Mover.
setVelocity	MoverVector	Velocity that is send to the Planar Mover.
setAcceleration	MoverVector	Acceleration that is send to the Planar Mover.
referenceSystemOid	OTCID	Part or coordinate system id.

7.2.2.8.22 AddToGroup



Adds the Planar Mover to the given Planar Group.

Syntax


Definition:

```

METHOD AddToGroup
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    group           : MC_PlanarGroup;
END_VAR
    
```

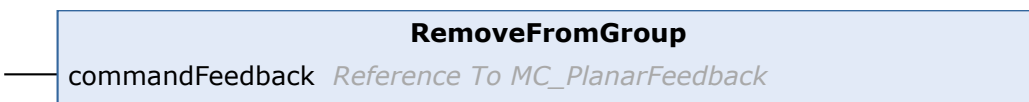
 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.

 **In/Outputs**

Name	Type	Description
group	MC_PlanarGroup [▶ 141]	The Planar Group that the Planar Mover joins.

7.2.2.8.23 RemoveFromGroup



Removes the Planar Mover from its current Planar Group, i.e. disables collision checks.

Syntax

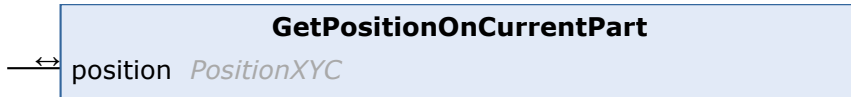
Definition:

```
METHOD RemoveFromGroup
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.

7.2.2.8.24 **GetPositionOnCurrentPart**




Sets the values of the given position to the movers position values on the current part.

Syntax

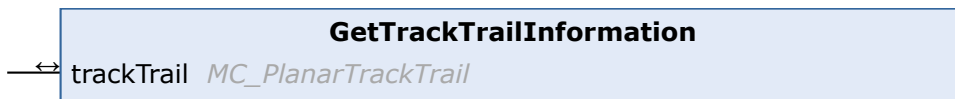
Definition:

```
METHOD GetPositionOnCurrentPart
VAR_IN_OUT
    position : PositionXYC;
END_VAR
```

 In/Outputs

Name	Type	Description
position	PositionXYC	The position on the planar part.

7.2.2.8.25 **GetTrackTrailInformation**




Sets the values of the given track trail to the movers internally planned track trail.

Syntax

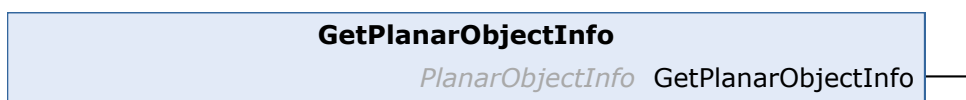
Definition:

```
METHOD GetTrackTrailInformation
VAR_IN_OUT
    trackTrail : MC_PlanarTrackTrail;
END_VAR
```

 In/Outputs

Name	Type	Description
trackTrail	MC_PlanarTrackTrail	Track trail determining along which tracks the mover will move with the current mvoement on track.

7.2.2.8.26 **GetPlanarObjectInfo**



Returns mover object info (type: mover, id: OID of nc mover).

Syntax

Definition:

