DSM Motion Program Variables

Objectives

UPON COMPLETION OF THIS MODULE, YOU SHOULD BE ABLE TO:

- Change PAC controller variables from an Operator interface.
- Copy the PAC controller variables to a DSM Parameter Variable.
- Examine the DSM Parameter values on an OI
- Execute a DSM Motion Program using the Variables.

Overview

This module will demonstrate the use of variables in motion programs stored in a DSM motion controller module for the RX3i PLC. This implementation provides a flexible method of setting and changing various motion program settings from either an operator interface or from the PLC logic program.

This workshop exercise will demonstrate how to change and monitor five DSM parameter variables, however the methodology can be extended to allow as many as 215 DSM parameters to be used in one motion program. The RX3i maintains up to ten motion programs and 40 subroutines in a single file for each DSM module as part of the hardware configuration.

Motion program parameters may be used as variables for motion settings such as acceleration, velocity, move distance, dwell time and more. As an example, this lab motion program contains instructions for a move to a location, a time delay dwell and a return to the zero position using five parameters. Using variables in this example motion program allows the operator the ability to control all aspects of this point-to-point move without changing the PAC controller or motion programs. This approach is very useful where different production lots use essentially the same axis movements but with different move distances, speeds, cycle times, etc.

Motion program variables may be changed at any time in the machine cycle however the new value will not be utilized until the motion program attempts to use the command containing the variable. In practice this means that new parameter values typically take effect on the next machine cycle.

DSM parameters are stored in volatile memory in the DSM module and will reset to zero value whenever the PAC controller is power cycled or if a new hardware configuration is stored to the RX3i. Most programmers construct PLC program logic to initialize the DSM parameters any time the PLC makes a stop to run transition.

Write and Verify the Value of a DSM Motion Program Parameter

The necessary steps to transfer an operator-entered value to the DSM motion program parameter are.

1. Write the OI variable to an RX3i variable (tag)
2. Copy the RX3i tag to a DSM Parameter
3. Optionally, verify the value written to the DSM parameter

Writing OI variables to PAC controller tags.

In the context of this workshop exercise, the user will enter a variable value on a QuickPanel View operator interface screen. Using the capability of the QuickPanel View a value may be displayed in the desired engineering unit and automatically scaled to the appropriate user unit for the DSM axis (i.e. displayed as a floating point metric or English standard value on the OI but written to the PLC as a DINT value matching the DSM axis scaling). Additionally, variable data entered on the QuickPanel View may be range checked to maintain the variable setting between desired limits.

The QuickPanel operator interface will write this new variable value (tag) to the RX3i. The RX3i will then copy this tag value to the DSM324 module as a DSM parameter. All DSM motion program parameters are typed as double integer (DINT) and must be written to the DSM324 as a DINT value. Typically the OI creates the RX3i tag as a type DINT.

Copying RX3i Tags to DSM Parameters.

There are two different methods available for the RX3i to copy a parameter value to the DSM324;

- Immediate Command – Write Parameter
- Load Parameter COMMREQ.

Each method will require a minimal amount of PLC application logic in order to execute.

Immediate Command – Write Parameter

Each DSM324 module used in the RX3i will use some of the mapped IO data memory in the RX3i controller. The DSM module uses %I (Discreet Input), %Q (Discreet Output), %AI (Word Input) and %AQ (Word Output) data types determined by the hardware configuration and the number of axis selected to operate on the module. For the Immediate Command interface we need only use the %AQ references. The RX3i hardware configuration will establish the beginning %AQ reference for a given DSM324 module.

Word Offset	*Demo Address*	*Description*	*Axis*
00	%AQ0021	Immediate Command Word	Servo 1
01-02	%AQ0022/23	Command Data	Servo 1
03	%AQ0024	Immediate Command Word	Servo 2
04-05	%AQ0025/26	Command Data	Servo 2
06	%AQ0027	Immediate Command Word	Servo 3
07-08	%AQ0028/29	Command Data	Servo 3
09	Axis not	Immediate Command Word	Servo 4
10-11	configured	Command Data	Servo 4

Table 2- 1. DSM Comand Word to Axis Map

The immediate command format uses the %AQ references mapped to the DSM324 and can change one DSM parameter for each configured axis per PLC sweep. This interface is typically used if a small number of parameters will be changed at one time (e.g. the operator changes a single set point).

Load Parameter COMMREQ

The Load Parameter COMMREQ copies a block of 16 DINT variables (tags) from RX3i controller memory to a block of consecutive DSM parameters at the specified parameter starting point. If required, multiple COMMREQ instructions may be executed in the same RX3i sweep. Each COMMREQ will contain confirmation that the write operation to the DSM was successful. This format is useful when the user wants to initialize all of the parameter values in an entire motion program (e.g. on machine power up) or a change of product. Recipe management functionality at the operator interface level (or higher) can be interfaced to the COMMREQ data structures to provide even more functionality when changing a given machine to run a different job lot.

Verify the Value of the DSM Parameter (Optional)

As an option the user may validate or display the value in a given DSM parameter using one of two methods:

- On-Demand Read (Immediate Command)
- Continuous Read (hardware configuration)

In both cases the DSM parameter value is copied into an RX3i %AI memory location as a DINT. If desired, the OI may read and display this value.

Each DSM configured axis will have two locations in the %AI memory map (User Selected Data 1 and User Selected Data 2) where user data will appear. Implementing either of the following methods will cause the desired parameter value to be refreshed in the PLC %AI memory each I/O sweep until a new command is entered or the RX3i PAC controller is reset.

On-Demand Read of a DSM Parameter Value

The immediate commands used to read a DSM parameter value are Select Return Data 1 (40h) and Select Return Data 2 (41h). These commands also require mode and offset.

- **Mode** - determines what type of information will be stored to the %AI memory location (e.g. mode 18h specifies parameter data).
- **Offset** - determines which specific DSM parameter number will be stored to the %AI memory location.

For example, the %AQ command Select Return Data 1, Parameter Mode (1840 hex) and Offset (1) would read the value of parameter 1 and store it in the %AI memory location for that axis User Select Return Data 1.

Continuous Read of a DSM Parameter Value

This feature is functionally similar to the User Select Return Data immediate command. It is specified in the RX3i controller hardware configuration and becomes the default data written to the User Select Return data %AI memory location each PLC sweep. The immediate command interface can modify the configured User Select Return data, however, on a power cycle or RX3i controller reset the configuration setting will be reinstated. For purposes of this lab, the DSM has been configured for continuous read of parameter numbers 1–5.

Lab Exercise – Working with DSM Motion Programs

Lab Objectives: This exercise is designed to help the user:

- Understand minimal logic required for a simple motion application
- Understand basic DSM motion program commands and operation
- Create logic to download parameters to the DSM motion controller
- Change a DSM stored motion program

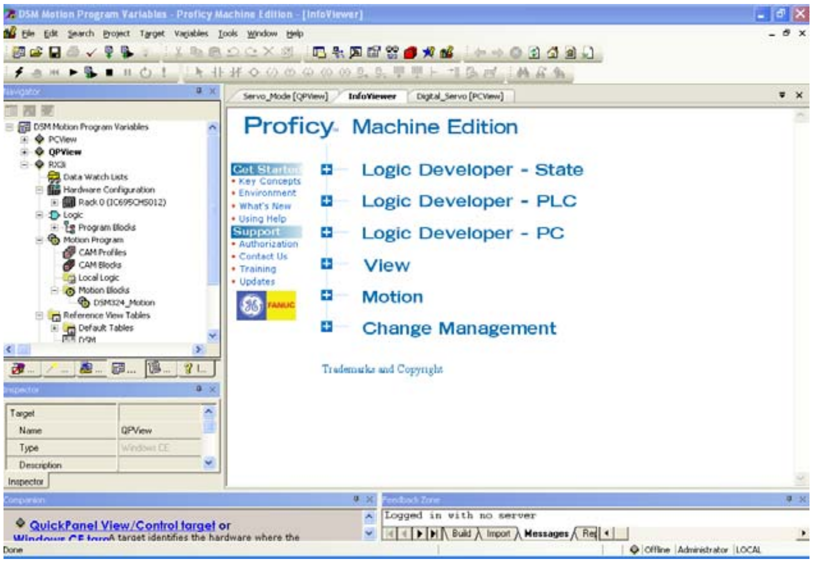
The RX3i controller and the QuickPanel View operator interface in this lab comprise a completely configured and programmed application allowing the user to move the servo motor in one of two operating modes:

- **Manual Mode** - uses the switches on the PLC Input Simulator module (IC694ACC300 – fourth card from the left) as the command interface (optional).
- **Auto Mode** - uses the QuickPanel View operator interface as the command interface.

Your task in this lab will be to understand the functionality provided in the RX3i and to add program logic to download five parameters to the DSM motion controller module. Time permitting you can explore the optional procedure on how to make a change to the motion program.

Procedure 1: Download Project to RX3i

1. Open the project titled "DSM Motion Program Variables" by double-clicking on this project name in the Navigator window. This should result in a workbench similar to the one shown here (same windows and toolbars):



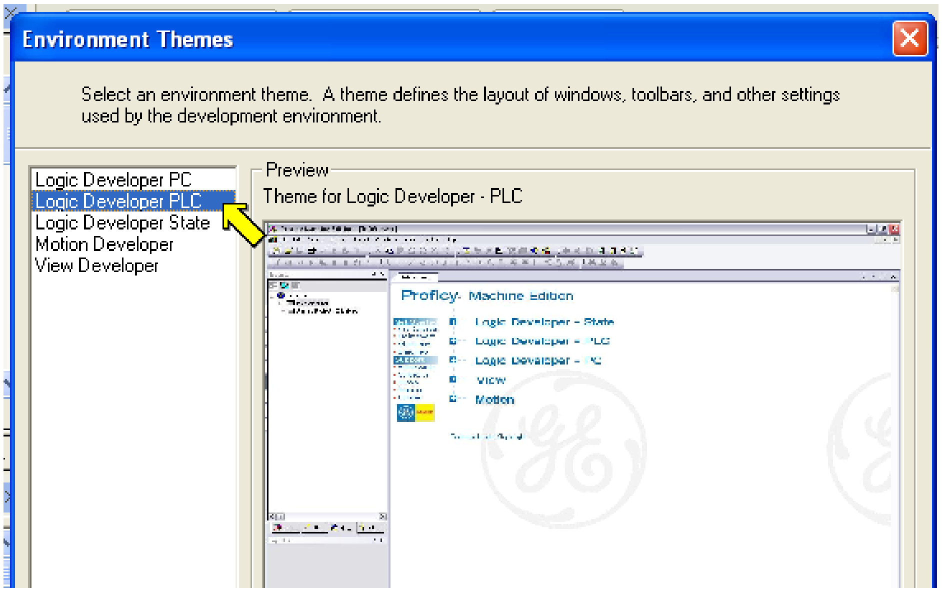
Logged in with no server

TIP!

Workbench windows can be opened, closed, moved and parked in many different ways to suit your individual work style. You can even auto-hide windows in order to leave more room for work on logic programs. The Machine Edition (ME) software will remember your custom arrangement when you re-open the workbench.

Sometimes, however, you may want to quickly restore the default layout or change to a layout for a different target (in order to get the right toolbars visible). This is most easily done through the Window/Apply Theme menu item.

