

在使用AC500 PLC作为EtherCAT®主机时，可以通过PS552-MC-E运动库提供的易于使用的功能块捕捉远程EtherCAT驱动器的位置。



引言

AC500 PLC（PM59x）可用于通过EtherCAT对ABB运动交流伺服驱动器进行实时运动控制。本应用说明接续了AN00205（AC500—EtherCAT入门指南）的话题，并详细介绍了如何使用Automation Builder定义适合于捕捉（锁存）轴位置的硬件和软件设置。位置捕捉的典型用途包括需要定位的应用程序（例如打印、贴标、分度输送机）以及精确寻零（请参阅AN00220以进一步了解位置捕获和寻零的使用方法）。

前提条件

您需要具备以下条件才能完成本应用说明：

- 5860或更高版本的Mint Workbench（参见www.abbmotion.com 以获取最新的下载和支持信息）
- MicroFlex e190或MotiFlex e180驱动器，内置5868.7或更高版本的固件，连接有数字输入1和2（即快速位置捕捉输入）— 请注意，本应用说明中的文字假定使用的是MicroFlex e190，但使用MotiFlex e180的过程相同
- 运行Automation Builder v2.1.1或更高版本的PC或笔记本电脑
- 已安装经许可的最新版本的ABB PLCopen运动控制库（3.2.0或更高版本的PS552-MC-E）的副本
- 带有CM579-ECAT通信模块的AC500 PM59x ETH PLC（CM579-ECAT模块必须运行2.6.9或更高版本，最好是4.3.0.2或更高的版本的固件— 有关如何检查该模块的详细信息，请与您当地的 ABB PLC支持团队联系，必要时进行更新）
- 将CM579-ECAT模块连接至EtherCAT伺服驱动器的以太网电缆
- 了解通过以太网进行的AC500 PLC和MicroFlex e190/MotiFlex e180驱动器的基本操作 - 如有必要，请参阅应用说明AN00205了解更多详细信息

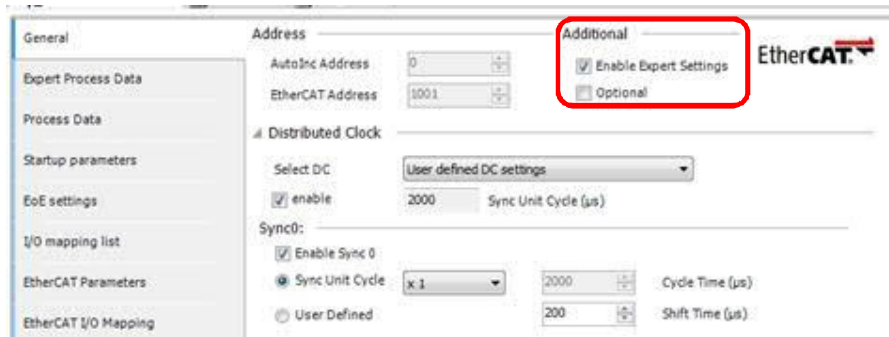
驱动器的设置和ESI文件

本应用说明假设您已经调试了ABB运动驱动器，并且它已加载了适当的固件（5868.7或更高版本）。也即是说，您已经通过调试向导定义了电机和应用设置，并为驱动器自动调节了（如有必要，进行微调）控制回路。可在相关的驱动器安装手册中找到关于驱动器调试的详细信息。本文还假设您至少已阅读并理解应用说明AN00205的内容。本应用说明的内容假设来自AN00205的项目将被用作新PLC代码的起点。然而，为方便起见，本应用说明的下载文件中包含一个已完成的项目档案，以便您将已完成的项目档案与相应的EtherCAT ESI/XML文件一起安装到您的Automation Builder设备存储库中。本应用说明中提供的示例项目包含一个可视化功能，允许更轻松地测试示例功能。

