Exercises: Contents

Exercise 1: Exploring the Hands-On Hardware

Objective: Explore the hardware used in this hands-on seminar, and execute the prebuilt test code modules that will be used to create a test sequence in TestStand.

Similar to the Windows Device Manager, which manages all peripherals connected to a Windows PC, Measurement & Automation Explorer, or MAX, manages all NI hardware and software. This application is installed with most NI software packages. In this exercise, you will look at the most used features of MAX including **Software** and **Devices and Interfaces**.

1. Open MAX by double clicking the icon on the desktop or navigating to **Start » All Programs » National Instruments » Measurement & Automation Explorer**.

2. After MAX launches, expand **My System » Devices and Interfaces**. You should be able to see any hardware attached to your local machine. If you do not you might need to expand the section **NI PXIe-1062.**

3. Your test system consists of a chassis, a controller and measurement modules. The chassis is the frame that holds the controller and instruments and allow them to communicate with each other. The controller has the processor and memory that allows you to execute programs, in fact it is a computer that can run either a standard Windows operating system, or a real-time operating system. The modules send and receive the physical signals to and from your devices under test. There might be more devices in your system but in this seminar you will use the following four instrument:

- **PXIe-5442 "Arb"** - Function generator used to generate a frequency sweep to test the filter response.
- **PXIe-5122 "Digitizer"** - Modular oscilloscope to measure filter response
- **PXI-4071 "DMM"** - Digital Multimeter used to test the LED function
- **PXI-2532 "Matrix" or "Matrix1_PXI2532"-** Switch module to rapidly change connections between units to maximize instrument reuse

4. Verify that you have these four instruments in your device list and then close MAX.

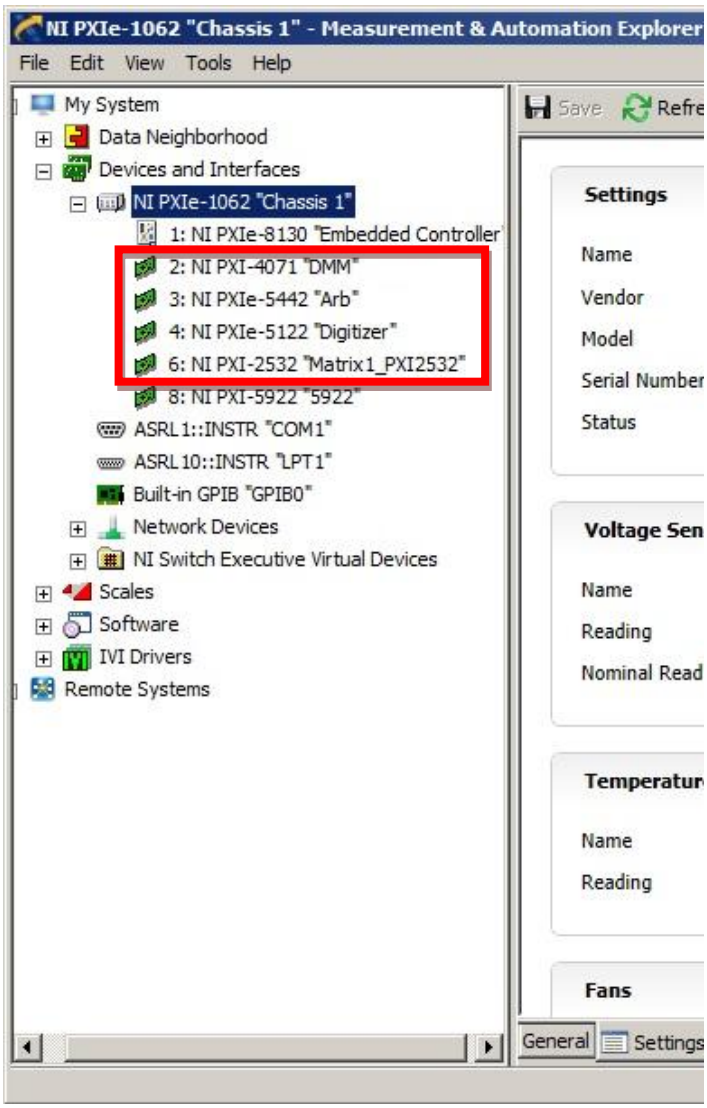


Figure 1. MAX configuration

Exercise 1A: Use NI-SWITCH Soft Front Panel to interactively make switch connections to the filter on DUT0

1. First, make sure that the following **PXIe-5442 Arb** and **PXIe-5122 Digitizer** signal connections have already been made, or make them now:

    a. The **CH0** SMB connector of the PXIe-5442 connects to **Left-most BNC** of the 5 DUT Switch Demo box using an SMB-112 cable.

    b. The **CH1** BNC connector of the PXIe-5122 connects to the **Upper-left BNC of the Dual-BNC panelette** of the 5 DUT Switch Demo box using a BNC cable.

    Next, do the same for the following connections:

    c. The **TB-2641** terminal block of the **PXI-2532** connects to the **demo box**. Make sure to screw the terminal block all the way in and that it is firmly seated.

    d. Plug in four of the **Filter/LED Demo DUTS** into **DUT0, DUT1, DUT2,** and **DUT3** of the 5 DUT Switch Demo box.

    e. Make sure that the **jumper** on each DUT is set to **NONE**.

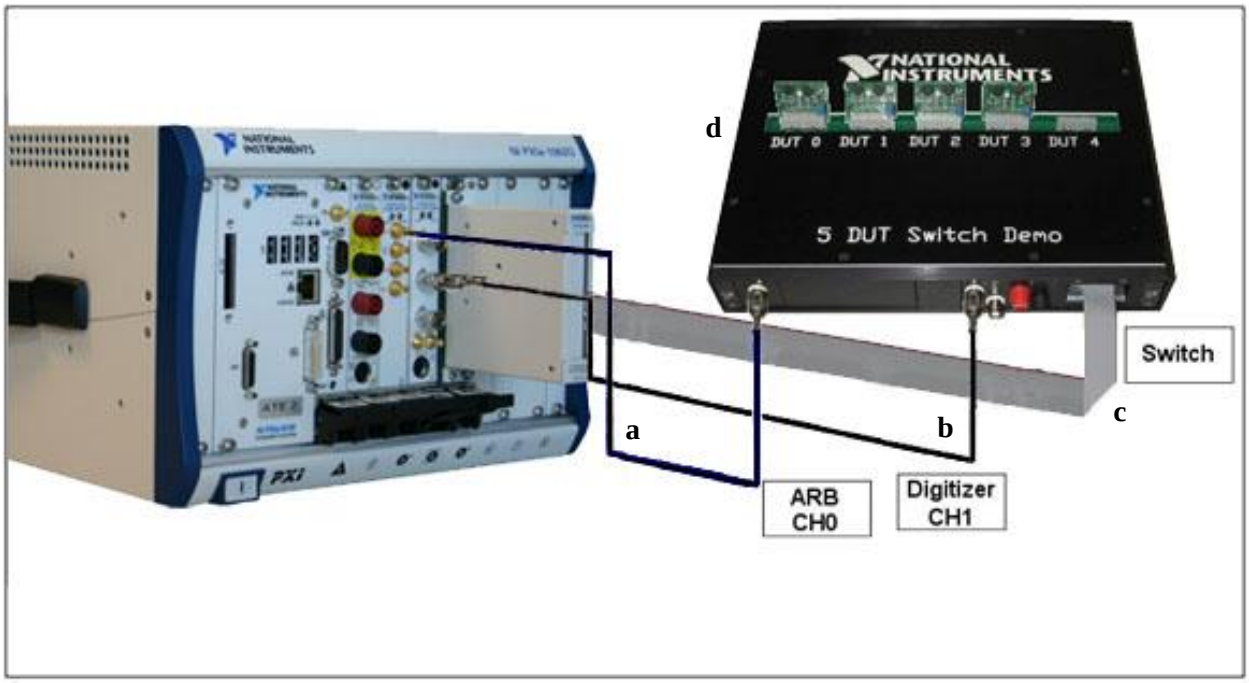


Figure 2 Signal Connections

2. Now you will interactively make switch connections to test the filter on DUT0 in the system using the **NI-Switch Soft Front Panel**. Select **Start » All Programs » National Instruments » NI-SWITCH » NI-SWITCH Soft Front Panel.**

3. Set the **Active Device** to **Matrix** and **Topology** to **2532/1-Wire 8x64 Matrix**.

4. Click on the **Schematic** tab. Notice that there is a graphical display of an 8x64 matrix. This is the current configuration of the PXI-2532 512-Crosspoint matrix. This means that there are 8 rows and 64 columns. We will cover switching in more detail during the next exercise, but for this exercise you

should know that all the instruments and test points are connected to the columns of the matrix. The rows are only used to connect columns together.

5. The switch connections required to connect the instruments to the Filter on DUT0 are listed on the following chart.

Arb to **DUT0 Filter input**	**c1** to **c25**
Digitizer to **DUT0 Filter output**	**c3** to **c24**

6. You will make the connections in the following figure interactively using the Switch Soft Front Panel.

    a. Click on the intersection of **r0** and **c1**. This connects c1 (Arb) to row 0.

    b. Next, click on the intersection of **r0** and **c25**. This connects c25 (DUT0 Filter input) to row 0, which in turn connects to c1 (Arb).

    c. Click on the intersection of **r1** and **c3**. This connects c3 (Digitizer) to row 1.

    d. Next, click on the intersection of **r1** and **c24**. This connects c24 (DUT0 Filter output) to row 1, which in turn connects to c3 (Digitizer).

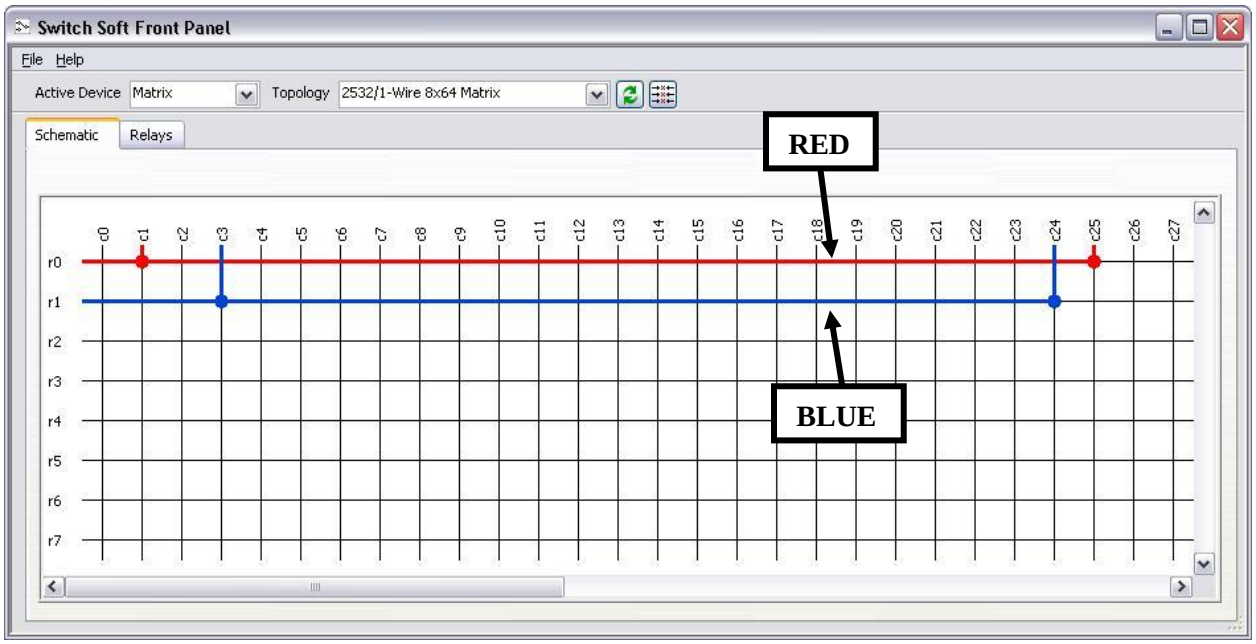


Figure 3 Switch Connections

    e. Close the NI-Switch Soft Front Panel window.