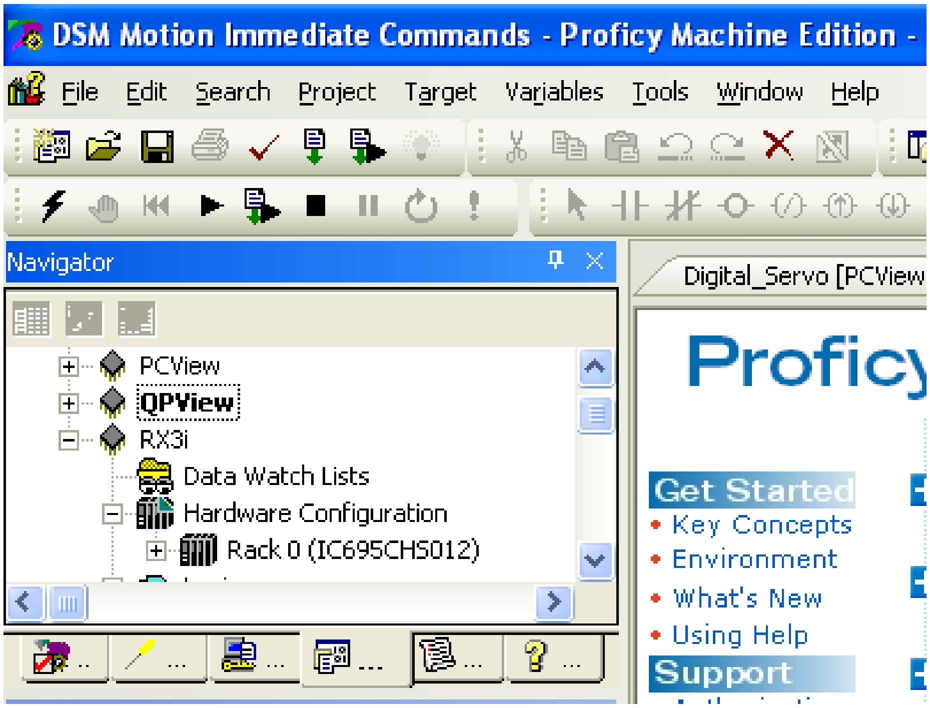
2. If the screen does not look like the above, use the Window/Apply Theme command to open the following selection window:
3. Select Logic Developer PLC and press “OK”.



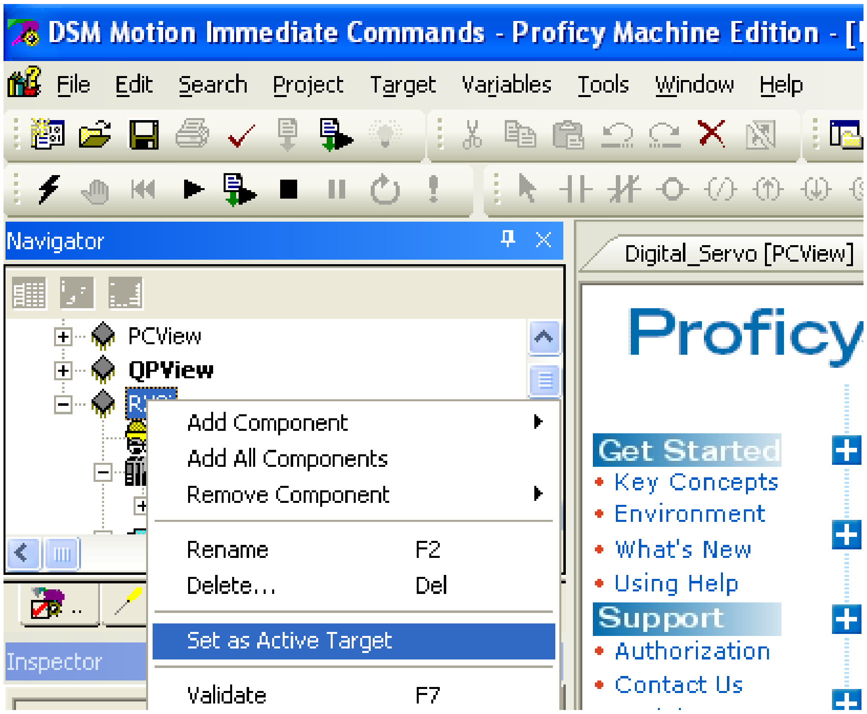
Select an environment theme. A theme defines the layout of windows, toolbars, and other settings used by the development environment.

Theme for Logic Developer - PLC

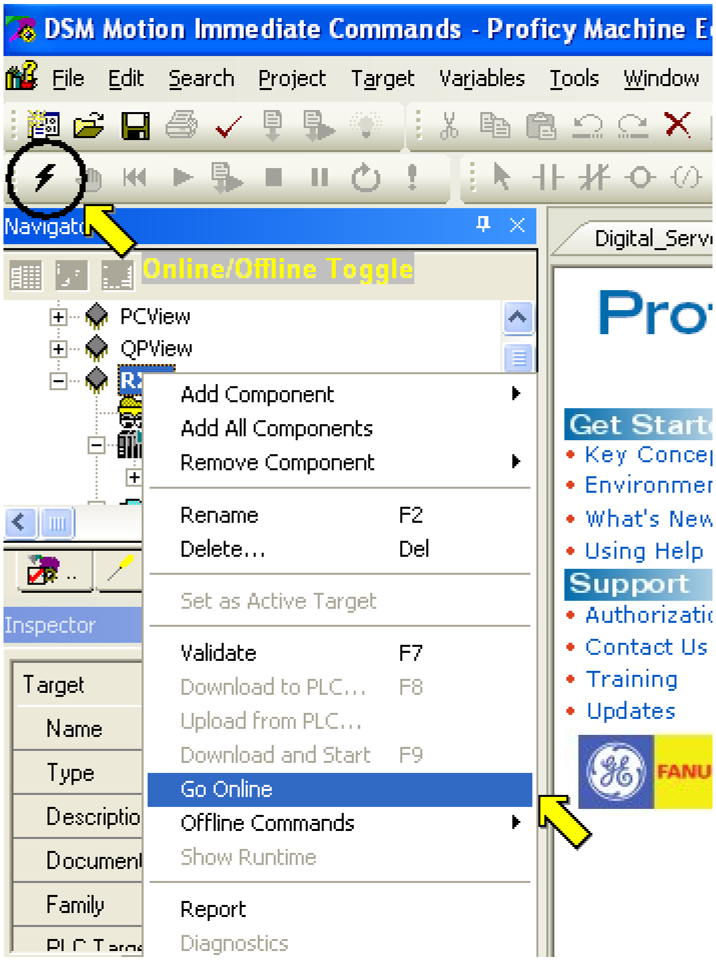
4. Collapse the targets as necessary in the Navigator window by left clicking on the “-“ next to them until you can see the RX3i target as shown below



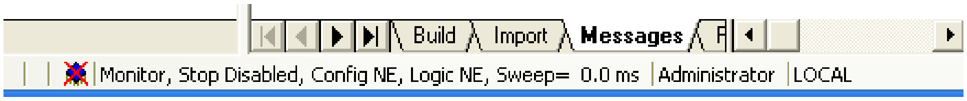
5. If the RX3i target is not shown in bold type, right-click on it and select “Set as Active Target”. Setting the active target tells Machine Edition which target you will be commanding with the toolbar buttons and allows us to go online and view the logic in the target.



6. Now go online with the RX3i by selecting the lightning bolt on the toolbar. Alternatively, you can right-click on the target name and selecting “Go Online”.



7. Examine the status bar at the bottom of the ME window. If the diamond to the left of RX3i target name appears in blue this indicates that the target is connected, but Logic Developer-PLC is in “Monitor Mode”. (See bottom right hand corner of workbench window for mode indication as shown below.)

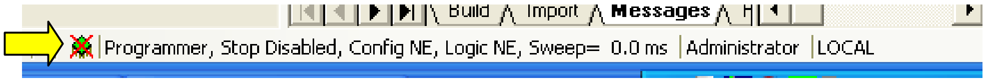


Monitor, Stop Disabled, Config NE, Logic NE, Sweep= 0.0 ms | Administrator | LOCAL

TIP!
Toggle between Monitor and Online Mode using the green hand icon.

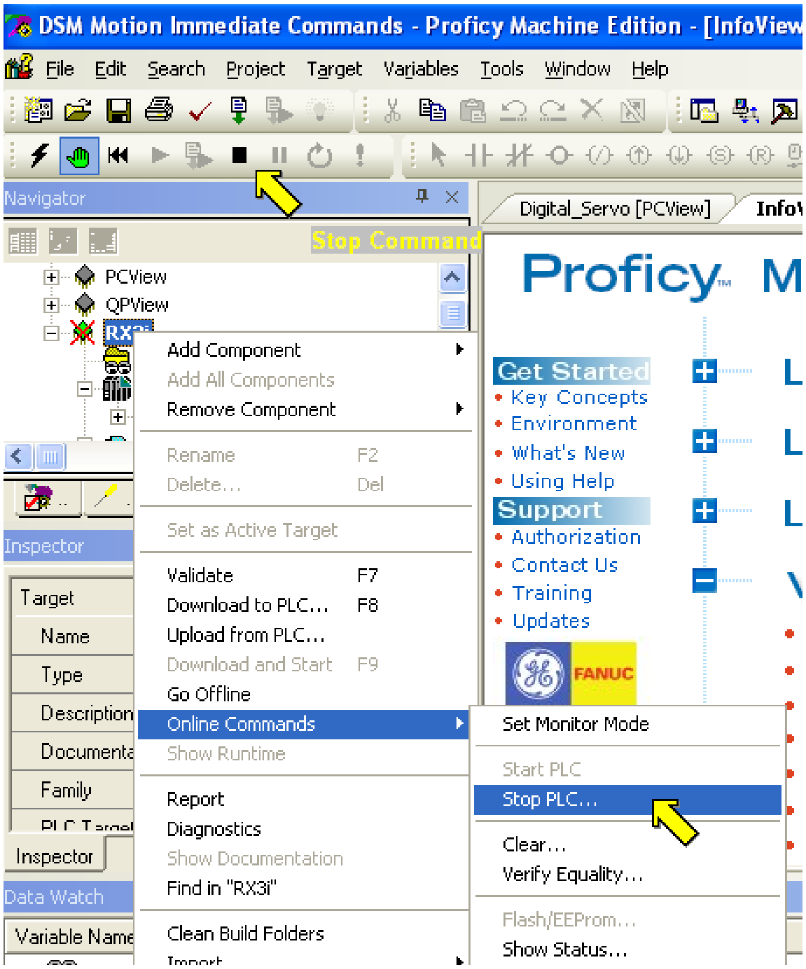
Note that the active target status indicator (diamond shape) appears on this line also. The red X indicates that the logic or configuration in the PLC is not equal to what is in the ME project. Further information is shown by the status line text including the run/stop/fault mode and the sweep (scan) time.

If the status indicator is blue, left-click on the toolbar hand symbol to toggle the mode to “Programmer”. The status indicator will turn green.



Programmer, Stop Disabled, Config NE, Logic NE, Sweep= 0.0 ms | Administrator | LOCAL

8. If the PLC is in Run mode, you will have to stop it in order to download the configuration. Programs can only be downloaded in run mode if the program name matches the one in the folder. To stop the PLC, use the stop button on the toolbar ■ or right-click the RX3i target and select Online Commands/Stop PLC.



TIP!

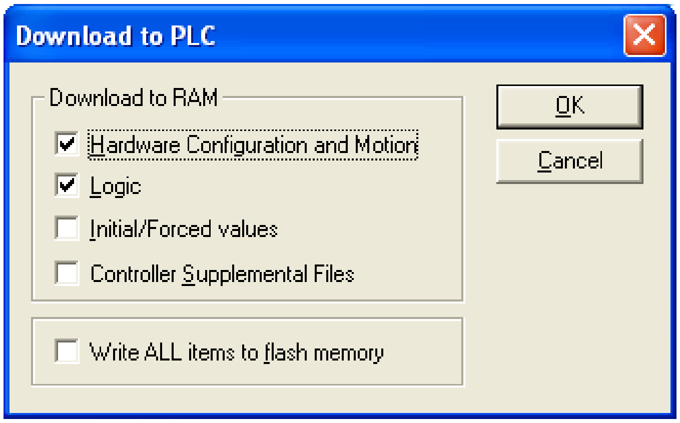
Use toolbar icons for quick access to “Download”…

or “Download and Start” commands…

WARNING!
Toolbar icons always operate on the Active Target!

