

通过易于使用的高速锁存功能捕获位置、编码器或步进器值。



概述

整个 e100、e180 和 e190 系列的产品均有能力运行 Mint 应用程序，有时可能需要（例如：为了注册申请）捕获（锁存）受控轴的位置。本应用说明文档描述如何捕获 Mint 中的位置（或编码器或步进器）的值以及如何通过专用 Mint 中断/事件访问这些数值。

如果读者尚不熟悉 NextMove e100，建议先阅读应用说明文档 AN00162（转向 NextMove e100）与 AN00187（NextMove e100 入门指南）。

NextMove e100 控制器

NextMove e100 是一种可编程运动控制器，能够在本地控制 3 个伺服轴与 4 个步进轴，除此之外，还可通过 EPL 最多控制 32 个定位驱动轴。NextMove e100 可设置 8、12 或 16 个轴（取决于零件编号）来进行插补运动。其它轴可添加为“受控节点”（即这些轴自动设置点对点运动）或“仅监控”节点（例如：这些轴运行自己的 Mint 程序）。

NextMove e100 可使用两类捕获/锁存。

- 本地轴数据锁存
- 远程 e100 轴数据锁存

本说明书首先讲述本地轴数据锁存。本地轴通过 Mint Workbench 中的系统设置向导设置。最多可添加 3 个本地（模拟）伺服轴（每一个分别使用一条板载编码器通道）与 4 个本地步进轴（每一个分别使用一条板载步进与方向通道）。

在系统设置向导中，用户可向本地轴分配轴编号，之后会使用这些轴编号指代轴的位置。本地轴使用的编码器或步进/方向通道编号用于在 Mint 程序中指代编码器（如果是伺服轴）或步进器（如果是步进轴）。

为本地轴设置快速位置锁存时，必须在 Mint 应用程序中添加代码（无法进行图形化设置）。该代码一般包含在 Mint 启动块中，因为设置几乎不会在运行过程中发生变化，但就算需要将其包含在主应用程序码中，也不会造成任何问题。NextMove e100 有四个快速位置捕获输入（0..3），因此其中之一一般用于捕获快速位置/编码器/步进器值。

本地伺服轴锁存设置示例（使用锁存通道 0）：

```
LATCHSOURCE (0)      = _IsAXIS_POSITION
LATCHSOURCECHANNEL (0) = 2
LATCHTRIGGERMODE (0) = _ltmINPUT
LATCHTRIGGERCHANNEL (0) = 1
LATCHTRIGGEREDGE (0) = _ltePOSITIVE_EDGE
LATCHMODE (0)       = _lmaUTO_ENABLE
```

以上代码设置数字输入（因为 LATCHTRIGGERCHANNEL 的设置值是 1）来捕获轴 2 的位置（因为 LATCHSOURCECHANNEL 的设置值是 2）。锁存在数字输入出现上升（正）沿时发生。LATCHMODE 的设置使锁存机制在锁存某些数据后自动重新启动.....但必须先要在程序中启用锁存（一般在刚好需要时启用锁存）.....

```
LATCHENABLE (0) = 1
```

LATCHENABLE 被设为 1 时，锁存通道启用。以上括号中的 0 指示的是锁存通道的编号。NextMove e100 上最多可能有 32 条锁存通道（0...31）。使用的通道也与捕获到锁存值时调用的 Mint 事件/中断有关，因此，该示例使用.....

Event LATCH0

...作为中断程序来处理收到的新锁存值。

例如：

Event LATCH0

```
?#2 "Latched position is ", LATCHVALUE (0)
```

```
End Event
```

通过 USB 连接 Workbench 的，捕获到的轴位置会被打印在 Workbench 终端窗口中。LATCHVALUE 关键词存储捕获到的数据（可能是位置、编码器或步进器值，取决于锁存通道的设置）。请注意，使用的通道与事件声明中使用的通道一致。

编码器输入 0（可用于捕获位置或编码器）与步进器通道 0 有“特殊功能”。这些数据**最多可被捕获四次**（例如：可使用输入 0 通过某条锁存通道捕获编码器 0，使用输入 1 通过另一条锁存通道另行捕获编码器 0）。所有其它位置/编码器/步进器值仅可通过一条锁存通道捕获。

除能够利用输入 0 至 3 捕获快速数据外，还可利用数字输出 0 至 3 捕获快速位置等。这通过 LATCHTRIGGERMODE 关键词（using _ltnOUTPUT instead of _ltnINPUT）设置。

