Using the Selector in the PTi210

Objective
Demonstrate the usage of the Selector feature.

Overview
The PTi210 module provides a unique means of externally selecting and initiating module functions. This feature is called a “Selector” and it is found under the “I/O Assignments-Selector” view of the PowerTools Studio programming software as shown in Figure 1 below.

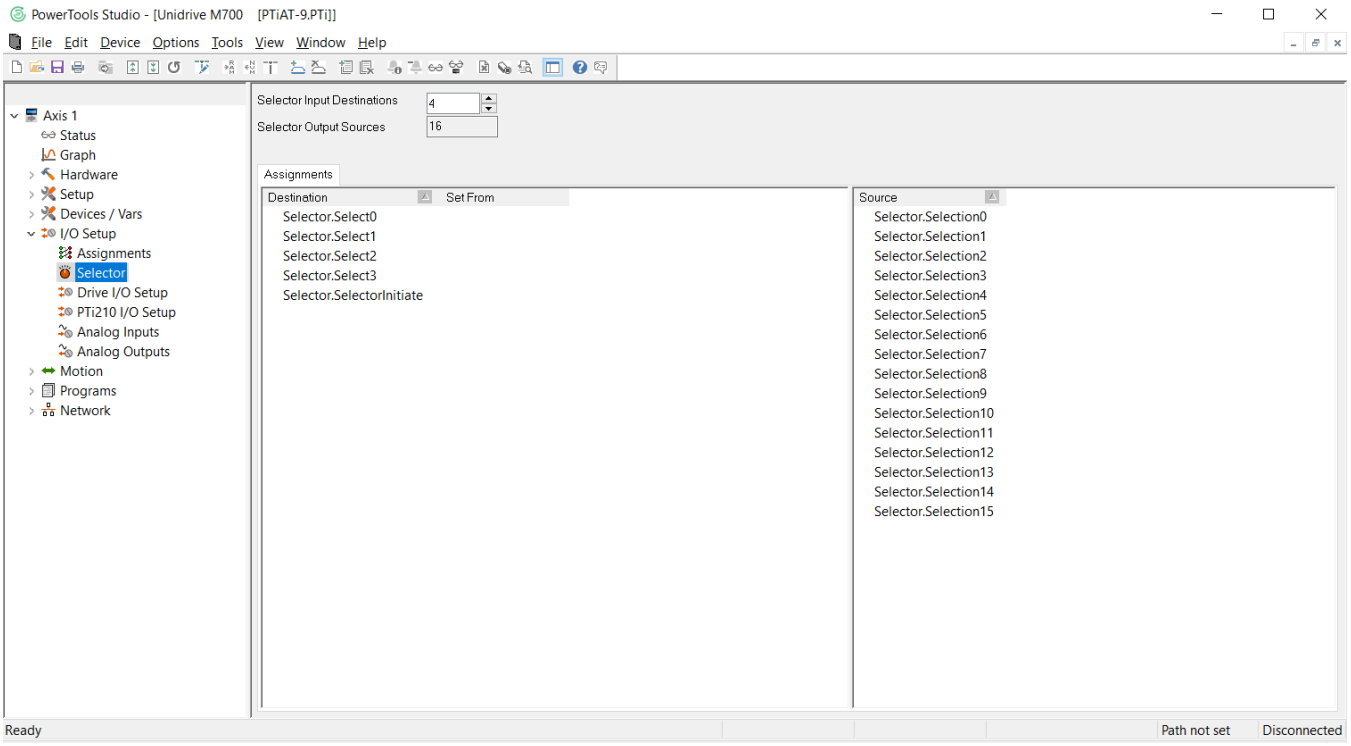

Figure 1

The Selector is used to conserve the quantity of digital inputs required to initiate common PTi210 functions like Jogging, Homing, Indexing and Motion Programs. For example, consider an application in which there is a requirement for:

1. A Home Sensor
2. A CW Overtravel Limit Switch
3. A CCW Overtravel Limit Switch

Control Techniques a division of Nidec Motor Corporation
7078 Shady Oak Road, Eden Prairie, MN 55344-3505 USA
T: +1 952 995-8000 www.controltechniques.us

4. A Stop Input
5. A Jog+ Input
6. A Jog- Input
7. 10 different Motion Programs

This example would require 16 external inputs into the drive and attached PTi210 in order to operate. Since the M700 with PTi210 module provides a maximum of 9 digital input points (6 on the base M700 drive and 3 more on the PTi210 module itself), it would be impossible to fulfill the input requirements for the application without using the Selector. The Selector would be configured to reduce the number of digital inputs required while not sacrificing any of the application's functionality.

