LAB: Demonstrating Kinetix Motion

Key Topics Covered in this Lab:

In this lab, you will learn how to easily demonstrate the Kinetix™ Advantage for motion control to your customers. You can show them how in about 5 minutes, they can have their motion control system completely configured, downloaded and running with just one simple tool; RSLogix 5000™. Key topics covered include:

- Understanding the RSLogix 5000 controller organizer.
- Enabling Kinetix motion in your Logix™ system.
- Configuring and commissioning your Kinetix motion system.

Required Equipment & Software

- RSLogix 5000 (V9 or greater). V16 is used in this revision.
- ControlLogix SERCOS demo unit (1796-CL11).

Let's begin. If you have ever been responsible for commissioning a control system that included motion, you may remember that it was not easy. In fact, you probably had to deal with a logic controller, the motion controller, a network to make these two separate items communicate, several servo drive amplifiers, and lots of wiring and cables. Then you had to learn to use a separate software package for each component, all fighting over the same serial port on your PC. Most people agree that even a simple, low-axis count motion system takes about 1-2 weeks to properly commission out of the box.

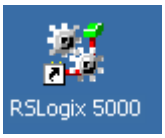
Kinetix from Rockwell Automation is the solution you've been looking for. Kinetix integrated motion is a combination of architecture, world class Allen-Bradley motion products, and motion application expertise. It's ideal for applications in packaging, material handling, assembly, converting, and many others. With its broad range of control instructions, multi-tasking operating system, multi-processing support, and high-performance SERCOS interface motion module, Kinetix meets the most demanding sequential and motion control needs.

Powerful yet easy to use RSLogix 5000 is the only software necessary for both motion and sequential control applications, eliminating the need for multiple programming tools. Just add and configure motion axes and drives using the wizard-based configuration tools, insert any of the 38 motion instructions in the ladder application program, and create simple or complex motion profiles with the advanced graphical motion profile editor. RSLogix 5000 provides the power you need in an easy to use format.

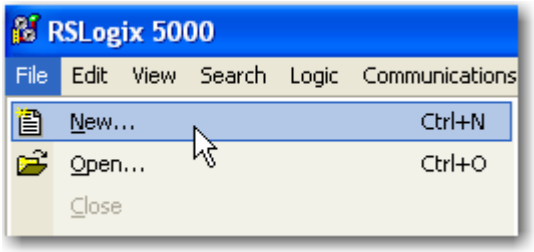
Let's begin this exercise. Here is summary of the steps that you will take to easily demonstrate motion control in the RSLogix 5000 environment:

- Open RSLogix 5000 and create a new application.
- Drop your processor in the correct slot and select your software version.
- Give the file a name and save it.
- Briefly discuss the controller organizer and its features.
- Enable motion by checking the CST box.
- Create your motion group.
- Drop your SERCOS motion module in the correct slot; leave as default.
- Drop your servo drive with the correct node address.
- Create your axis of motion and associate it with the drive.
- Select your servo motor from the list.
- Drag the axis into the motion group.
- Download to your controller.
- Right-click on the axis to begin Motion Direct Commands.
- Enable and jog the axis.

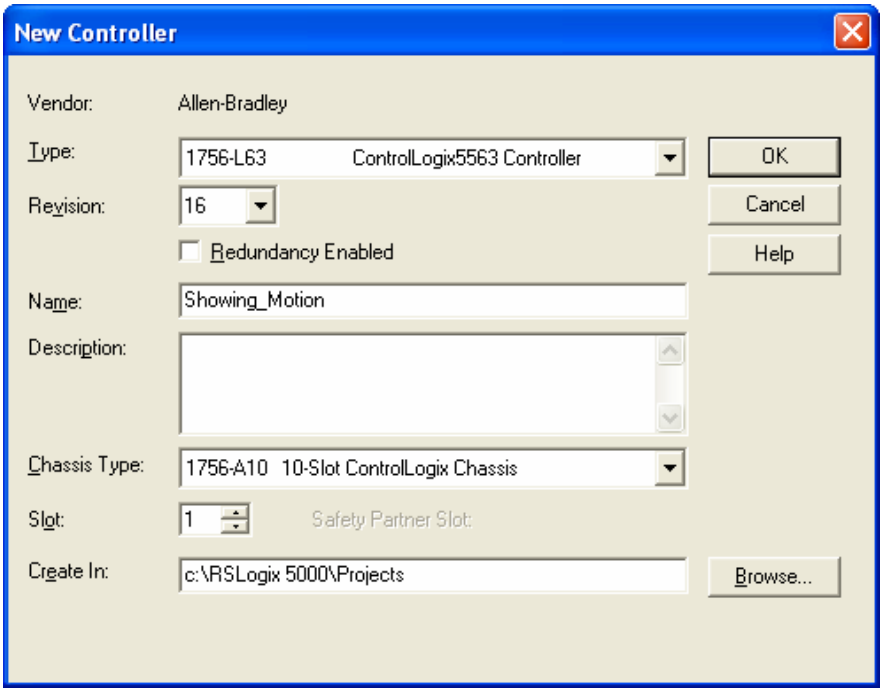
1. Start RSLogix 5000 software either by double-clicking on the **RSLogix 5000** shortcut that may be on your desktop or by navigating through the Windows command tree to **Start → Programs → Rockwell Software → RSLogix 5000 Enterprise Series → RSLogix 5000**.



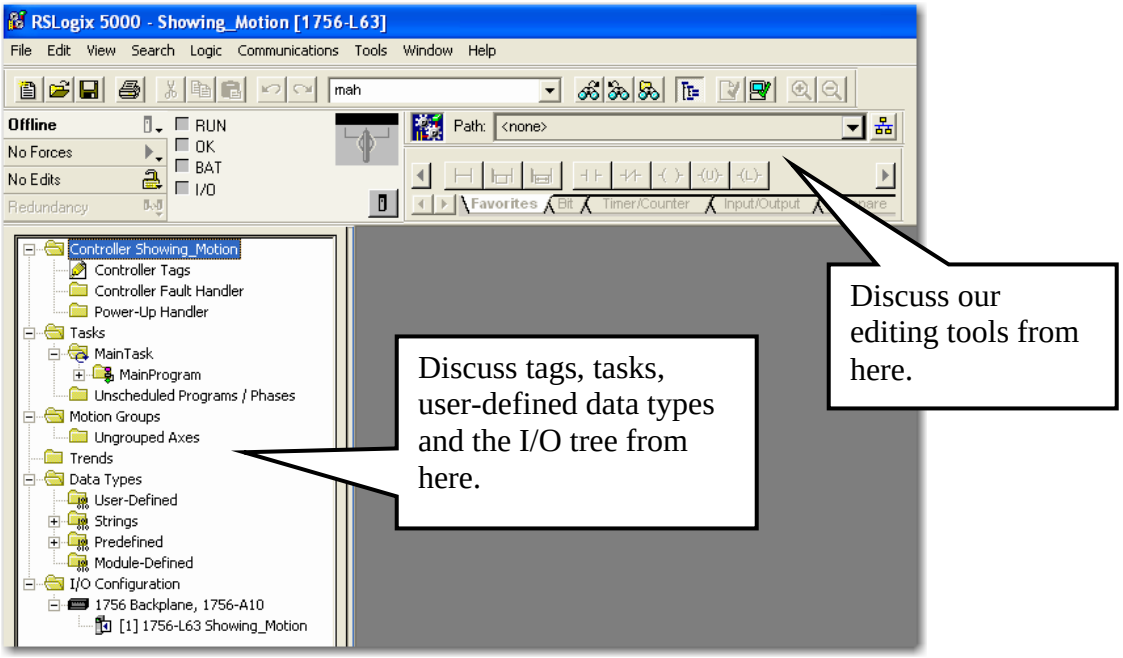
2. Select **File → New.**



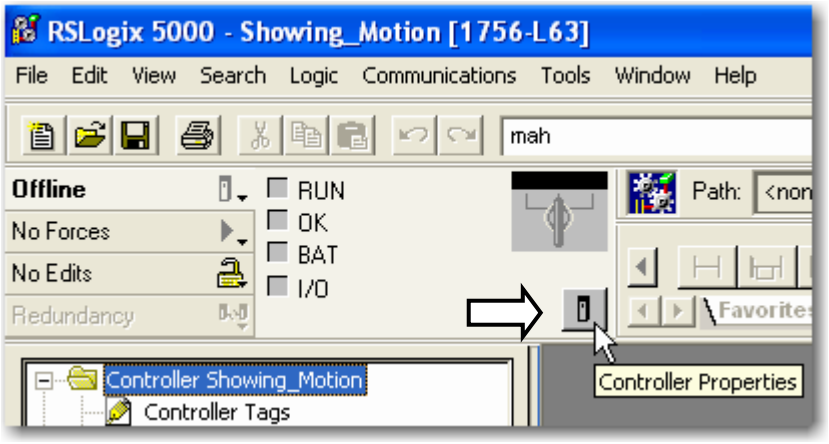
3. Select the **processor type** that you have in your demo unit, select **software revision** that you currently have, give the file a **name** and select the proper **slot address** for your processor. Then **press OK** to accept. **(For our exercise choose 1768-L43 with using your software version. Chassis slot will be blank.)**



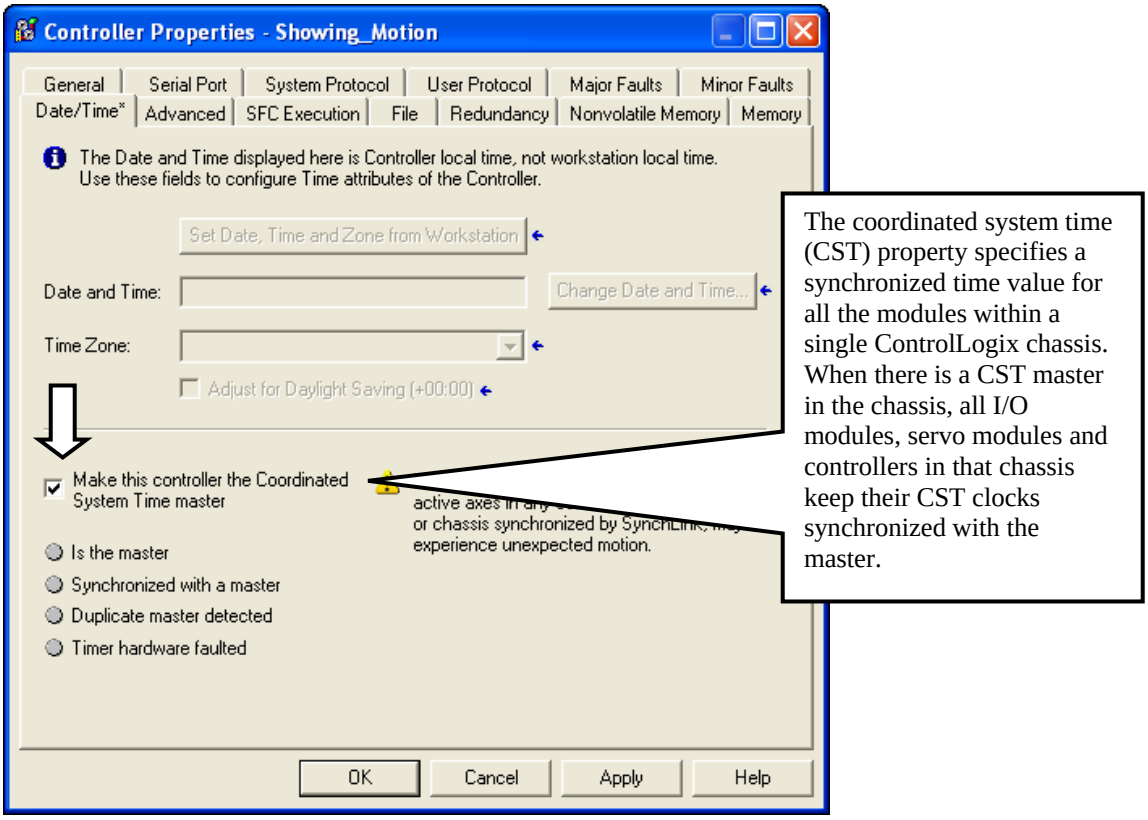
4. From here, based on your experience and knowledge of the RSLogix 5000 environment, you may to discuss the aspects of the controller organizer, troubleshooting windows editing tools.