添加用于位置捕捉的过程数据对象

如果希望按照下面的内容添加所需的過程数据对象（PDO），以访问远程驱动器上的快速位置捕捉数据，请打开来自AN00220的项目。或者，也可以使用我们预先准备好的项目，我们已经完成相关的代码等。用新的名称保存项目，然后双击驱动器图标并选择“从设备”选项卡。

快速位置捕捉数据通过驱动器对象（被称为“Touchprobe”对象）来访问。要使用touchprobe数据，我们需要添加一些额外的过程数据对象。因此，如下文所示，我们必须首先选中“启用专家设置”复选框。



在选择此选项后，Automation Builder会在右窗格中提供一个名为“专家过程数据”的附加选项卡 — 选择此选项卡。当您在右侧窗格的输出和输入之间进行选择时，Automation Builder会显示PLC和驱动器之间的现有PDO映射设置。

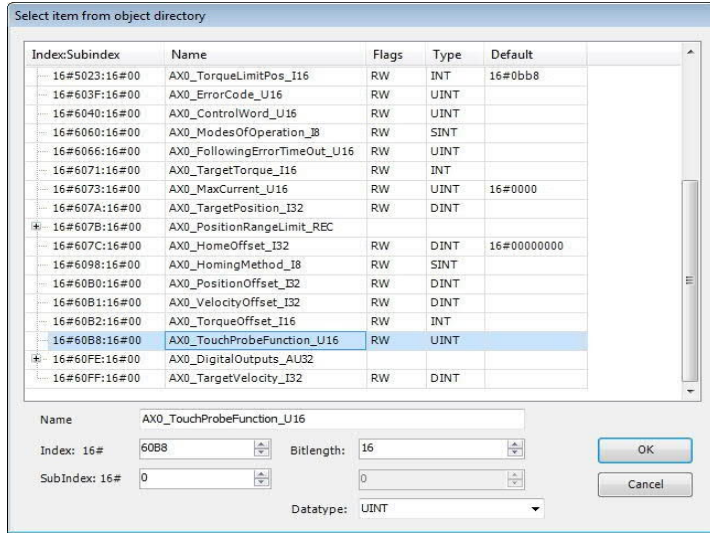
Index	Size	Name	Flags	SM
16#1600	8.0	Outputs	M	2
16#1A00	24.0	Inputs	M	3

Index	Size	Offs	Name	Type
16#6040:00	2.0	0.0	AX0_ControlWord_U16	UINT
16#607A:00	4.0	2.0	AX0_TargetPosition_I32	DINT
	6.0			

Index	Size	Name	Flags	SM
16#1600	8.0	Outputs	M	2
16#1A00	24.0	Inputs	M	3

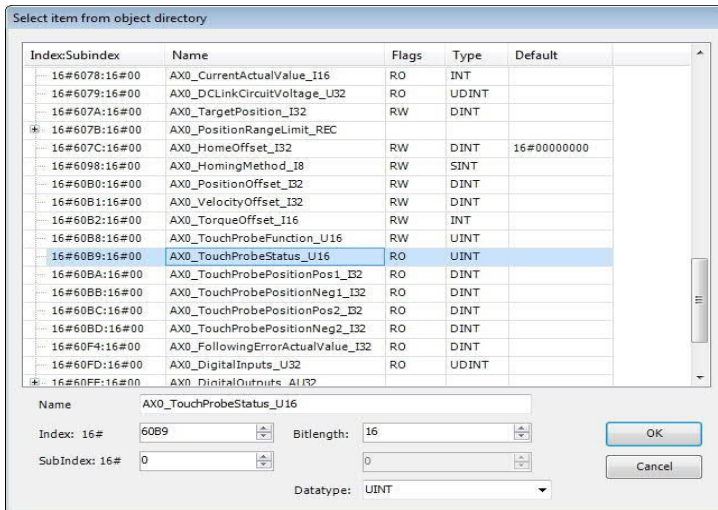
Index	Size	Offs	Name	Type
16#6041:00	2.0	0.0	AX0_StatusWord_U16	UINT
16#6064:00	4.0	2.0	AX0_ActualPosition_I32	DINT
	6.0			