详见 Mint 帮助文件“锁存”。

MicroFlex e190 / MotiFlex e180

现在，在远程 Ethernet Powerlink（EPL）驱动运行锁存。这与本地锁存非常类似，但锁存必须在驱动上设置（即驱动锁存快速位置/编码器数据，然后通过 Ethernet Powerlink 网络将数据传给 NextMove e100）。











驱动有四条锁存通道（0 至 3），其中最多有两条可被映射回 NextMove e100。e180 与 e190 驱动有两个快速数字输入（1 和 2），还能够利用数字输出 0 或 1，甚至主通用编码器输入的 Z 脉冲锁存数据。

可使用以下任何方式在驱动上设置锁存通道：

- 使用驱动上保存的 Mint 程序（内容与之前 NextMove e100 的 Mint 程序的内容类似）
- 利用驱动的参数表保存锁存设置
- 利用“重定向”技术，通过 NextMove e100 上保存的 Mint 程序设置驱动的锁存

驱动在独立应用程序中使用（即不存在 NextMove e100）或驱动作为仅监控节点被添加到 EPL 网络且运行自己的 Mint 应用程序时，使用驱动上的 Mint 程序。

如果您倾向于“可视化”设置方法，那么利用驱动的参数表保存锁存设置较为方便。

Parameter	Active
LatchEnable (LatchChannel 0)	 1
LatchInhibitTime (LatchChannel 0)	 0 ms
LatchInhibitValue (LatchChannel 0)	 0.0000
LatchMode (LatchChannel 0)	 0x0001
LatchSource (LatchChannel 0)	 Axis position
LatchSourceChannel (LatchChannel 0)	 0
LatchTriggerChannel (LatchChannel 0)	 1
LatchTriggerEdge (LatchChannel 0)	 Positive edge
LatchTriggerMode (LatchChannel 0)	 Digital input
LatchValue (LatchChannel 0)	 0.0000

但是，最灵活的驱动锁存设置方法是利用 NextMove 的 Mint 程序中的重定向技术，这使用户能够“查看”锁存设置（设置不会被隐藏在驱动的参数表中），还使应用程序能够在必要时更改设置（例如：应用程序可能需要切换在上升沿时捕获快速数据或在上下降时捕获快速数据）。

为使用 NextMove 的 Mint 程序中的重定向技术，必须先声明将用于访问远程驱动上的参数/命令的“控制器”变量。以下示例假设要设置的是作为 EPL 网络上节点 3 的 e190 驱动上的锁存。

有两种方法声明控制器变量/将控制器变量初始化：

```
Dim cAxis3 As Controller = {_busETHERNET, 3}
```

或

```
Dim cAxis3 As Controller
cAxis3.nBus = _busETHERNET
cAxis3.nNode = 3
```

声明控制器变量后，可使用控制器访问远程驱动。用于设置锁存的命令与上述用于设置 NextMove e100 上的本地锁存的命令完全相同。对于本地锁存，我们有效地在本地驱动发送命令，但对于远程锁存，我们从 NextMove e100 远程发送命令...