Exercise 1B: Run the provided LabVIEW VI to verify that the UUT and modular instruments respond as expected.

Launch LabVIEW by selecting **Start » All Programs » National Instruments LabVIEW x.x.**

Tip: *National Instruments LabVIEW x.x* refers to the latest version of LabVIEW on the system.

1. Open the prepared filter test created in LabVIEW located here:

   C:\ Seminars\Automated Test System with NI TestStand\Exercises\Ex1 Filter Test.vi

2. Select **Window » Tile Up and Down** so that both front panel and block diagram are visible.

   Notice that the VI consists of two windows: the *front panel* and the *block diagram*. The front panel is the graphical user interface of the VI. You build the front panel using controls and indicators, which are the interactive input and output terminals of the VI, respectively. After you build the front panel, you add code using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code, also known as G code or block diagram code. Front panel objects appear as terminals on the block diagram.

3. Execute the filter test by pressing the **Run** button on either the front panel or the block diagram.

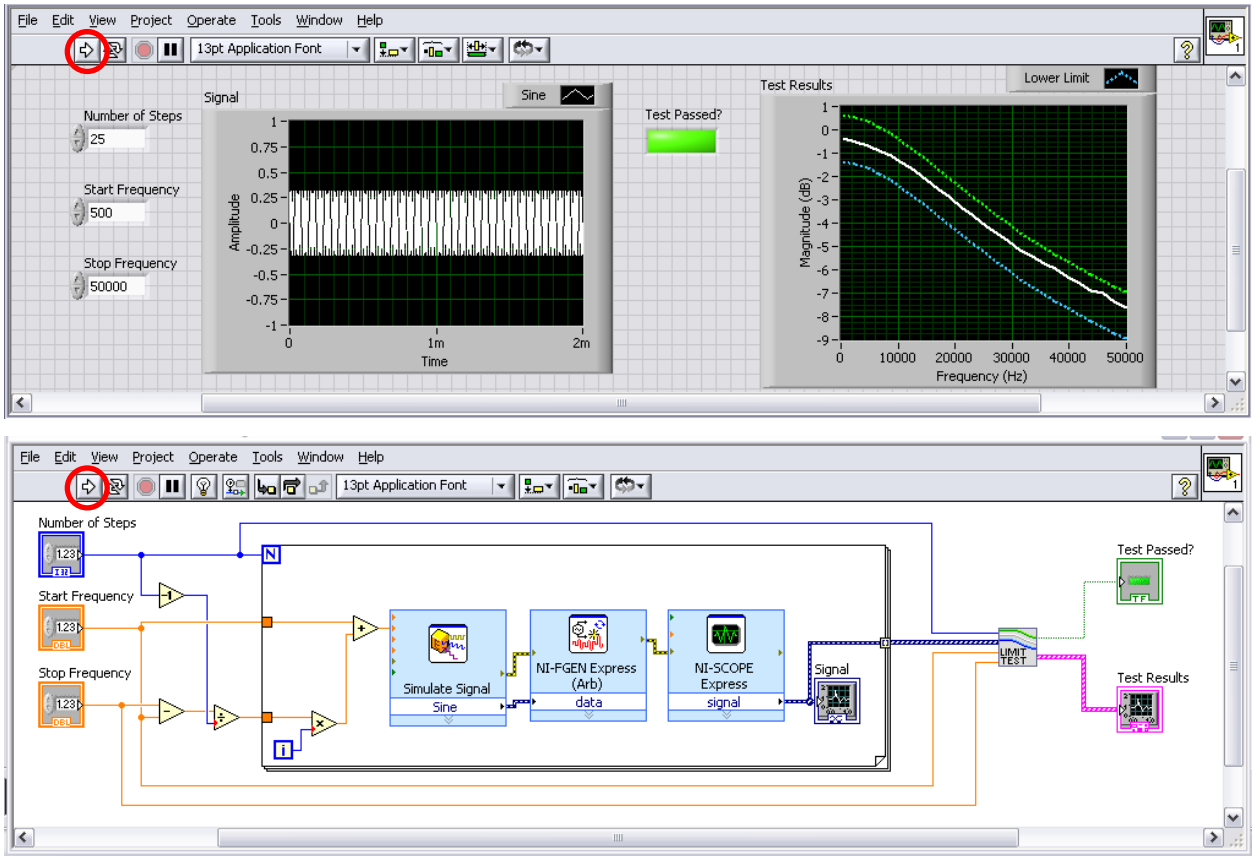


Figure 4. Test results with good DUT connected

4. The DUT has a jumper that can be moved to force the filter test to fail. If you move the jumper to "**C4**" or "**R2**" the filter will be bypassed and the test will fail. Try running the test with the jumper moved to simulate a bad filter.

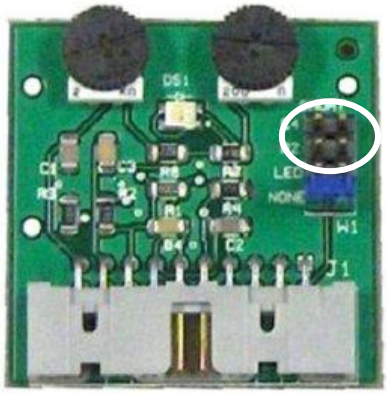


Figure 5. Location of jumpers on DUT

5. Replace the jumper when you are finished.

End of Exercise 1

Exercise 2: Introduction to the TestStand Environment

Objective: Explore an example sequence file and run it using **Execute»Test UUTs**.

Description

1. Log in to the TestStand Sequence Editor.
   - Select **Start»All Programs»National Instruments»TestStand 20*xx*»Sequence Editor** to launch the sequence editor.
   - When the Login dialog box appears, select the default, administrator, from the **User Name** control and leave the **Password** textbox empty, as shown in Figure 1.
   - Click **OK** in the Login dialog box.

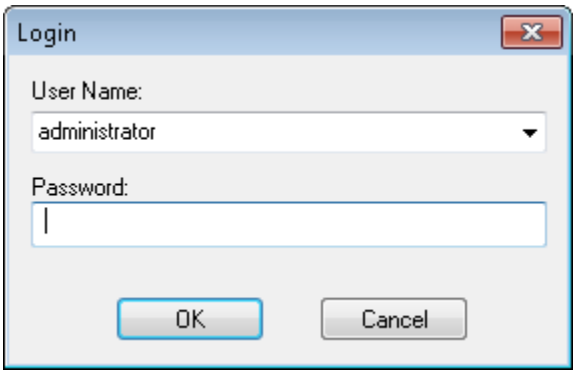


Figure 1. Login Dialog Box

2. Open a sequence file.
   - Select **File»Open File…** and navigate to the `<TestStand Public>\Examples\Demo\LabVIEW\Computer Motherboard Test` directory.

     Note: Directory names in angle brackets, such as `<TestStand Public>`, refer to abbreviated folder paths. For example, `<TestStand Public>` refers to the directory path that starts at your root directory and leading to `Users\Public\Documents\National Instruments\TestStand` for **Windows 7/Vista** or `\Documents and Settings\All Users\Shared Documents\National Instruments\TestStand` for **Windows XP**.
   - Open the Computer Motherboard Test Sequence.seq file. Figure 2 shows the TestStand window that opens.

     Additional Information
     The Computer Motherboard Test Sequence.seq sequence file simulates testing parts of a computer motherboard such as the RAM, ROM, and keyboard. This sequence includes tests written in LabVIEW, but you can write the tests in LabWindows/CVI or other programming environments.