注意，这些是从PLC的角度进行的操作（即，输出是PLC发送到驱动器的值，输入是PLC从驱动器接收到的值）。突出显示输出，并在右下窗格的“关闭”列中选择“6.0”，右键单击并选择“插入...”。滚动可用对象列表，并选择DS402_TouchProbeFunction_U165对象（索引60B8）。



单击OK把这个对象添加到输出PDO映射列表中。这个对象允许AC500 PLC配置驱动器的操作，使快速输入（1和2）或来自电机编码器的Z脉冲可以用于捕捉（锁存）轴位置。

现在在右上窗格中选择“输入”，然后在“关闭”列中再次选择“6.0”，右键单击并选择“插入...”。这一次选择DS402_TouchProbeStatus_U16对象（索引60B9），并单击OK将其添加到输入PDO映射中。这个对象允许PLC检测驱动器上快速锁存的状态。



现在重复这个过程，以添加另一个输入PDO映射。但是，这一次我们需要选择“TouchProbePosition”对象。您将注意到，列表中有四个可能的对象...

- DS402_TouchProbePositionPos1_I32（索引60BA）
- DS402_TouchProbePositionNeg1_I32（索引60BB）
- DS402_TouchProbePositionPos2_I32（索引60BC）
- DS402_TouchProbePositionNeg2_I32（索引60BD）

与POS1和NEG1相关的对象与驱动器上的数字输入1（或编码器Z脉冲）捕获的快速锁存值直接相关。

与POS2和NEG2相关的对象与驱动器上的数字输入2（或编码器Z脉冲）捕获的快速锁存值直接相关。

(POS与上升沿锁存数据有关，NEG与下降沿锁存数据有关)。

我们通常只选择应用程序所需的对象，以保持最小数量的映射数据（例如，对于定位应用程序，我们可能只想使用连接到驱动器上数字输入1的定位传感器的上升沿，因此我们选择DS402_TouchProbePositionPos1_I32）。但在当前的例子中，我们将添加所有对象，因此重复“插入...”过程，直到所有四个对象都包含在内。在添加完这些对象后，输入PDO映射应该与下图所示相似（顺序并不重要）。

PDO Content (16#1A00):				
Index	Size	Offs	Name	Type
16#6041:00	2.0	0.0	AX0_StatusWord_U16	UINT
16#6064:00	4.0	2.0	AX0_ActualPosition_I32	DINT
16#60B9:00	2.0	6.0	AX0_TouchProbeStatus_U16	UINT
16#60BA:00	4.0	8.0	AX0_TouchProbePositionPos1_I32	DINT
16#60BB:00	4.0	12.0	AX0_TouchProbePositionNeg1_I32	DINT
16#60BC:00	4.0	16.0	AX0_TouchProbePositionPos2_I32	DINT
16#60BD:00	4.0	20.0	AX0_TouchProbePositionNeg2_I32	DINT
		24.0		

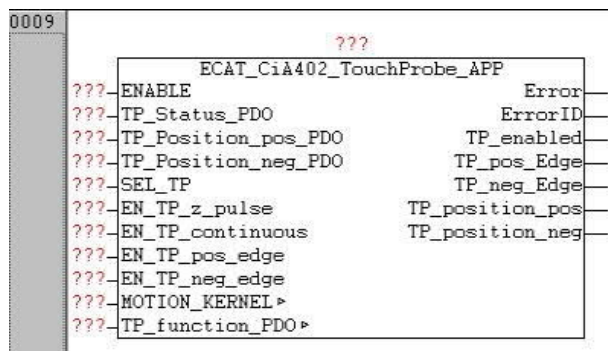
现在选择Automation Builder中的EtherCAT I/O映射选项卡。我们需要为我们添加的新PDO映射分配变量名。