5. Open the **controller properties** screen by clicking on the controller icon.

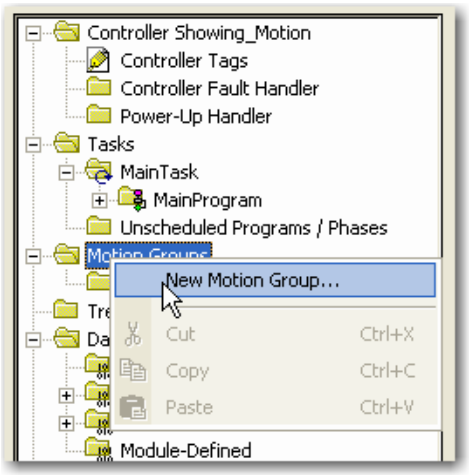


6. To enable motion in your system, select the **Date/Time tab** and check the box for **"Make the controller the Coordinated System Time master."** This allows synchronizing between axes. **Press OK**.

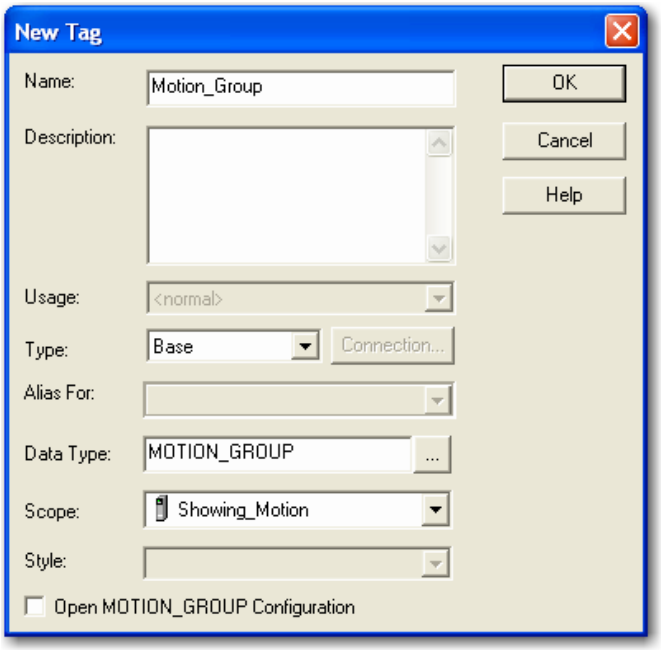


7. Another requirement for motion control is a **Motion Group**. A Motion Group is a collection of all of the active axes in your system. Inactive axes are kept outside of the group in the Ungrouped Axes folder. Simply drag axes in and out of the Motion Group offline to activate and deactivate axes. You can also issue Motion Group commands in your program that affect all active axes. Instead of programming individual Stop commands for each axis, you can program one Motion Group Stop command to stop them all at once with one step. See your Instruction Help for more details and more group commands.

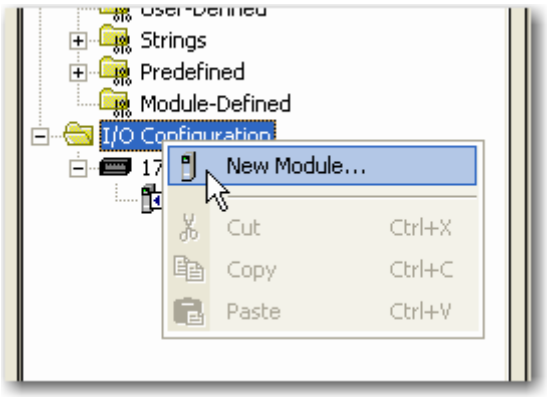
8. Let's create our Motion Group. Right-click on the **Motion Group folder** in the controller organizer and choose **New Motion Group**.



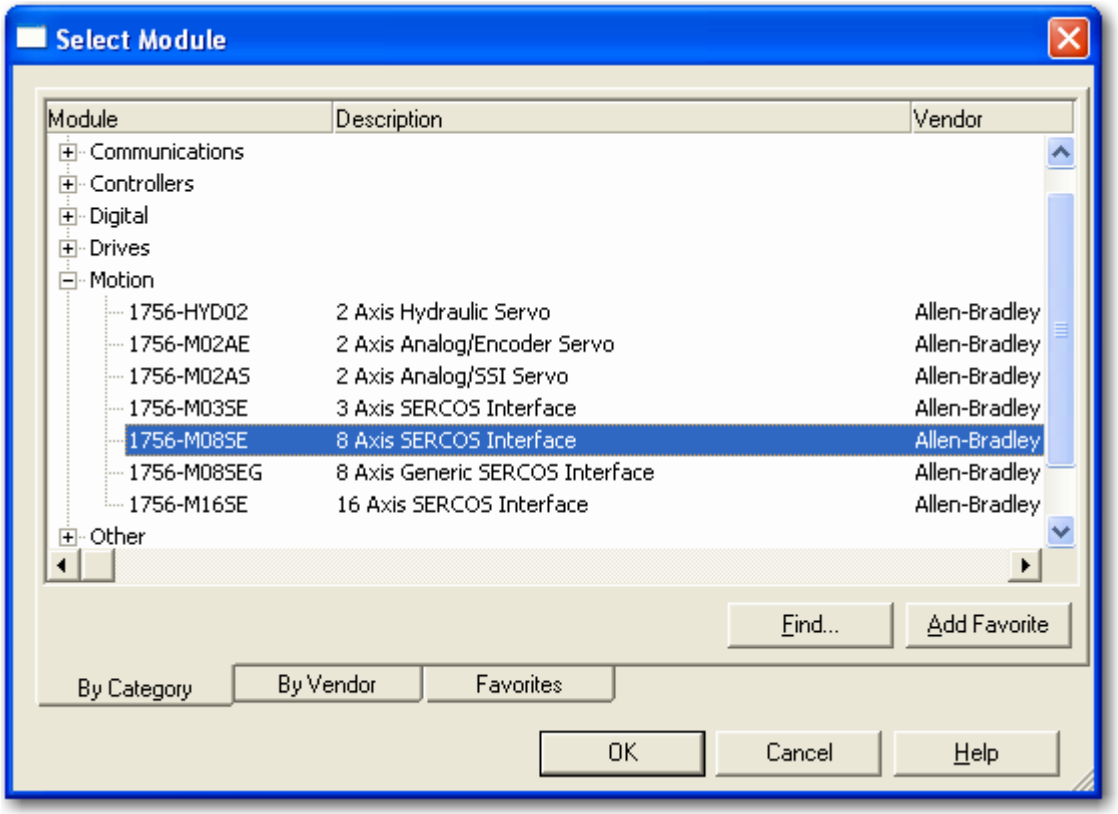
9. Name your Motion Group **'Motion_Group'** and use all the default parameters. **Press OK.**



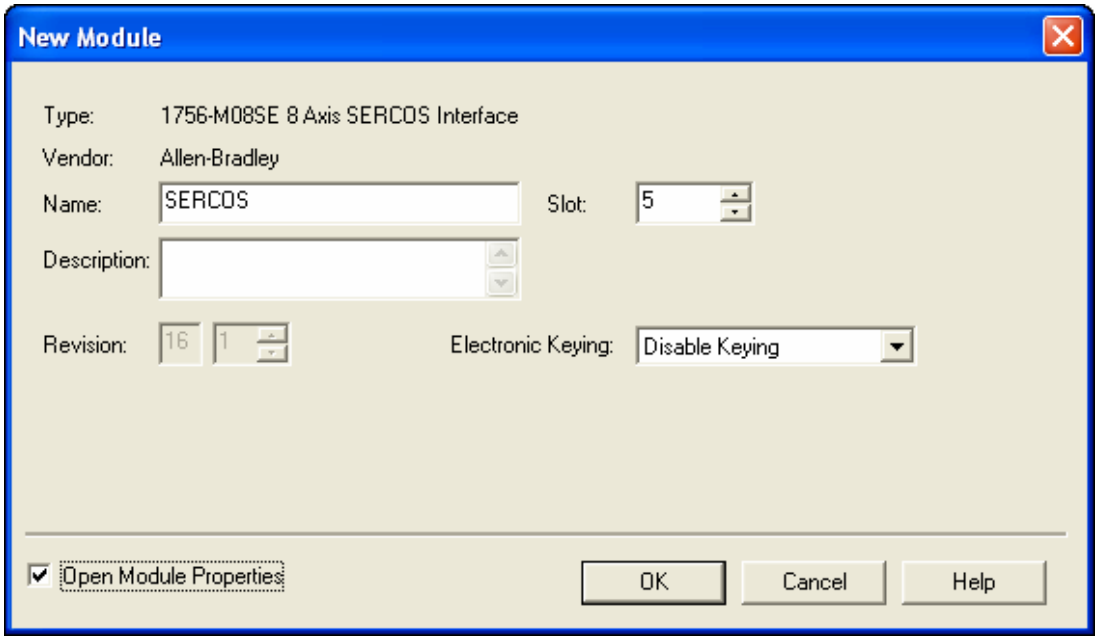
10. Next, we have to add our motion control hardware to the I/O configuration list. Right-click on the **I/O Configuration folder** in the controller organizer and choose **New Module.**



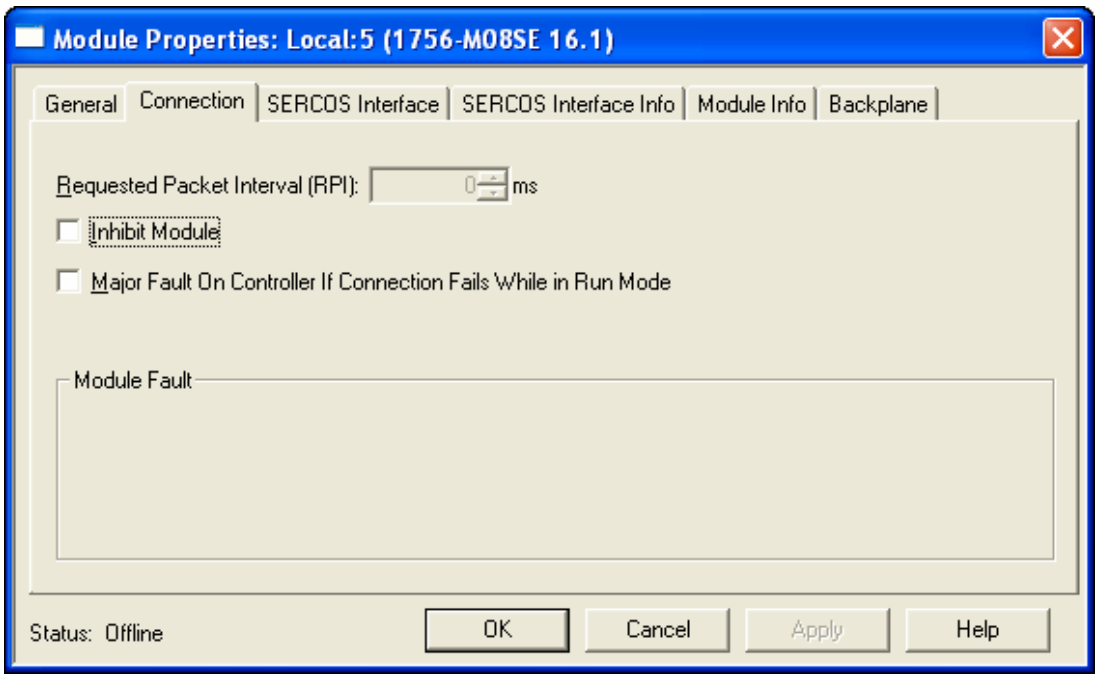
11. Expand the **'Motion'** module selection list by clicking on the "+" sign beside it. Discuss all of the motion modules currently available. When finished, choose the **1756-M08SE module** and **press OK. (For our exercise choose 1768-M04SE and place in upper BUS slot 2. Note the Ethernet IP module and SERCOS card count right to left from the processor.)**