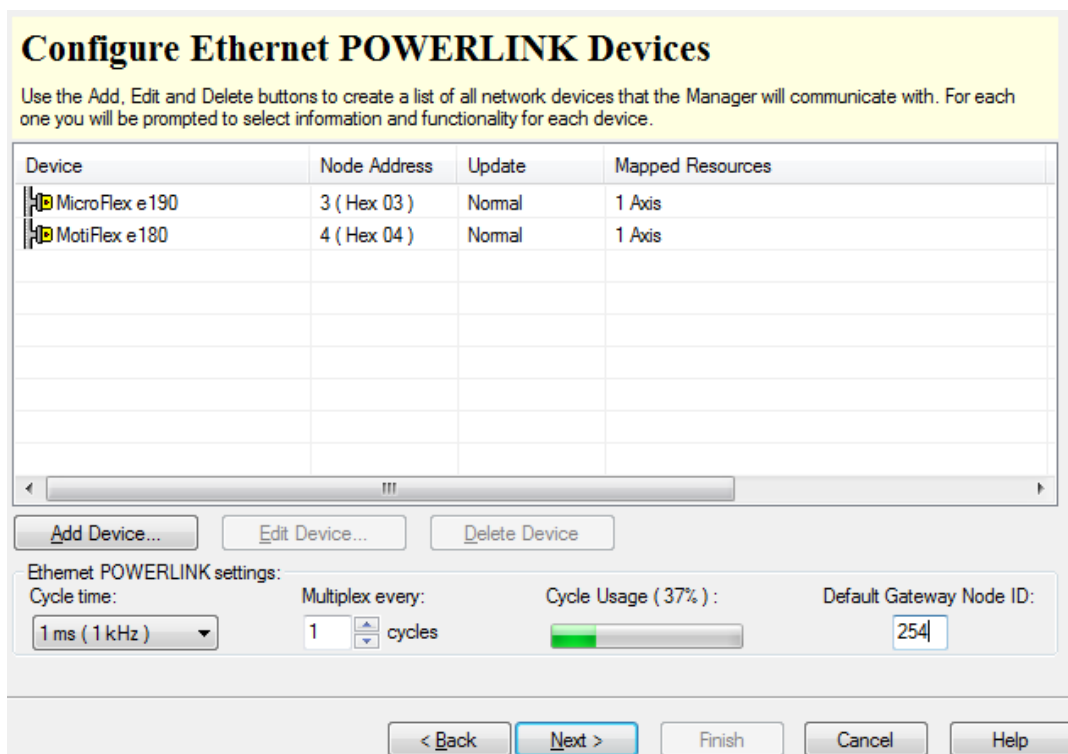
```
cAxis3 -> LATCHSOURCE (0)      = _IsAXIS_POSITION
cAxis3 -> LATCHSOURCECHANNEL (0) = 0
cAxis3 -> LATCHTRIGGERMODE (0)  = _ltmINPUT
cAxis3 -> LATCHTRIGGERCHANNEL (0) = 1
cAxis3 -> LATCHTRIGGEREDGE (0)  = _ltePOSITIVE_EDGE
cAxis3 -> LATCHMODE (0)        = _lmAUTO_ENABLE
```

我们利用变量名称 (cAxis3) 加重定向符号 (->) 的方式指示我们访问的是远程驱动上的命令/参数。以上示例设置驱动上的锁存通道 0 来从驱动上的数字输入 1 捕获位置。

与本地锁存一样，必须先启用锁存，但在使用远程锁存时，LATCHENABLE 实际是启用接收驱动发送的锁存数据（以及远程启用锁存），因此是本地命令（即无需对 LATCHENABLE 命令使用重定向技术）。

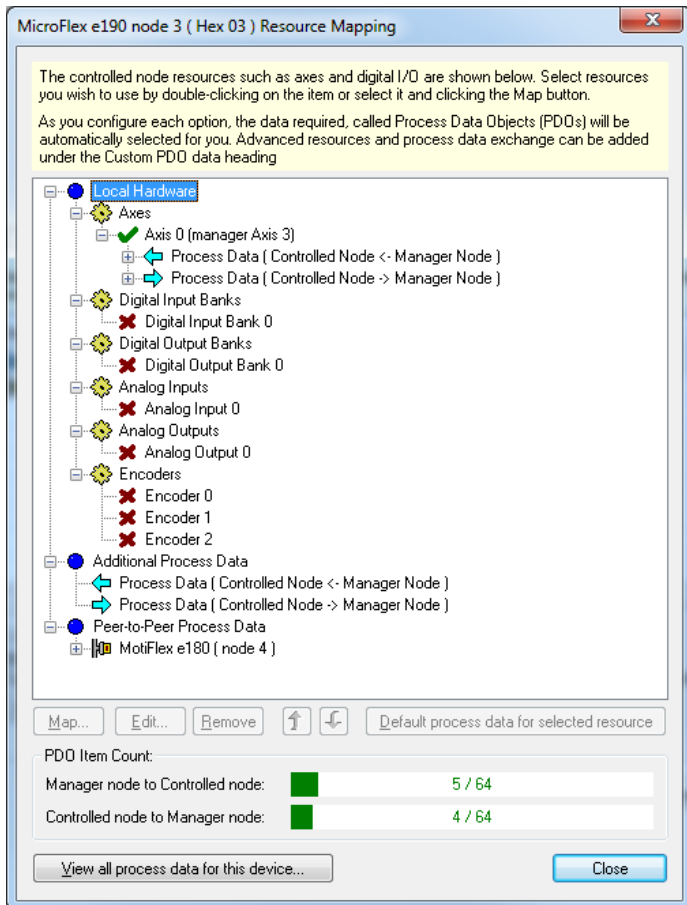
发送该命令之前，需要设置从驱动到 NextMove 的远程锁存数据传输/映射。这一过程会决定 NextMove 中用于接收数据的通道（即本示例已设置驱动上的锁存通道 0，但需将其映射回 NextMove e100 上 32 条锁存通道中的某条通道）。