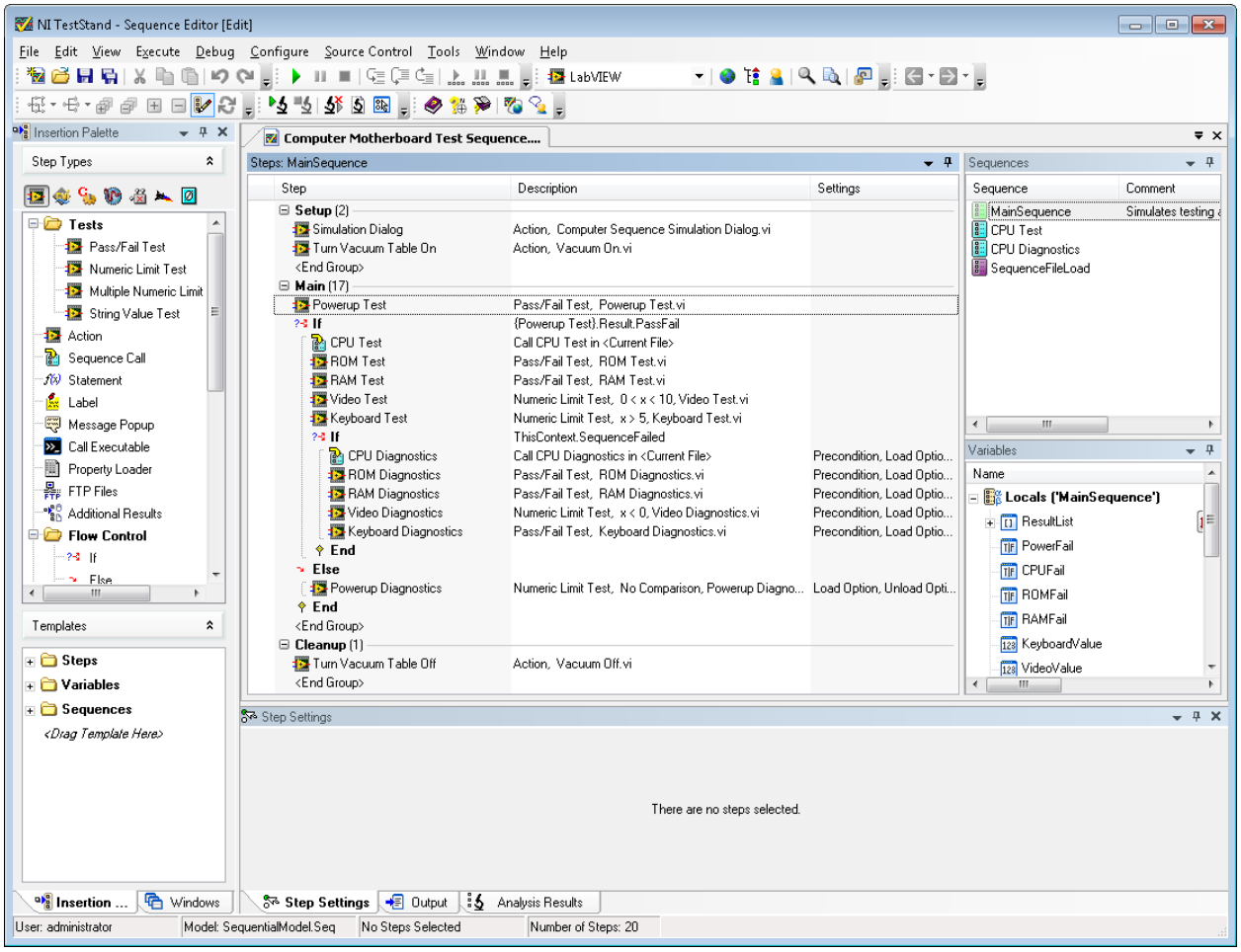


Figure 2. Sequence File in the Sequence Editor

3. Explore the sequence file.

   - In the Steps pane, located in the center, select the **ROM Test** step. Observe that the Step Settings pane at the bottom displays the settings for the step. The VI icon next to the step indicates that the step calls LabVIEW code.
   - In the Sequences pane, select **CPU Test** sequence. Observe that the Steps pane displays the steps in the CPU Test sequence.

     Additional Information
     The Computer Motherboard Test Sequence.seq file contains three sequences. The MainSequence calls the CPU Test and CPU Diagnostics sequences as subsequences.

   - In the Sequences pane, select the **MainSequence** sequence.

4. Confirm that the execution process model is correct for this exercise. In this case, you will use the Sequential execution process model.

   - Select **Configure » Station Options** and select the **Model** tab.
   - Use the *Station Model* drop-down to select **SequentialModel.seq** or click the browse button, to go to C:\Program Files\National Instruments\TestStand x.x\Components\NI\Models\TestStand Models and select **SequentialModel.seq**.

5. Execute a sequence using the Single Pass execution entry point.

   Additional Information
   Both the Single Pass and the Test UUTs execution entry points will execute the target sequence file and generate a report. The primary difference between the two is that the Test UUTs execution entry point provides prompts and a looping mechanism to test multiple UUTs in succession while Single Pass does not. The Test UUTs execution entry point repeats the sequence until the operator clicks **Stop** in the UUT Information dialog box.

   - Click the **Toggle Analyze File Before Executing** button on the toolbar to disable the TestStand Sequence Analyzer. This button now indicates that analysis is disabled, as shown in Figure 3.

Analysis Enabled:  **Analysis Disabled:** 

Figure 3. Toggle Analyze File Before Executing Button

   - Select **Execute»Single Pass**.

     Additional Information
     Because the computer you are using does not have equipment to test a computer motherboard attached, this sequence simulates the tests. Use the simulator dialog box select which tests should pass or fail. The selections you make in this dialog box are passed to the LabVIEW VIs that execute the tests. The VIs use this information to determine whether to return a passing result or a failing result.

   - When the Motherboard Test Simulator dialog box appears, disable all the checkboxes.
   - Click **OK** to exit the Motherboard Test Simulator dialog box.
   - When the execution completes, notice that the Execution window displays on the Report pane a report that describes the test results. The sequence and all of the steps have a status of **Passed** because you did not enable any items in the simulator dialog box.

     Additional Information
     The Single Pass Execution entry point executes a process model around the sequence. In the current configuration, the primary purpose of the process model is to generate a report. However, the process model can perform other operations. For example, if you enable database logging, the process model would log the test results to a database.

   - When you finish viewing the report, right-click the top of the **Single Pass** tab of the Report pane and select **Close**.

     Note: You can also dismiss all completed execution tabs using the shortcut <Ctrl-D>.

6. Execute a sequence using the Test UUTs Execution entry point.

- Select **Execute»Test UUTs**. When the UUT Information dialog box appears, enter **2-2A** in the **Enter UUT Serial Number** textbox.

  Additional Information
  Use the UUT serial number to identify each UUT on which you run the test sequence. You can enter any or no value, but consider specifying a descriptive identifier so you can accurately associate the test results with the UUT in the report.

- Click **OK** to exit the UUT Information dialog box.
- When the Motherboard Test Simulator dialog box appears, verify that none of the checkboxes are enabled.
- Click **OK** to exit the Motherboard Test Simulator dialog box.
- When execution completes, notice that a UUT Result dialog box displays the status of the test.
- Click **OK** to exit the UUT Result dialog box.
- When the UUT Information dialog box appears, enter **2-2B** in the Enter UUT Serial Number textbox.
- Click **OK** to exit the UUT Information dialog box.
- When the Motherboard Test Simulator dialog box appears, enable the CPU test checkbox.
- Click **OK** to exit the Motherboard Test Simulator dialog box.
- When execution completes, notice that the UUT Result dialog box displays a failed result.
- Click **OK** to exit the UUT Result dialog box.
- When the UUT Information dialog box appears, click **Stop**.
- When the test report displays, scroll through the report and review the entries for each UUT you tested.
- When you finish viewing the report, close the completed execution windows by pressing <Ctrl-D>.
- Go to **File»Close Computer Motherboard Test Sequence.seq** to close the example.

End of Exercise 2

Exercise 3: Add Tests Using NI TestStand Sequence Editor

Objective: To create a TestStand sequence and add steps to the MainSequence that tests a filter and LED on the device under test (DUT). Figure 1 shows the results of this exercise.

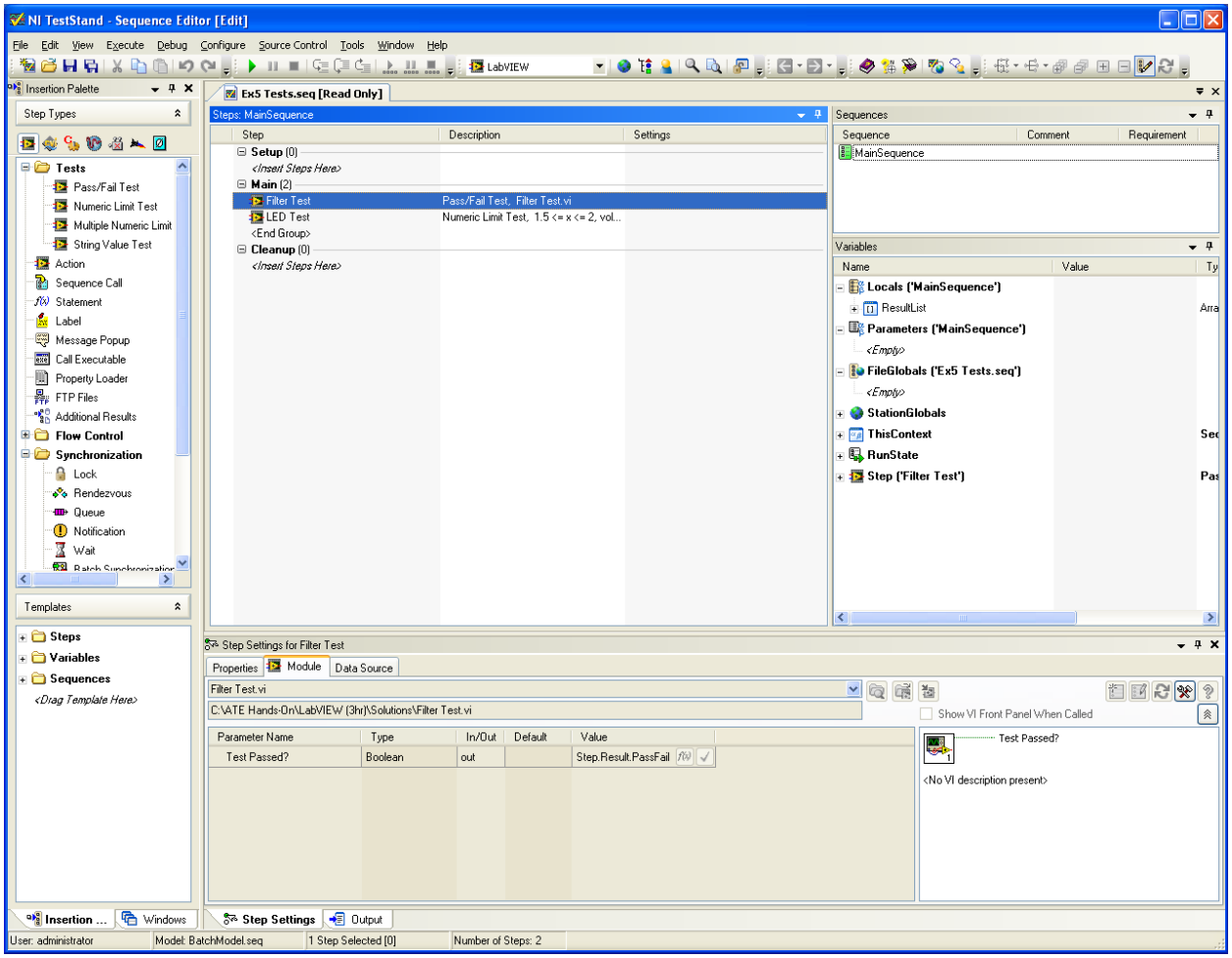


Figure 4. Completed exercise

Hardware: Make sure that the following **PXI-4071 DMM** signal connections have been made, or make them now.

1. The **HI (INPUT V Ohms)** connector of the PXI-4071 connects to **Red banana receptacle of the 5 DUT Switch Demo box** using a red banana cable.
2. The **LO (INPUT V Ohms)** connector of the PXI-4071 connects to **Black banana receptacle of the 5 DUT Switch Demo box** using a black banana cable.

The following connections should already be made and will remain the same:

- The **CH0** SMB connector of the PXIe-5442 connects to **Left-most BNC** of the 5 DUT Switch Demo box using an SMB-112 cable.
- The **CH1** BNC connector of the PXIe-5122 connects to the **Upper-left BNC of the Dual-BNC panelette** of the 5 DUT Switch Demo box using a BNC cable.
- The **TB-2641** terminal block of the demo box connects to the **PXI-2532**. Make sure to screw the terminal block all the way in and that it is firmly seated.
- Plug in four of the **Filter/LED Demo DUTS** into **DUT0, DUT1, DUT2,** and **DUT3** of the 5 DUT Switch Demo box.