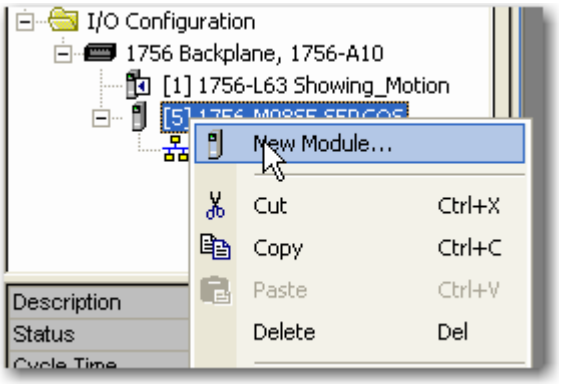
12. Name your module **'SERCOS'** and select the **proper slot** for your demo unit. Always **disable keying** to minimize errors during demonstrations. **Press OK** when complete. **(Our 1768-M04SE resides in upper bus slot 2.)**



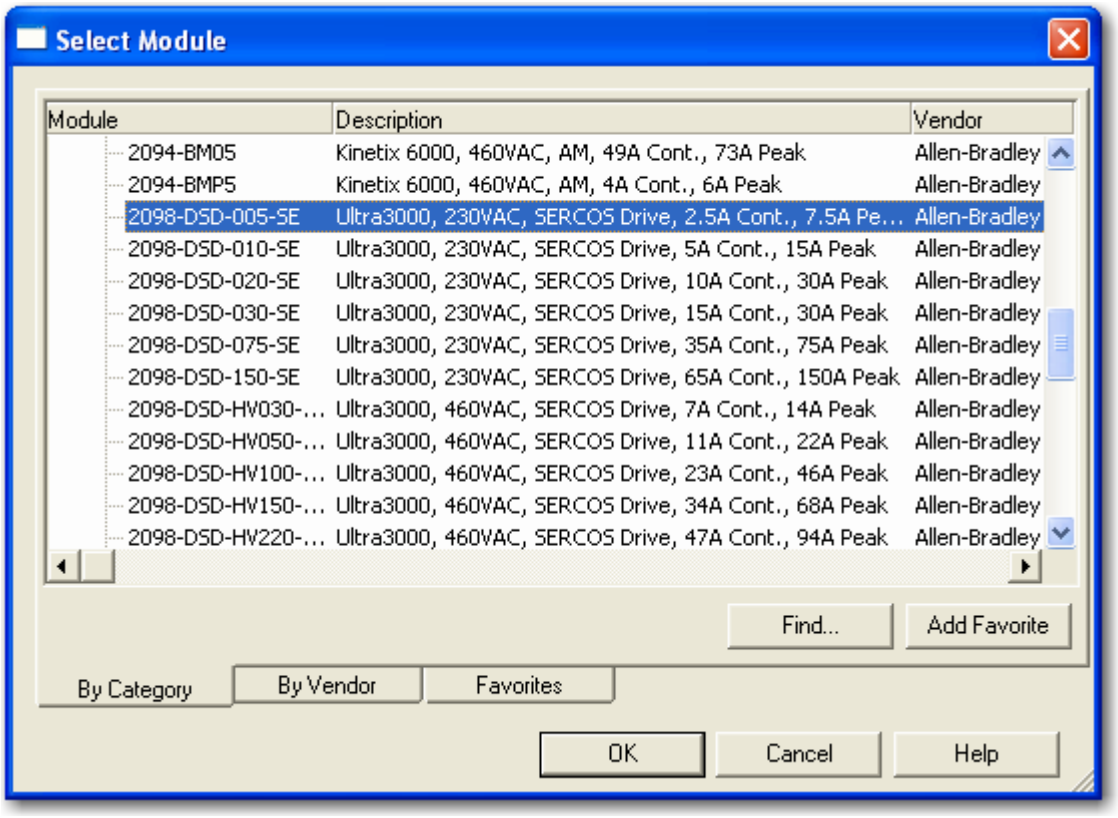
13. When the module properties dialogue opens, you can discuss the various tabs, or simply **press OK** to choose all of the default values. *This is an important feature of RSLogix 5000. The user can generally accept the defaults as usable values for their configuration.*



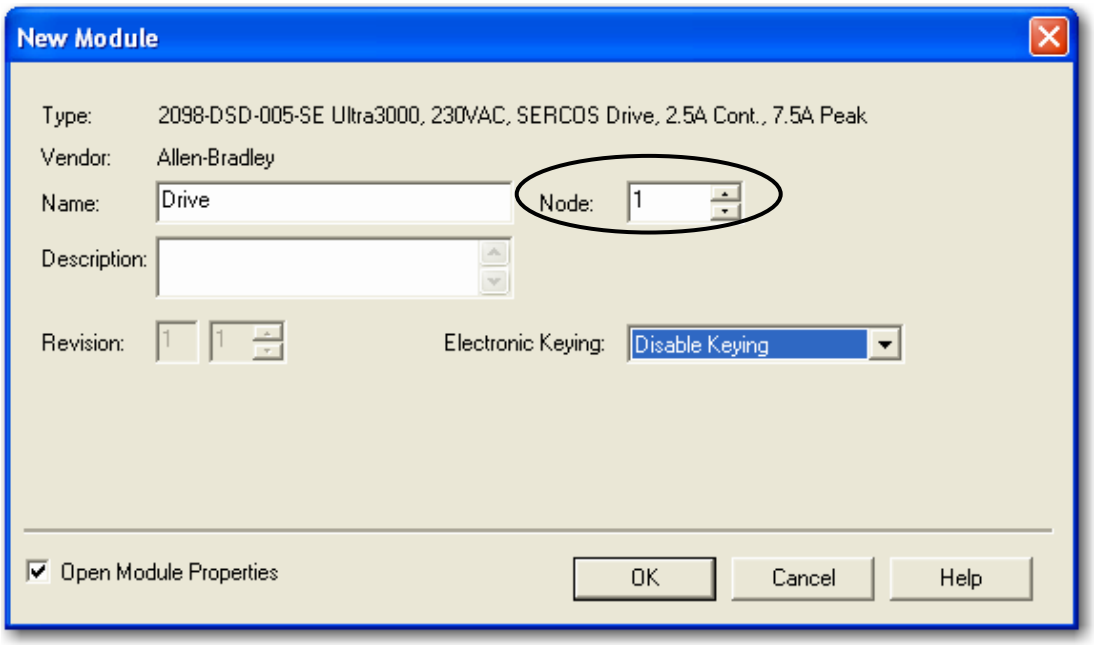
14. To add the SERCOS drive to your SERCOS motion control module, simply right-click on the **1756-M08SE (For us 1768-M04SE) module** and select **New Module.**



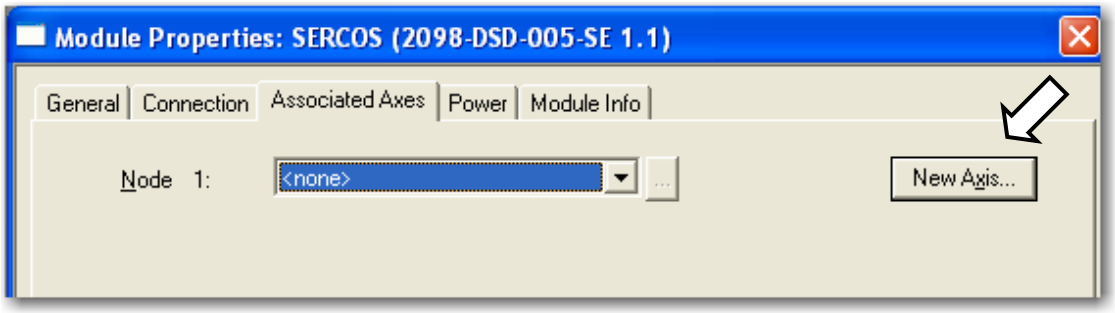
15. Expand the Drives selection list by clicking on the "+" sign beside it. Discuss all of the drive options available. When complete, select the first **Ultra 3000** drive in the list; the **'2098-DSD-005-SE'** and **press OK.** This is the catalog number of the drive in your demo unit. You may want to make a note of this, because this number is hard to see in your demo unit.



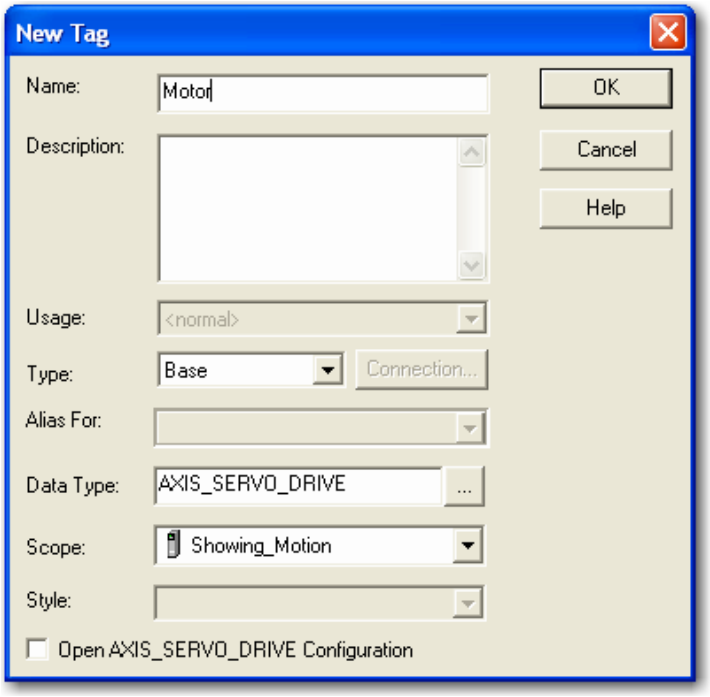
16. Name the drive **'Drive'** and select the **node address** that is dialed in on rotary switches on the front of the drive. Choose **'Disable Keying'** and **press OK.**



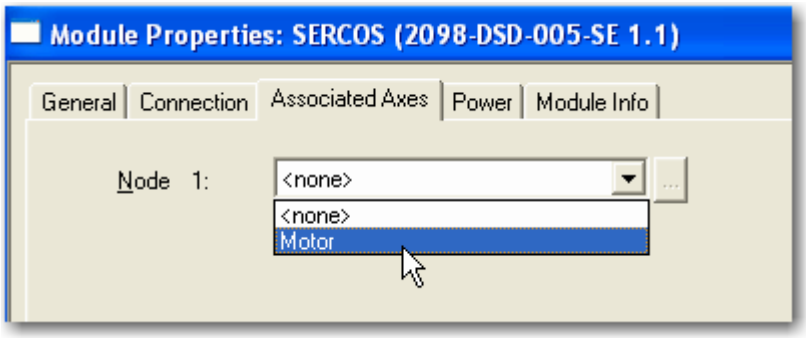
17. When the module properties dialogue opens, select the **Associated Axes** tab. This is where you link your drive to the axis of motion. We haven't created the axis of motion, yet, so select the **New Axis** button.