为设置锁存映射，先连接 NextMove e100，然后启动系统设置向导。假设您已设置好一个或多个远程 e180 或 e190 轴（如必要，请参阅 AN00187，了解详细信息）。点击“上传控制器中的设置”（Upload configuration from controller），然后点击“下一步”（Next）查看已设置的 EPL 设备的列表。



高亮显示需要的驱动，然后点击“编辑设备...”（Edit Device...）按钮访问该驱动的设置（该示例选择节点 3）。

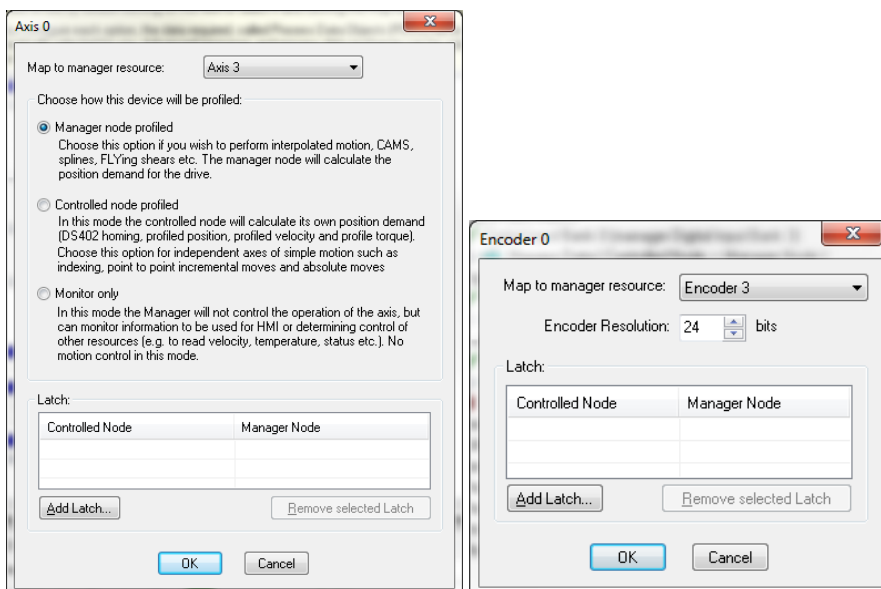
Workbench 将首先显示轴的资源映射...



现在，必须决定希望收到远程轴的锁存位置值还是锁存编码器值。

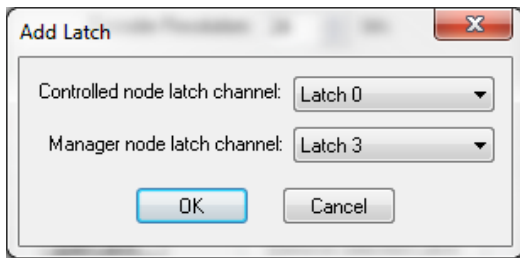
如果希望将锁存位置值发送回 NextMove e100，则双击对话框中的“轴 0（管理器轴...）”（Axis 0（manager axis ...））。

如果希望将编码器位置值发送回 NextMove e100，则双击对话框中的“编码器 0（管理器编码器...）”（Encoder 0（manager Encoder...））。