```
METHOD GetPlanarObjectInfo : PlanarObjectInfo
```

Return value

[PlanarObjectInfo](#) [[▶](#) [126](#)]

7.2.2.9 MC_PlanarPart

A Planar Part object represents the area that Planar Movers can move on. It contains information about the stator objects.

Do not call the main FB directly. Only use the available methods.

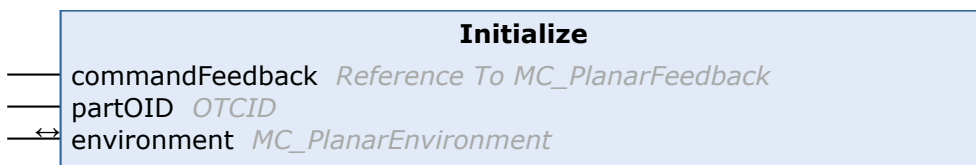
Methods

Name	Description
Initialize [▶ 158]	Initialize the Planar Part, i.e. connecting it to the MC via its OID.
ActivatePosition [▶ 159]	Activates the position given by part position index.
ForceEnable [▶ 159]	Removes force disable and enables the part.
AllowEnable [▶ 160]	From now on the part can be enabled until ForceDisablePart is called.
ForceDisable [▶ 160]	Disables the part and keeps it disabled until AllowEnabledPart is called.
Reset [▶ 160]	Resets the part.
GetPosition [▶ 161]	Sets the values of the given position to the parts position values.

Required License

TC3 Planar Motion Base

7.2.2.9.1 Initialize



Initialize the Planar Part, i.e. connecting it to the MC via its OID.


Syntax

Definition:

```
METHOD Initialize
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    partOID         : OTCID;
END_VAR
VAR_IN_OUT
    environment     : MC_PlanarEnvironment;
END_VAR
```

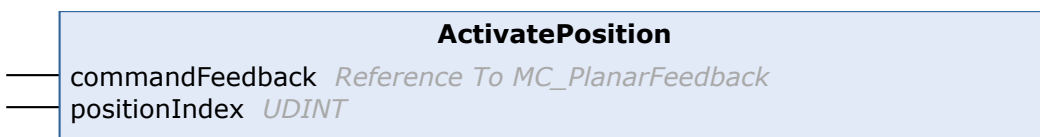
 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
partOID	OTCID	OID of the part.

 In/Outputs

Name	Type	Description
environment	MC_PlanarEnvironment [▶ 133]	Environment the part is in.

7.2.2.9.2 ActivatePosition



Activates the position given by part position index.

Syntax

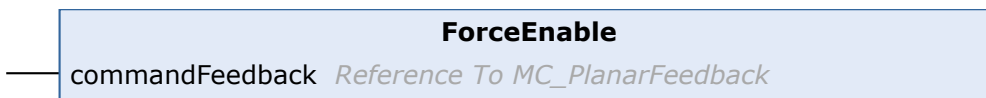
Definition:

```
METHOD ActivatePosition
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
    positionIndex   : UDINT;
END_VAR
```

 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.
positionIndex	UDINT	Index of the position.

7.2.2.9.3 ForceEnable



Removes force disable and enables the part.

Syntax

Definition:

```
METHOD ForceEnable
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback	The feedback object for the command.

7.2.2.9.4 AllowEnable

AllowEnable

commandFeedback *Reference To MC_PlanarFeedback*

From now on the part can be enabled until ForceDisablePart is called.

Syntax

Definition:

```
METHOD AllowEnable
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

7.2.2.9.5 ForceDisable

ForceDisable

commandFeedback *Reference To MC_PlanarFeedback*

Disables the part and keeps it disabled until AllowEnabledPart is called.

Syntax

Definition:

```
METHOD ForceDisable
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

7.2.2.9.6 Reset

Reset

commandFeedback *Reference To MC_PlanarFeedback*

Resets the part.

Syntax

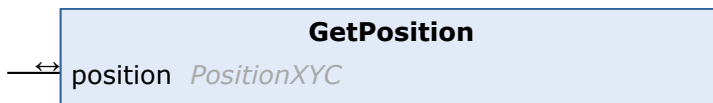
Definition:

```
METHOD Reset
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.

7.2.2.9.7 GetPosition




Sets the values of the given position to the parts position values.

Syntax

Definition:

```
METHOD GetPosition
VAR_IN_OUT
    position : PositionXYC;
END_VAR
```

 **In/Outputs**

Name	Type	Description
position	PositionXYC	The position of the Planar Part.

7.2.2.10 MC_PlanarTrack

A track within a plane which Planar Movers can follow. Planar Movers on the track automatically avoid collisions with each other. The Planar Track can consist of several consecutive segments and be joined with other Planar Tracks at its start/end.

Do not call the main FB directly. Only use the available methods.

 **Methods**

Name	Description
Clear [▶ 162]	Clears the geometric information of the Planar Track.
AppendPosition [▶ 163]	Appends a position to the Planar Track.
AppendLine [▶ 163]	Appends a line to the Planar Track.
AppendCircle [▶ 164]	Appends a circular arc to the Planar Track.
CloseLoop [▶ 164]	Closes the loop of the Planar Track, no further part can be appended.
StartFromTrack [▶ 165]	Sets the other Planar Track's endpoint as start point of this Planar Track, transition is smooth. The other Planar Track is blocked for further changes (until it is cleared).

Name	Description
EndAtTrack [▶ 165]	Appends a smooth transition from the end of this Planar Track to the other Planar Track's start point. The Planar Track is blocked for further changes (until it is cleared).
StartFromTrackAdvanced [▶ 166]	Sets the other Planar Track's endpoint as start point of this Planar Track, transition is smooth. The other Planar Track is blocked for further geometrical changes (until it is cleared).
EndAtTrackAdvanced [▶ 167]	Appends a smooth transition from the end of this Planar Track to the other Planar Track's start point. The Planar Track is blocked for further geometrical changes (until it is cleared).
Enable [▶ 167]	Starts enabling the Planar Track.
Disable [▶ 168]	Starts disabling the Planar Track.
Reset [▶ 168]	Starts resetting the Planar Track.
GetArcLengthClosestTo [▶ 168]	Calculate the arc length value where the Planar Track is closest to a geometry's center point.
GetPositionAt [▶ 169]	Get a position on the Planar Track at a specific arc length value.
GetLength [▶ 169]	Returns the Planar Track's length, -1 return value indicates no connection to Nc Track.
GetPlanarObjectInfo [▶ 170]	Returns track object info (type: track, id: OID of nc track).
Update [▶ 170]	Updates internal state of the object, must be triggered each cycle.
AddToGroup [▶ 170]	Adds the Planar Track to the given Planar Group.
RemoveFromGroup [▶ 171]	Removes the Planar Track from its current Planar Group, i.e. disables collision checks.

Required License

TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3.1.4024.12 Advanced Motion Pack V3.1.10.11	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

7.2.2.10.1 Clear

Clear	
commandFeedback	Reference To MC_PlanarFeedback

Clears the geometric information of the Planar Track.

Syntax

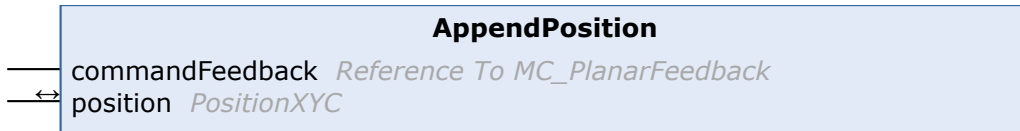
Definition:

```
METHOD Clear
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶ 136]	The feedback object for the command.

7.2.2.10.2 AppendPosition



Appends a position to the Planar Track.

Syntax

Definition:

```
METHOD AppendPosition
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    position        : PositionXYC;
END_VAR
```

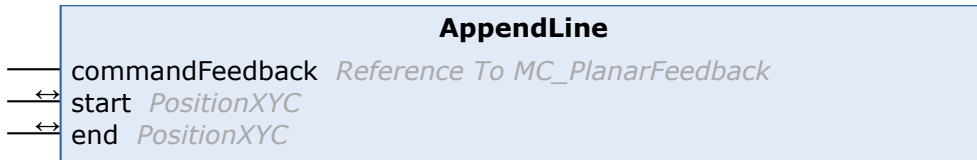
Inputs

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.

In/Outputs

Name	Type	Description
position	PositionXYC	Position that is the new endpoint of the Planar Track.

7.2.2.10.3 AppendLine



Appends a line to the Planar Track.