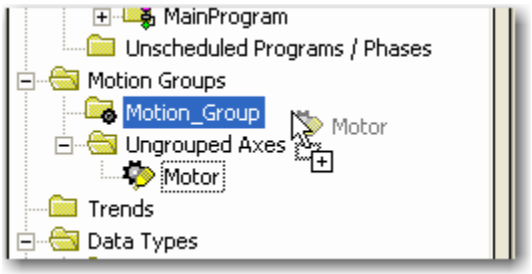
18. Name your axis **'Motor'** and use all of the default values. **Press OK.**



19. When you return to the Drive module properties screen, choose the **drop-down** selection next to the Node entry. Select your **'Motor'** axis and then **press OK.**

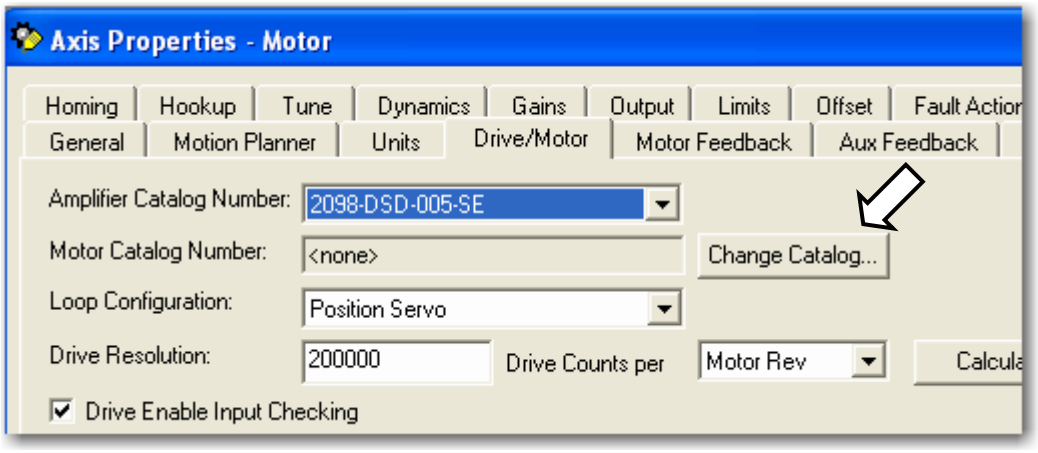


20. Your axis now appears in the Ungrouped Axes folder. Click on the axis, and **drag it** up into the **Motion Group** to activate it.

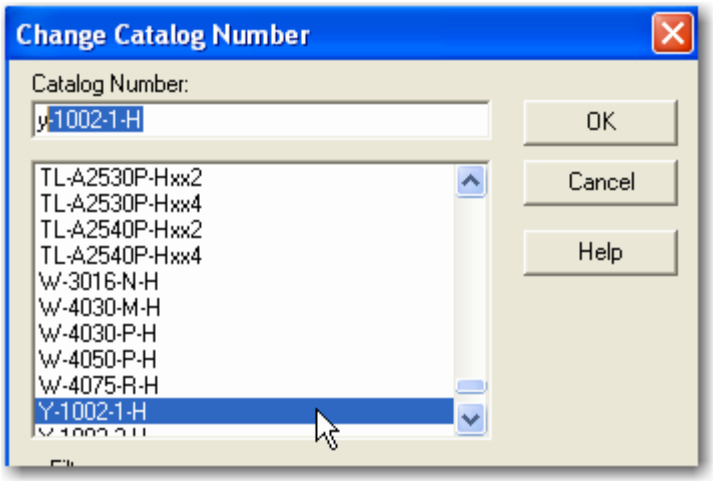


21. Double-click on your axis to open the **Axis Properties** tabs. Discuss some of these tabs and as many of the features that you are comfortable discussing. Some things you might want to learn more about and promote are the unit conversion **Calculator** on the **Drive/Motor** tab, as well the diagnostic and tuning tools found on the **Hookup and Tune** tabs.

22. We must complete our axis configuration by selecting a servo motor. Click on the **Drive/Motor** tab. Notice that the Motor Catalog Number selection is empty. Press the **Change Catalog** button.

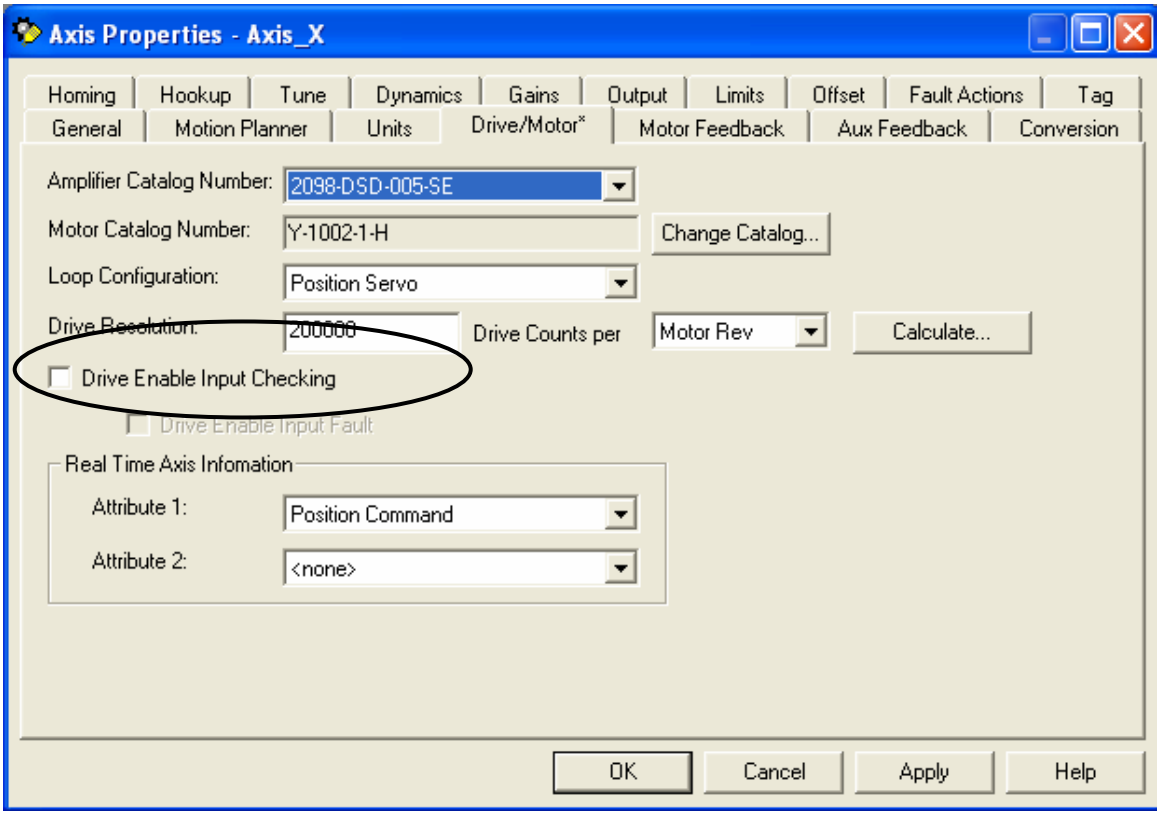


23. When the motor list pops up, you can select your motor several different ways. You can filter by nameplate voltage, product family or feedback type. You can scroll down the list until you see what you want. Since our demo unit contains a "Y" series servo motor, simply type a **'Y'** and double-click on the first "Y" motor in the list. This is the **Y-1002-H** servo motor.



NOTE: Depending on how your demo unit has been internally wired, you may be required to turn on an output in the controller that is physically connected to the Ultra 3000 drive's 24Vdc enable input before commanding the drive. To avoid this problem, perform the next step:

24. **Uncheck** the box for "**Drive Enable Input Checking**" as shown below. This means that the drive will run without the standard 24Vdc enable input present on the control I/O wiring board. **<u>This is not recommended for field usage</u>**.



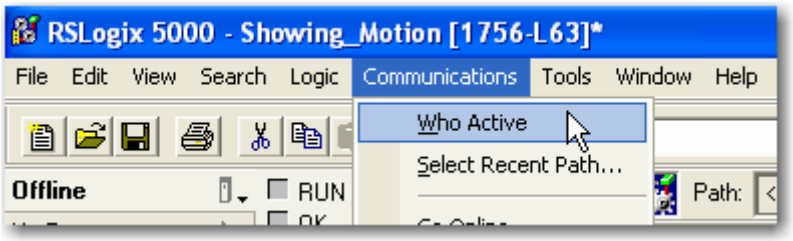
25. Close the **Axis Properties** box for your axis by pressing **OK.**

IMPORTANT: Be sure to point out how easy it was to completely configure your motion system, due to our drop-down menus and selection lists. In fact, usable tuning values were just PRE-LOADED into your axis when you selected the drive and motor. *For applications with load to inertia ratios higher than 1:1*, simply enter your actual ratio (which can calculated using Motion Analyzer) in the space provided on the Outputs tab of your Axis Properties and new values will be loaded to better suit your application. Other vendors might require you to sort through hundreds of parameters, know which ones need populated and know what values to populate them with.

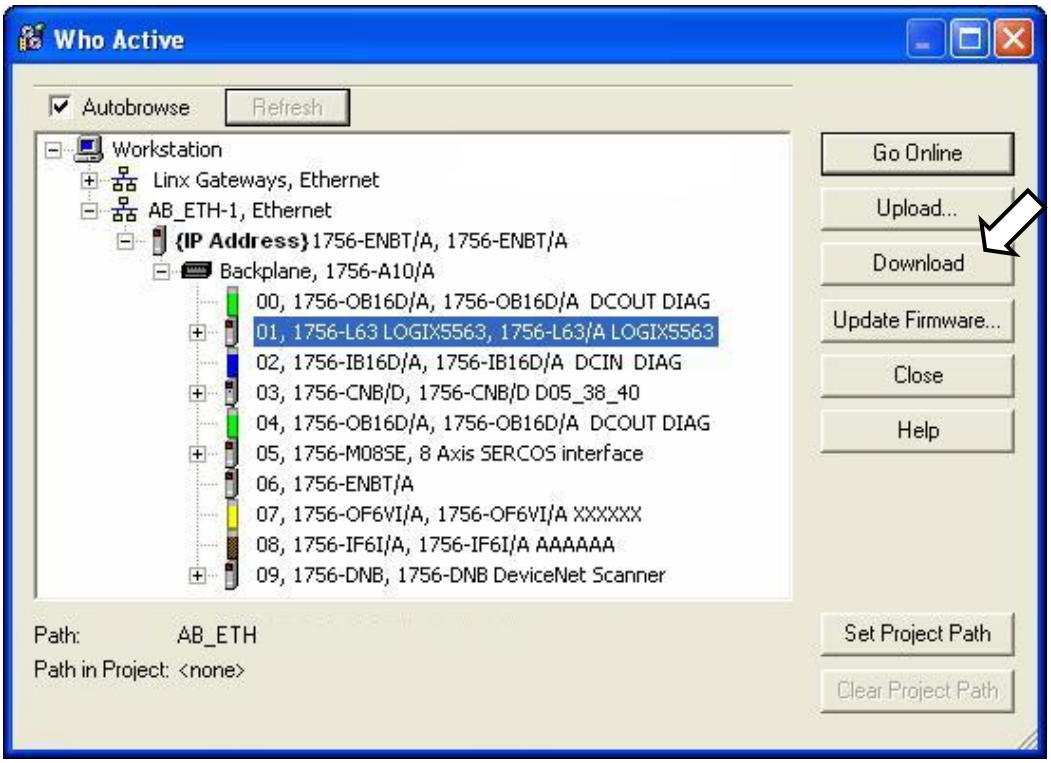
26. At this point, you are ready to download your motion system to your demo unit.