Solution Summary

Program the Selector to utilize 3 *Selector Input Destinations*. In the Assignments view, attach the *Selector.Select0* through *Selector.Select2* Destinations to Drive Inputs 4-6. Attach the *Selector.SelectInitiate* Destination to Drive Input 3. Program *Selector.Selection0* through *Selector.Selection3* to be used as *Program.X.Initiate* where "X" corresponds to User Programs numbered 0-3. Program *Selector.Selection4* as the *Jog.PlusActivate* selection. Program *Selector.Selection5* as the *Jog.MinusActivate* selection. Program *Selector.Selection6* as the *Stop* selection. Program *Selector.Selection7* to activate DriveOutput.1. Program Jog0, Indexes 0-3. Create 4 Motion Programs (0-3) that incorporate a single Index in each program with an End instruction.

How the Selector Works

The Selector is composed of 4 separate parts:

Selector Input Destinations

Selector Select Destinations

Selector Selections

Selector Selector Initiate

Figure 2 shows a schematic view of the Selector. The Selector works like a rotary selector switch. On the input side of the Selector, there are Selector.Select Destinations. The quantity of Selector.Select Destinations determines the maximum number of Selector.Selections that can be output from the Selector.

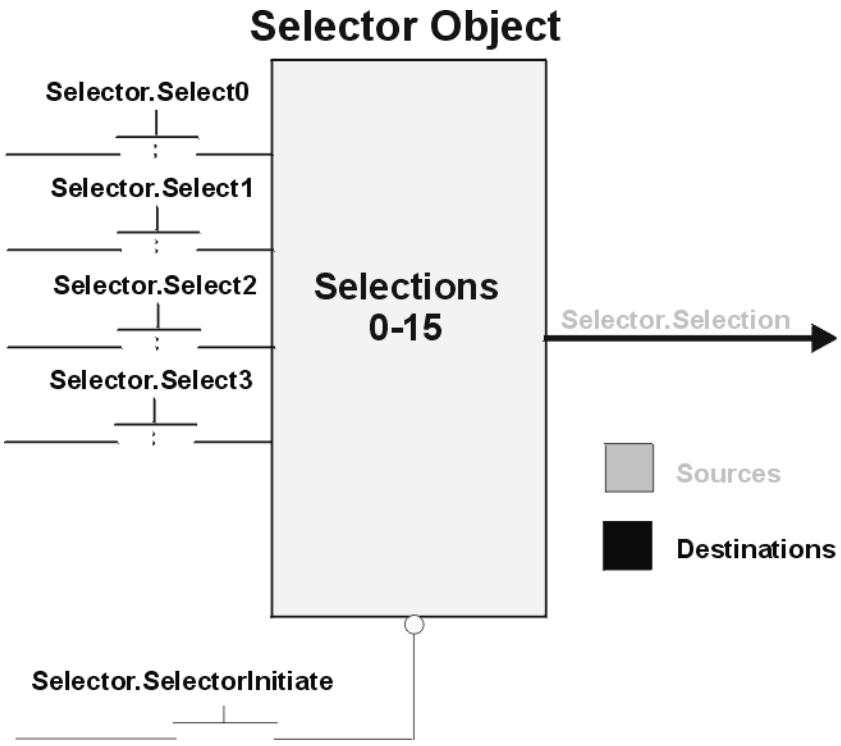
Selector Object
Selector.Select0
Selector.Select1
Selector.Select2
Selector.Select3
Selections
0-15
Selector.Selection
Sources
Destinations
Selector.SelectorInitiate

Figure 2

One of the keys to understanding the Selector is to understand the correlation between the state of the Selector.Select Destinations and the actual selection itself. The Selector.Select Destinations are interpreted by the module as a *binary* input pattern. Selector.Select0 is always the Least Significant Bit and therefore has a binary weight of 2^0 or 1. By activating the Selector.Select0 Destination, the Selector will output Selection 1, not Selection 0. The example in Figure 3 shows a total of 4 Selector.Select Destinations. This will yield up to 16 Selector.Selections (2^4 =16). The Selector.Select Destinations are normally attached to digital inputs on either the M700 drive or the PTi210 module. However, one of the more powerful features of the PTi210 module is the user's ability to attach these Destinations to any Source in the Assignments screen.

By the same token, the Selector.Selection Sources can be attached to any Destination in the Assignments screen. In this example, the Selector.Selections are going to be attached to Program.Initiate Destinations, the Jog.PlusActivate Destination, the Jog.MinusActivate Destination, the Stop Destination and the DriveOutput.1 Destination.