Syntax

Definition:

```
METHOD AppendLine
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    start           : PositionXYC;
    end             : PositionXYC;
END_VAR
```

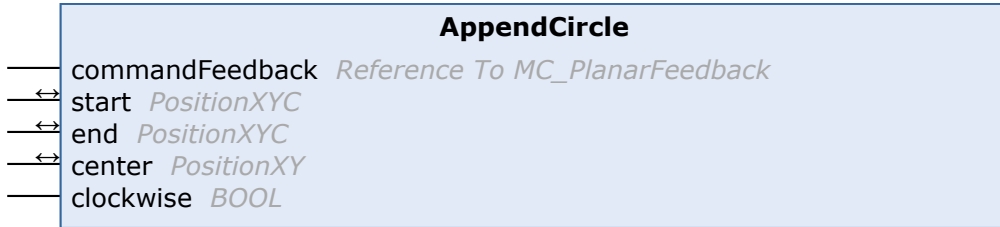
Inputs

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.

 In/Outputs

Name	Type	Description
start	PositionXYC	Start position of the line.
end	PositionXYC	End position of the line, this position is the new endpoint of the Planar Track.

7.2.2.10.4 AppendCircle



Appends a circular arc to the Planar Track.

Syntax


Definition:

```

METHOD AppendCircle
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    start           : PositionXYC;
    end             : PositionXYC;
    center          : PositionXY;
END_VAR
VAR_INPUT
    clockwise       : BOOL;
END_VAR
    
```

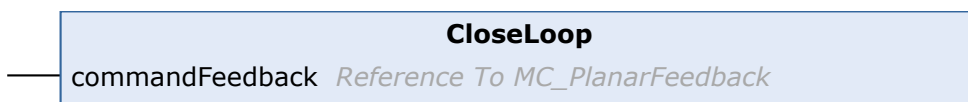
 Inputs

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.
clockwise	BOOL	Indicates if the clockwise circle is appended.

 In/Outputs

Name	Type	Description
start	PositionXYC	Start position of the circular arc.
end	PositionXYC	End position of the circular arc, this position is the new endpoint of the Planar Track.
center	PositionXY	Center of the circular arc.

7.2.2.10.5 CloseLoop



Closes the loop of the Planar Track, no further part can be appended.

Syntax

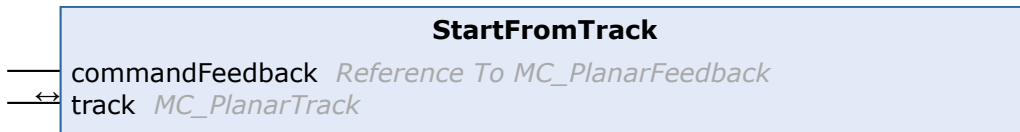
Definition:

```
METHOD CloseLoop
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 Inputs

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.

7.2.2.10.6 StartFromTrack



Sets the other Planar Track's endpoint as start point of this Planar Track, transition is smooth. The other Planar Track is blocked for further changes (until it is cleared).


Syntax

Definition:

```
METHOD StartFromTrack
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    track : MC_PlanarTrack;
END_VAR
```

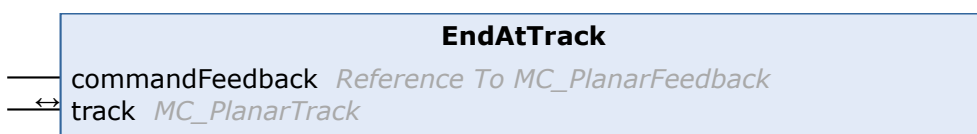
 Inputs

Name	Type	Description
commandFeedback	Reference To <u>MC_PlanarFeedback</u> [▶ 136]	The feedback object for the command.

 In/Outputs

Name	Type	Description
track	<u>MC_PlanarTrack</u> [▶ 161]	The other Planar Track.

7.2.2.10.7 EndAtTrack



Appends a smooth transition from the end of this Planar Track to the other Planar Track's start point. The Planar Track is blocked for further changes (until it is cleared).


Syntax

Definition:

```
METHOD EndAtTrack
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    track           : MC_PlanarTrack;
END_VAR
```

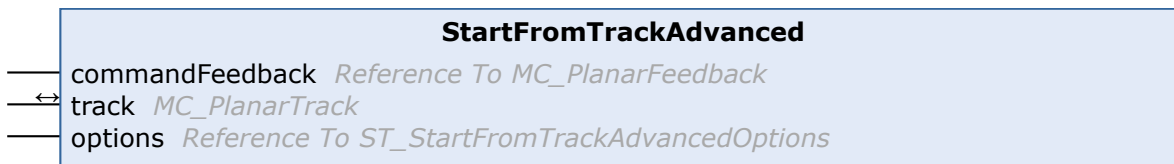
 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

 **In/Outputs**

Name	Type	Description
track	MC_PlanarTrack [▶_161]	The other Planar Track.

7.2.2.10.8 StartFromTrackAdvanced



Sets the other Planar Track's endpoint as start point of this Planar Track, transition is smooth. The other Planar Track is blocked for further geometrical changes (until it is cleared).


Syntax

Definition:

```
METHOD StartFromTrackAdvanced
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    track           : MC_PlanarTrack;
END_VAR
VAR_INPUT
    options         : Reference To ST_StartFromTrackAdvancedOptions;
END_VAR
```

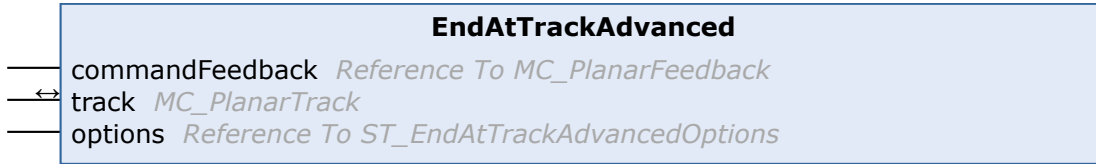
 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.
options	Reference To ST_StartFromTrackAdvancedOptions [▶_132]	Options for the connection, i.e. which connections are closed.

 In/Outputs

Name	Type	Description
track	MC_PlanarTrack [▶_161]	The other Planar Track.

7.2.2.10.9 EndAtTrackAdvanced



Appends a smooth transition from the end of this Planar Track to the other Planar Track's start point. The Planar Track is blocked for further geometrical changes (until it is cleared).


Syntax

Definition:

```
METHOD EndAtTrackAdvanced
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    track           : MC_PlanarTrack;
END_VAR
VAR_INPUT
    options         : Reference To ST_EndAtTrackAdvancedOptions;
END_VAR
```

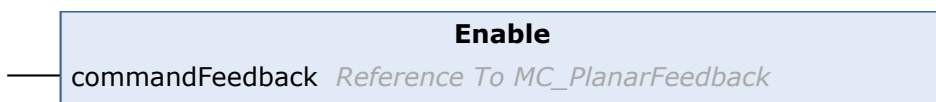
 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.
options	Reference To ST_EndAtTrackAdvancedOptions [▶_127]	Options for the connection, i.e. which connections are closed.

 In/Outputs

Name	Type	Description
track	MC_PlanarTrack [▶_161]	The other Planar Track.

7.2.2.10.10 Enable



Starts enabling the Planar Track.

Syntax

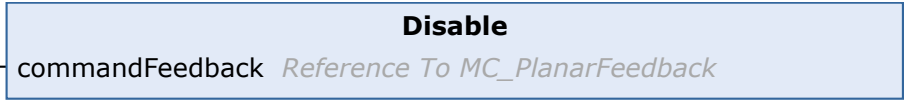
Definition:

```
METHOD Enable
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

7.2.2.10.11 Disable



Starts disabling the Planar Track.