下图显示了分配给新touchprobe对象的变量。

Object Name	Variable	Channel	Address	Type	Description
MicroFlex_e150	wAxisOTPFfunction	AX0_TouchProbeFunction_U16	%QW1.4	UINT	AX0_TouchProbeFunction_U16
MicroFlex_e150	wAxisOTPStatus	AX0_TouchProbeStatus_U16	%IW1.4	UINT	AX0_TouchProbeStatus_U16
MicroFlex_e150	diAxisOTPPos1	AX0_TouchProbePositionPos1_I32	%ID1.3	DINT	AX0_TouchProbePositionPos1_I32
MicroFlex_e150	diAxisOTPNeg1	AX0_TouchProbePositionNeg1_I32	%ID1.4	DINT	AX0_TouchProbePositionNeg1_I32
MicroFlex_e150	diAxisOTPPos2	AX0_TouchProbePositionPos2_I32	%ID1.5	DINT	AX0_TouchProbePositionPos2_I32
MicroFlex_e150	diAxisOTPNeg2	AX0_TouchProbePositionNeg2_I32	%ID1.6	DINT	AX0_TouchProbePositionNeg2_I32
MicroFlex_e150	wAxisQControlWord	AX0_ControlWord_U16	%QW1.0	UINT	AX0_ControlWord_U16
MicroFlex_e150	diAxisQTargetPos	AX0_TargetPosition_I32	%QD1.1	DINT	AX0_TargetPosition_I32
MicroFlex_e150	wAxisQStatusWord	AX0_StatusWord_U16	%IW1.0	UINT	AX0_StatusWord_U16
MicroFlex_e150	diAxisQActualPos	AX0_ActualPosition_I32	%ID1.1	DINT	AX0_ActualPosition_I32

再次保存项目，然后双击Automation Builder中的程序图标启动Codesys。单击“更新”接受对硬件配置的更改。

选择程序中的最后一个梯级，右键单击并选择“网络（后）”以将新梯级添加到程序中。单击“新建框”工具栏按钮，输入新功能块的名称ECAT_CiA402_TouchProbe_APP（或通过键盘上的F2，使用输入助手查找并选择此功能块）。



此功能块允许PLC配置驱动器上的快速中断（touchprobe）操作。功能块输入具有以下功能：

启用	将此设置为TRUE，以启用远程驱动器上的touchprobe（锁存）。
TP_Status_PDO	它必须被链接到分配给touchprobe状态对象的变量（在我们的例子中是wAxis0TPStatus）。
TP_Position_pos_PDO	它必须被链接到由SEL_TP输入选择的touchprobe的上升沿锁存变量（例如，如果SEL_TP设置为1，则为diAxis0TPPos1）。
TP_Position_neg_PDO	它必须被链接到由SEL_TP输入选择的touchprobe的下降沿锁存变量（例如，如果SEL_TP设置为1，则为diAxis0TPNeg1）。
SEL_TP	选择功能块是控制驱动器上的touchprobe 1还是touchprobe 2
EN_TP_z_pulse	选择是否由编码器的z脉冲捕捉锁存。如果设置为FALSE，驱动器将根据为相关数字输入定义的设置锁存位置（例如，Touchprobe 1将锁存来自数字输入1的信号）
EN_TP_continuous	选择每次发生快速输入（或z脉冲，取决于配置）时，驱动器是否应继续锁存位置。如果设置为FALSE，驱动器将锁存一个值，并且在禁用再重新启用touchprobe之前不会再次锁存
EN_TP_pos_edge	选择是否使用相关快速输入（或z脉冲，取决于配置）的上升沿来存储TP_position_pos功能块输出中的值
EN_TP_neg_edge	选择是否使用相关快速输入（或z脉冲，取决于配置）的下降沿来存储TP_position_neg功能块输出中的值
MOTION_KERNEL	它必须与驱动器所使用的CMC_MOTION_KERNEL_REAL功能块（例如kerAxis0）的实例相关联。
TP_Function_PDO	它必须链接到分配给touchprobe函数PDO的变量（例如wAxis0TPFunction）。