NOTE: It is assumed that you have already established communications between your PC and your ControlLogix demo unit. This means that you have:
• Set your TCP/IP or serial settings on your PC to match those on your ControlLogix demo unit. • Configured an appropriate driver in RSLinx Classic. • Connected your PC to your demo unit with an Ethernet cross-over cable or switch or with a serial (null-modem) cable.

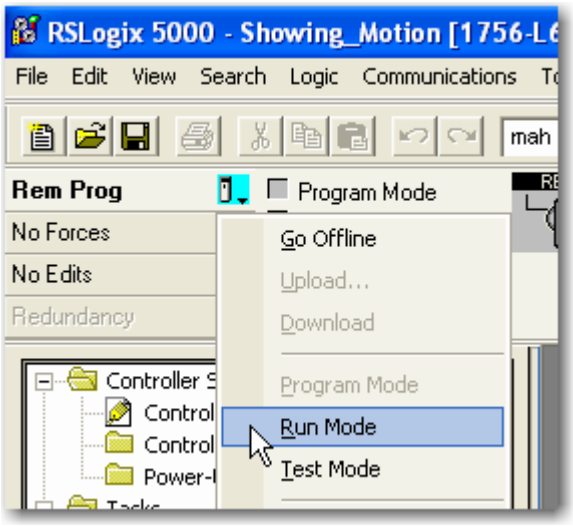
27. From the top menus, select **Communications** and **Who Active.**



28. Expand the RSLinx driver named AB_ETHIP-1. The **IP Address** for the **1756-ENBT (for us 1768-ENBT)** can be obtained from the scrolling LED display on the front of the module. Drill through (expand) the 1756-ENBT, the backplane and then the processor in slot. Select **Download**.

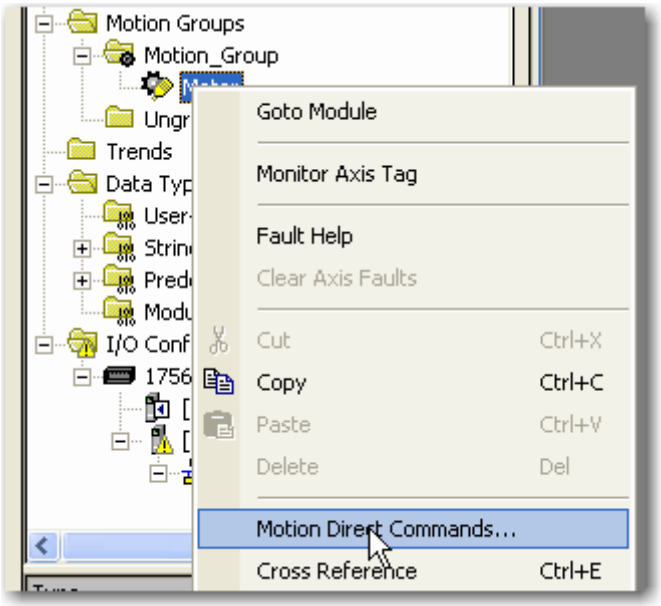


29. Put your ControlLogix controller in **Rem Run** mode**.** You can do this with the key switch on front of the processor, or through the software, as shown below. Be sure to return the **key switch** to the **CENTER** position (Rem) when done.

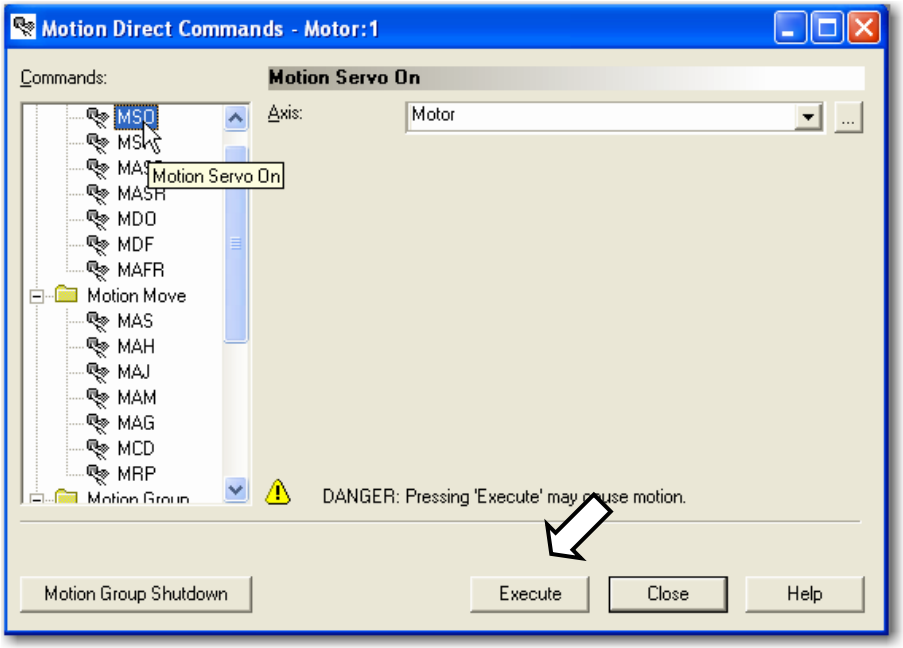


30. **Wait** for the SERCOS drive to "phase up" or count up from "1" to "4." Then we are ready to command motion.

31. Right-click on your **'Motor'** axis and choose **Motion Direct Commands**.

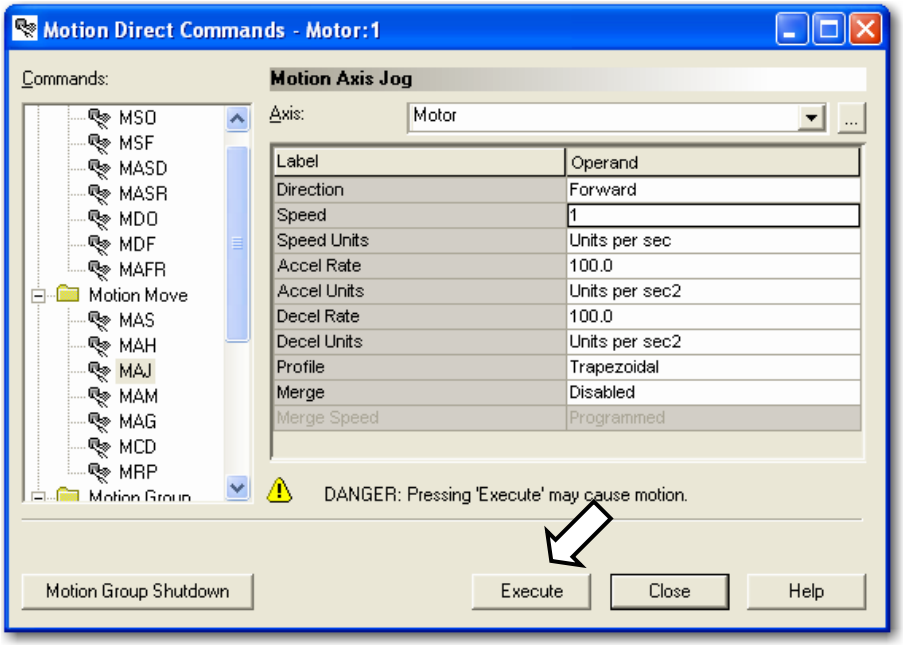


32. To enable the axis of motion, highlight the **MSO command** and then **press Execute.**



33. The servo drive is now enabled, and power is applied to the motor. The blinking green **'Module Status' LED** on the Ultra 3000 drive is now solid, and the motor resists you from spinning it manually.

34. You can now make it spin under control if you highlight the **MAJ instruction** in the Motion Direct Command window. Change the **Speed** variable from **'0' to '1'** and leave everything else at default. **Press Execute.**



35. Your motor is now spinning, and you've completely configured, downloaded and proven motion in about 5 minutes, using only **ONE software package**; RSLogix 5000.

36. This concludes the Kinetix motion *configuration* demonstration.

Ask your customer some important questions:
• Have you had to configure and program motion systems before? • Was this easier than you expected? • Does this look like it could save you time and money? • Would you like to know more?
At this point, you can bring your motion expert in to analyze and quote any applications, and discuss the technical aspects of the application.

Continue with the Extra Task, below!

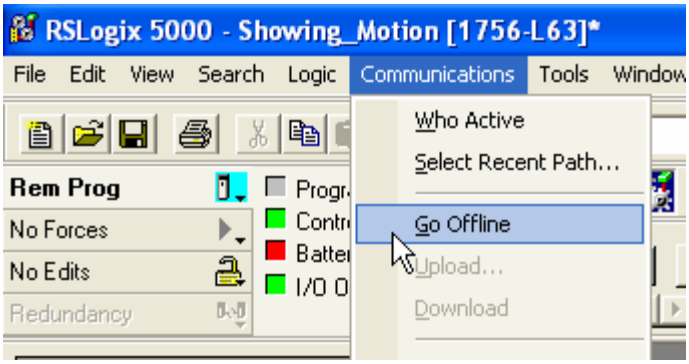
Extra Task: Programming Motion Commands

In this lab, you will learn how to easily program motion control commands in ladder logic. You will explore several of the existing (38) motion commands, and see just how simple it is to use them in your program. The most important news for your customers is that you are still programming in the familiar RSLogix™ 5000 environment, using relay ladder logic side-by-side with your existing sequential control. Key topics covered include:

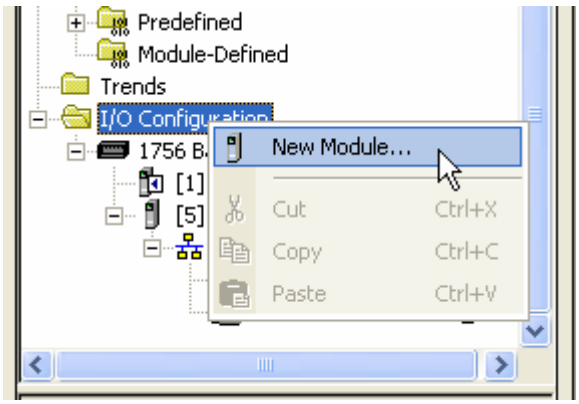
Let's begin this exercise. Here is summary of the steps that you will take to easily write motion code using ladder logic in the RSLogix™ 5000 environment:

- Write ladder code in our motion routine.
- Perform a jog command on our axis.
- Perform an incremental move on our axis.

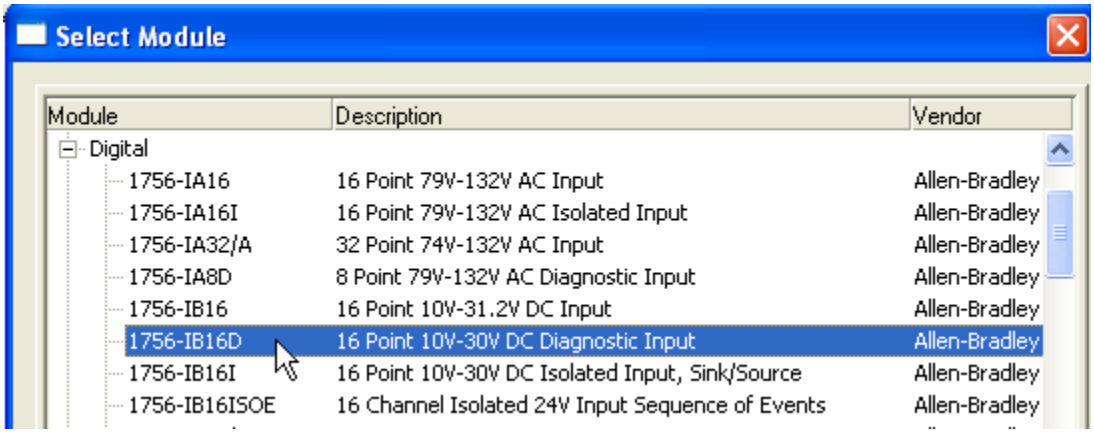
37. To simplify things while writing our code, let's go **OFFLINE** from our motion system. Select **Communications** and **Go Offline** from the top menus.