The Selector.SelectorInitiate is used to initiate the selection that is output from the Selector. The PLC timing for this operation would be for the PLC to update and maintain the Selector.Select lines with the correct binary pattern for the selection that is desired, and then to pulse the Selector.SelectInitiate line to produce an output from the Selector. Care should be taken to understand that the Selector.Select Destinations are *level* sensitive and the Selector.SelectorInitiate Destination is *edge* sensitive. If the PLC activates Selector.Select0 and Selector.Select1 and then pulses the Selector.SelectInitiate line, then Selector.Selection3 ($2^0 + 2^1$=3) will be output from the Selector.

To further illustrate the importance of properly timed Inputs into the Selector object, refer to the timing diagram in Figure 3.

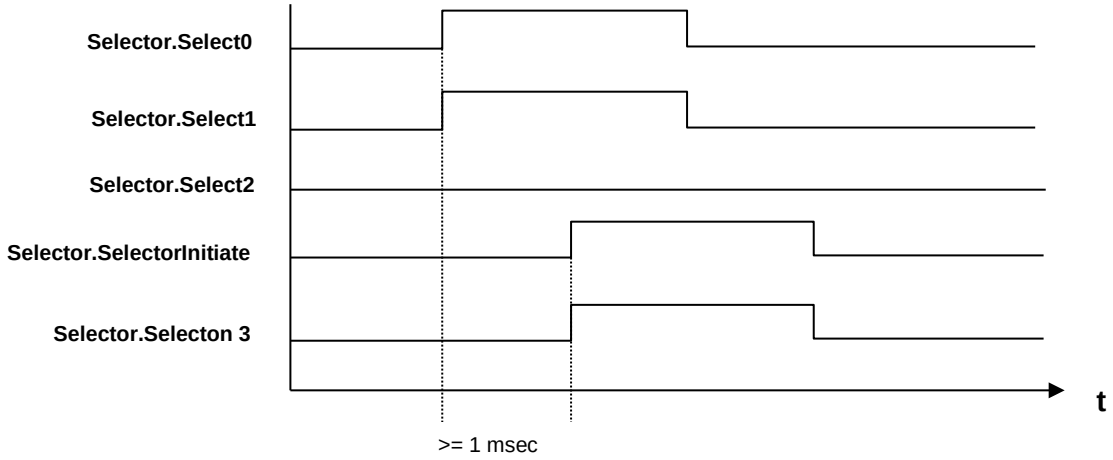

Figure 3

Of particular importance is the fact that the Selector.SelectorInitiate Destination must be present for a time equal to or greater than the user's Trajectory Update setting. This setting is found in the Setup view of the PowerTools Studio software. It is also important to note that the Selector.Selection3 Source is true for the same period of time that the Selector.SelectorInitiate Destination is true.

I/O Setup-Selector

To program the Selector, begin by opening the Selector view. The Selector view is found by expanding the "I/O Setup" view in PowerTools Studio software.

The maximum number of selections that the Selector can provide is determined by the value entered into the “Selector Input Destinations”. The default value of 4 in the Selector Input Destinations field will provide 16 selections for the Selector to use (2^4 =16). For this example, change the number of Selector Input Destinations to 3 as shown in Figure 4.