9. Once the PLC is stopped, select “Download and Start” from the right-click menu or using the toolbar icon. You will be presented with a choice as show at right:

10. Select “Hardware Configuration and Motion” and “Logic” as shown and press “OK” to begin the download.



Download to PLC

Download to RAM

☑ Hardware Configuration and Motion

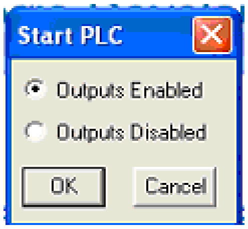
☑ Logic

☐ Initial/Forced values

☐ Controller Supplemental Files

☐ Write ALL items to flash memory

11. When downloading has completed another dialog will allow you to start the PLC with outputs enabled or disabled. Select “Outputs Enabled” and press OK.

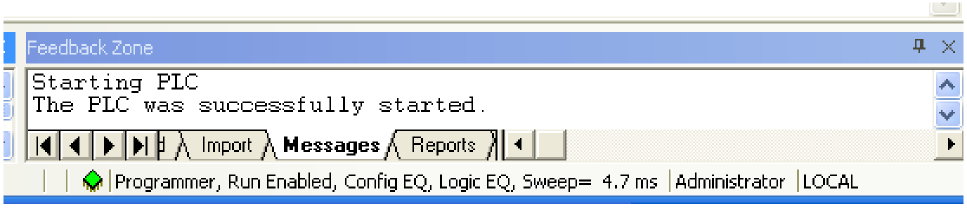


Start PLC

● Outputs Enabled

○ Outputs Disabled

The Feedback Zone Messages tab and status bar should appear as follows:



Feedback Zone

```
Starting PLC
The PLC was successfully started.
```

Programmer, Run Enabled, Config EQ, Logic EQ, Sweep= 4.7 ms | Administrator | LOCAL

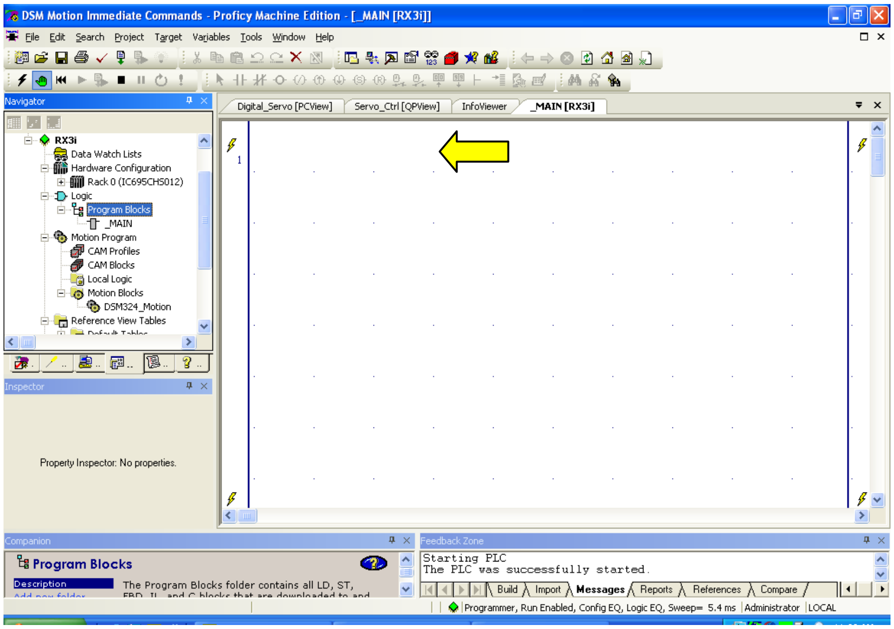
Tip!

Feedback Zone window will open automatically whenever you ask ME to do anything that requires completion or error messages

The Feedback Zone has many other uses as you may note by selecting the various tabs at the bottom of the window. One of the most helpful is the “References” tab where you will automatically see all uses of any variable (in all targets in the project!) that may be selected in any other window and you may go to each point of use by double clicking on the listed instance.

If there were errors, these would also show up in the Feedback Zone and double-clicking on each error message would open the required window to show you the problem. The Companion window (if open) would give you an explanation for the error. The Companion window may be opened by selecting the Star Icon on the toolbar or by pressing Shift + F11. The Companion window is open by default in the PLC Theme.

The RX3i is now configured and ready to interact with the viewers. Note that there is no program in the controller. If desired, you can confirm this by opening the _MAIN program block and noting that it shows no logic. In order to view this, simply expand the RX3i target node in Navigator window, expand Logic node, expand Program Blocks and double-click on _MAIN.



Property Inspector: No properties.

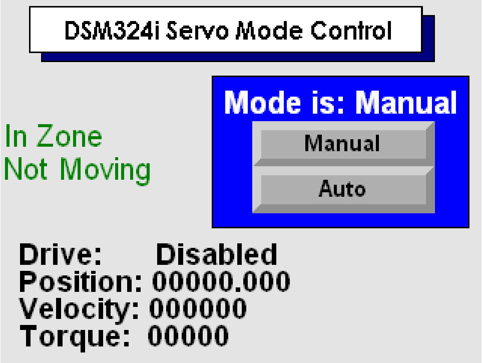
The Program Blocks folder contains all LD, ST, FBD, IL, and C blocks that are downloaded to and

```
Starting PLC
The PLC was successfully started.
```

Programmer, Run Enabled, Config EQ, Logic EQ, Sweep= 5.4 ms | Administrator | LOCAL

The yellow lighting bolt indicates ME is online with the controller. If the logic displayed were not equal to the logic in the PLC, the lighting bolt would be joined by a not equal symbol.

Procedure 2: Explore Current Application Functionality



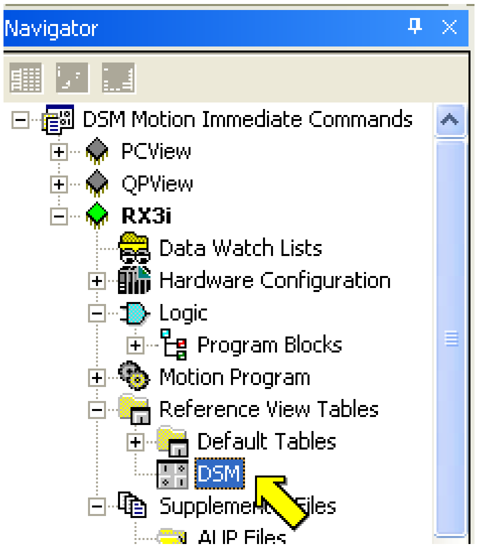
DSM324i Servo Mode Control

Mode is: Manual

In Zone
Not Moving

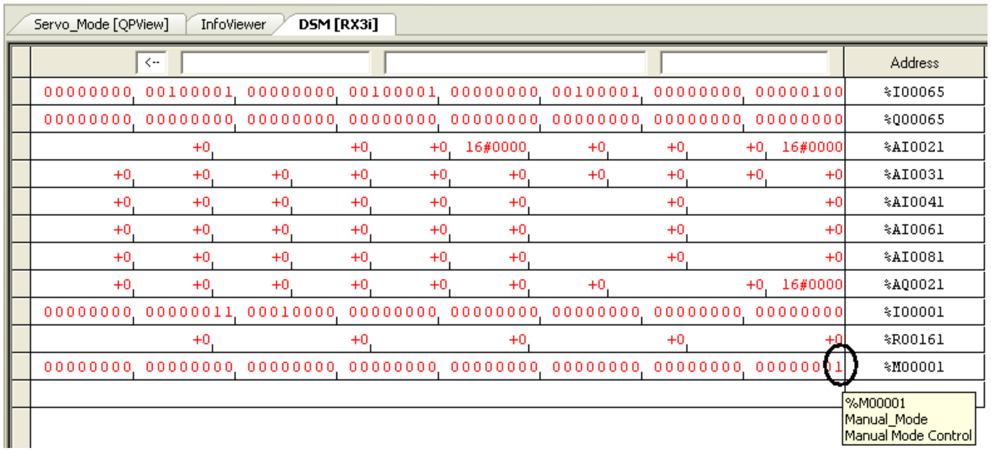
Drive: Disabled
Position: 00000.000
Velocity: 000000
Torque: 00000

1. Right-click on the QPView target in the Navigator window and choose Download and Start. When the download has completed, you should see the "DSM324i Servo Mode Control" screen on QuickPanel View.



2. Press the Auto and Manual buttons a few times to see the mode selection change. These buttons are setting or resetting a single bit in the %M area (%M00001). In order to view this, expand the Reference View Tables node in the Navigator window and double-click (or right-click and select "Open") on the predefined DSM table.

This will open a preformatted table created for this project to give a comprehensive view of all of the inputs and outputs mapped to this card by the configuration as well as other references used in the application.



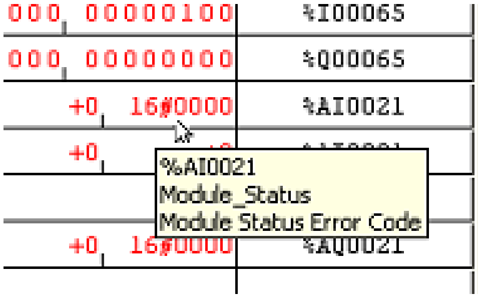
```
                     <--                                                         Address
 00000000  00100001  00000000  00100001  00000000  00100001  00000000  00000100  %I00065
 00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  %Q00065
             +0              +0      +0 16#0000      +0      +0      +0 16#0000  %AI0021
     +0      +0      +0      +0      +0      +0      +0      +0      +0      +0  %AI0031
     +0      +0      +0      +0      +0      +0              +0              +0  %AI0041
     +0      +0      +0      +0      +0      +0              +0              +0  %AI0061
     +0      +0      +0      +0      +0      +0              +0              +0  %AI0081
     +0      +0      +0      +0      +0      +0      +0              +0 16#0000  %AQ0021
 00000000  00000011  00010000  00000000  00000000  00000000  00000000  00000000  %I00001
             +0              +0              +0              +0              +0  %R00161
 00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000001  %M00001

                                                                                 %M00001
                                                                                 Manual_Mode
                                                                                 Manual Mode Control
```

TIP!

You may create as many custom Reference View Tables for each PLC target in the project as you wish. These are stored in the project folder (which can be downloaded to the PACSystem controller if desired). Each may contain a mix of any reference types as needed, in any order and each reference may be displayed in several different formats.

Notice that specific analog data has been formatted for hexadecimal (16#0000 format) or double integer display (double wide spacing) while the rest are shown in the default signed integer format. Hovering over any value in the table will display a tool tip with the address, name and description of the variable that has been assigned in the project.