  NOTE: Make sure that the **jumper** on each DUT is set to **NONE**.

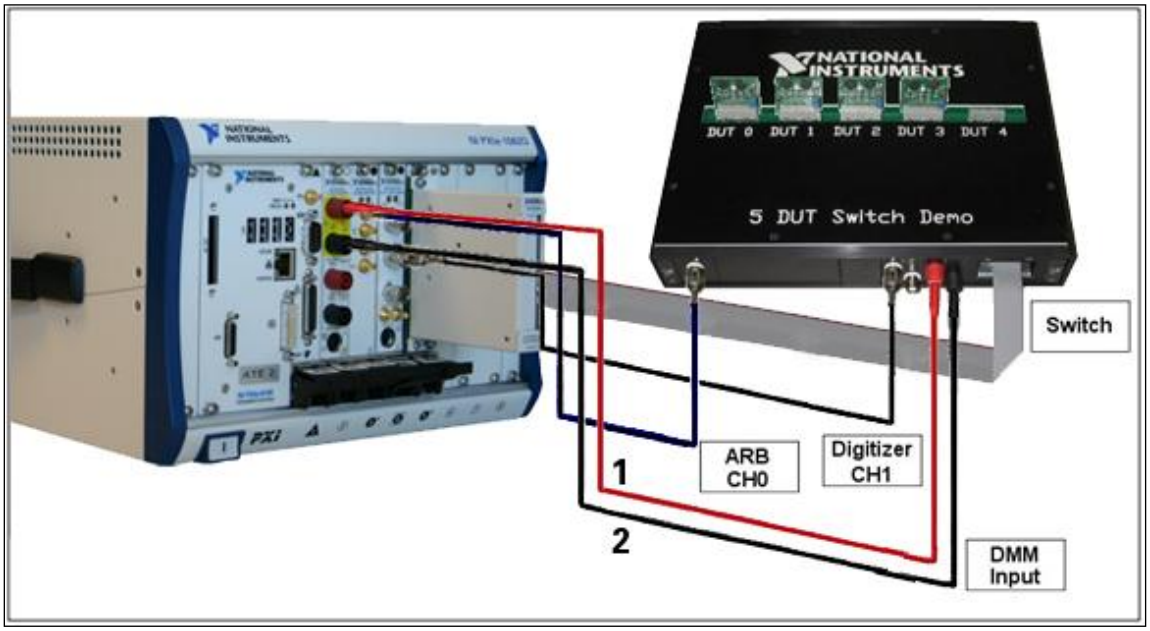


Figure 5. Hardware Connections

Exercise 3A: Create a TestStand sequence and add the Filter Test step

1. If TestStand is not already launched, select **Start » All Programs » National Instruments TestStand x.x » Sequence Editor**

2. Close all open sequences and create a new one by pressing **File » New » Sequence File**, or *Ctrl+N*.

3. First, you will select the LabVIEW adapter which allows you to call and run LabVIEW VI code modules from TestStand. It should be selected since the previous exercise, but if it is not, select the **LabVIEW** adapter from the *Adapter* ring control.

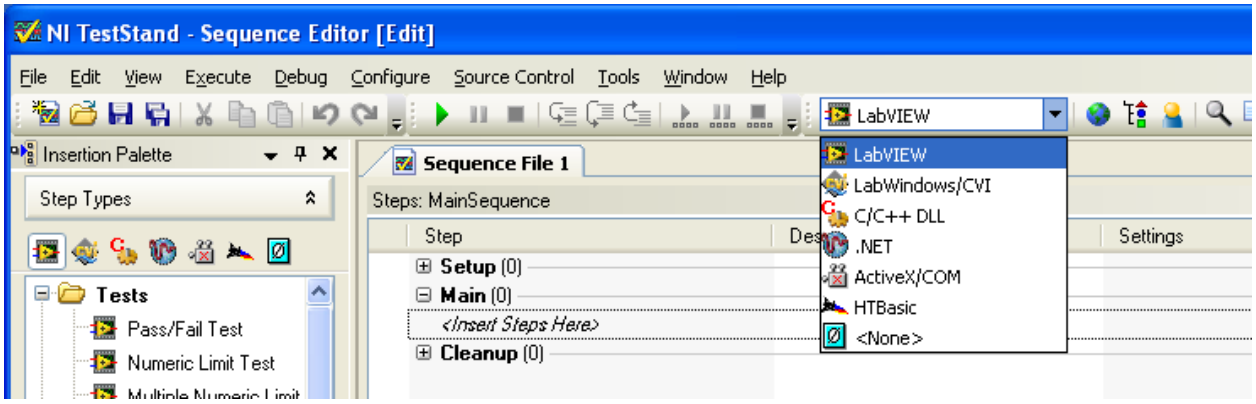


Figure 4 Adapter Ring Control

4. Confirm that the execution process model is correct for this exercise. In this case, you will use the Sequential execution process model.
    a. Select **Configure » Station Options** and select the **Model** tab.
    b. Use the *Station Model* drop-down to select **SequentialModel.seq** or click the browse button, to go to `C:\Program Files\National Instruments\TestStand x.x\Components\NI\Models\TestStand Models` and select **SequentialModel.seq**.

5. Now, you will insert your first step in the test sequence, which will test the filter on the DUT. The test will return a Boolean value for the pass/fail result. To test this Boolean value we will use a Pass/Fail test.
6. Expand the **Tests** group in the **Insertion Palette,** highlighted with the solid line in Figure 5, to show all the available tests.
7. Drag the **Pass/Fail** test from Insertion Palette to the **MainSequence** as shown in Figure 5.

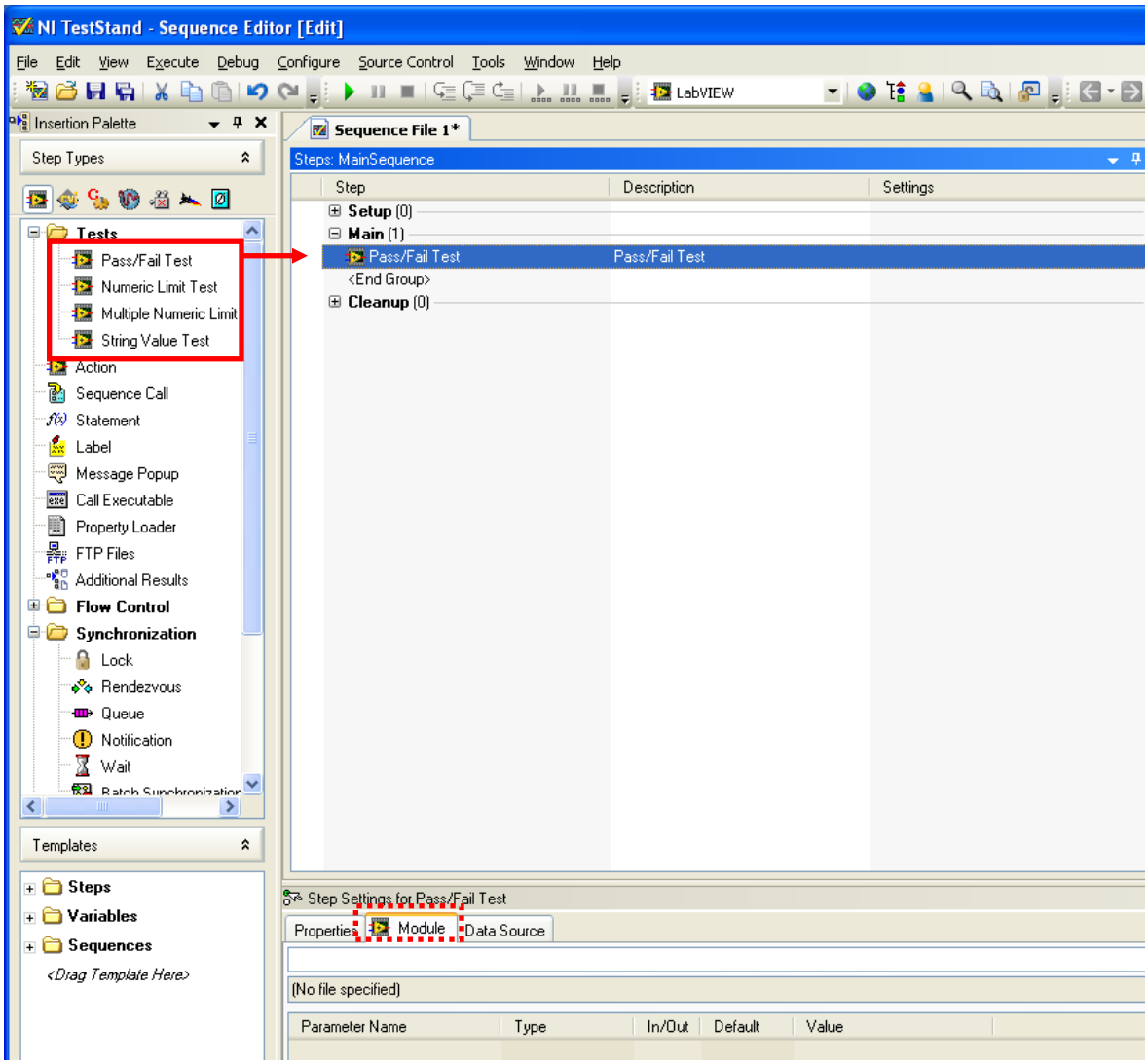


Figure 5 Addition of Filter Test step

8. Rename the step **Filter Test.**
9. Now that you've selected the type of step, you will need to specify the LabVIEW VI this step will call. You will use the LabVIEW VI that you saved earlier in this exercise as the code module for this test.
    a. Select the **Module** tab in **Step Settings** pane highlighted with a dashed line in Figure 5.

b. Click the **File Browse** button  on the right side of the blank file name selection.

c. Select the `Filter Test.vi` file from your Exercises directory and click **OK**.

   This is the same VI as you saw in Exercise 1, but it has been resized to only show the signal graph.

d. If you are prompted with File FilterTest.vi Not Found dialog window, leave the default selection of "Add the directory containing the file you selected to the list of search directories", and click **OK**.

10. Now that you have selected the VI, you will need to define where to store the test result in TestStand.

    a. Notice that on the right side of the window there is now an image of the VI icon with an output terminal called "Test Passed?". This is something that has been preconfigured in LabVIEW.

    b. Click the Expression Browse button, *f(x)* button  , to the right of the *Value* field.