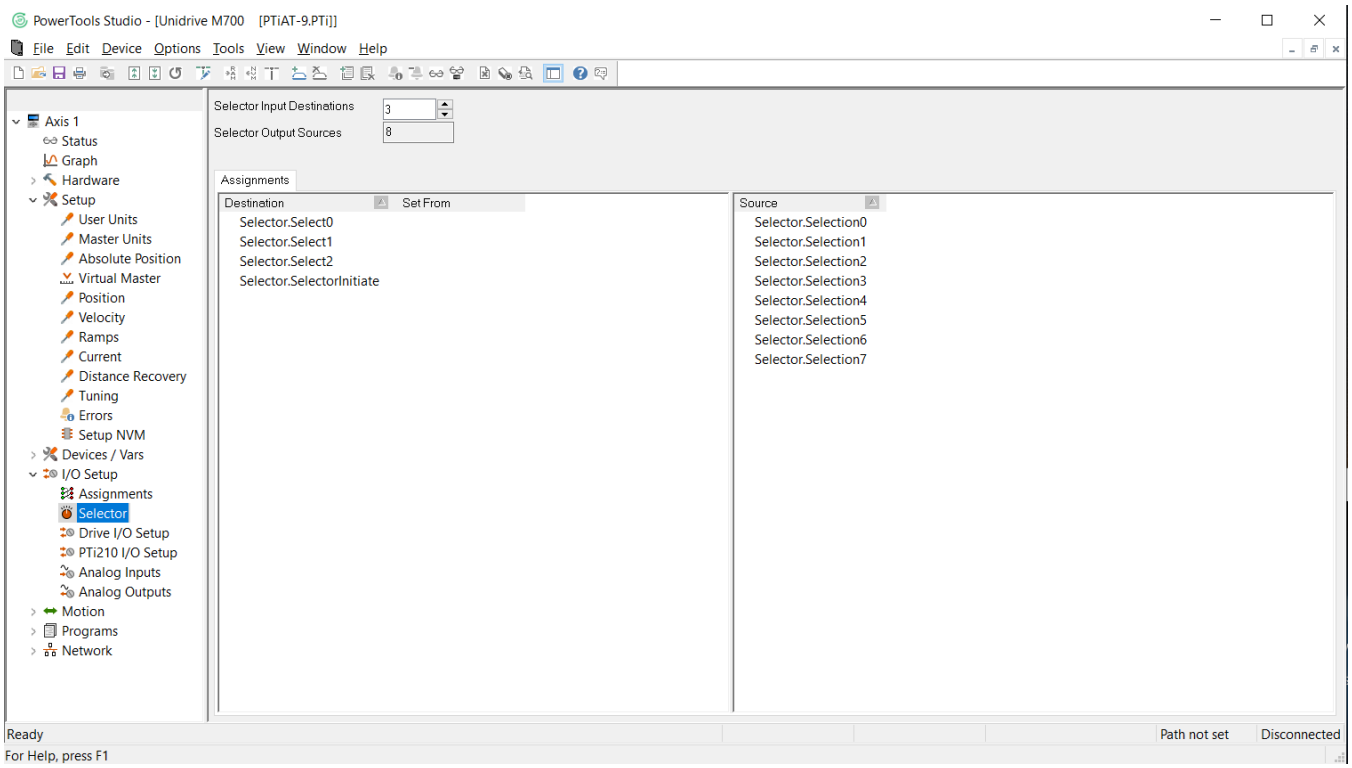

Figure 4

I/O Setup-Assignments

To utilize the Selector feature with the red simulator box, begin by opening the “Assignments” view of the software. Complete the necessary assignments as described below.

1. Expand the Selector Destinations.
2. Expand the Inputs Source
3. Drag the Destination *Selector.Select0* across the screen and attach it to *DriveInput.4*.
4. Drag the Destination *Selector.Select1* across the screen and attach it to *DriveInput.5*.
5. Drag the Destination *Selector.Select2* across the screen and attach it to *DriveInput.6*.
6. Drag the Destination *Selector.SelectorInitiate* across the screen and attach it to *DriveInput.3*.

Your Assignments screen should look like the picture in Figure 5 at this point.