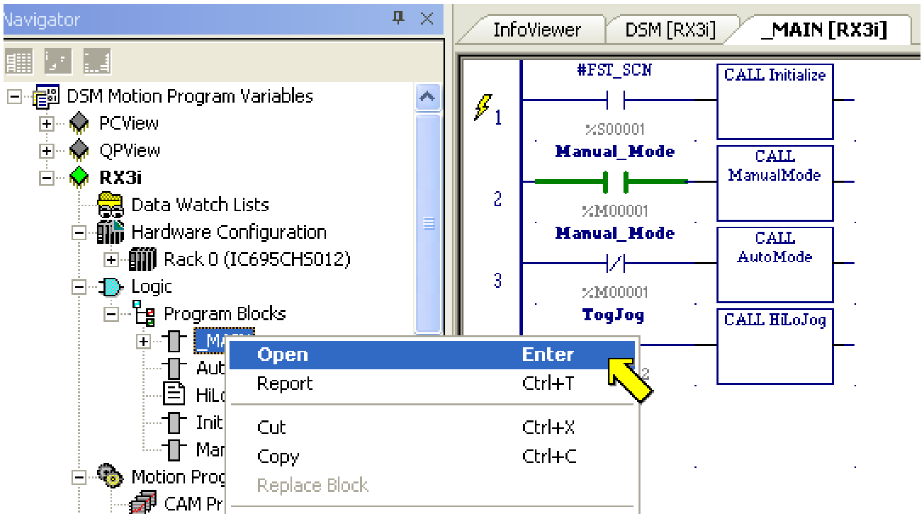


```
000  00000100     %I00065
000  00000000     %Q00065
 +0   16#0000     %AI0021
 +0
                  %AI0021
                  Module_Status
                  Module Status Error Code
 +0   16#0000     %AQ0021
```

3. Select Manual mode and try controlling the axis using the Input Simulator Module (IC694ACC300) switch inputs. The 16 switches are mapped to %I00001 - %I00016 and, if desired, may be viewed in the DSM Reference View Table. The switch functions are shown in the table to the right. The bold listings may be used with current functionality; the others will have to wait until you have added the programmed parameters functionality later in the lab. Try Jog Plus (switch 5) and Jog Minus (switch 6), try Drive Enable/Disable (switch 3) and then try the Toggle High/Low jog (switch 15). If you try Find Home (switch 4) without enabling the drive an error will be generated that you can reset using Clear Errors (switch 16).

Switch	*Function*
1	Abort All Moves
2	Feedhold
3	**Enable Drive**
4	**Find Home**
5	**Jog Plus**
6	**Jog Minus**
7	Execute Program 1
8	Load Parameters
15	**Toggle High/Low Jog**
16	**Clear Errors**

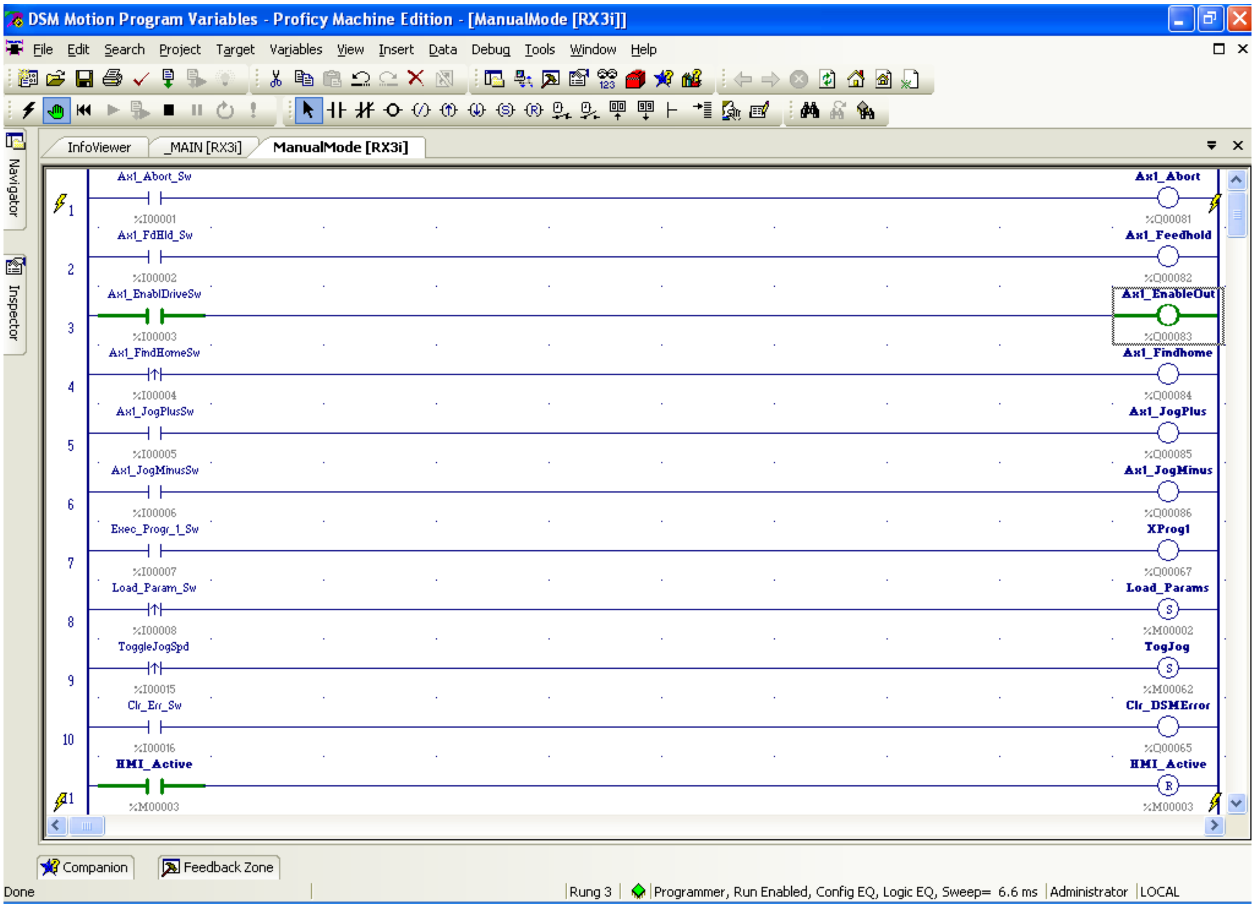
4. Now expand the RX3i target Logic/Program Blocks and double click (or right-click and choose Open) to open the _MAIN program block.



Tip!

You have a great deal of control of how the ladder work area is displayed. If you left-click somewhere in the work area (i.e. make the ladder the active window) the context-sensitive menu at the top of the workbench changes to include a View selection. You may wish to explore all of menu items available there to make your view of the logic more comfortable and informative for you. Some of these selections have hot keys (Zoom = Ctrl+/Ctrl-) or right-click selections (Adjust Cell Width).

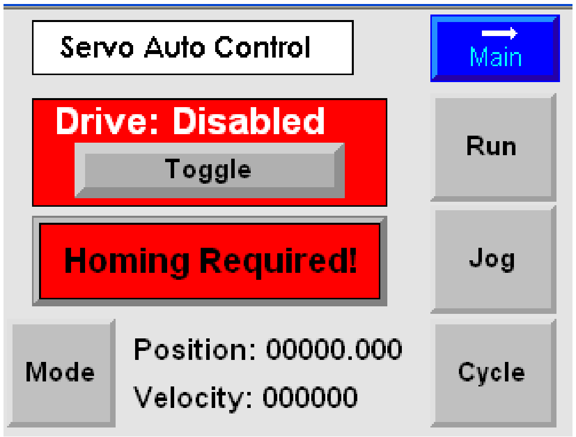
5. With Manual mode selected on the operator interface, double click on the “CALL ManualMode” block on the _MAIN (RX3i) tab to view the manual mode logic on the ManualMode (RX3i) tab. Try some Manual switches and watch as %I00003 sets Q00082, eleven rungs shown in the screenshot below constitute the entire Manual mode logic for this application.



Done | Rung 3 | Programmer, Run Enabled, Config EQ, Logic EQ, Sweep= 6.6 ms | Administrator | LOCAL

Tip!

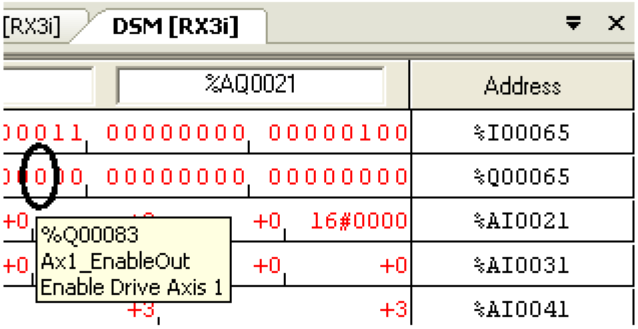
Another nice feature of the Machine Edition interface is the ability to auto-hide windows around the central workspace. Each of these windows has a small pushpin icon in the upper right corner next to the X close symbol. Left-clicking on the pushpin toggles it between vertical (auto-hide OFF) and horizontal (auto-hide ON). When auto-hide is on, that window will collapse to the side or bottom of the screen until you mouse over it again, at which time you can view and select any item just as you can when the auto-hide is off.



6. Now from the Main screen of the operator interface change to Auto mode and observe that the inputs on the ManualMode (RX3i) tab no longer control the outputs. This is because the ManualMode (RX3i) logic block is not being scanned.

7. Now select “Auto Ops” pushbutton and view the “Servo Auto Control” screen.

8. Toggle the Drive: Enabled/Disabled button as you view the %Q0083 bit turning off and on in the DSM Reference View Table.



```
           %AQ0021                          Address
00011  00000000  00000100                   %I00065
0000   00000000  00000000                   %Q00065
+0     +0     +0  16#0000                   %AI0021
+0            +0       +0                   %AI0031
       +3              +3                   %AI0041
```

%Q00083
Ax1_EnableOut
Enable Drive Axis 1

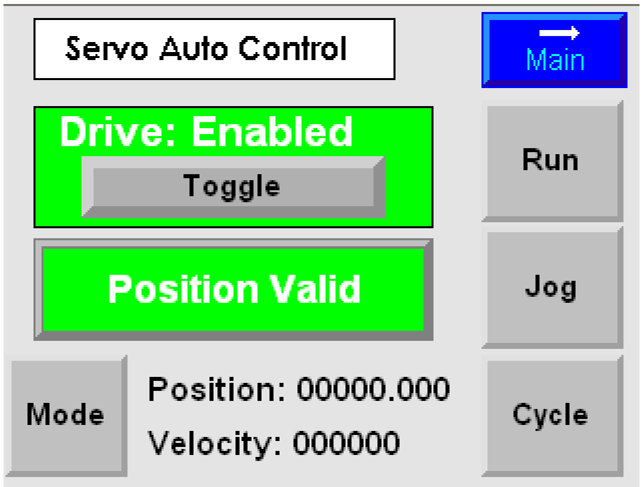
Notice that the bit immediately above %Q00083 toggles on and off also. This is the *Drive Enabled* bit (%I00083), which gives an indication that the servo really is enabled. This bit will remain off even when the *Enable Drive* output bit is on when, for instance, there is a servo fault.

The lower six bits in this byte command the functions:

- %Q00081 = Abort All Moves
- %Q00082 = Feed Hold (Pause Program)
- %Q00083 = Enable Drive
- %Q00084 = Find Home
- %Q00085 = Jog Plus
- %Q00086 = Jog Minus

All of these command bits are used in this lab, both in Manual logic that you have already seen and in operator interface screens where they are simply set momentarily by pushbuttons (except *Enable Drive* and *Feedhold*, which are toggled). The *Clear Error* bit (%Q00065) is also set momentarily by the operator interface and by input simulator switch 16 in order to reset error conditions.