38. We will need some digital inputs to control our motion, so let's add an input module to the configuration tree. In the **Controller Organizer** window on the left, scroll all the way down to the bottom and right-click on the **I/O Configuration** folder. Select **'New Module.'**

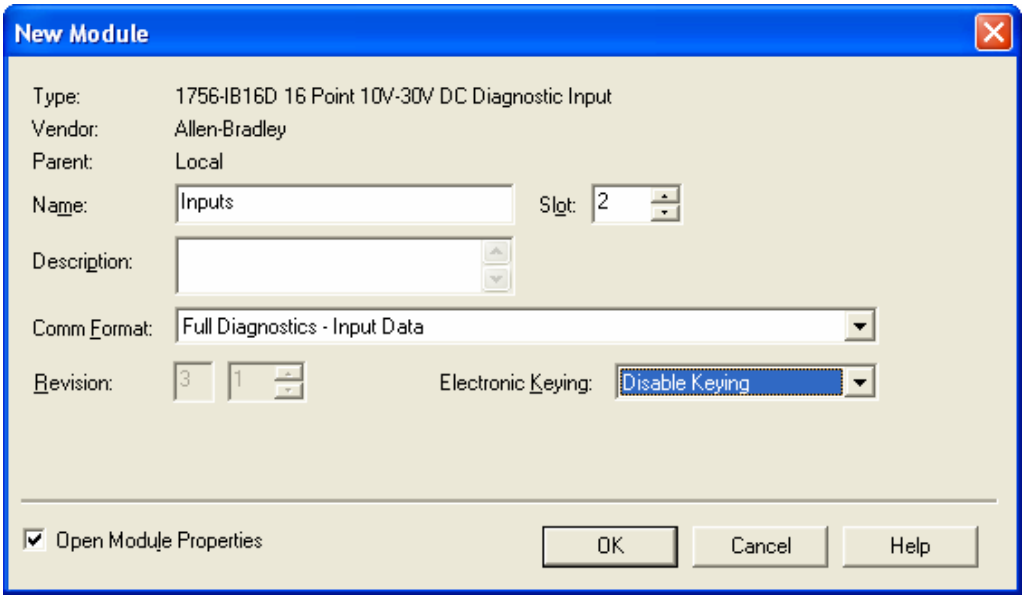


39. In the **Select Module** window, expand the **Digital** section by pressing the **'+'** sign. Select the **1756-IB16D (For our exercise choose 1769-IQ16F)** module and press **OK**.



40. Confirm **Major Revision 3.1** and press **OK**.

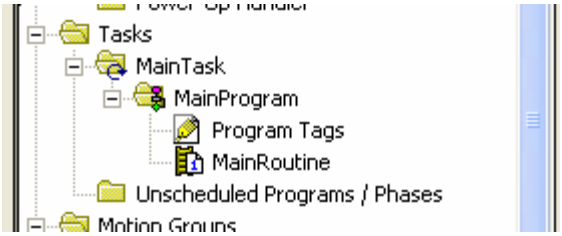
41. Enter the name **'Inputs'** for your module, set the Slot to **'1'** and **'Disable Keying.'** Press **OK** to close and save.



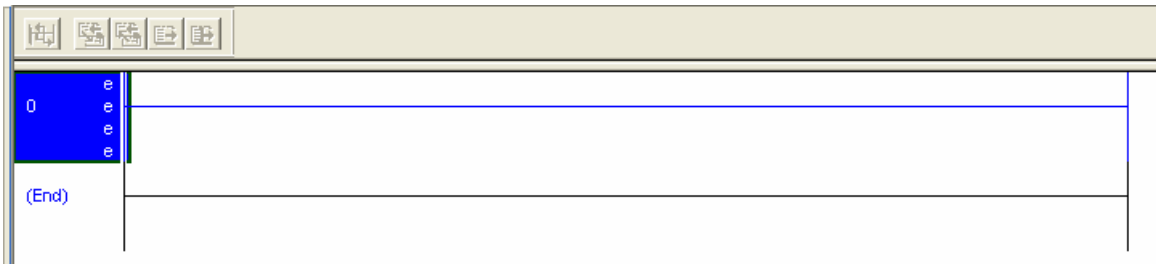
42. Press **OK** to close the **Module Properties** window and use the default values.

Note: You could continue to add existing modules to the configuration but you do not have to. The processor will pay attention only to modules in the configuration.

43. Scroll back up to the top of the **Controller Organizer** window. Press the '+' signs next to the **Tasks** folder, then **MainTask** and **MainProgram** folders to expand to our **MainRoutine**. By default, this is in ladder code.

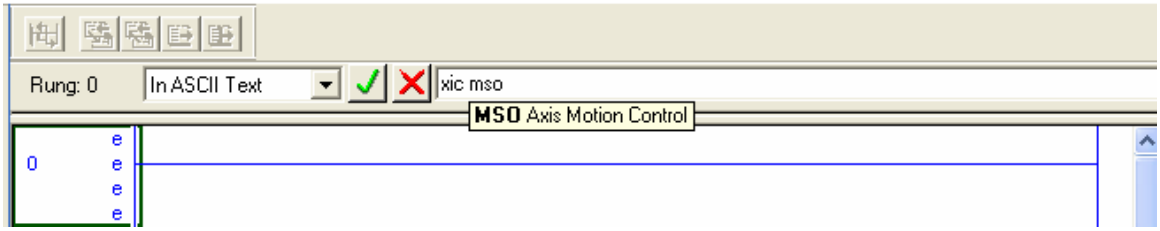


44. Double-click on the **MainRoutine** and notice that a simple, empty rung is provided by default. This is where we will add our motion code.

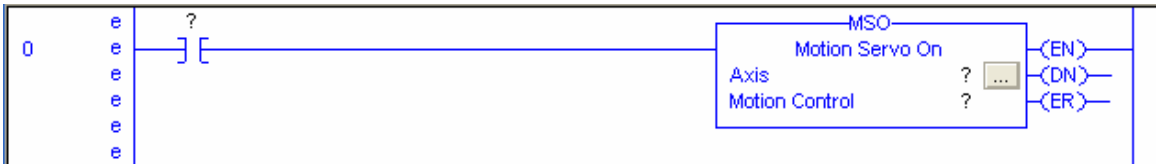


45. Before we can command any motion, we have to program an **MSO** (Motion Servo On) command that turns on power to the motor. We also need to program the **MSF** (Motion Servo Off) command, so we can turn off power to the motor.

46. You can enter ladder in several different ways in RSLogix™ 5000. One of them is by simply typing in the commands manually, if you know them. With Rung 0 still highlighted, type **'xic mso'** to enter an **MSO** command conditioned by a Boolean bit (**XIC**). Press **Enter** when complete.

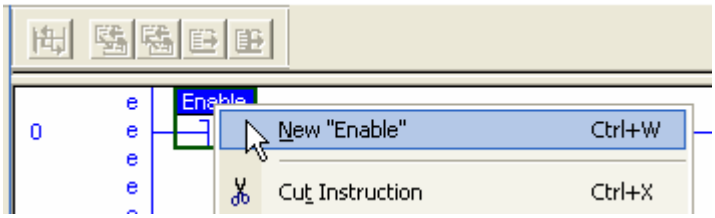


47. Now your new rung looks like this:



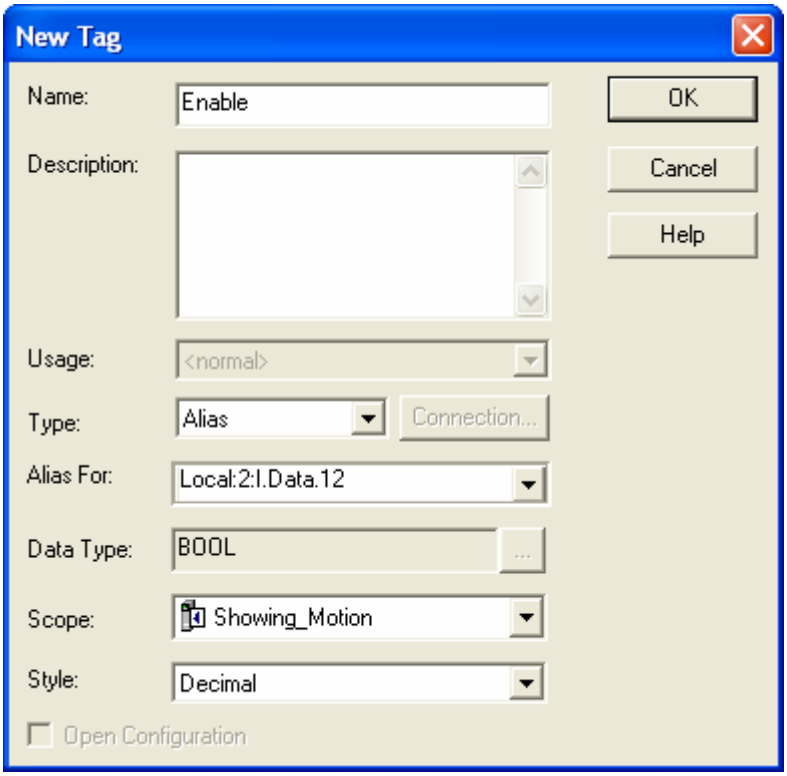
48. Click on the question mark **(?)** above the normally open **(XIC)** instruction, type **'Enable'** and press **Enter.**

49. Let's create the **'Enable'** tag and associate it to one of our inputs on the demo box. Right-click on Enable and select 'New "Enable"'.



50. Set its Type to 'Alias.'

51. In the **Alias For** section, use the drop-down menu, expand the **Local:1:I** section and select **Data**. Use the drop-down arrow within the tool to further select bit **'14'** (which is the first toggle switch on the left of our demo box) and make it a controller-scoped tag. Press **OK** to close the New Tag window.



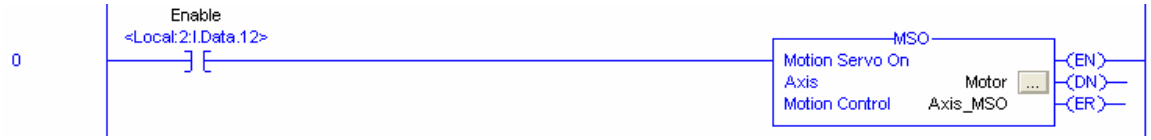
Note: Your alias for will be local 1:I Data 14

52. Next, the **MSO** command has two parameters that need to be filled out. The first one tells the controller which axis to turn on, and the second one is simply the instruction tag we have to create for this particular command.

53. Double-click on the **'?'** next to the **Axis** parameter. Use the tag browser to select **'Motor'.**

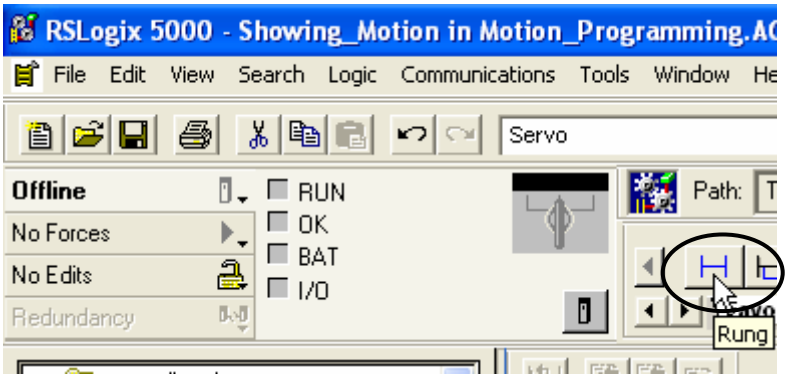
54. In the **Motion Control** entry, type '**Axis_MSO'** and press **Enter**.