Syntax

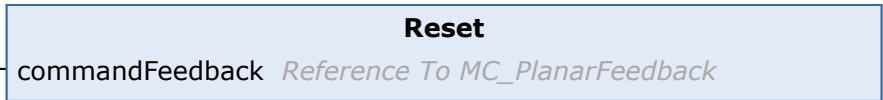
Definition:

```
METHOD Disable
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

7.2.2.10.12 Reset



Starts resetting the Planar Track.

Syntax

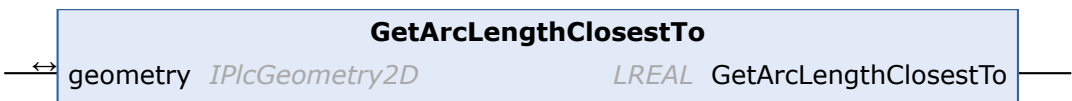
Definition:

```
METHOD Reset
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
```

 **Inputs**

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

7.2.2.10.13 GetArcLengthClosestTo




Calculate the arc length value [mm] where the Planar Track is closest to a geometry's center point.

Syntax

Definition:

```
METHOD GetArcLengthClosestTo : LREAL
VAR_IN_OUT
    geometry : IPlcGeometry2D;
END_VAR
```

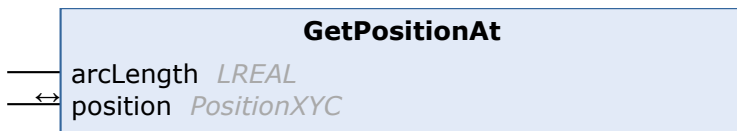
 **In/Outputs**

Name	Type	Description
geometry	IPlcGeometry2D	The geometry to check the arc length for.

 **Return value**

LREAL

7.2.2.10.14 GetPositionAt



Get a position on the Planar Track at a specific arc length value.


Syntax

Definition:

```
METHOD GetPositionAt
VAR_INPUT
    arcLength : LREAL;
END_VAR
VAR_IN_OUT
    position : PositionXYC;
END_VAR
```

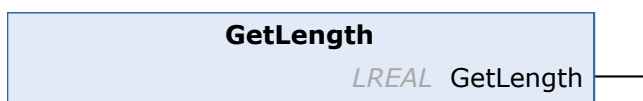
 **Inputs**

Name	Type	Description
arcLength	LREAL	Arc length value where the position is evaluated.

 **In/Outputs**

Name	Type	Description
position	PositionXYC	The position at the specified arc parameter.

7.2.2.10.15 GetLength




Returns the Planar Track's length [mm], -1 return value indicates no connection to Nc Track.

Syntax

Definition:

METHOD GetLength : LREAL

 **Return value**

LREAL

7.2.2.10.16 GetPlanarObjectInfo**GetPlanarObjectInfo***PlanarObjectInfo* GetPlanarObjectInfo

Returns track object info (type: track, id: OID of nc track).

Syntax

Definition:

METHOD GetPlanarObjectInfo : PlanarObjectInfo

 **Return value**PlanarObjectInfo [[▶ 126](#)]**7.2.2.10.17 Update****Update**

Updates internal state of the object, must be triggered each cycle.

Syntax

Definition:

METHOD Update

7.2.2.10.18 AddToGroup**AddToGroup** **commandFeedback** *Reference To MC_PlanarFeedback* **group** *MC_PlanarGroup*

Adds the Planar Track to the given Planar Group.

Syntax

Definition:


```

METHOD AddToGroup
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
VAR_IN_OUT
    group           : MC_PlanarGroup;
END_VAR

```

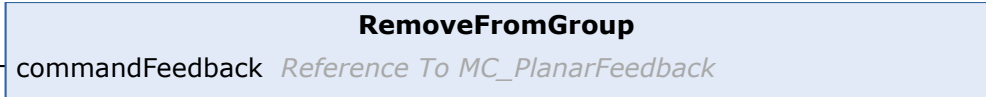
 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

 In/Outputs

Name	Type	Description
group	MC_PlanarGroup [▶_141]	The Planar Group that the mover joins.

7.2.2.10.19 RemoveFromGroup



Removes the Planar Track from its current Planar Group, i.e. disables collision checks.

Syntax

Definition:

```

METHOD RemoveFromGroup
VAR_INPUT
    commandFeedback : Reference To MC_PlanarFeedback;
END_VAR
    