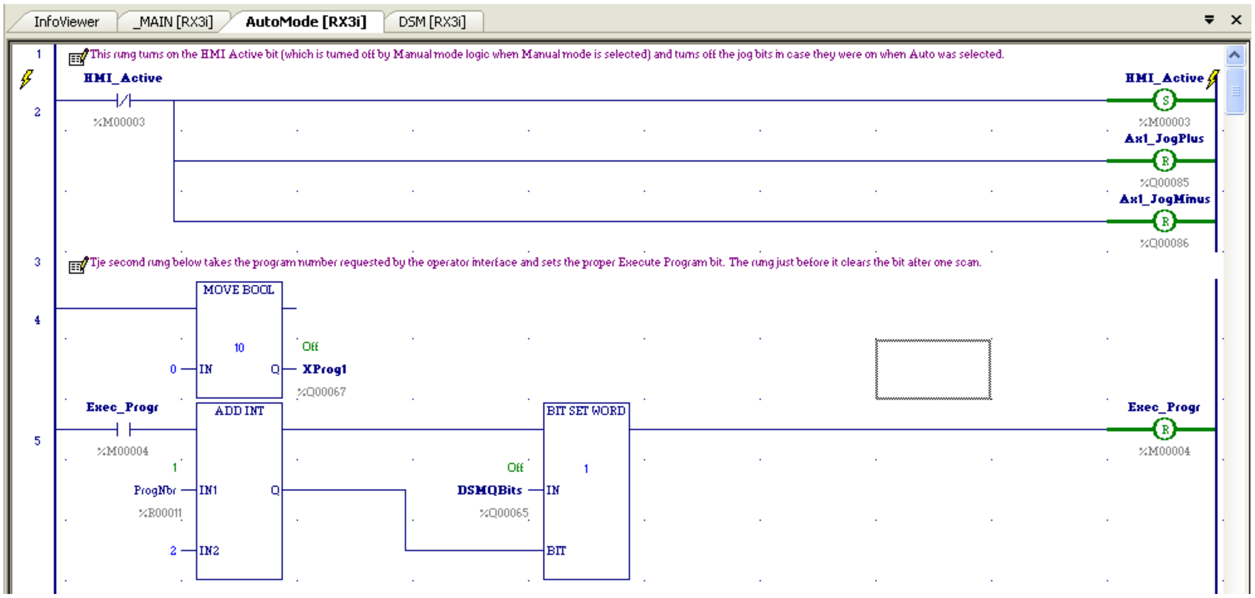
When the servo is initially powered on the *Position Valid* bit (%I00082) will be off, indicating that the servo has not completed a homing cycle or a *Set Position* command. The operator interface uses this bit to flash a "Homing Required!" message on a pushbutton. With the drive enabled, Pressing this button will start the DSM's automatic homing routine and move the motor to locate the encoder marker pulse. The DSM can also be configured to home to a switch or a switch plus the encoder marker. When homing completes, the screen will display "Position Valid", %I00082 will be on and the screen will look like this:



Position: 00000.000
Velocity: 000000

You may now feel free to try out the Run and Jog screens. These are fully functional and their operation should be self-explanatory.

9. Now click on the _MAIN (RX3i) tab and double-click on "CALL AutoMode" to view the AutoMode (RX3i) logic.



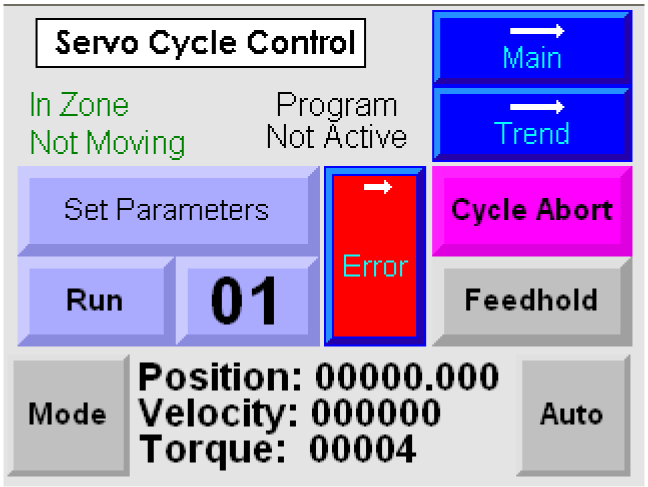
This rung turns on the HMI Active bit (which is turned off by Manual mode logic when Manual mode is selected) and turns off the jog bits in case they were on when Auto was selected.

Tje second rung below takes the program number requested by the operator interface and sets the proper Execute Program bit. The rung just before it clears the bit after one scan.

Notice that there are only three rungs of logic. The first is an initialization rung for Auto mode. The third rung triggers the proper *Execute Program X* bit when the operator interface selects one of the ten DSM stored motion programs and the second rung clears that selection after one scan. (Our application only has one stored program, so the second and third rungs would also be unnecessary if the operator interface "Execute" button set the *Execute Program 1* bit directly.)

The simplicity of this logic is due to the power of the DSM command interface. You may explore the details in the companion lab "DSM Motion Immediate Commands".

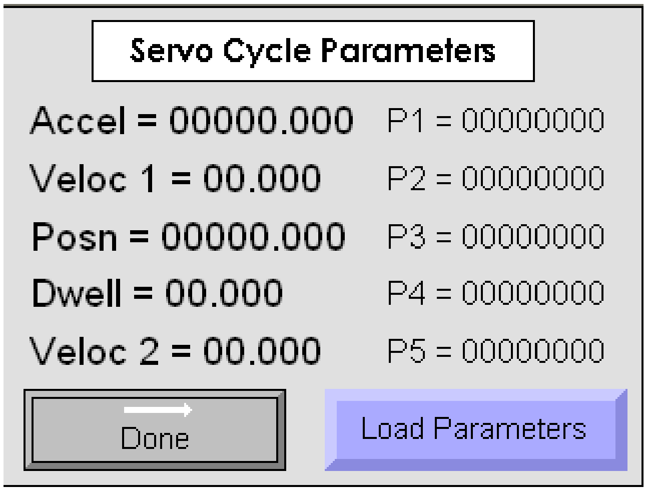
10. From the "Servo Auto Control" screen make sure the drive is enabled and "homed" (Position Valid displayed). If the position shown is not zero, press the "Position Valid" button again to re-home the servo. Now select "Cycle". The Servo Cycle Control screen will be displayed.



Position: 00000.000
Velocity: 000000
Torque: 00004

11. Now select "Set Parameters" to view the "Servo Cycle Parameters" screen.

    This screen shows the five parameters used in this lab to change motion variables. The operator interface is programmed to download these values to registers in the RX3i in preparation for transfer to DSM. The "Load Parameters" button sets a bit (which is also set by Switch 8 when Manual mode is active) to signal the RX3i logic program to pass these parameter values to the DSM.



Servo Cycle Parameters

Accel = 00000.000 P1 = 00000000

Veloc 1 = 00.000 P2 = 00000000

Posn = 00000.000 P3 = 00000000

Dwell = 00.000 P4 = 00000000

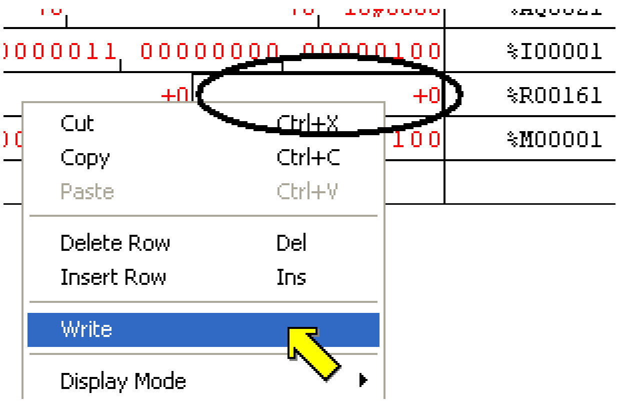
Veloc 2 = 00.000 P5 = 00000000

12. Before we add the logic required to pass parameter values to the DSM, we will perform a manual download in order to see what actually needs to take place in the logic.

    The values displayed in the left column on the Servo Cycle Parameters screen are tied to memory register locations visible in DSM Reference View Table at %R00161/162 through %R00169/170. These are the five double integer parameters we will pass to the DSM.

    Select %R00161/62 in the DSM RVT and enter a value. (Left-click to select and begin entering a number or right-click and select "Write" from the menu)

    Verify that the number you entered appears next to the Accel label on the operator interface.



```
)0000011  00000000  00000100   %I00001
      +0              +0       %R00161
)0                   100       %M00001
```

13. Now enter a number for Accel from the operator interface by touching a number in the left column and entering a value. (If you enter a number that is too big for the display it will show ##### instead of the number.)

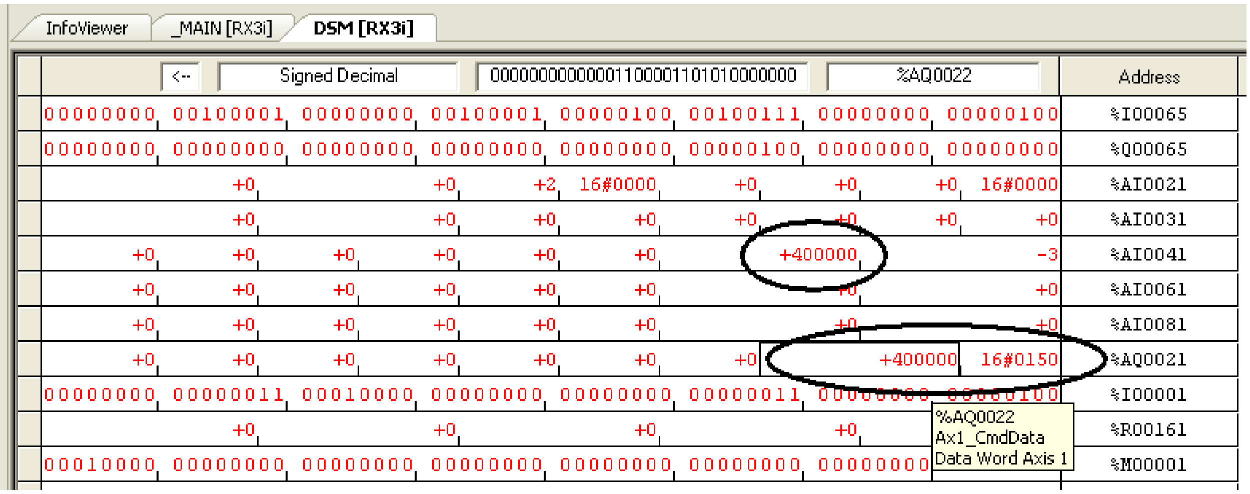
    Verify that the number entered on the display appears in the Reference View Table. This is a simple operator interface function requiring no logic in the PLC.

14. Now we will write a parameter value into the DSM manually through the Reference View Table by using an immediate command. The command number for Load Parameter Immediate is 50h (hexadecimal). The upper byte of the command number must contain the number of the parameter to be written to. So, for our application, we must write values with the following command numbers:

    - Parameter 1 = 150h
    - Parameter 2 = 250h
    - Parameter 3 = 350h
    - Parameter 4 = 450h
    - Parameter 5 = 550h

The command word interface for axis 1 consists of a command word at %AQ0021 and command data at %AQ0022/23. Notice that the RVT formatting for %AQ0021 is hexadecimal and the formatting for %AQ0022/23 is double integer. This makes our task very easy. Select %AQ0021 and enter 150h. Select %AQ0022/23 and enter 400000.