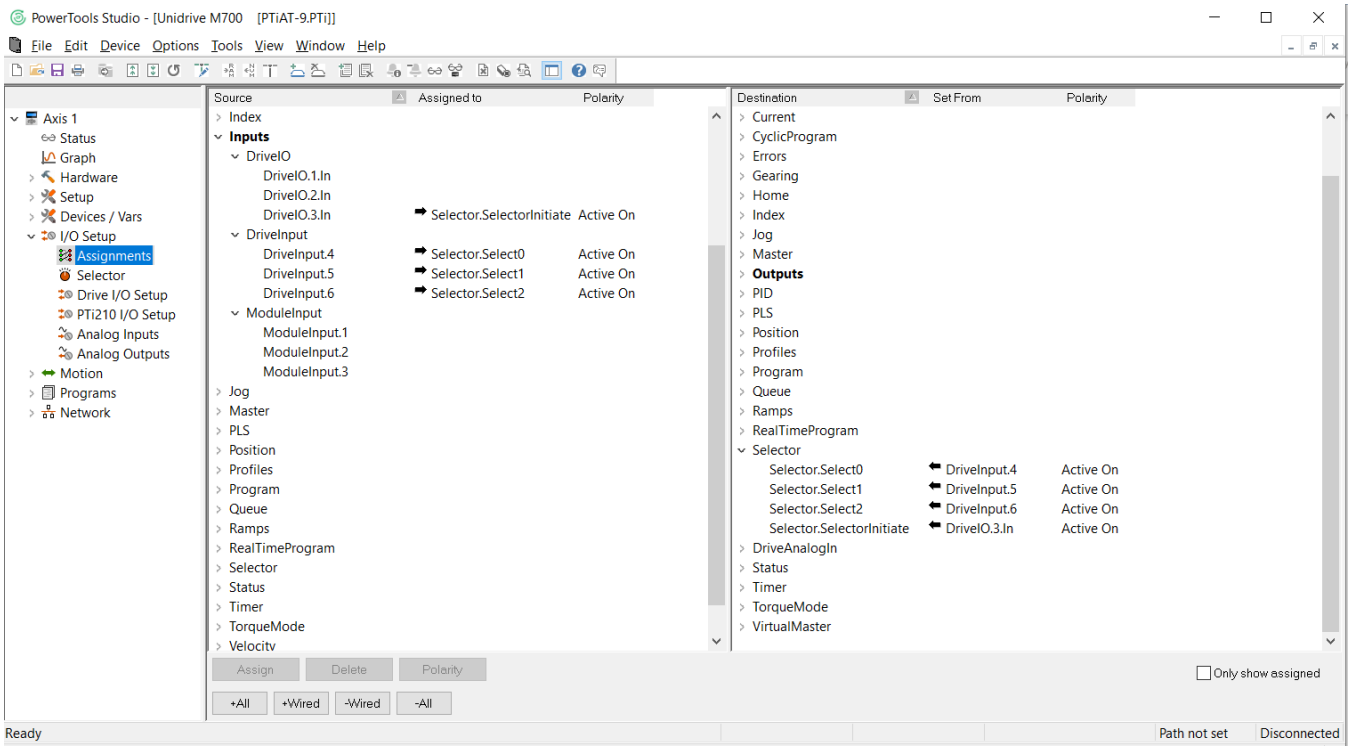

Figure 5

Before the Assignments screen can be completed, you must add 3 more Programs to your application. To add programs, follow these steps:

1. Click on the “Programs” level in the PowerTools Studio hierarchy.
2. Click the “Add New Program” icon on the software toolbar 3 times to add 3 new programs as shown in Figure 6.

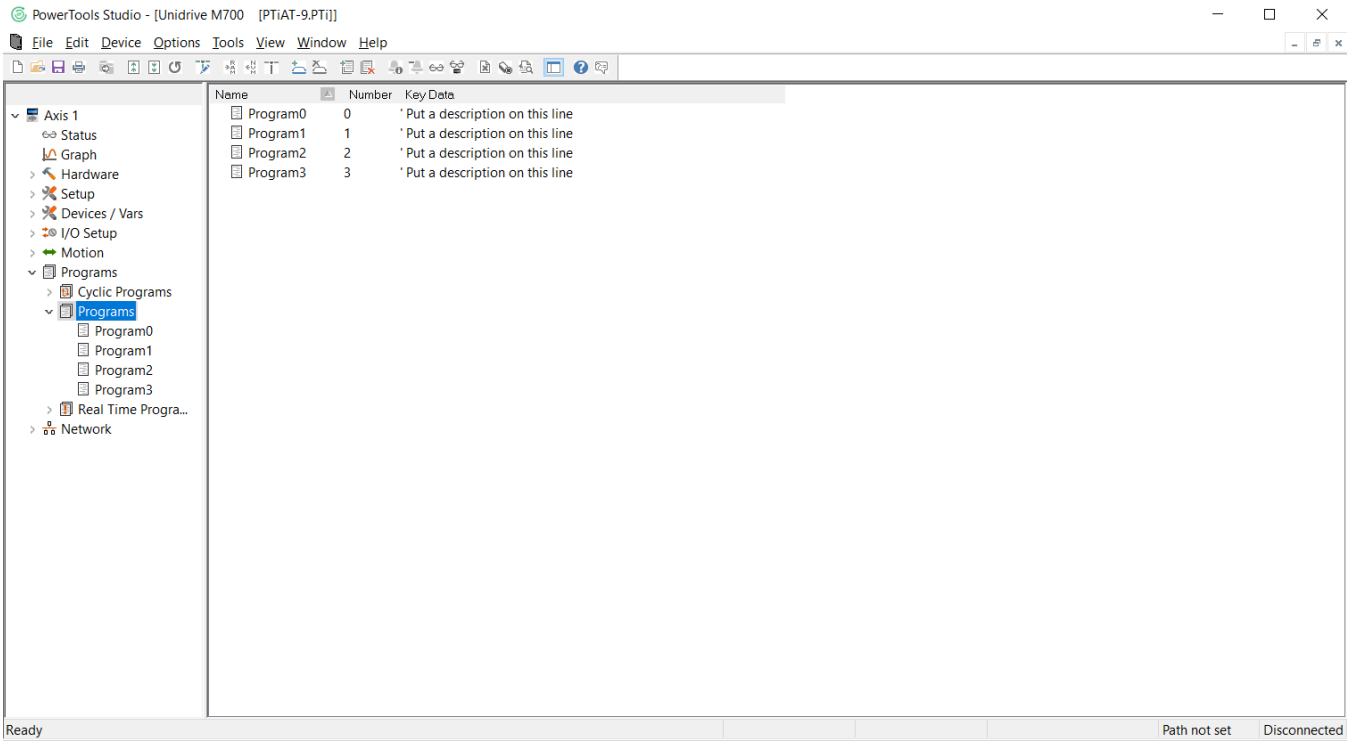
PowerTools Studio - [Unidrive M700 [PTiAT-9.PTi]]
File Edit Device Options Tools View Window Help
Axis 1
Status
Graph
Hardware
Setup
Devices / Vars
I/O Setup
Motion
Programs
Cyclic Programs
Programs
Program0
Program1
Program2
Program3
Real Time Progra...
Network
Name Number Key Data
Program0 0 ' Put a description on this line
Program1 1 ' Put a description on this line
Program2 2 ' Put a description on this line
Program3 3 ' Put a description on this line
Ready
Path not set
Disconnected