```

 Inputs

Name	Type	Description
commandFeedback	Reference To MC_PlanarFeedback [▶_136]	The feedback object for the command.

7.2.2.11 MC_PlanarTrackTrail

A list of distinct tracks each starting at the ending vertex of its predecessor.

Do not call the main FB directly. Only use the available methods.

 Methods

Name	Description
Clear [▶_172]	Clears the TrackTrail.
AddTrack [▶_172]	Adds a track to the TrackTrail. The track should start at the end vertex of the currently last track.
GetTrackOidAt [▶_172]	Returns the oid of the track at the specified index [1, ..., size] in the trail. Zero is returned on error.

Required License

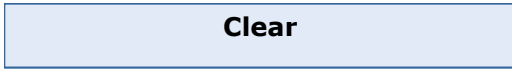
TC3 Planar Motion Base

System Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT V3.1.4024.12	PC or CX (x64)	Tc3_Mc3PlanarMotion, Tc3_Physics

Development environment	Target system type	PLC libraries to include
Advanced Motion Pack V3.1.10.11		

7.2.2.11.1 Clear



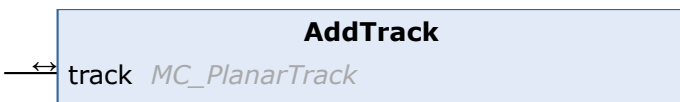
Clears the TrackTrail.

Syntax

Definition:

```
METHOD Clear
```

7.2.2.11.2 AddTrack



Adds a track to the TrackTrail. The track should start at the end vertex of the currently last track.

Syntax

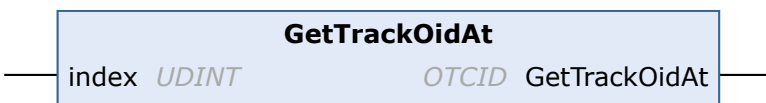
Definition:

```
METHOD AddTrack
VAR_IN_OUT
    track : MC_PlanarTrack;
END_VAR
```

In/Outputs

Name	Type	Description
track	MC_PlanarTrack [▶ _161]	The track to be added to the end of the TrackTrail.

7.2.2.11.3 GetTrackOidAt



Returns the oid of the track at the specified index [1, ..., size] in the trail. Zero is returned on error.


Syntax

Definition:

```
METHOD GetTrackOidAt : OTCID
VAR_INPUT
    index : UDINT;
END_VAR
```

Inputs

Name	Type	Description
index	UDINT	Index of the track in the trail [1, ..., size].

 Return value

OTCID

8 技术支持和服务

倍福公司及其合作伙伴在世界各地提供全面的技术支持和服务，对与倍福产品和系统解决方案相关的所有问题提供快速有效的帮助。

下载搜索器

我们的下载搜索器包含我们供您下载的所有文件。您可以通过它搜索我们的应用案例、技术文档、技术图纸、配置文件等等。

可供下载的文件格式多种多样。

倍福分公司和代表处

若需要倍福产品的本地支持和服务，请联系倍福分公司或代表处！

倍福遍布世界各地的分公司和代表处地址可在倍福官网上找到：<http://www.beckhoff.com.cn>

该网页还提供更多倍福产品组件的文档。

倍福技术支持

技术支持部门为您提供全面的技术援助，不仅帮助您应用各种倍福产品，还提供其他广泛的服务：

- 技术支持
- 复杂自动化系统的设计、编程和调试
- 以及倍福系统组件的各种培训课程

热线电话： +49 5246 963-157
电子邮箱： support@beckhoff.com

倍福售后服务

倍福服务中心提供所有售后服务：

- 现场服务
- 维修服务
- 备件服务
- 热线服务

热线电话： +49 5246 963-460
电子邮箱： service@beckhoff.com

倍福公司总部

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

电话： +49 5246 963-0
电子邮箱： info@beckhoff.com
网址： www.beckhoff.com

Trademark statements

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.

更多信息:

www.beckhoff.com/TF5430

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
电话号码: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

