

Application note

Fast position capture

AN00223

Rev B (EN)

Capture position, encoder or stepper values via simple to use high speed latching features



Overview

The complete range of e100, e180 and e190 products are capable of running Mint applications and it may be necessary at times to capture (latch) the position of a controlled axis (e.g. for registration applications). This application note describes how it is possible to capture position (or even encoder or stepper) values in Mint and access this data via special purpose Mint interrupts/events.

If the reader is not already familiar with NextMove e100 it is recommended that they familiarise themselves with the contents of application notes AN00162 (Moving to NextMove e100) and AN00187 (Getting started with NextMove e100) beforehand.

NextMove e100 controller

The NextMove e100 is a programmable motion controller capable of controlling up to 3 axes of servo and 4 axes of stepper locally. In addition up to 32 positioning drive axes can be controlled over EPL. 8, 12 or 16 axes (depending on the part number) can be profiled by the NextMove e100 to perform interpolated moves. The remaining axes can be added as 'controlled nodes' (i.e. they profile their own point to point motion) or as 'monitor only' nodes (e.g. they run their own Mint programs).

There are two types of capture/latching that can be used on a NextMove e100

- Latching of local axis data
- Latching of remote e100 axis data

We will look at latching local axis data initially. Local axes are configured via the System Configuration wizard in Mint Workbench. It is possible to add up to 3 local (analogue) servo axes (that each use one of the on-board encoder channels) and 4 local stepper axes (that each use one of the on-board step and direction channels).

The System Configuration wizard allows the user to allocate axis numbers to these local axes and it is this axis number that is used when referring to the position of the axis. The encoder or step/direction channel number used by a local axis is the number that is used when referring to either encoder (for a servo axis) or stepper (for a stepper axis) in a Mint program.

To configure a fast position latch for a local axis we must add some code to our Mint application (there is no way to graphically configure this). This is typically included in the Mint startup block as the configuration very rarely changes at run-time, but there's no problem with including it in the main application code if needed. NextMove e100 has four fast position capture inputs (0 to 3) so one of these is typically used to capture the fast position/encoder/stepper value.

Example latch configuration for local servo axis (using latch channel 0):

```
LATCHSOURCE(0)    = _lsAXIS_POSITION
LATCHSOURCECHANNEL(0) = 2
LATCHTRIGGERMODE(0) = _ltmINPUT
LATCHTRIGGERCHANNEL(0) = 1
LATCHTRIGGEREDGE(0) = _ltePOSITIVE_EDGE
LATCHMODE(0)      = _lmAUTO_ENABLE
```

This code configures a digital input (in this case input 1 because of the setting for LATCHTRIGGERCHANNEL) to capture the POSITION of Axis 2 (it is Axis 2 because of the setting for LATCHSOURCECHANNEL). The latch occurs on the rising (positive) edge of the digital input. The setting for LATCHMODE in this case causes the latch mechanism to automatically re-enable after latching some data....but we must first enable the latch in our program somewhere (we wouldn't typically enable the latch until just before it is needed)....

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

Setting LATCHENABLE to 1 enables the latch channel. In the example above the zero in brackets indicates the latch channel number. There are 32 possible latch channels on a NextMove e100 (0 to 31). The channel used also relates to the Mint event/interrupt that will be called when a latch value is captured....so for our example above we would use....

Event LATCH0

...as the interrupt routine to process receipt of a new latch value.

Example:

Event LATCH0

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

End Event

This sample would print the captured position of our axis to the Workbench terminal window if Workbench is connected via USB. The LATCHVALUE keyword stores the captured data (which could be a position, encoder or stepper value depending on the configuration of the latch channel). Note that the channel used by LATCHVALUE matches the channel used in the event declaration.

Encoder input 0 (which can be used to capture position or encoder) and Stepper channel 0 have "special functionality". It is possible to capture these quantities **up to four times** (e.g. Input 0 could be used to capture Encoder 0 via one latch channel and Input 1 could also be used to capture Encoder 0 separately via a second latch channel). All other Position/Encoder/Stepper values can only be captured via a single latch channel.

As well as being able to use inputs 0 to 3 to capture fast data it is also possible to use digital outputs 0 to 3 to capture fast position etc... This is configured via the LATCHTRIGGERMODE keyword (using _ltmOUTPUT instead of _ltmINPUT).

Refer to the Mint help file topic 'Latching' for further details.

MicroFlex e190 / MotiFlex e180











Now we will examine operation of latches on remote Ethernet Powerlink (EPL) drives. This is very similar to local latching, but now the latch configuration must be made on the drive itself (i.e. fast position/encoder data is latched by the drive itself and then passed via the Ethernet Powerlink network to the NextMove e100).