Figure 6

Return to the Assignments screen to finish the configuration of the Assignments. To continue, follow these steps:

1. Expand the “Selector” Sources.
2. Expand the “Program” Destinations, then expand the Program0-2 Destinations.
3. Drag the Program.0.Initiate Destination across the screen and attach it to the Selector.Selection0 Source.
4. Drag the Program.1.Initiate Destination across the screen and attach it to the Selector.Selection1 Source.
5. Drag the Program.2.Initiate Destination across the screen and attach it to the Selector.Selection2 Source.
6. Drag the Program.3.Initiate Destination across the screen and attach it to the Selector.Selection3 Source.
7. Expand the “Jog” Destination.
8. Drag the Jog.PlusActivate Destination across the screen and attach it to the Selector.Selection4 Source.
9. Drag the Jog.MinusActivate Destination across the screen and attach it to the Selector.Selection5 Source.
10. Expand the “Ramps” Destinations.
11. Drag the “Stop” Destination across the screen and attach it to the Selector.Selection6 Source.
12. Expand the “Outputs” Destinations.
13. Drag the DriveOuput.1 Destination across the screen and attach it to the Selector.Selection7 Source.

Figure 7 shows the completed Assignments screen for this example.

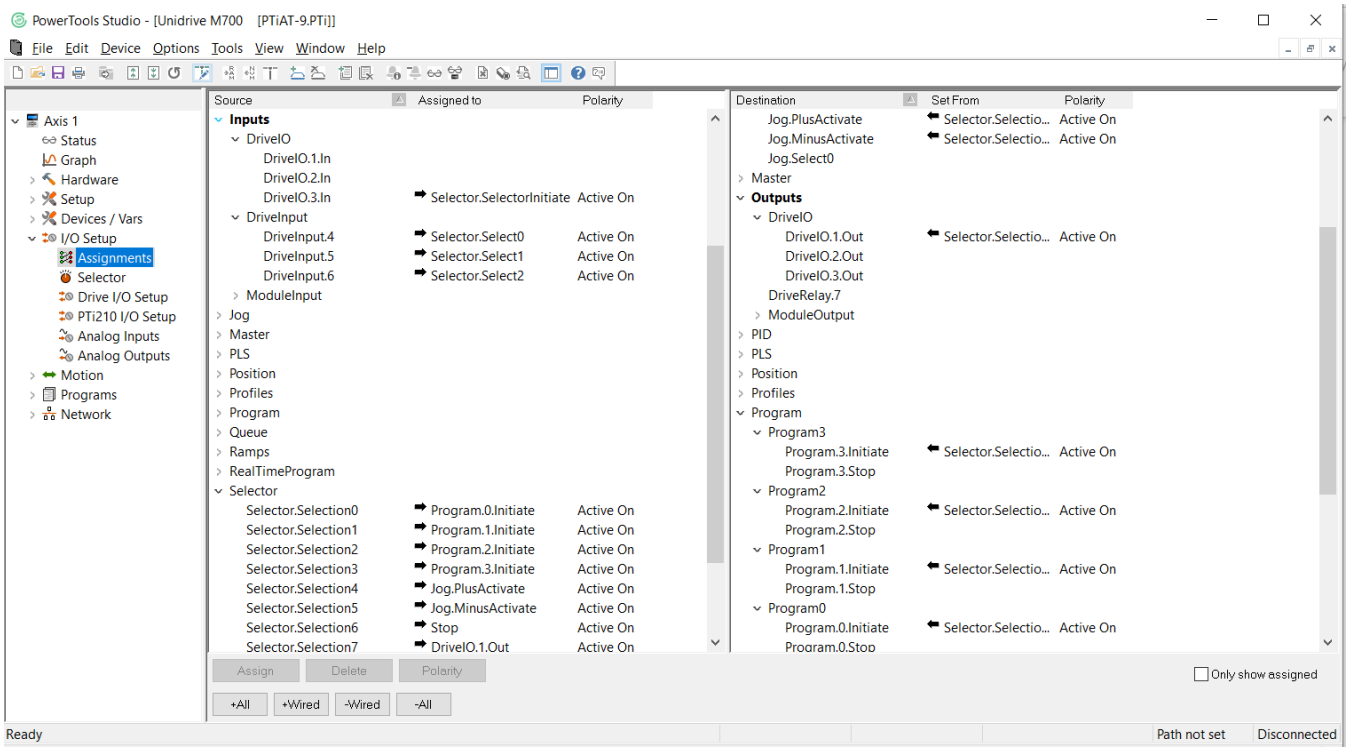

Figure 7

A Truth table for the binary selection process is shown in Figure 8. The Selector.SelectorInitiate input (DriveInput.4) would be pulsed to initiate each selection.

Drive Input #4 State	Drive Input #5 State	Drive Input #6 State	Selection #
0	0	0	0
1	0	0	1
0	1	0	2
1	1	0	3
0	0	1	4
1	0	1	5
0	1	1	6
1	1	1	7

Figure 8

Motion-Jogs-Jog0

Configure Jog0 according to Figure 9 below.

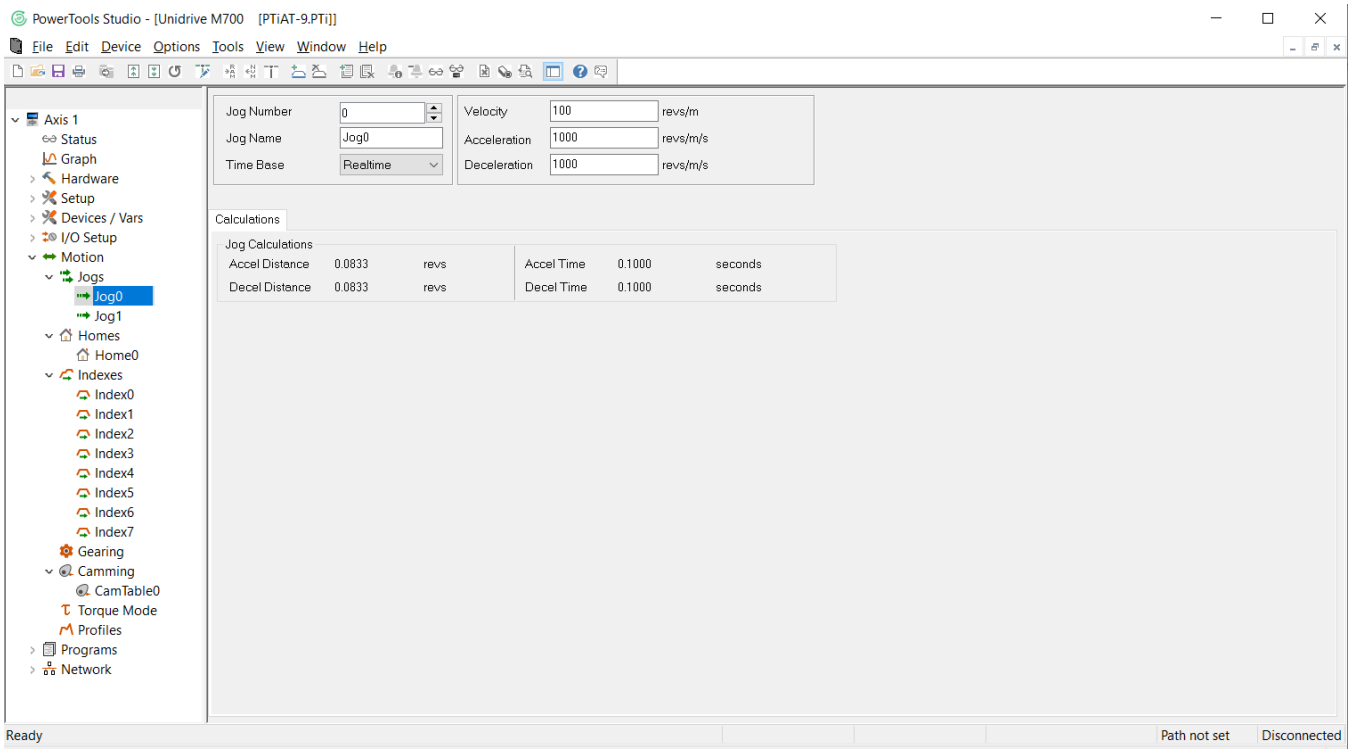

Figure 9

Motion-Indexes

Expand the Indexes and then configure 4 different Indexes to use in this example application. Each of the 4 different Programs will contain a single Index. A suggested configuration is outlined in Table 1.