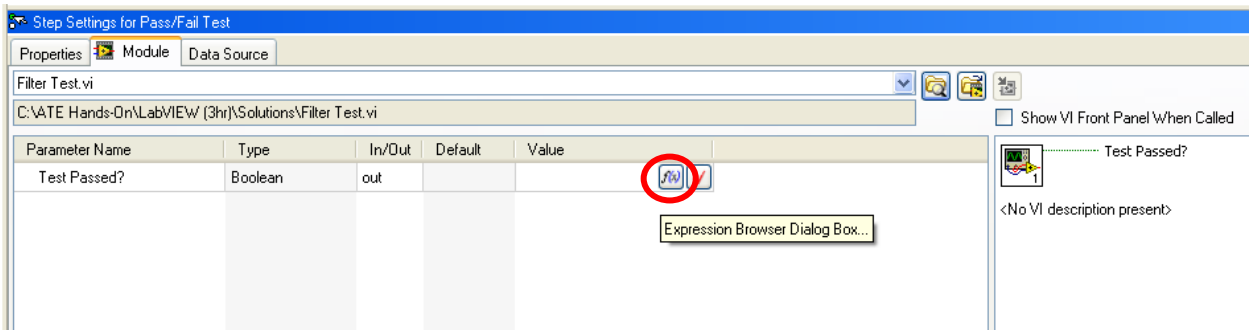


Figure 6 Step Settings for Filter Test window

c. This is the *Expression Browser* dialog box, which you can use to interactively build an expression by selecting from lists of variables, properties, operators, and functions.

d. In this case, navigate the properties of the step and select `Step.Result.PassFail`. Click **Insert** to select that property.

e. `Step.Result.PassFail` is the TestStand property that will now contain the Test Passed value.

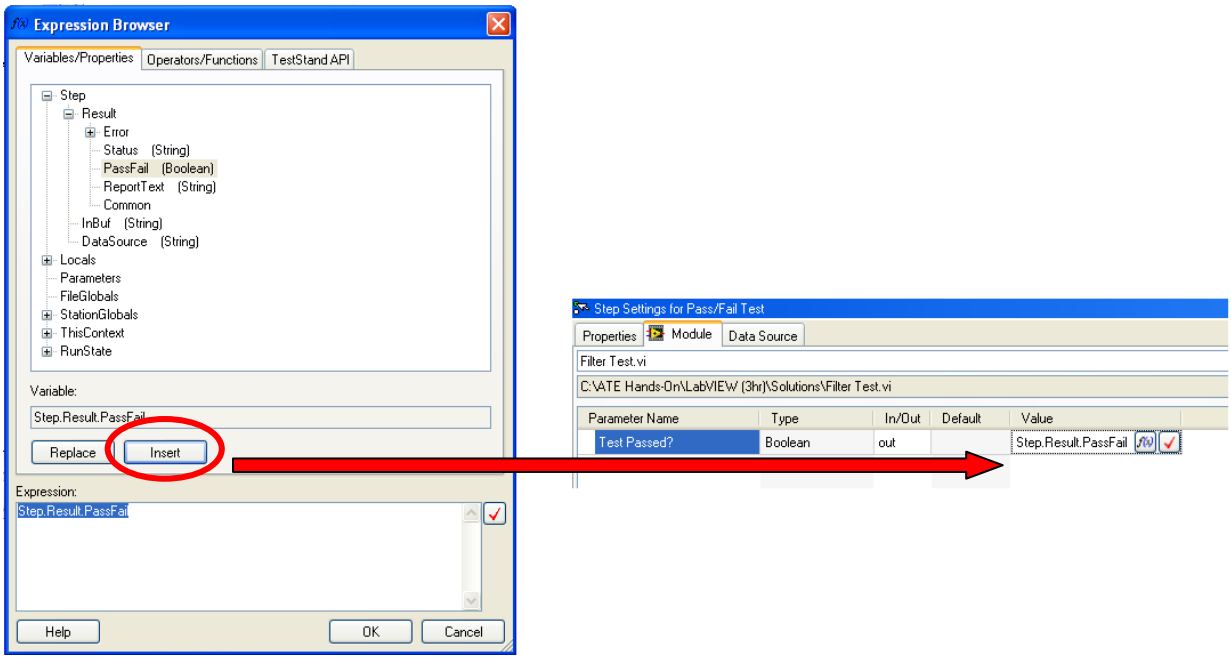


Figure 7 Use Expression Browser to build an expression

f. Click **OK** to return to the Sequence Editor.

11. You have finished adding the first step to your test sequence.

Exercise 3B: Add the LED Test step and interactively connect routes

1. Next, you will add a second step which will test the LED on the DUT. The test will return a numeric value for the voltage that you will want to compare to limits, so the type of test you want is a numeric limit test.

    a. Drag the Numeric Limit test from the **Insertion Palette** to the **MainSequence** as shown in Figure 8.

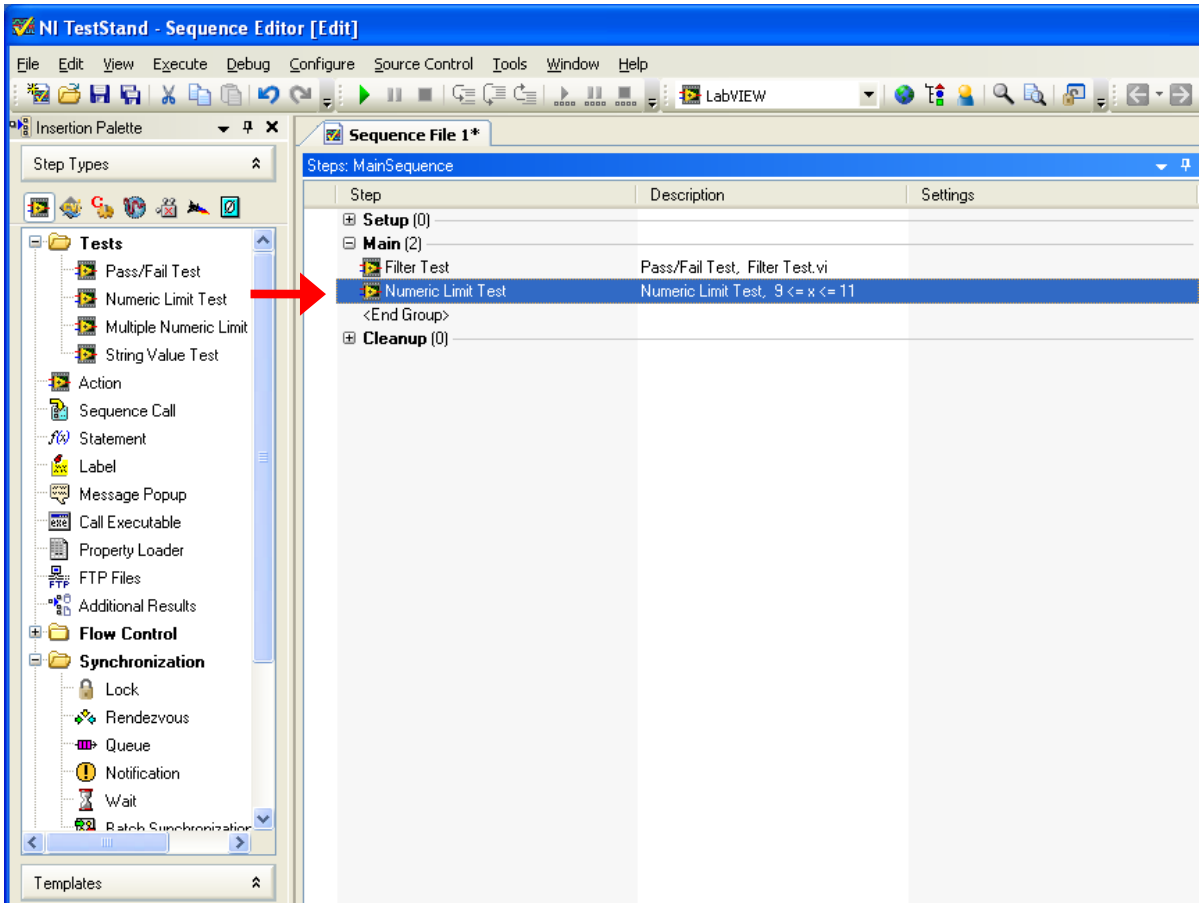


Figure 8. Addition of LED Test step

    b. Rename the step **LED Test.**

2. Just as you did for the previous test step, you will need to specify which LabVIEW VI this step calls.

    a. Select the **Module** tab in the Step Settings for LED Test window.

    Tip: LabVIEW Express VIs can be called directly by TestStand. In this case, you will create a test step yourself using an niDMM Express VI to specify acquisition properties and view the resulting data in the dialog box before inserting the VI into your sequence.

    b. Click on the **Select Express VI** icon  in the top-right corner of the **Module** tab

    c. Select **Measurement I/O » NI-DMM » NI-DMM/Switch Express**.

d. Configure the NI-DMM Express dialog box as shown in Figure 9.

Configuration Option	Value
Measurement function	Diode

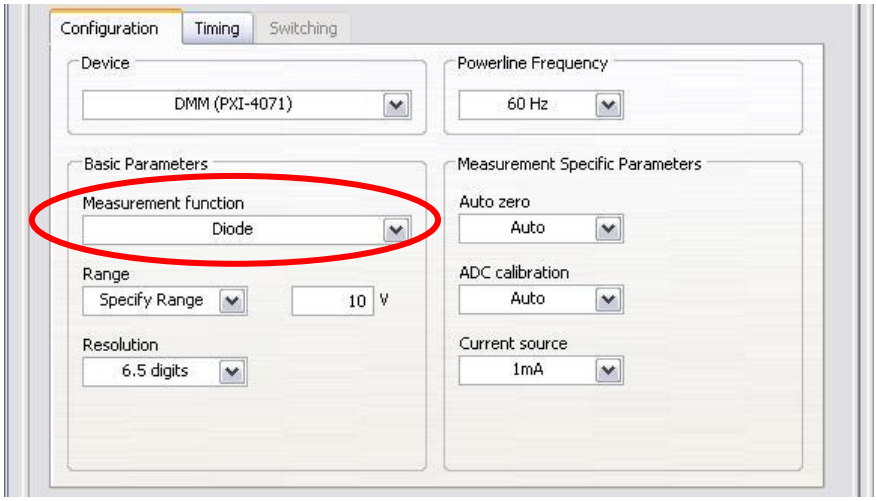


Figure 9 NI-DMM/Switch Express Dialog Box

IMPORTANT: If you were to run the NI-DMM Express VI now, you would receive a message (OVLD V Diode) indicating that the signal is over range. This is because the DMM hasn't been connected to the LED and it is measuring infinite resistance. So, to quickly prototype our system, you need to use the switch to connect the DMM to the LED on DUT0. You will connect the PXI instruments to the LED and filter on DUT0 in the next step.

e. Leave the NI-DMM Express VI open and continue.

3. In a previous exercise, you interactively made switch connections to test the filter on DUT0 using the **NI-Switch Soft Front Panel**. This time, you will make the necessary switch connections using a higher-level switch configuration environment, called **NI Switch Executive**, to connect PXI instruments to both the filter and the LED on DUT0.

   a. Select **Start » All Programs » National Instruments » Switch Executive » MAX with NI Switch Executive.**

   b. Double click on **BuildATE**.