55. Create the tag for Axis_MSO by **right-clicking** on it, selecting **'New "Axis_MSO"'**. Confirm the defaults and press **OK** to close. Your rung should have no errors.

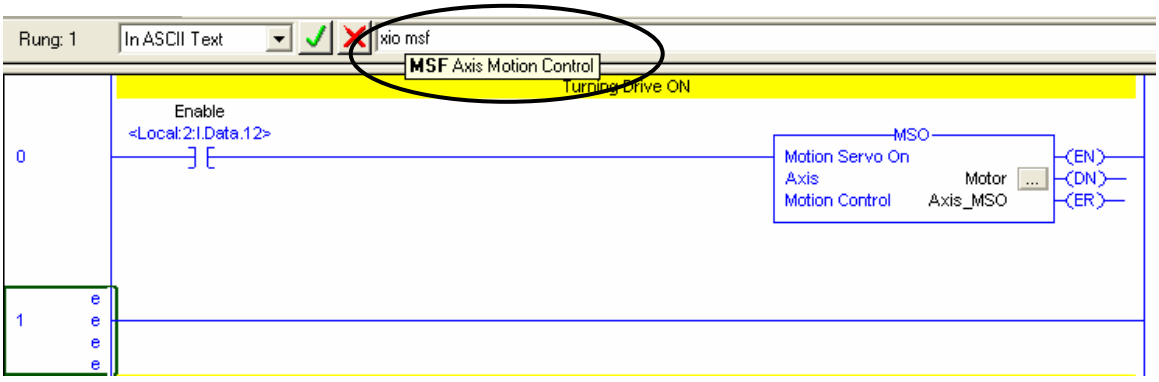


Now that we can turn the axis **on**, let's program the **off** command.

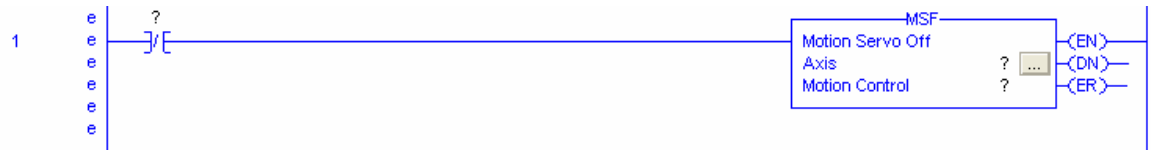
56. With rung 0 highlighted, press the **Add New Rung** button at the top of the screen.



57. With rung 1 highlighted, type **'xio msf'** to enter an **MSF** (Motion Servo Off) command conditioned by a *different* Boolean bit (**XIO**), the Examine If Open. Press **Enter** when complete.

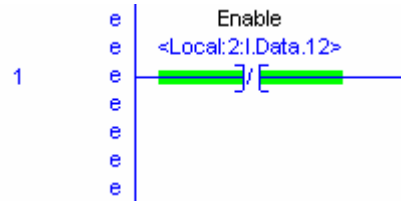


58. Now your new rung looks like this:



59. Click on the question mark **(?)** above the normally closed **(XIO)** instruction, type **'Enable'** and press **Enter.**

60. Since the **'Enable'** tag has already been created, we don't need to do anything else.

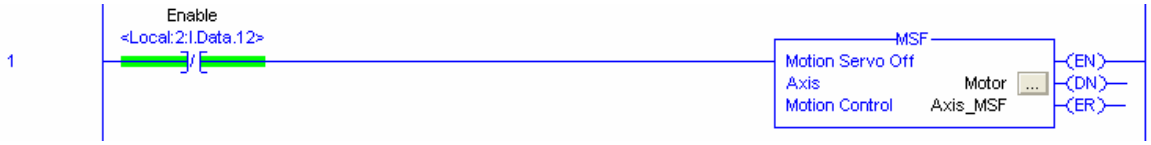


Next, let's fill out the **MSF** command's required parameters.

61. Double-click on the **'?'** next to the **Axis** parameter. Use the tag browser to select **'Motor'.**

62. In the **Motion Control** entry, type '**Axis_MSF'** and press **Enter**.

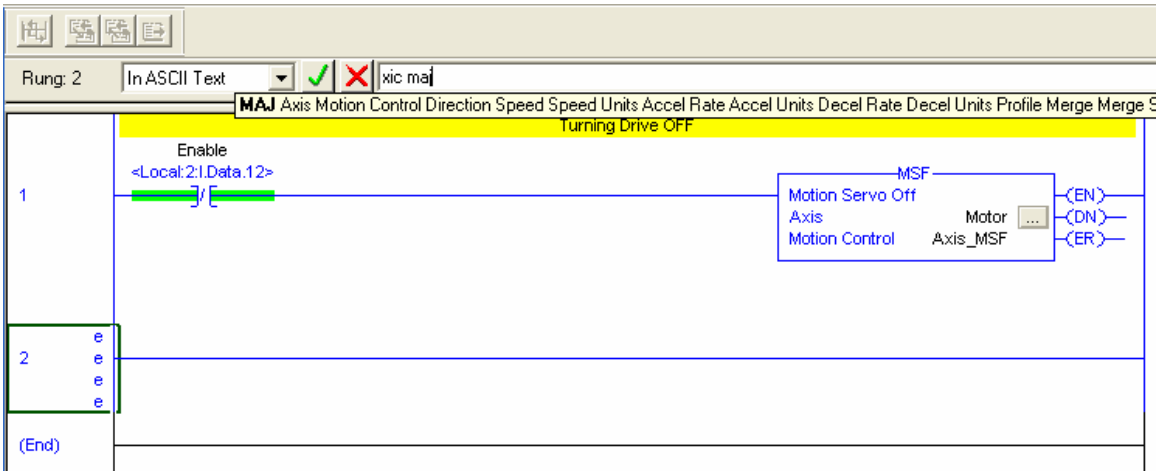
63. Create the tag for Axis_MSF by **right-clicking** on it, selecting **'New "Axis_MSF"'**. Confirm the defaults and press **OK** to close. Your rung should have no errors.



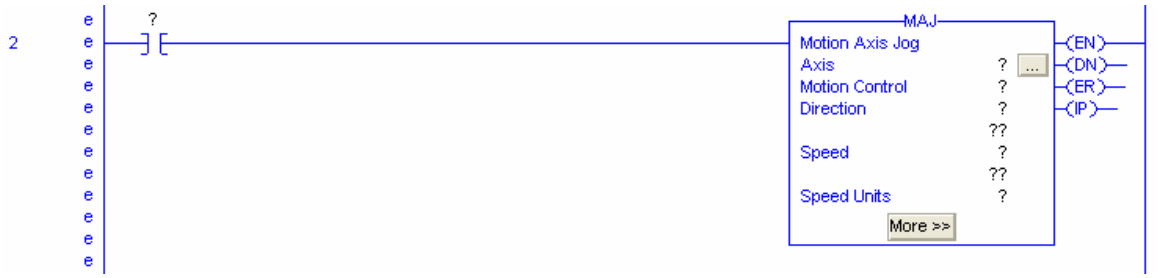
Now that we can turn the axes on and off, let's program the jog commands.

64. A **jog command** in motion is just an open-ended **velocity** (or speed) command with no end position specified. In fact, *if you don't program a stop command to end the jog, it will continue forever*. Therefore, let's program and jog and an **associated stop command**.

65. Click on the **(End)** rung below rung 1 and type **'xic maj'** to enter a jog command (**MAJ**) conditioned by a Boolean bit (**XIC**). Rung 2 gets created automatically. Press **Enter** when complete.



66. Now your new rung looks like this:

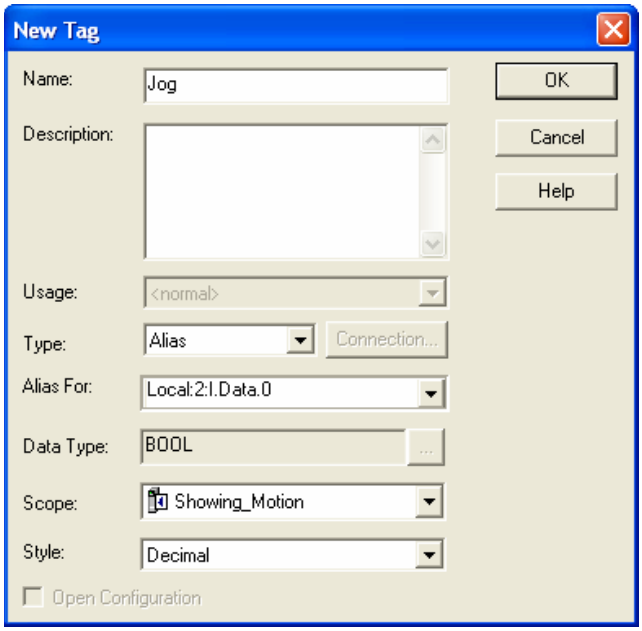


67. Click on the question mark **(?)** above the normally open **(XIC)** instruction, type **'Jog'** and press **Enter.**

68. Let's create the 'Jog' tag and associate it to one of our inputs on the demo box. Right-click on Jog and select 'New "Jog"'.

69. Set its Type to '**Alias**.'

70. In the **Alias For** section, use the drop-down menu, expand the **Local:1:I** section and select **Data**. Use the drop-down arrow within the tool to further select bit **'0'** and make it a controller-scoped tag. Press **OK** to close the New Tag window.



71. The **MAJ** command has many parameters that need to be filled out. The first one tells the controller which axis you wish to jog. Select **'Motor'** from the tag browser.

72. In the **Motion Control** entry, type '**Axis_MAJ'** and press **Enter**.

73. Create the tag for Axis_MAJ by **right-clicking** on it, selecting **'New "Axis_MAJ"'**. Confirm the defaults and press **OK** to close.

74. The next parameter is **Direction**. For a jog command, Direction can be set to **forward** (use a '0') or **reverse** (use a '1'). You can also use constants to keep your code well documented. For simplification, just enter a **'0'** for forward jogging.

75. The next parameter is the **Speed** for our jog, which can be hard-coded or a variable. For simplification, just enter a **'5'** for this value.

76. The next parameter is **Speed Units,** which can be in our scaled units per second or a percentage of our motor's maximum. From the drop-down menu for this parameter, choose '**Units per sec**.' Click on the **More** button to see the remaining parameters.

77. The next parameter is the **Accel Rate** to get our jog up to speed, which can be hard-coded or a variable. For simplification, just enter a **'50'** for this value.

78. The next parameter is **Accel Units,** which can be in our scaled units per second squared or a percentage of our motor's maximum. From the drop-down menu for this parameter, choose '**Units per sec2**.'