Touchprobe功能块的输出指示发生有效锁存（例如TP_pos_Edge）的时间，以及快速锁存位置的值（例如TP_position_pos）。然后，将根据PLC应用代码的要求使用这些输出（例如，将捕捉的位置与预期值进行比较，以计算定位校正值）。

重要提示

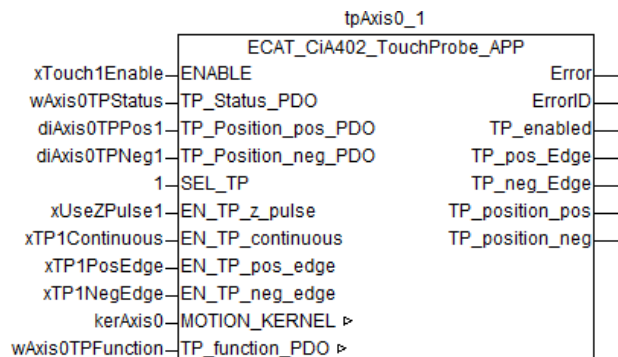
Touchprobe功能块必须包含在与EtherCAT数据循环同步的程序中。我们的PLC项目示例仅使用一个程序对象，它由与EtherCAT循环相关联的任务调用，因此满足这一要求。在大多数应用程序中，主应用程序逻辑可能被编码在单独的循环PLC任务中（运动内核、参数和DS402功能块位于与EtherCAT相关的任务/程序中）。因此，记住将touchprobe功能块包含在与内核功能块相同的程序元素中非常重要。

当touchprobe功能块在“连续”模式下运行时，指示接收新值的输出“TP_pos_edge”和“TP_neg_edge”仅在一个EtherCAT周期内设置为真。因此，如果功能块不在EtherCAT的相关处理范围内，则这些状态将丢失。

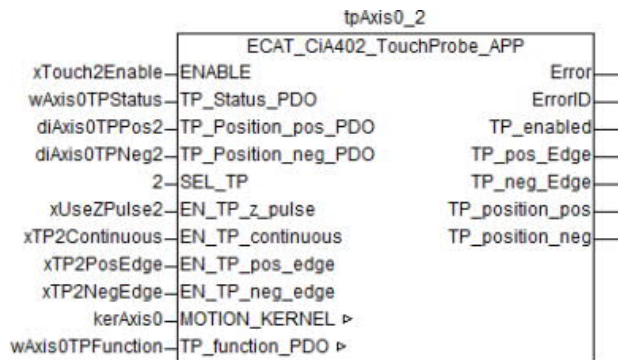
同时要注意，如果调整了轴的位置（例如，通过MC_SetPosition），锁存值本身的“TP_position_pos”和“TP_position_neg”将被修改。因此，如果应用程序需要实际锁存/存储这些值，则应该添加代码以将这些值转移到其他程序变量中（或数据中（如果需要存储数据队列））。

在本应用说明中，我们将为所有配置输入分配变量，然后我们可以通过CoDeSys强制执行这些输入，以测试各种锁存选项（本应用说明包含的示例项目通过可视化来帮助执行测试）。