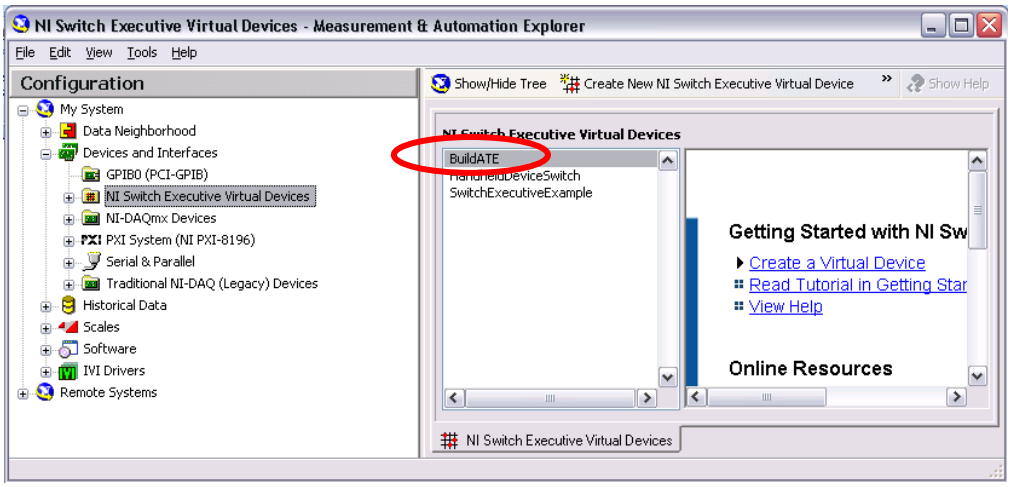


Figure 10 NI Switch Executive

c. Click **Show/Hide Tree** to maximize the view of the virtual switch device.

This virtual switch device has been created specifically for this demo. All of the channels have been named, and all of the routes and route groups have been configured.

d. Click on **Test Panel…**.

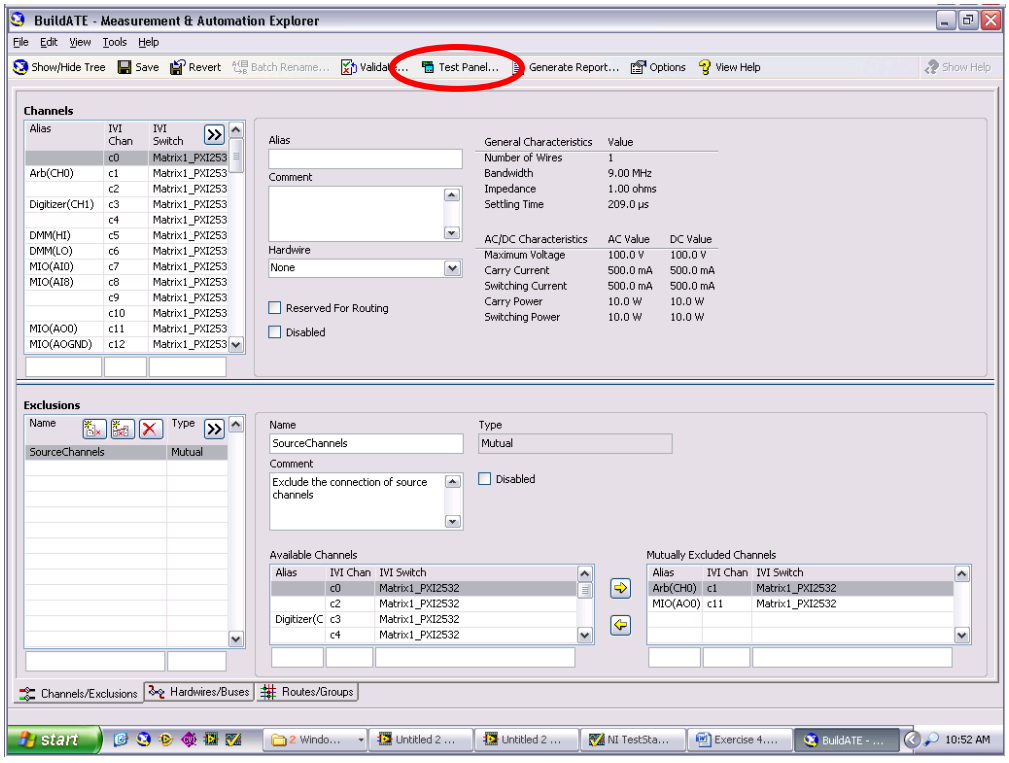


Figure 11 BuildATE Virtual Device

The NI Switch Executive Test Panel allows you to interactively select and test individual routes and route groups associated with a virtual device. In this case, the virtual device is called BuildATE.

To connect the filter on DUT0:

e. In the Routes and Route Groups view, scroll down to the first route group called **FilterTest_DUT0**.

f. Click on the **Connect Selected Route or Route Group** button. The FilterTest_DUT0 route group has now been connected.

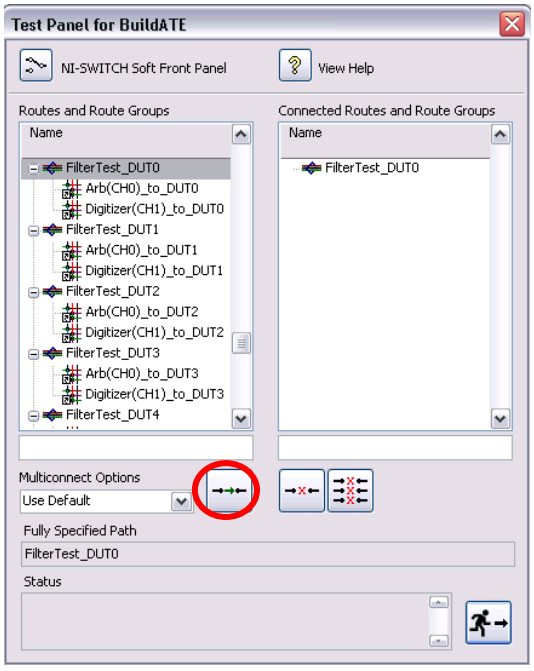


Figure 12 Test Panel for 5DUTSwitchDemo

To connect the LED on DUT0:

g. Scroll down to the route group called **LED1Test_DUT0** and select it.

h. Click on the **Connect Selected Route or Route Group** button. The LED1Test_DUT0 route group has now been connected.

i. Now click on the Launch Debug Panel Button 

j. Now click on the **NI-SWITCH Soft Front Panel** button  to view the routes that you just connected.

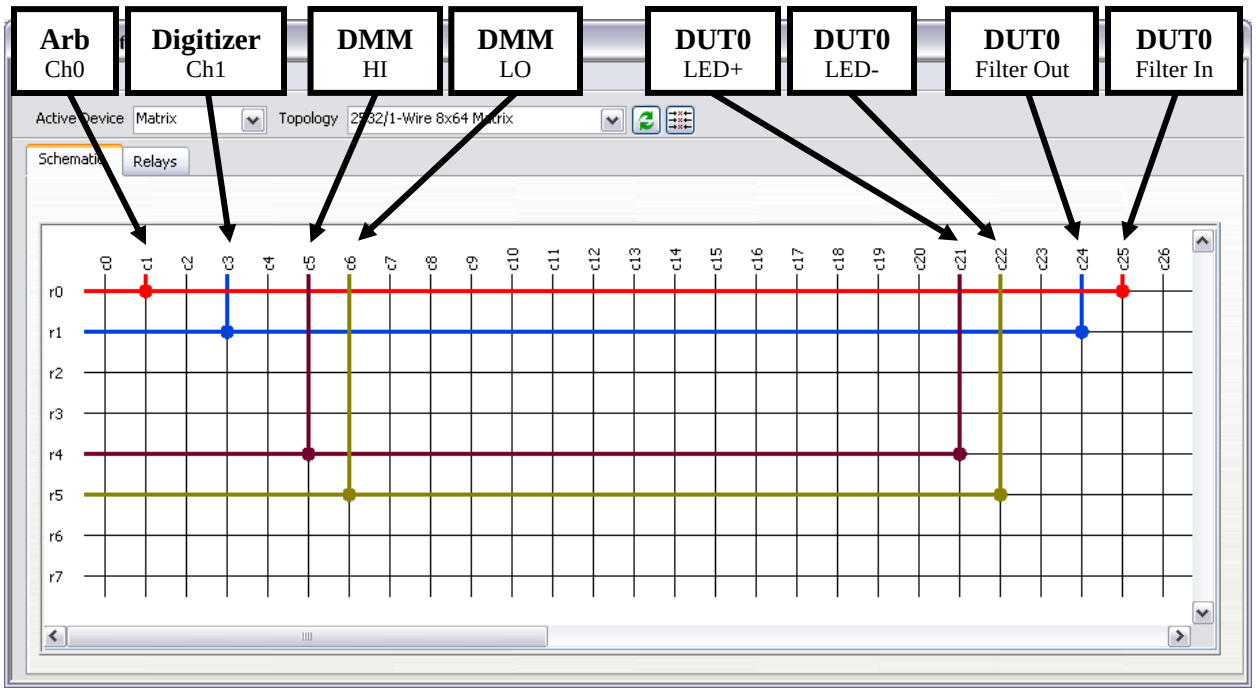


Figure 13 Switch Front Panel showing configurations

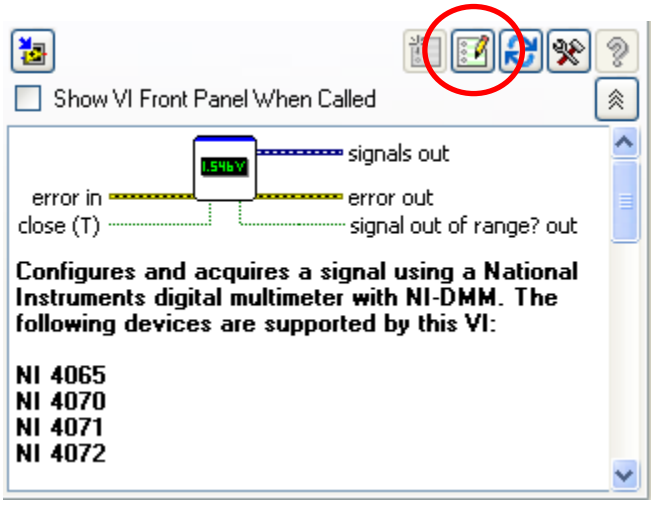
k. Notice the physical switch connections that were made when you connected the routes. The FilterTest_DUT0 route connected the same routes that you connected manually in the previous exercise. This is how NI Switch Executive simplifies route connections.

l. **Close** the NI-SWITCH Soft Front Panel window.

m. Click the **Exit Test Panel** button  to close the test panel. The route groups will stay connected.

n. **Close** the Measurement & Automation Explorer window.