These drives have four latch channels (0 to 3) of which only two at most can be mapped back to the NextMove e100. The e180 and e190 drives also feature two fast digital inputs (1 and 2) as well as being able to also latch data using digital outputs 0 or 1, or even the Z pulse of the main universal encoder input.

To configure the latch channels on these drives we can either:

- Use a Mint program (with content similar to that shown earlier for the NextMove e100) stored on the drive
- Use the drive's parameter table to store the latch configuration
- Use a technique called "redirection" to setup the drive's latch configuration from the Mint program stored on the NextMove e100

We would use a Mint program on the drive itself when the drive is being used in a standalone application (i.e. there is no NextMove e100) or if the drive is added to an EPL network as a monitor only node and running its own Mint application. Using the drive's parameter table to store the latch configuration is a convenient way to setup the latches if you're more comfortable with a "visual" setup method....

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

However, the most flexible way to configure the drive latches is via redirection in the NextMove's Mint program. This allows the latch configuration to be "viewed" by the user (rather than hidden in the drive's parameter file) as well as allowing the application code to change the configuration if necessary (e.g. maybe the application needs to switch between rising edge and falling edge of the digital input being used to capture the fast data).

To use redirection in the NextMove's Mint program we must first declare a variable of type 'Controller' that will be used to access parameters/commands on our remote drive. For this example we will assume we are configuring the latches on an e190 drive configured as node 3 on our EPL network....

There are two ways to declare/initialise the controller variable:

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

Or

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

Once the controller variable is declared we can use it to access the remote drive. The commands to configure the latch are exactly the same as we used earlier to configure a local latch on the NextMove e100, we are effectively issuing them locally on the drive, but doing this remotely from the NextMove e100....

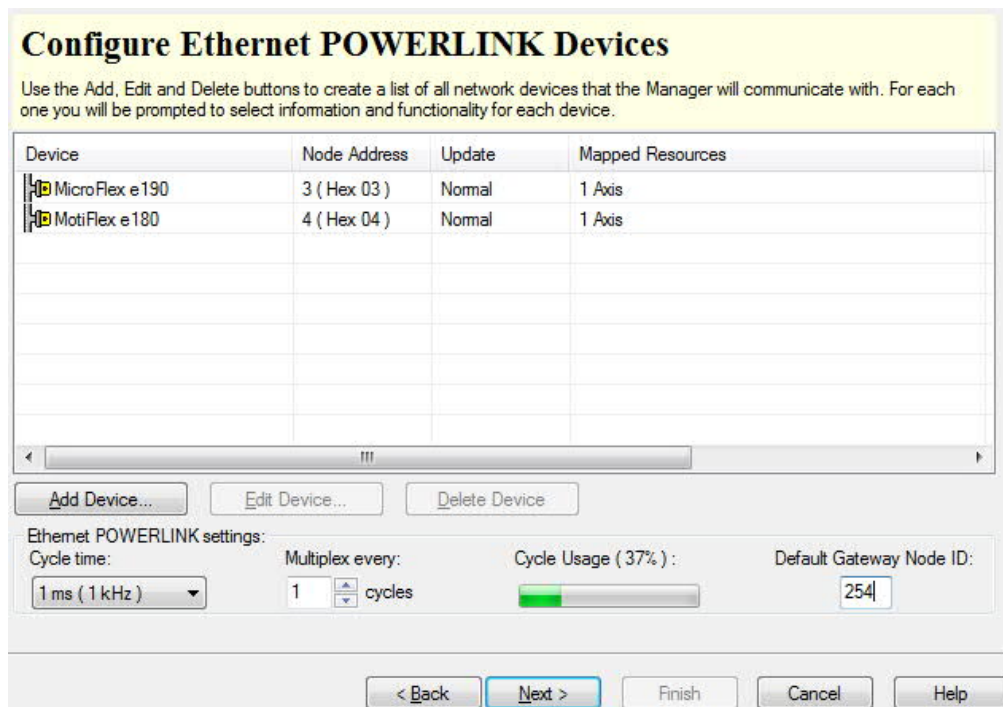
```
cAxis3 -> LATCHSOURCE(0)      = _IsAXIS_POSITION
cAxis3 -> LATCHSOURCECHANNEL(0) = 0
cAxis3 -> LATCHTRIGGERMODE(0) = _lrmINPUT
cAxis3 -> LATCHTRIGGERCHANNEL(0) = 1
cAxis3 -> LATCHTRIGGEREDGE(0) = _lrmPOSITIVE_EDGE
cAxis3 -> LATCHMODE(0)        = _lrmAUTO_ENABLE
```

We use the variable name (cAxis3 in this case) followed by the redirection symbol (->) to indicate we are accessing commands/parameters on the remote drive. In this example above we have configured latch channel 0 on the drive to capture position from digital input 1 on the drive.