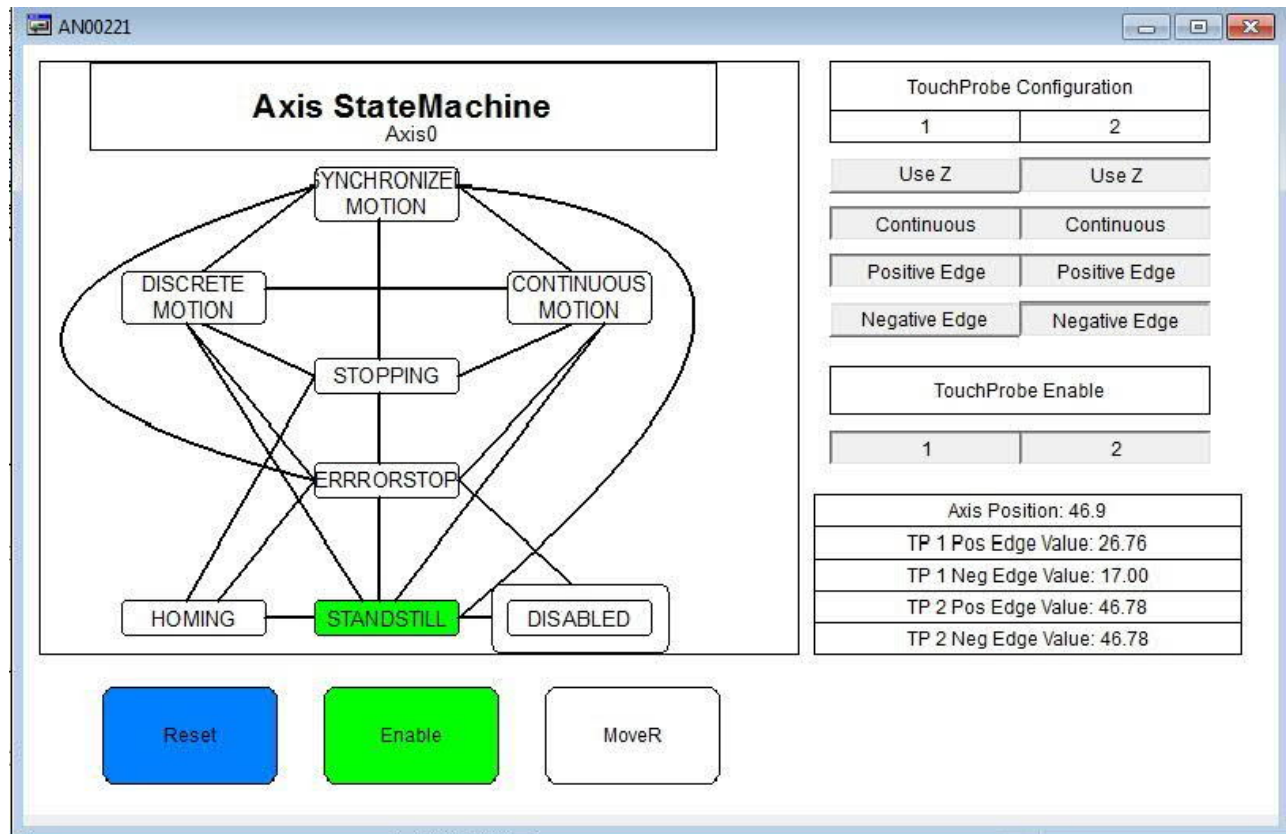
下图显示了我们针对名为tpAxis0_1的块的Touchprobe配置功能块所做的最终设置。



现在右键单击此梯级添加新网络（后）。单击“新建框”按钮，为新块再次输入 ECAT_CiA402_TouchProbe_APP。输入tpAxis0_2的参数/变量，如下所示：



这是测试驱动器的touchprobe操作所需的全部代码。登录到PLC并下载此代码进行测试。使用CoDeSys强制执行各种配置和启用参数，并查看两个touchprobe功能块的输出。或者，也可以使用本应用说明中包含的项目的可视化工具，可以通过它测试各种touchprobe模式-记住，要测试与快速输入相关的模式，必须将这些数字输入连接到ABB运动驱动器上。



也可以使用touchprobe对象来执行非常精确的寻零 - 有关更多详细信息，请参阅应用说明AN00220。

联系我们

要了解更多信息，请联系您当地的ABB代表，或使用以下一种方式：

new.abb.com/motion
new.abb.com/drives
new.abb.com/drives/drivespartners
new.abb.com/PLC

EtherCAT®是由德国倍福自动化有限公司许可的注册商标和专利技术。

© ABB公司，2014年，版权所有。保留所有权利。技术规格如有变更，恕不另行通知。