4. Return to the NI-DMM Express dialog box by clicking on the 'Configure VI' icon under the module tab within TestStand, as shown below:



5. Select **Run Once**. If the LED is connected properly and is functional, the displayed value will be around **1.6 Volts**. Also notice that the diode on LED0 is lit.

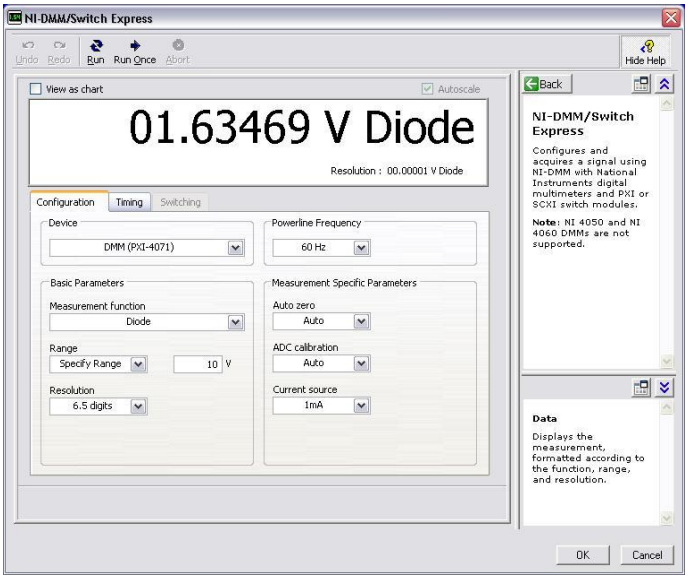


Figure 14 NI-DMM/Switch Express Dialog Box

6. Click the **OK** button to return to the Sequence Editor.
7. Next you will enter an expression that specifies where TestStand stores the numeric data that the VI outputs.

    a. In the *Value* field of the *signal out* terminal, type in `Step.Result.Numeric`.

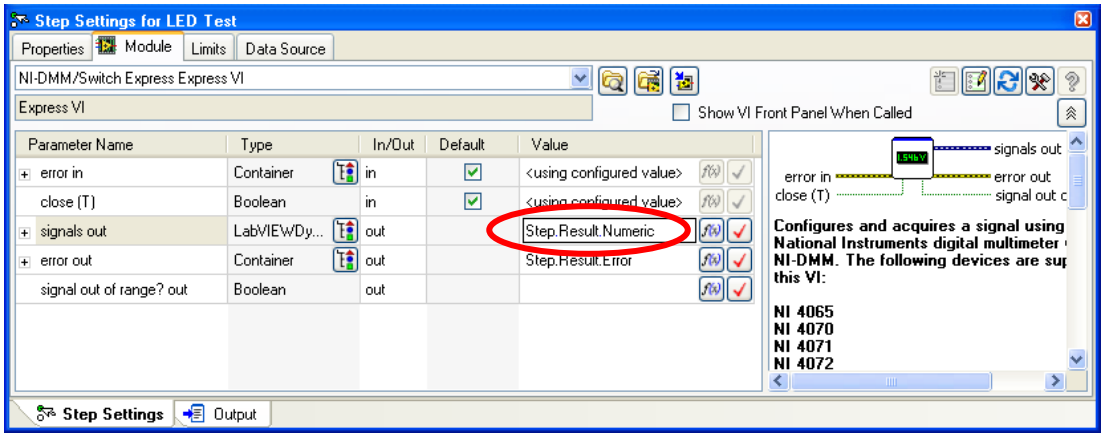


Figure 15 Step Settings for LED Test window

    b. `Step.Result.Numeric` is the TestStand property that will now contain the measured DMM value which will be compared to the limits we will define in the next step.

8. Now that the VI has been configured, you will want to define what values are within range and allow the test to pass.

    a. Click the **Limits** tab in the Step Settings for LED Test window.

    b. Set the limits in the Edit Numeric Limit Test dialog box to the values shown in Figure 16. Setting the limits in this way will have this step fail only if the returned measurement, the voltage value, is outside of 1.5V and 2.0V, which effectively tests for shorts and opens.

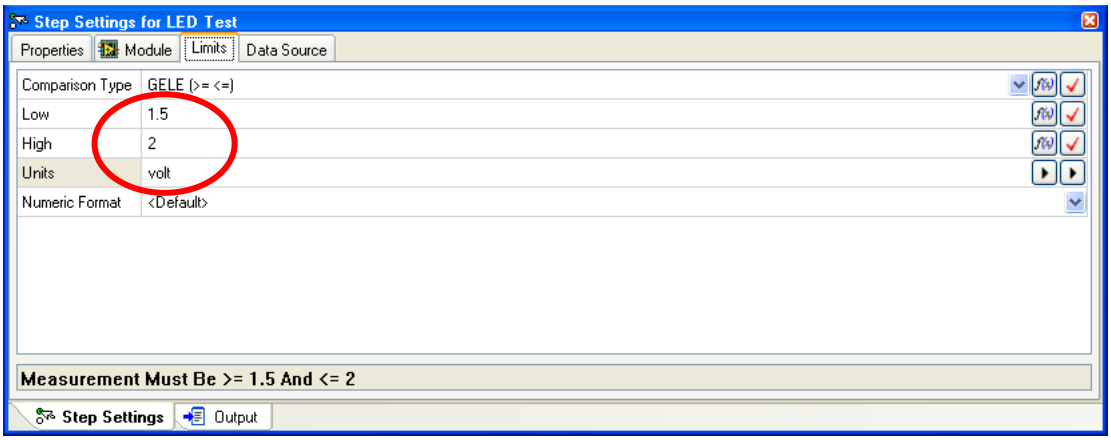


Figure 16 Step Settings for LED Test Dialog Box

9. You have finished adding the second step to your test sequence. To save the sequence file, select **File » Save**. Browse to `C:\Seminars\Automated Test System with NI TestStand\Exercises` and save the sequence as `Ex3 Tests.seq`.

10. Now that the tests are added and the proper switches have been connected manually, you can execute your test sequence. Select **Execute » Single Pass**. If prompted, select **Yes** to save the sequence file.

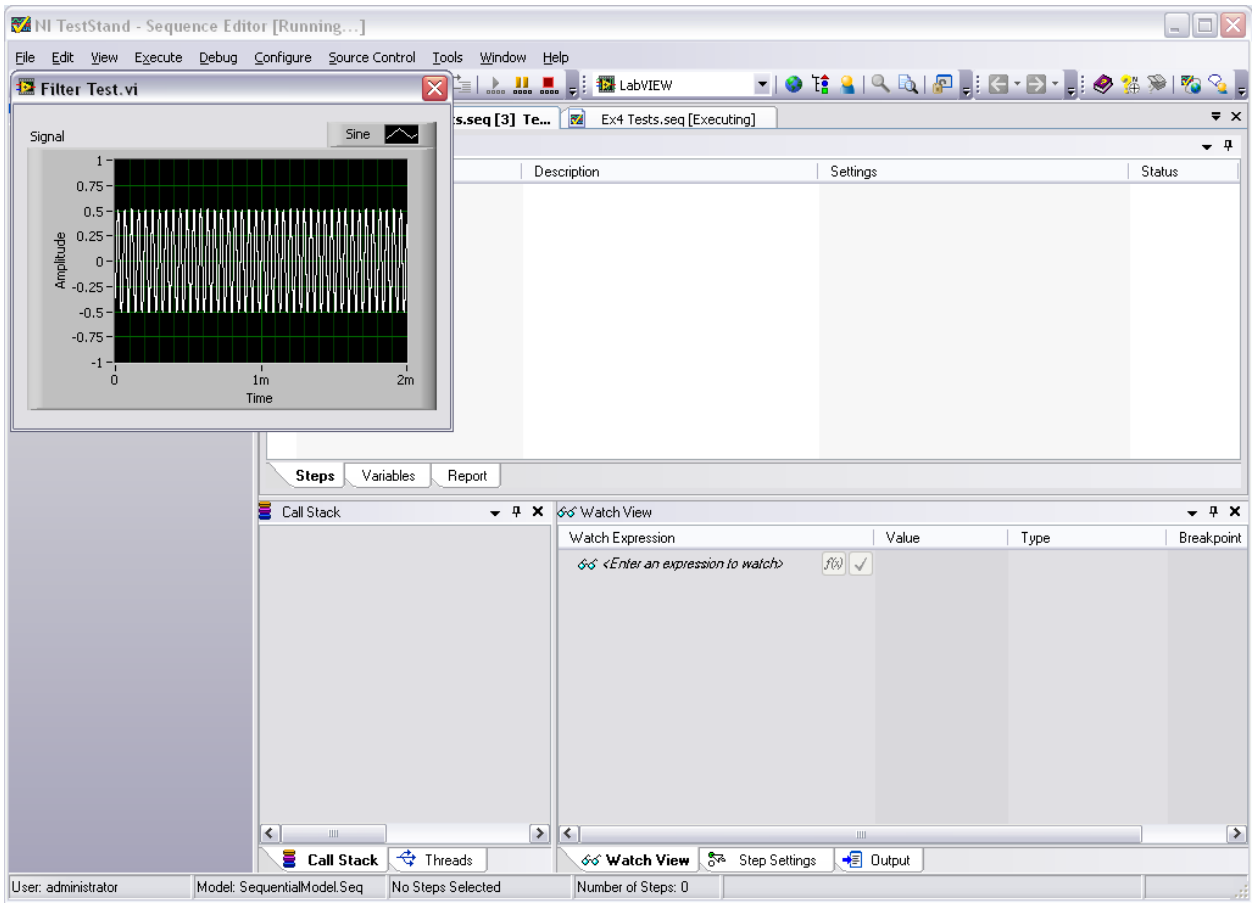


Figure 17 Sequence Editor Running

11. View the test results. Notice the status of the tests and the result of the LED Test measurement is displayed.

    NOTE: The LED will remain on after the test is complete.