Index #	Index Type	Index Distance	Index Velocity	Index Accel	Index Decel
0	Incremental	10 Revs	100 RPM	1000 rev/min/sec	1000 rev/min/sec
1	Incremental	20 Revs	200 RPM	1000 rev/min/sec	1000 rev/min/sec
2	Incremental	30 Revs	300 RPM	1000 rev/min/sec	1000 rev/min/sec
3	Incremental	40 Revs	400 RPM	1000 rev/min/sec	1000 rev/min/sec

Table 1

Programs-Program0

Create Program0 according to Figure 10 below.

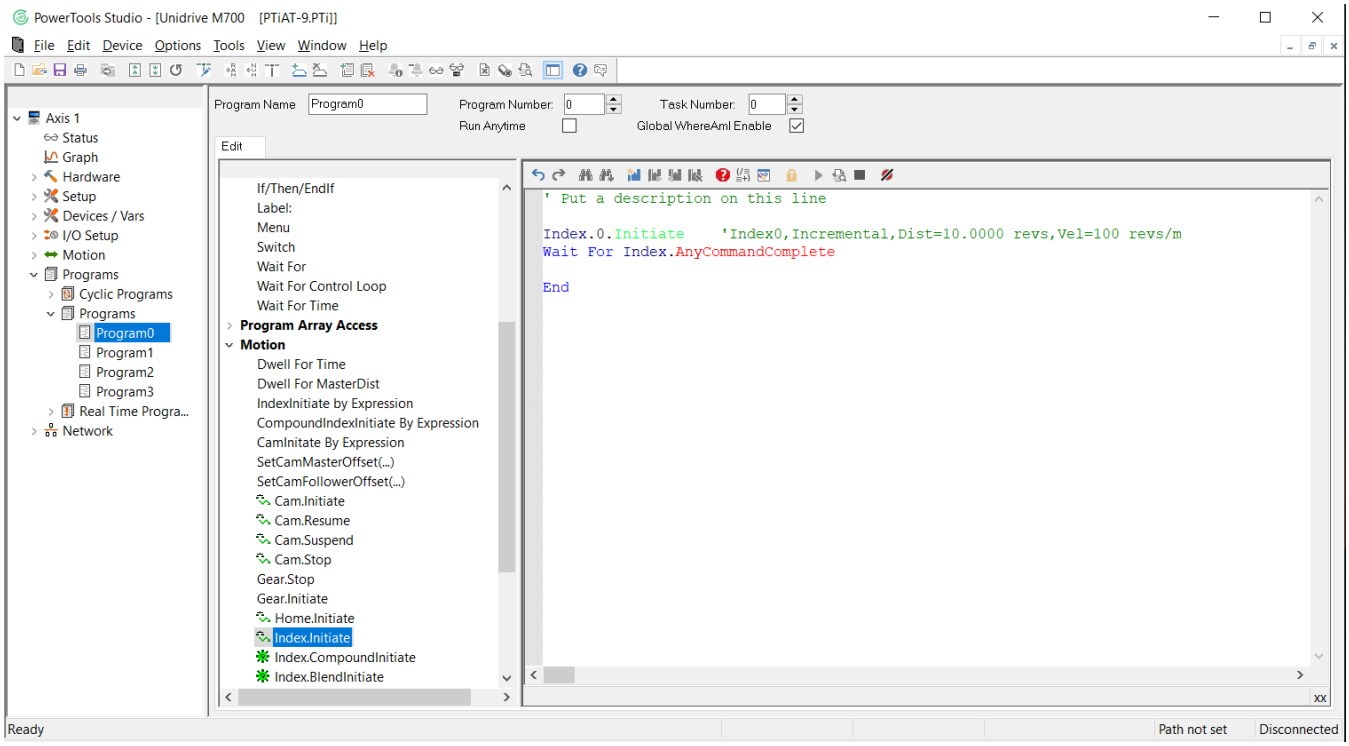

Figure 10

To complete the programming for this example application, configure Programs 1, 2 and 3 exactly as shown in Figure 10 with the exception of using Indexes 1, 2 and 3 in place of Index0 in each additional Program.

When completed, you should have 4 different Programs (Program 0 – Program 3). Each Program will contain a single Index and an End instruction.