Your RVT should now look like this:



```
          <--    Signed Decimal     00000000000001100001101010000000    %AQ0022          Address
00000000 00100001 00000000 00100001 00000100 00100111 00000000 00000100   %I00065
00000000 00000000 00000000 00000000 00000000 00000100 00000000 00000000   %Q00065
                +0                +0       +2  16#0000      +0      +0      +0  16#0000   %AI0021
                +0                +0       +0       +0      +0      +0      +0       +0   %AI0031
       +0       +0       +0       +0       +0       +0         +400000               -3   %AI0041
       +0       +0       +0       +0       +0       +0              +0               +0   %AI0061
       +0       +0       +0       +0       +0       +0              +0               +0   %AI0081
       +0       +0       +0       +0       +0       +0      +0         +400000  16#0150   %AQ0021
00000000 00000011 00010000 00000000 00000000 00000011 00000000 00000100   %I00001
                +0                +0                +0              +0  %AQ0022            %R00161
00010000 00000000 00000000 00000000 00000000 00000000 00000000  Ax1_CmdData         %M00001
                                                                Data Word Axis 1
```

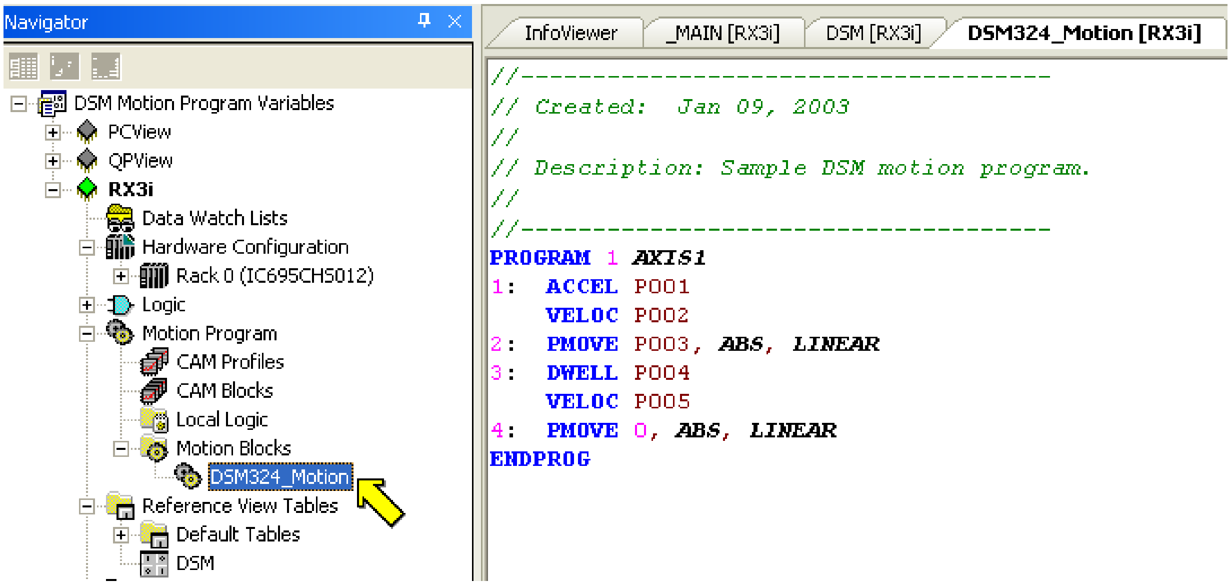
Notice that the value passed to parameter 1 shows up in %AI0043/44 and on the operator interface screen beside the P1 label. The parameter 1 value has been configured in the DSM to write to that address and the operator interface screen simply displays the value stored in this memory location. Similarly, the other four parameter values are also configured for continuous display in the Reference View Tables and on this operator interface screen.

Now try writing a different value into %AQ0022/23. Notice that it is immediately loaded to the DSM and fed back though %AI0043/44. This illustrates a very important consideration that is critical to successful loading of the parameters through the PLC program logic: **the DSM sees any change to the Command Word OR to the Command Data as a new command.** Therefore it is good programming practice to always write the command and data in the same PLC sweep. When entering commands in a Reference View Table, you should always enter data while the command word is zero (null command) and then enter the command word only **<u>after</u>** the data is already in place.

TIP!

It is very important to realize that DSM sees any change to the Command Word OR to the Command Data as a new command. Therefore it is recommended programming practice to always write the command and command data in the same PLC sweep.

15. Expand the RX3i node in the Navigator windows to Motion Program/Motion Blocks and double-click on “DSM324_Motion”.



```
//---------------------------------------
// Created:  Jan 09, 2003
//
// Description: Sample DSM motion program.
//
//---------------------------------------
PROGRAM 1 AXIS1
1:  ACCEL P001
    VELOC P002
2:  PMOVE P003, ABS, LINEAR
3:  DWELL P004
    VELOC P005
4:  PMOVE 0, ABS, LINEAR
ENDPROG
```

This is the Motion Program file that is downloaded to the DSM each time power cycles or whenever a PLC configuration is downloaded. In this lab we have only one motion program in the file, but there could be up to ten motion programs and 40 subroutines for each DSM module.

This program contains two moves. The first move is an absolute position move (PMOVE) to a position specified in parameter 3 (P003) using the acceleration specified in parameter 1 (P001) and velocity specified in parameter 2 (P002). Then there is a time delay (DWELL) in milliseconds specified by parameter 4 (P004). Finally, the second PMOVE returns the servo to absolute position zero at a velocity specified by parameter 5 (P005).

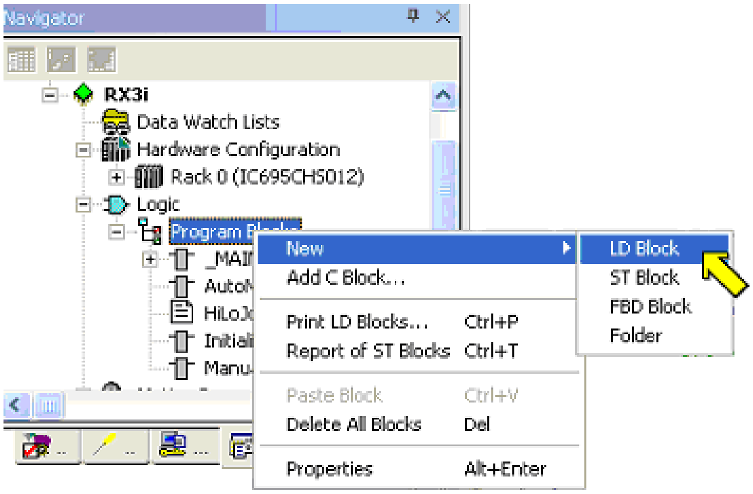
NOTE

This demo is configured for 1000 user units = 1 turn of the motor. Therefore, velocity units are 0.001 rev/second and acceleration units are 0.001 rev/second/second. Maximum velocity of the servo in the demo system is about 90000 = 90 rev/sec = 5400 rpm.

16. (Optional Step) Download the parameter values shown in the table below to DSM manually and then manually run program 1. You will have to trigger the move by using Run button on the “Servo Cycle Control” screen or by setting Manual mode and using switch 7 on the Input Simulator module (IC694ACC300).

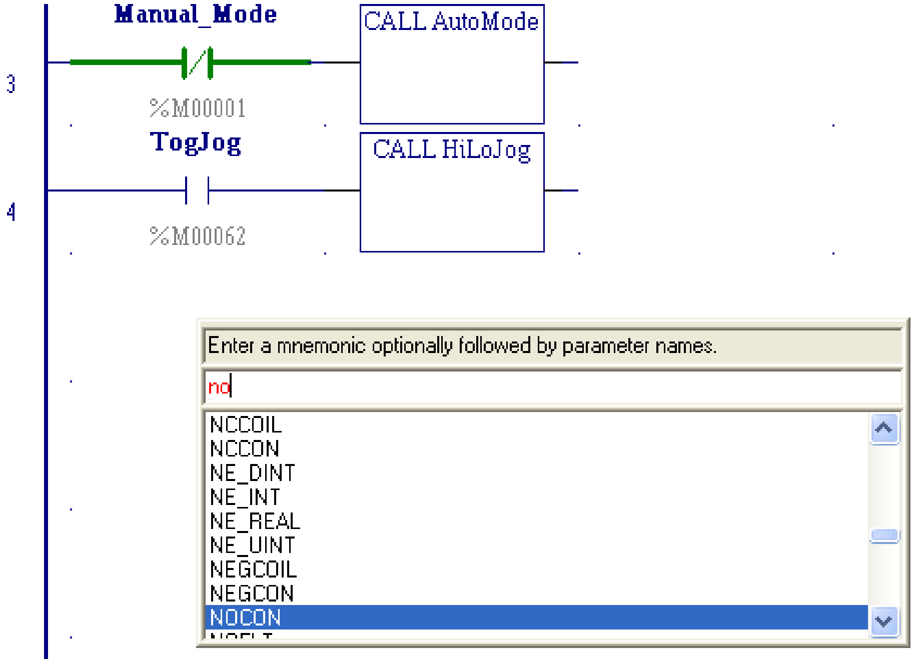
Command in %AQ0021	Data in %AQ0022/23
150	400000
250	40000
350	500000
450	2000
550	90000

Procedure 3 (Optional): Add Parameter Download Logic



1. Now that you have exercised all of the functionality of the application, it is time to add the parameter download logic. We start by creating a new logic block named "LoadParam". In the Navigator expand the RX3i node, right-click the Logic/Program Blocks and select New/LD Block. (The logic could also be created in Structured Text or Function Block Diagram, but ladder is well-suited to the task.)

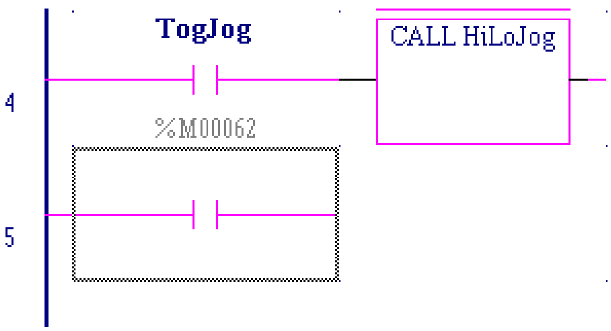
2. After naming the block, we want to call it from the _MAIN program whenever the bit Load_Params (%M00002) is set by the Manual logic or by the operator interface. To do this, left-click in the space directly under the TogJog contact in _MAIN. Press "Enter" and type in "no".

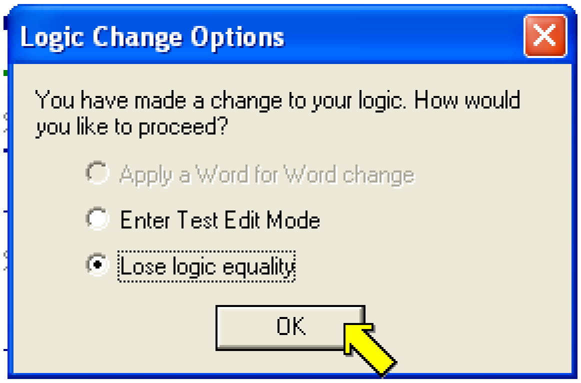


Enter a mnemonic optionally followed by parameter names.

no

As you can see, this selects the normally open contact (NOCON) instruction. Press “Enter” to place this instruction in a new rung connected to the left rail.




Logic Change Options

You have made a change to your logic. How would you like to proceed?

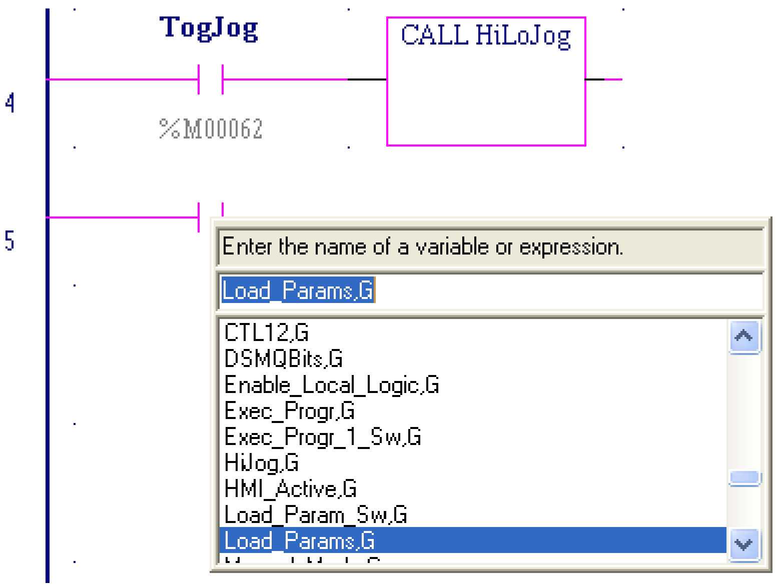
○ Apply a Word for Word change

○ Enter Test Edit Mode

● Lose logic equality

If you are online, you will then get a “Logic Changes Options” dialog that offers the options of “Enter Test Edit Mode” and “Lose logic equality”. Test Edit is a feature of PACSystems controllers that allows the original logic plus a block with modified logic to reside in the controller at the same time and allows you to toggle between the two for debug/test purposes. In this instance it will probably be easier to lose logic equality.