Just as with the local latching, we must also enable the latch initially. However, when using remote latching this LATCHENABLE is actually enabling receipt of the latch data from the drive (as well as enabling the latch remotely) and is therefore a local command (i.e. we do not need to use redirection for the LATCHENABLE command).

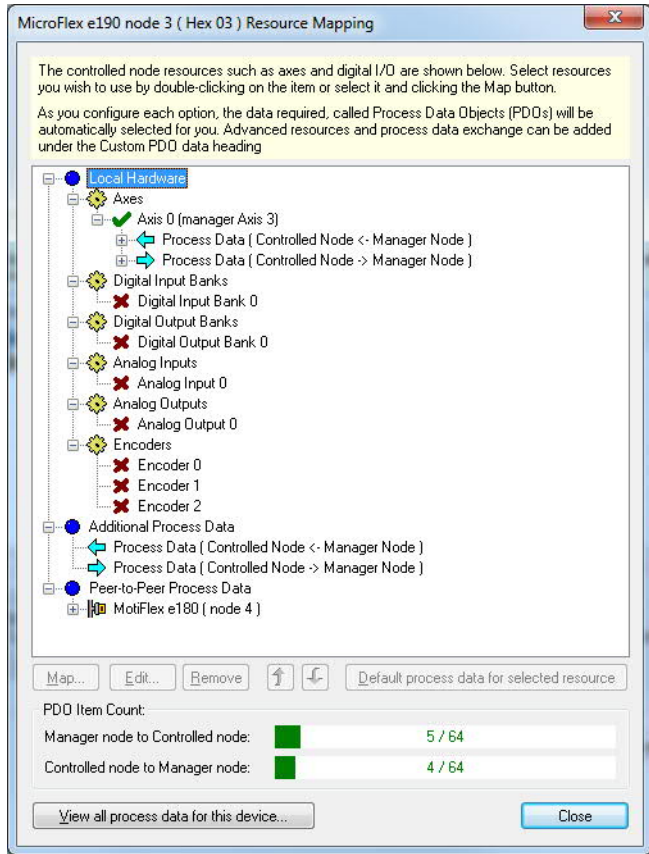
Before we can issue this command though, we need to setup the transfer/mapping of the remote latch value from the drive to the NextMove. This process will define the latch channel used on the NextMove to receive this data (i.e. we have configured latch channel 0 on the drive in this example, but we need to map this back to one of the 32 latch channels on the NextMove e100).

To configure the latch mapping, connect to the NextMove e100 and start the System Configuration wizard. Assuming you already have one or more configured remote e180 or e190 axes (see AN00187 for further details if necessary) click on 'Upload configuration from controller' and click on 'Next' to view the list of EPL devices configured.



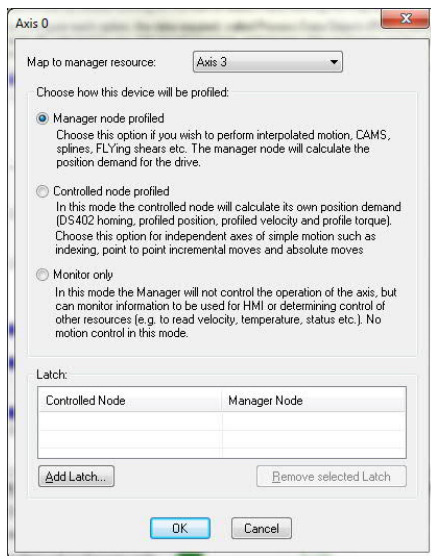
Highlight the appropriate drive and click on the 'Edit Device...' button to access the configuration for that specific drive (we have selected node 3 for this example).

Workbench will initially show the resource mappings for the axis...



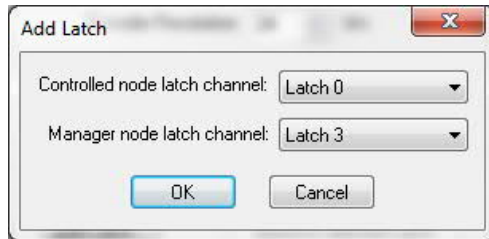
We must now decide if we want to receive a latched POSITION value from the remote axis or a latched ENCODER value. If we want a latched POSITION value to be passed back to the NextMove e100, double-click the 'Axis 0 (manager axis ...)' entry in the dialog.

If we want a latched ENCODER value to be passed back to the NextMove e100, double-click the 'Encoder 0 (manager Encoder...)' entry in the dialog.