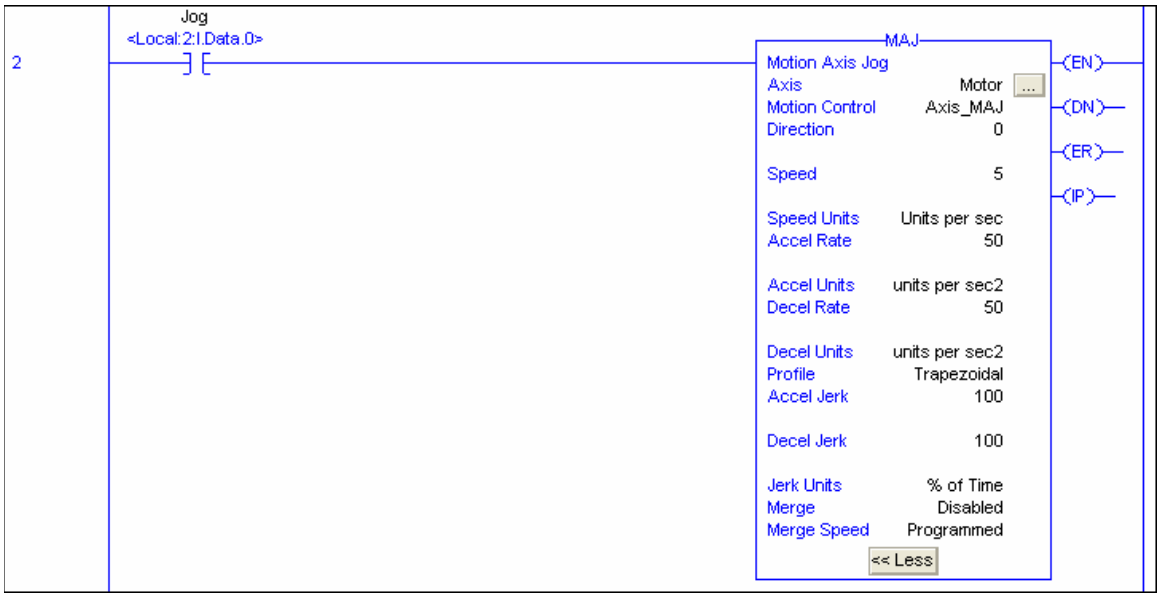
79. Similarly for **Decel Rate,** just enter a **'50'** for this value and use '**Units per sec2**' for the **Decel units**.

80. The next parameter is **Profile**. Our motion controller can make a simple Trapezoidal move profile (most common) or S-Curve moves (to ease mechanical demands on the simple). Use the drop-down menu to select **'Trapezoidal'** for our jog.

81. New parameters introduced in V16 of RSLogix 5000 allow us to control the rate of change of acceleration (and if you remember your Calculus class!) which is known as '**Jerk**.' Our Trapezoidal Profile setting above disregards this setting, but enter a '**100**' for **Accel Jerk** and for **Decel Jerk**.

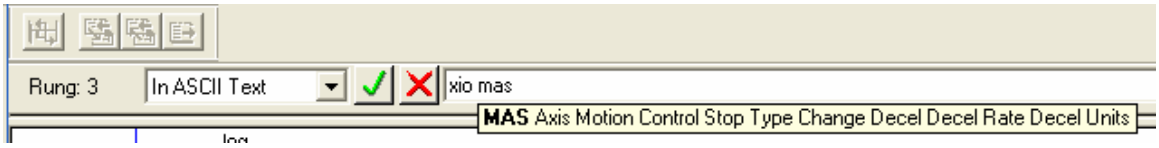
82. Use the drop-down menu to set the **Jerk Units** to **'% of Time**'.

83. Finally, set the **Merge** parameter to **'Disable'** and the **Merge Speed** value to **'Programmed'** each from drop-down menus to complete the setup for the jog command. Your rung should have no errors.

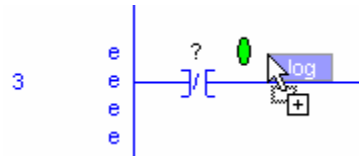


84. As mentioned above, we now have to program an **MAS** or Motion Axis Stop command to terminate the open-ended jog command.

85. Once again, highlight the **(End)** rung and type **'xio mas'** to add a normally closed condition with a stop command.



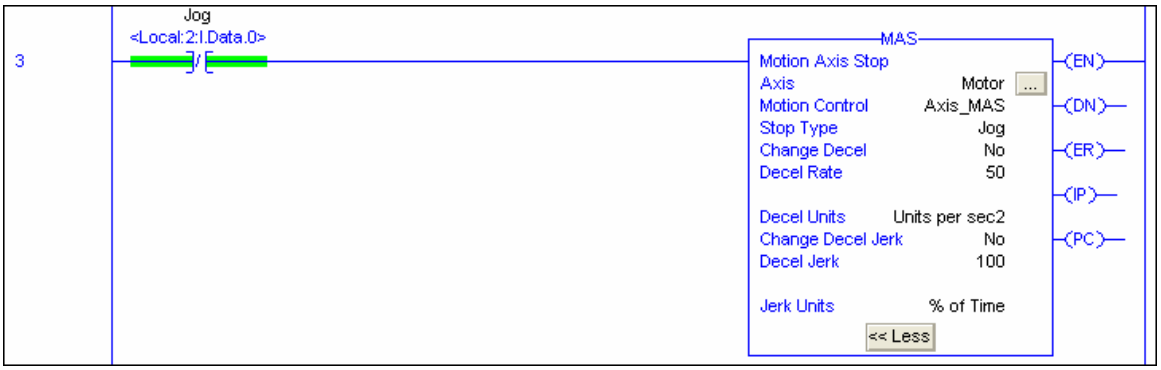
86. Drag and drop the **Jog** tag from rung 2 to the XIO on **rung 3.** This will cause the jog command to a stop when we let off of the jog push button.



87. In the **MAS** command, use the tag browser to set the **Axis** value to **'Motor.'**

88. Create an **'Axis_MAS'** instruction tag (type, right-click, etc) for the command and use the drop-down menu to set the **Stop Type** to **'Jog'.**

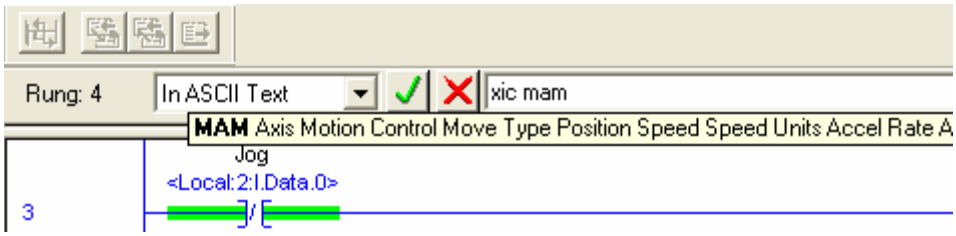
89. Set the rest of the **MAS** parameters as follows:



90. Your rung should have no errors. **Save** your work.

Finally, let's add our code to make a **move** with our **Motor** axis.

91. Click on the **(End)** rung below rung 3 and type **'xic mam'** to enter a move command (**MAM**) conditioned by a Boolean bit (**XIC**). Rung 4 gets created automatically. Press **Enter** when complete.



92. Now your new rung looks like this:

93. Click on the question mark **(?)** above the normally open **(XIC)** instruction, type **'Move'** and press **Enter.**

94. Create the **'Move'** tag and associate it to input **DI1** on the demo box.

95. The MAM command has many parameters that need to be filled out. Select **'Motor'** from the tag browser for the **Axis** value.

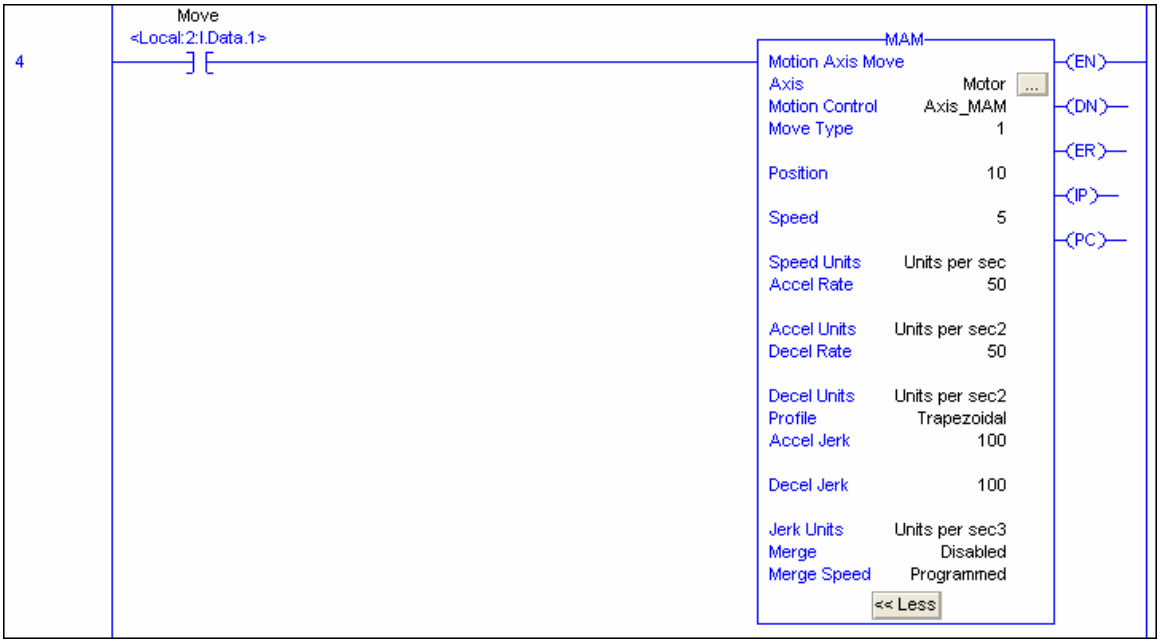
96. In the **Motion Control** entry, type '**Axis_MAM'** and press **Enter**. Create the tag for it as before.

97. The next parameter is **Move Type**. For a move command, we can make absolute or incremental moves, as we saw in the Motion Direct Commands sections. Setting the move type to '0' calls an absolute move, where using a '1' calls an incremental move. There are many more values that you could use here as well, and you would normally use a constant to keep your code well documented.

98. Type in a **'1'** for the **Move Type** value. If you opened the instruction help for a move (MAM) command, you would get a list of what the available move types are, and what value to use for each (you do not have to do this). The '1' that we entered tells is to make an incremental move, which literally means "wherever you are, go an additional number of turns."

99. Unlike the open-ended jog command, the **move** command requires an end position. Let's command our axis to move 10 revolutions when we press the move button. Enter a **'10'** for the **Position** value.

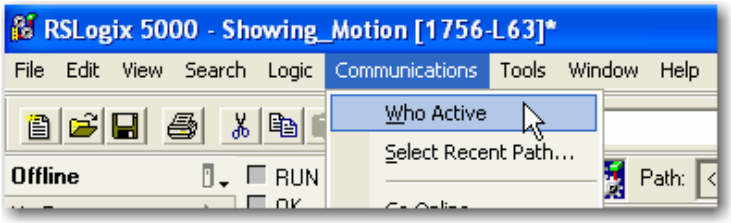
100. The rest of the parameters are similar to that of the jog command, so fill out the remaining values **as follows**:



101. Your rung should again have no errors. **Save** your work.

102. At this point, you are ready to **download** your motion code to your demo unit. Make sure that the **DI14** (Enable) switch on the demo unit is in the **Left position** (off). You may also want to rotate the motor so that the white line on the rotating disk is pointing straight up so you can easily count revolutions.

103. From the top menus, select **Communications** and **Who Active.**



104. Browse for the **controller in slot (1**). The **IP Address** for the **1756-ENBT(1768-ENBT)**can be obtained from the scrolling LED display on the front of the module. The **IP Address** is also listed on the **White Label** fastened on the front of your ENBT module. Select **Download**.

105. Select **Download** again.

106. Answer 'Yes' to the prompt to place your controller back in the **Rem Run**.

107. **Wait** for the Ultra 3000 drive to "phase up" or count up from "1" to "4." Then we are ready to control our axes of motion.

108. Turn the **DI14** (Enable) switch to the **right**, and you should hear the axis make a clicking sound to enable. Verify this by noting that the drive's **"Module Status" LED** has gone from blinking to solid, and you can no longer move the motor.

109. While watching the instructions on rungs 2 and 3, press the **DI0** (Jog) push button the demo box. The **Motor** axis should begin spinning at 5 revolutions per second. It should stop when you release the push button.

110. While watching the instruction on rung 4, press the **DI1** (Move) push button the demo box. The **Motor** should quickly move to **10 revolutions** and stop.

111. Pressing this button **again** should have the same effect, since we are making incremental moves. Try this.

112. Verify this and go **OFFLINE** using **Communications → Go Offline.**

This concludes the motion code programming portion of the lab.