3. You now need to add the name or the address of the variable to be associated with the contact, in this case, Load_Params. With the NO contact still selected type Load_Params. The variable list auto-fill feature will immediately present a list of defined variables matching this description and you can cursor down to pick it from the list.

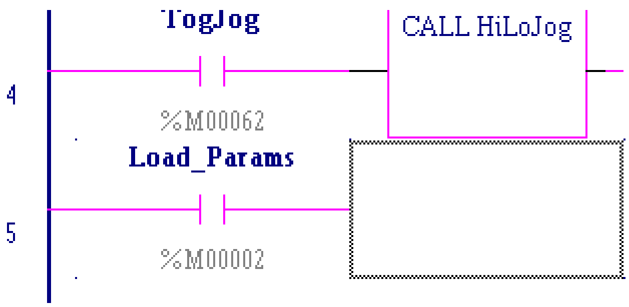


Enter the name of a variable or expression.

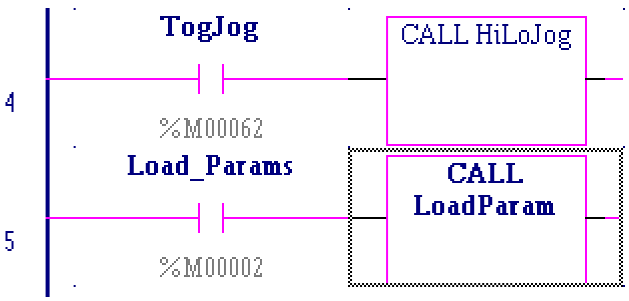
Load_Params,G

CTL12,G
DSMQBits,G
Enable_Local_Logic,G
Exec_Progr,G
Exec_Progr_1_Sw,G
HiJog,G
HMI_Active,G
Load_Param_Sw,G
Load_Params,G

4. Once the proper variable appears in the upper selection box, press “Enter”. The variable will be selected and the cursor box will move to the right waiting for a new instruction to be keyed in.



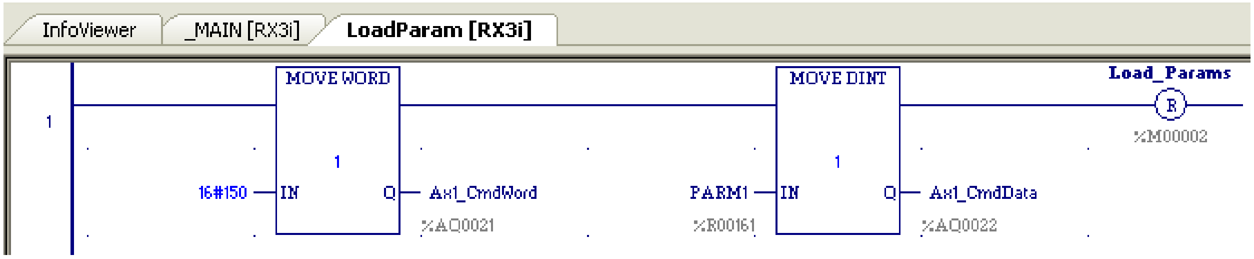
5. Press “c” (to select the CALL instruction) and then “Enter”. Then press “L” (to select “LoadParam” from the list of blocks) and “Enter”.



6. When this logic is downloaded, the controller will scan the LoadParam block whenever %M0002 is on. Existing logic turns this bit on. Logic we will add in the block must turn it off after loading the parameters to DSM.

   We begin by opening the block. Double-click on the CALL instruction.

7. Lets begin with logic that performs a single parameter download. As we saw in manually writing the values to the Reference View Table, this is very simple. We write a command number and a data value. After you have done that, reset the %M00002 bit. The logic will look like this:



There are a number of ways to place these logic elements – keyboard only, mouse only or a combination of the two. You can actually place the cursor at the upper left corner and enter the entire rung in one string by typing “move_word 1 16#150 21aq;move_dint 1 parm1 22aq;resetcoil 2m”. It is generally easier, however, to use the instruction auto-complete pick list and the variable pick list as illustrated above.

In this case, perform the following steps: (Do not type the quotes in the instructions.)

- Place cursor in upper left corner at rung 1.
- Type “mov”, cursor down to MOVE_WORD and press “Enter”.
- Cursor left and down to the IN parameter location.
- Type “16#150” and press “Enter”. Cursor jumps to the Q parameter location.
- Type “21aq” and press “Enter”. Cursor jumps to the rung continuation point.
- Type “mov”, cursor down to MOVE_DINT and press “Enter”.
- Cursor left and down to the IN parameter location.
- Type “parm1” and press “Enter”. Cursor jumps to the Q parameter location.
- Type “22aq” and press “Enter”. Cursor jumps to the rung continuation point.
- Type “res” and press “Enter”.
- Type “2m” and press “Enter”.

8. Download the logic. Once the download is complete select the “Servo Cycle Parameters” screen and enter a value for the Accel data. Press the “Load Parameters” button and confirm that the Accel value you entered appears as the P1 parameter.

9. Now we need to add logic to load the other parameters. We can only load one parameter per scan for each of the configured axes (up to four) on the DSM module. Therefore, we will send the five parameters in five successive scans. Begin by modifying the rung as follows:

Operation	**Keystrokes**
Replace the reset coil with a SETCOIL.	Right-click/Replace Instruction (or left-click and Shift-Enter). Type "set" and press "Enter".
Replace the coil %M00002 variable with a new variable we'll call "P1Loaded" and assign it to the first available %T reference.	With the SETCOIL selected, type "P1Loaded,%t" and press "Enter".
Prevent this operation from happening on subsequent scans by placing a P1Loaded NCCON (normally closed contact) at the beginning of the rung.	Select the cell at the beginning of the rung 2, type "nc" and cursor down to NCCON. Press "Enter". Type "p" and cursor down to "P1Loaded". Press "Enter".

10. Now copy rung 1 to rung 2. As with most Windows interfaces there are multiple ways to do this.

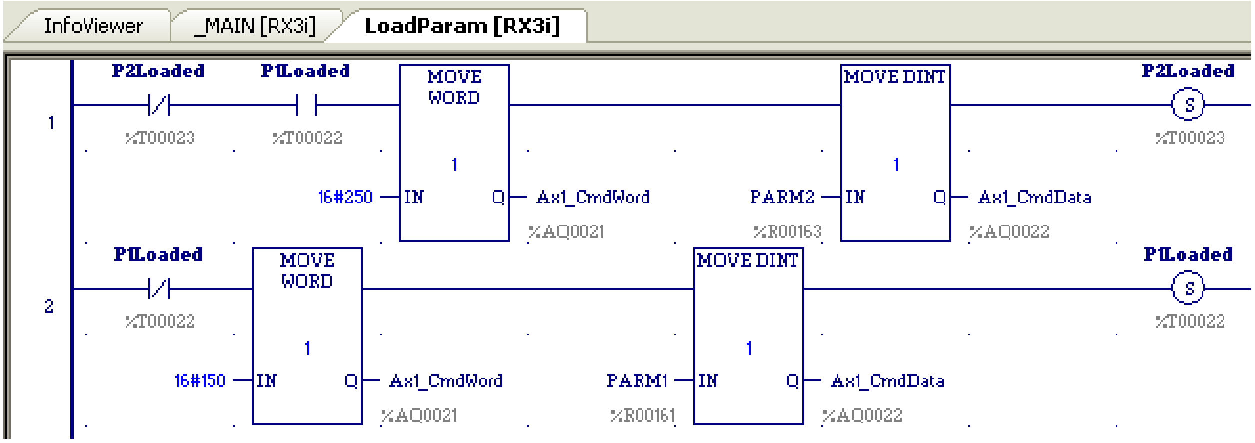
- You can left-click to the left of the rung number to select the rung, right-click/copy or Ctrl-C copy, left-click anywhere below rung 1 and right-click/paste or Ctrl-V paste.
- Or, you can left-click to the left of the rung number to select the rung then select the highlighted rung to the right of the left rail and pull down to the next rung location while holding the Ctrl key down.

Either way you end up with a duplicate rung at rung 2.

11. Modify rung 1 as follows:

Operation	Keystrokes
Replace the coil P1Loaded variable with a new variable we'll call "P1Loaded" and assign it to the next %T reference.	With the SETCOIL selected, type "P2Loaded,%t" and press "Enter".
Replace P1Loaded variable on NCCON with P2Loaded.	With NCCON selected, type "p2", cursor down to P2Loaded and press "Enter".
Move the two MOVE instructions to the right to make room for another contact.	Left-click and drag from MOVE_WORD to MOVE_DINT instruction until both are highlighted. Then select highlighted area and drag to the right one cell width.
Insert NOCON between P2Loaded NC and MOVE_WORD. Assign P1Loaded to the contact.	Select blank space to the right of P2Loaded contact, type "no" and press "Enter". Type "p", cursor down to P1Loaded and press "Enter".
Change MOVE_WORD IN parameter to 16#250 and MOVE_DINT input parameter to PARM2.	Select the constant 16#150, type "16#250" and press "Enter". Select PARM1, type "p", cursor down to PARM2 and press "Enter".

The logic should now look like this:



As it now stands, rung 2 will execute the first scan when the block is called. On the second scan, rung 1 will execute. On a third scan, neither would execute.

12. Now we want to duplicate rung 1 three times and insert above rung 1. To do this highlight rung 1 and press Ctrl-C to copy the rung. Then press Ctrl-V repeatedly until you have four identical rungs at the top of the logic. Change these as you did rung 1 in the previous step.

    On Rung 3:

    - Change the coil and NC contact variable from P2Loaded to P3Loaded, assigning P3Loaded to the next %T reference.
    - Change P1Loaded contact to P2Loaded.
    - Change constant 16#250 to 16#350 and PARM2 to PARM3.

    On Rung 2:

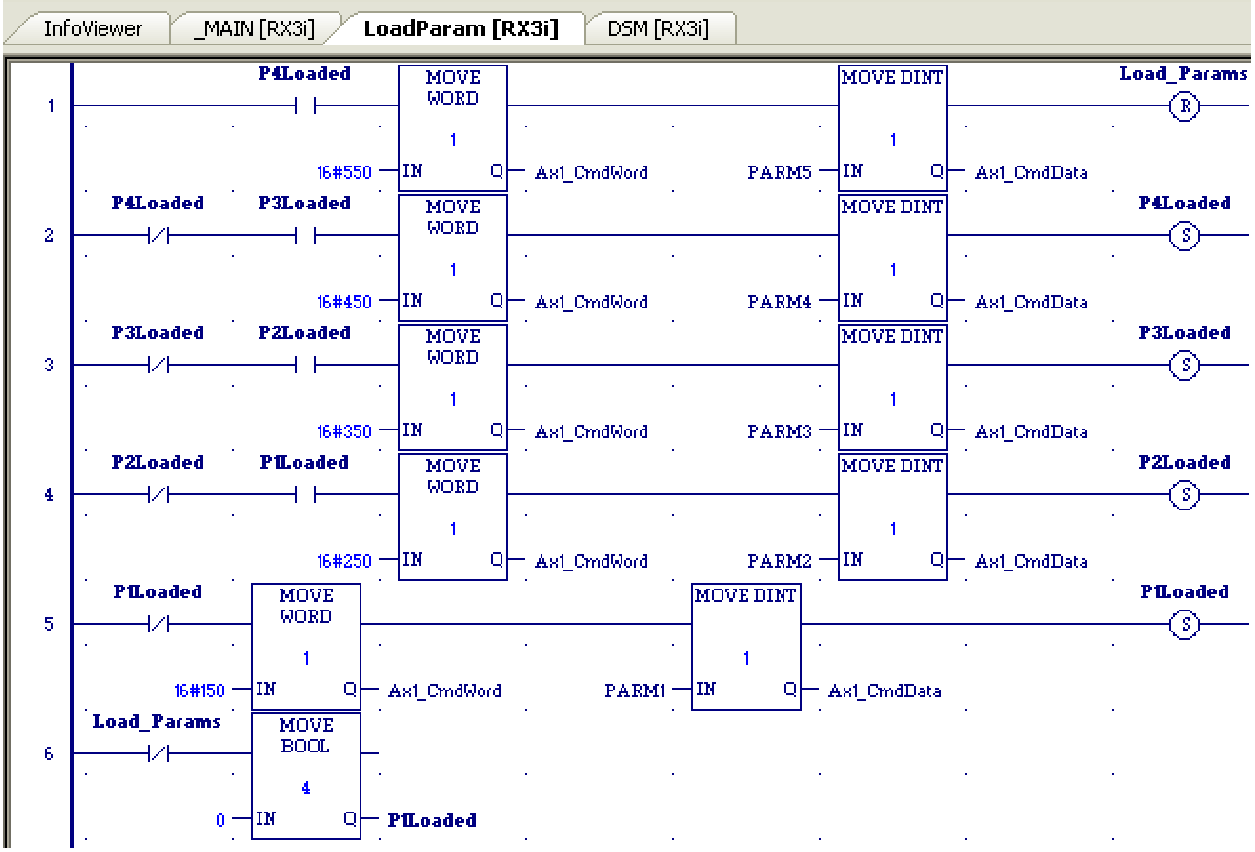
    - Change the coil and NC contact variable from P2Loaded to P4Loaded, assigning P4Loaded to the next %T reference.
    - Change P1Loaded contact to P3Loaded.
    - Change constant 16#250 to 16#450 and PARM2 to PARM4.

13. Rung 1 will be the last of the data load rungs to execute and will be slightly different than the others.

    - Change from a SETCOIL to a RESETCOIL with reference %M00002, just as in the original one parameter logic. When this rung executes, the load routine is over and the bit that calls the block must be reset.
    - Delete P2Loaded normally closed contact at the beginning of the rung. Replace with a horizontal line. The easiest way to do this is to highlight the contact and press "Delete" and hold down the Ctrl key and cursor to the right.
    - Change P1Loaded contact to P4Loaded.
    - Change constant 16#250 to 16#550 and PARM2 to PARM5.

    This top rung will execute last on the last scan. It will load parameter 5 from %R00169/70 to the DSM and reset the Load_Params variable (%M00002) so that the block does not execute on the next scan.

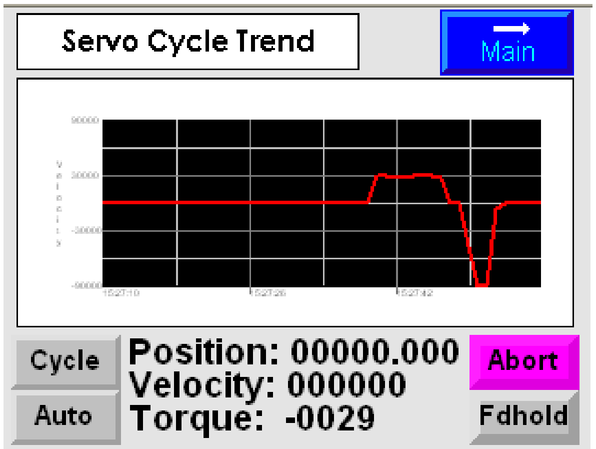
14. Now we have to add one more rung to the bottom of the logic that will clear the %T bits that have guided the scan-by-scan loading of the parameters. This rung will execute on the last scan after rung 1 resets Load_Params. Place the cursor below rung 5 at the left rail. Type "nc" and press "Enter". Type "2m" and press "Enter". Type "mov", cursor down to MOVE_BOOL and press "Enter". Type "4" and press "Enter". Cursor left and down to the IN parameter location, type "0" and press "Enter". The cursor will jump to the Q parameter location. Type "p", cursor down to "P1Loaded" and press "Enter". The logic will now look like this:



15. Download and verify the operation. Through the operator interface, enter Accel = 400, Veloc 1 = 30, Posn = 250, Dwell = 2 and Veloc 2 = 90. Press the "Load Parameters" button and verify that all parameters transfer and are displayed as entered into P1 through P5 on the right side of the screen.

16. Press "Done" to return to the "Servo Cycle Control" screen. Press "Run" and watch the trend as the move executes.

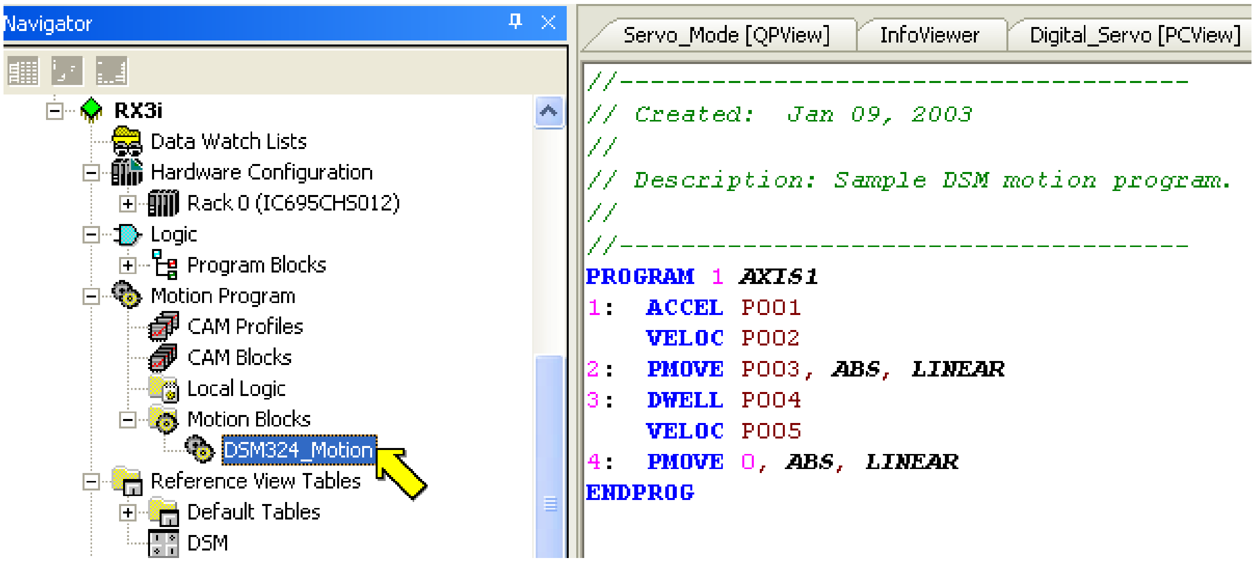
17. Return to the Cycle screen and press "Set Parameters" again. Change some of the parameters, load them and execute another cycle, verifying that the move did change in accordance with the new data. You should see move trends similar to this:


Servo Cycle Trend

Position: 00000.000
Velocity: 000000
Torque: -0029

Procedure 4 (Optional): Modify Stored Motion Program

1. Now we will explore how to actually change the motion program sequence. In this procedure we will add another step to the motion program sequence. First, open the motion program file DSM_Motion by expanding the RX3i node in Navigator window, expanding Motion Program/Motion Blocks and double- licking on the file name. This will open the following editor window:

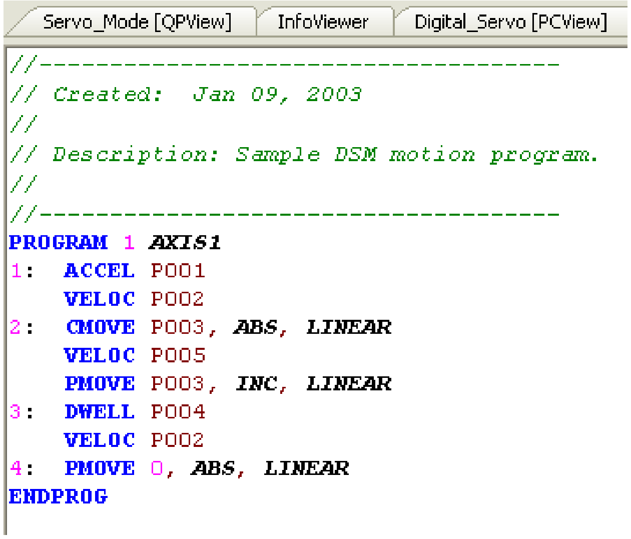


```
//---------------------------------------
// Created:  Jan 09, 2003
//
// Description: Sample DSM motion program.
//
//---------------------------------------
PROGRAM 1 AXIS1
1:  ACCEL P001
    VELOC P002
2:  PMOVE P003, ABS, LINEAR
3:  DWELL P004
    VELOC P005
4:  PMOVE 0, ABS, LINEAR
ENDPROG
```

2. We will change the move at block 2 to a blended move – Velocity P002 for the first part of the move and Velocity P005 for the last part. To do this:

   - Copy the line that begins with “2:” and paste/insert it below the same line.
   - Delete the “2:” from the beginning of the second copy of the PMOVE line.
   - Change the first PMOVE to a CMOVE.
   - Change the copied PMOVE from an “ABS” move to an “INC” move.
   - Copy the VELOC P005 and insert it between the CMOVE line and the first PMOVE line.
   - Change the VELOC P005 just before the last PMOVE to VELOC P002.

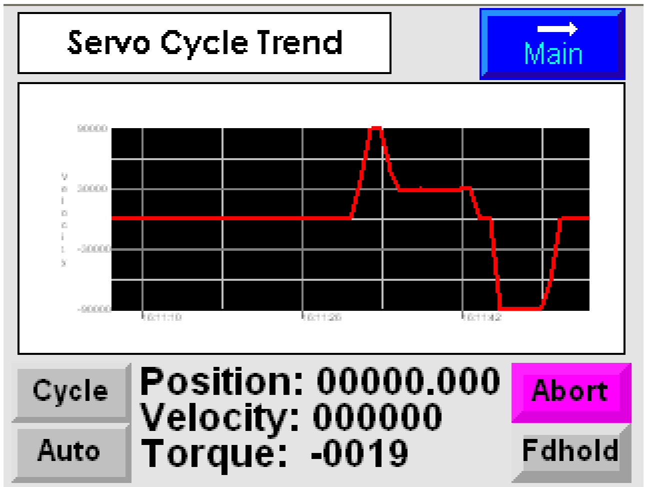
   The program should now look like this:



```
//---------------------------------------
// Created:  Jan 09, 2003
//
// Description: Sample DSM motion program.
//
//---------------------------------------
PROGRAM 1 AXIS1
1:  ACCEL P001
    VELOC P002
2:  CMOVE P003, ABS, LINEAR
    VELOC P005
    PMOVE P003, INC, LINEAR
3:  DWELL P004
    VELOC P002
4:  PMOVE 0, ABS, LINEAR
ENDPROG
```

   The program now commands a move at velocity P002 that reaches velocity P005 at position P003, continues until it moves an additional P003 distance at velocity P005, dwells P004 msec. and then returns to absolute position zero at velocity P002.

3. In order to download the modified stored program we must stop the CPU (since it is part of the configuration of the module). We do this by the same steps as in Procedure 1 above.
4. Try loading P001 = 400, P002 = 90, P003 = 250, P004 = 2, and P005 = 30. Run the program again and note the change in the velocity trend.



Position: 00000.000
Velocity: 000000
Torque: -0019

Review

This lab has demonstrated:

- The minimal logic required for simple manual control of a DSM servo axis.
- The even more minimal logic required for operator interface control of a DSM axis.
- How to create logic required for download of parameters to DSM.
- How to make changes to a DSM stored motion program.