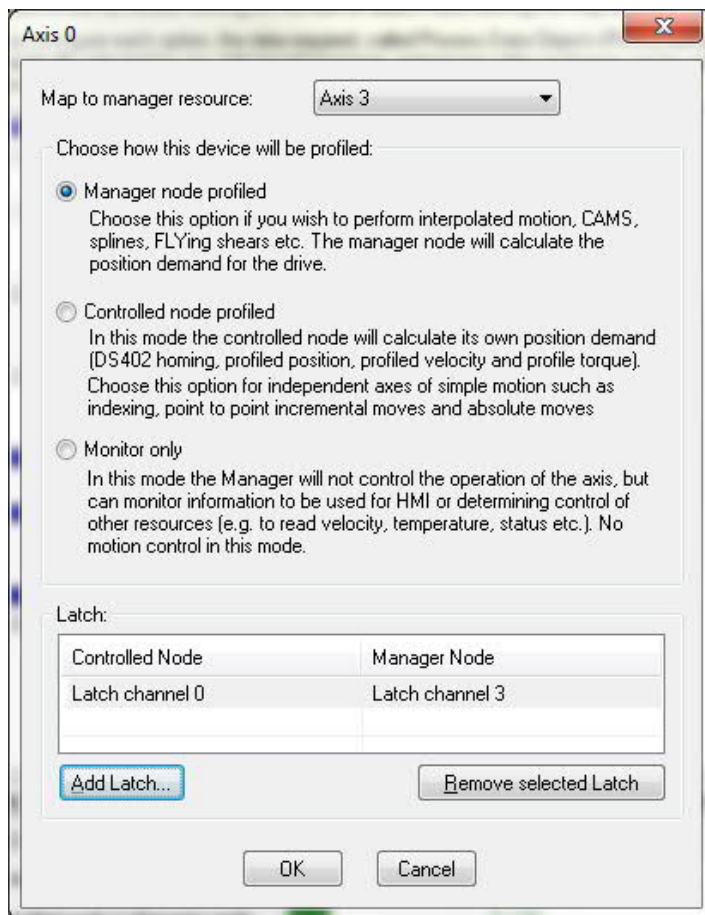
The screenshots above show the resulting dialogs (position latch mapping on the left, encoder latch mapping on the right).

In each case there is an 'Add Latch...' button. Click on this button...for our example we want to receive position data from the e190 drive and this is how we configured the drive's latch earlier so we have double-clicked the 'Axis 0.....' entry and then clicked on 'Add Latch...' in the Axis 0 dialog.



A small dialog appears that allows us to configure how the drive's latch channel is mapped back to the NextMove e100. In our example earlier we configured latch channel 0 on the drive to store the fast position (captured by digital input 1 on the drive). We have now mapped the drive's latch channel (0) back to the NextMove e100 as latch channel 3 (we selected 3 because this conveniently matches the drive's node address and axis number).

Click on OK. The relevant dialog (for Axis or Encoder data depending on what is being mapped) will now update to show the new mapping...



Click on OK again to accept this and then proceed with the rest of the System Configuration wizard until eventually the new Device Configuration File (DCF) can be saved and downloaded to the NextMove e100.

Now we have configured the local mapping we can access the latched data (position or encoder) via the associated latch event in our NextMove's Mint program. For example, as we mapped the drive's latch to latch channel 3 on the NextMove we would use...

Event LATCH3

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

End Event

And as before, to actually enable receipt of this data we would use the LATCHENABLE keyword and the relevant latch channel number....

LATCHENABLE(3) = 1

We can disable receipt of latch data from the drive at any time by setting this to zero...

LATCHENABLE(3) = 0

As we have already seen, each drive has two mappable latch channels so it is possible to map back any combination of latched encoder and position values (i.e. two position values, one position value and one encoder value or two encoder values).

Again, refer to the Mint help file topic 'Latching' and the range of keywords in the Help file starting with LATCH for more detailed information.

Latch timings (latency)

NextMove e100 (local latches):

Digital inputs 0 – 3 ; 1 μ s

Digital outputs 0 – 3 ; 16.67ns (although digital outputs are only updated every 1ms)

e180/e190 drives (local latches):

Digital inputs 1 / 2 ; 300 ns (incremental encoder), 62.5 μ s (other feedback types)

Digital outputs 0 / 1 ; 300 ns (incremental encoder), 62.5 μ s (other feedback types)

The time taken to call the Mint latch event (e.g. Event LATCH0) can be anywhere from 0 to 1 ms when using a local latch (or up to the defined EPL cycle time when using a remote latch), but this does not affect the accuracy of the latched data stored by either the NextMove e100 or remote EPL drive, only the time before this data (the LATCHVALUE) can be accessed and utilised.

Contact us

For more information please contact your local ABB representative or one of the following:

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

© Copyright 2017 ABB. All rights reserved.
Specifications subject to change without notice.