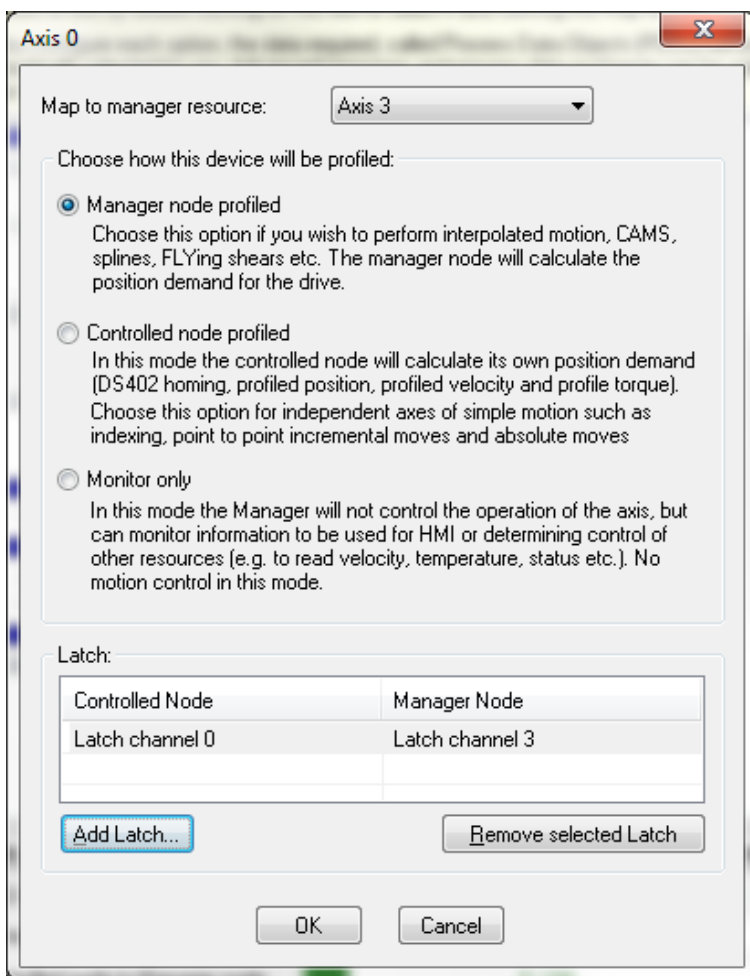
以上屏幕截图展示了后续显示的对话框（左侧是位置锁存映射，右侧是编码器锁存映射）。

两个对话框中均有“添加锁存...”（Add Latch...）按钮。点击该按钮...该示例希望接收 e190 驱动发送的位置数据，这也是之前设置的驱动锁存，因此双击“轴 0...”（Axis 0 ...），然后点击轴 0 对话框中的“添加锁存...”（Add Latch...）按钮。



小对话框显示，用于设置如何将驱动的锁存通道映射回 NextMove e100。以上示例设置了驱动上的锁存通道 0 来从保存快速位置（通过驱动上的数字输入 1 捕获）。该示例将驱动的锁存通道（0）映射回了 NextMove e100 的锁存通道 3（之所以选择 3 是因为该数字与驱动节点地址和轴编号一致）。

点击确定（OK）。相关对话框（轴或编码器，取决于映射对象）将更新，以显示新的映射...



点击确定（OK）接受更新后的映射，然后按系统设置向导完成其余步骤，直到最终可保存新设备设置文件（DCF）与将文件下载到 NextMove e100。

到此，本地映射设置完成，可通过 NextMove 的 Mint 程序中的相关锁存事件访问锁存的数据（位置或编码器）。以上示例将驱动锁存映射到 NextMove 的锁存通道 3，因此使用...

Event LATCH3

?#2 "Drive latched position is ", LATCHVALUE (3)

End Event

与之前相同，为实际启用数据的接收，使用 LATCHENABLE 关键词与相关锁存通道编号。

LATCHENABLE (3) = 1

可随时通过将该参数值设为零禁用锁存数据的接收。

LATCHENABLE (3) = 0

如前所示，每个驱动有两条可映射的通道，因此可映射锁存编码器与位置值的任意组合（即两个位置值、一个位置值与一个编码器值或两个编码器值）。

请参阅 Mint 帮助文件“锁存”以及其中以 LATCH 开头的关键词了解详细信息。

锁存时间（延时）

NextMove e100（本地锁存）：

数字输入 0 - 3: 1 μ s

数字输出 0 - 3: 16.67ns（尽管数字输入仅每 1ms 更新一次）

e180/e190 驱动（本地锁存）：

数字输入 1 / 2: 300 ns（增量编码器 0, 62.5 μ s（其它类型反馈）

数字输出 0 / 1: 300 ns（增量编码器 0, 62.5 μ s（其它类型反馈）

使用本地锁存时，用于调用 Mint 锁存事件（例如：Event LATCH0）的时间可以是 0 到 1 ms 之间的任意时间（或，使用远程锁存时，最长可以是定义的 EPL 周期时间），这不影响 NextMove e100 或远程 EPL 驱动保存的锁存数据的准确性，因为仅访问与利用该数据之前的时间（LATCHVALUE）。

联系我们

如需更多信息，请联系

您所在地的 ABB 代表或访问以下任一网站：

new.abb.com/motion

new.abb.com/drives

new.abb.com/drivespartners

new.abb.com/PLC

© 2017 年 ABB 版权所有。保留所有权利。
规格如有更改，恕不另行通知。