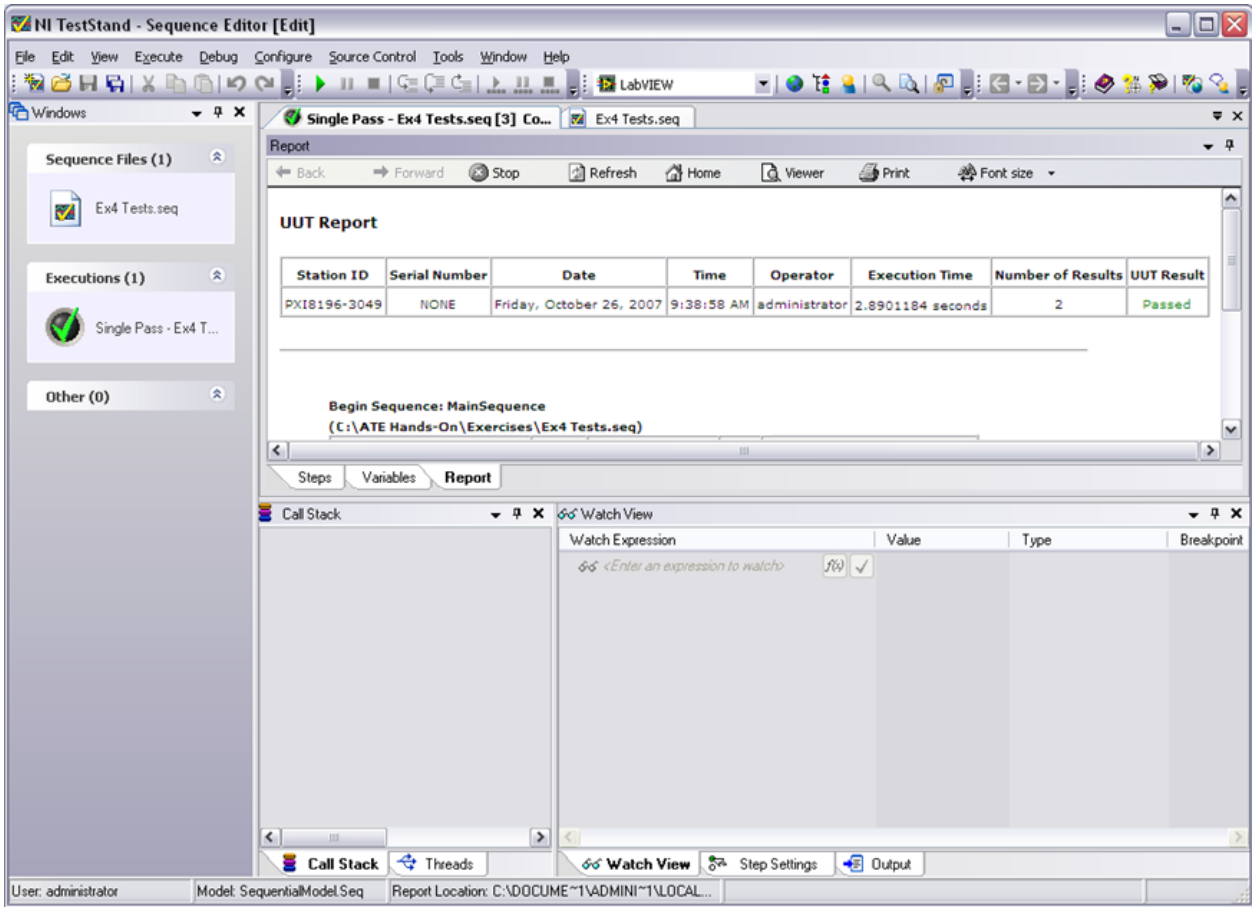


Figure 18 UUT Report

12. To run another test, click **Execute » Restart**.
13. Moving the jumper on DUT0 to a different position will make either the filter or LEDs fail. Move the jumper to **C4** and run the sequence again using **Execute » Restart**. View the test results to see that the Filter Test failed.

    Tip: Placing the jumper on the **C4** or **R2** jumper pins will cause the Filter Test to fail. Placing the jumper on the **LED** jumper pin will cause the LED Test to fail.

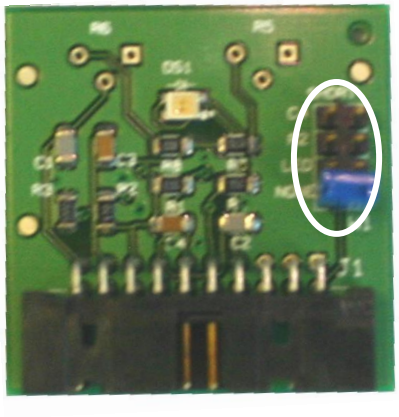


Figure 19 DUT jumpers

14. Place the jumper back on the **NONE** jumper pin when finished.
15. Close the Single Pass window by pressing the Close button on the window or selecting **Window » Close Completed Execution Displays** when you are finished.

End of Exercise 3

Exercise 4: Integrating Switching into the Sequence using NI Switch Executive

Objective: To add switching to each step such that the appropriate instruments connect to each component on each device under test (DUT). In this exercise, you will make use of the integration of NI Switch Executive in NI TestStand. Figure 1 shows the results of this exercise.

Additional Information: NI Switch Executive is an intelligent switch management and routing application. You can interactively configure and name switch modules, external connections, and signal routes using a graphical end-to-end route editor. This means you can preconfigure routes or autoroute signal endpoints on the switch. This exercise illustrates configuring a TestStand step to automatically perform the switching operations for the step based on a configuration created in NI Switch Executive.

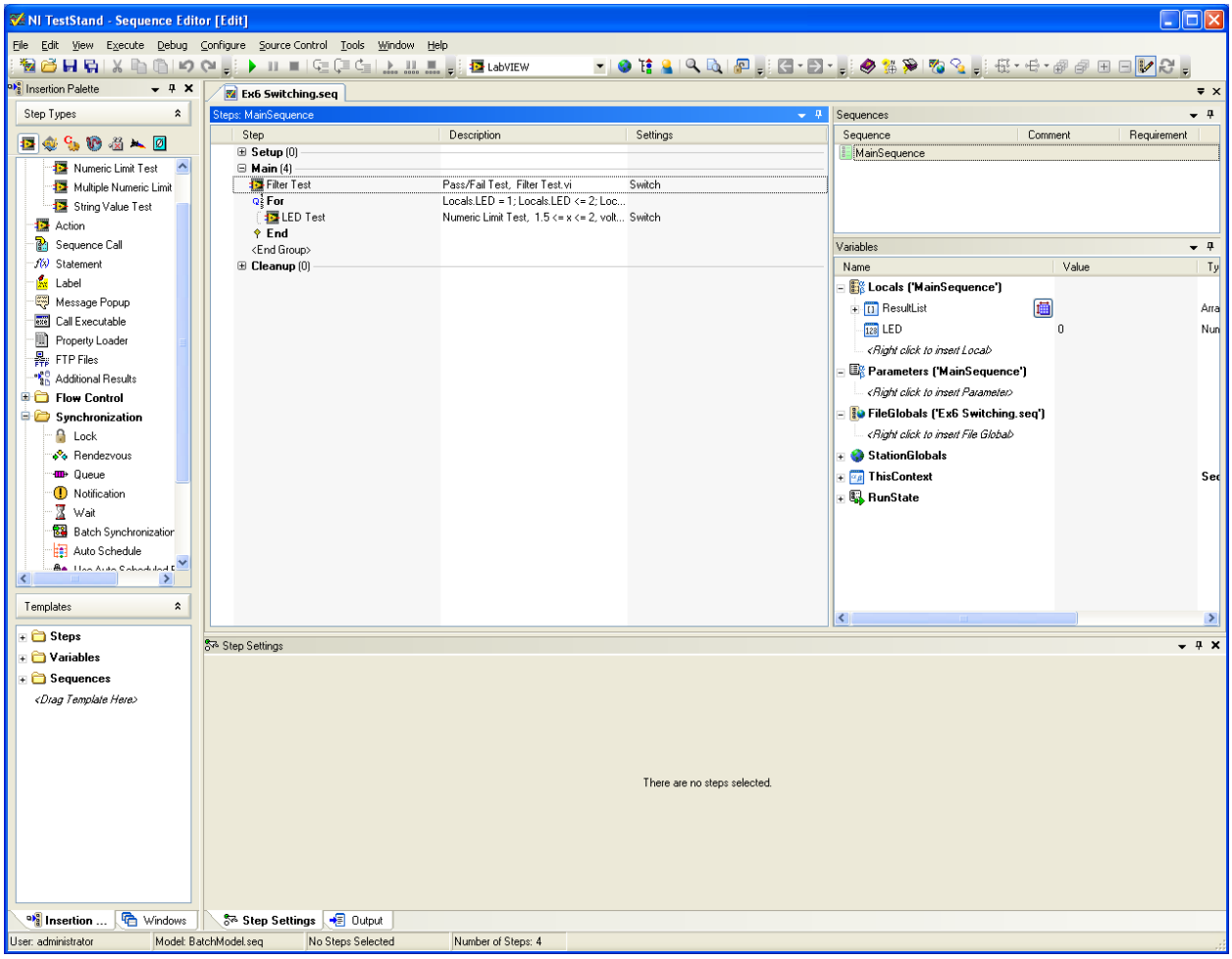


Figure 6. Completed exercise

Exercise 4A: Add switching to the Filter Test step

1. Follow these steps to make a working copy of the sequence that you created in the previous exercise.

    a. Open `Ex3 Tests.seq` in your Exercises folder. If you have not finished the last exercise, load the solution from `C:\Seminars\Automated Test System with NI TestStand\Solutions\Ex3 Tests.seq`

    b. Select **File » Save As…**.

    c. Go to your exercises folder, name the sequence as `Ex4 Switching.seq` and click **Save**.

In the previous exercise, you had to use Switch Executive route groups to connect the instruments to the DUT before you could execute the test sequence. With the tight integration of TestStand and Switch Executive, you can add switch routes to each step.

2. Now you will add a switch route to the first step in your sequence, the Filter Test.

    a. Click on the *Filter Test* step and select the **Properties** tab in the Step Settings for Filter Test window.

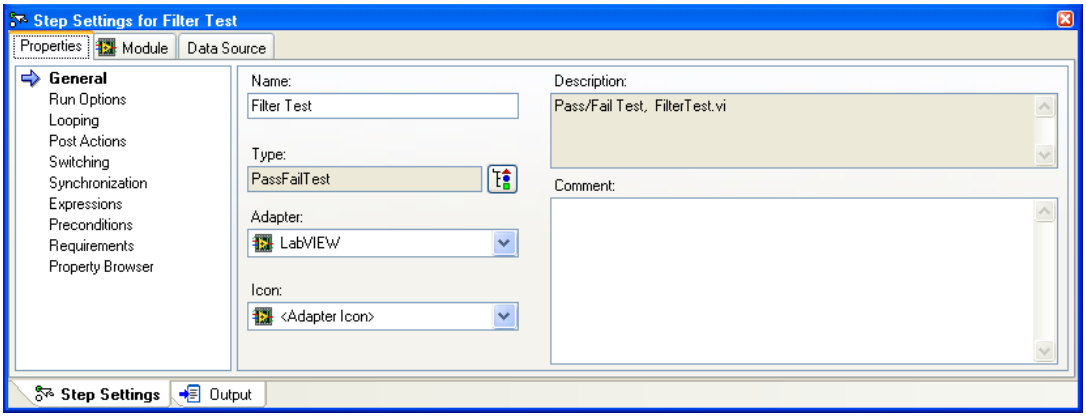


Figure 7. Properties Tab in Step Settings for Filter Test window

Tip: The Properties tab contains several options for configuring the step, including:

- When to load the step's module into memory
- What conditions will cause the step to execute
- Whether TestStand executes the step in a loop
- Ways to synchronize the step with other executions

b. Select **Switching**.

c. Select **Enable Switching**. Notice that the Filter Test step now has a "Switch" listed in the Flow Properties column.

d. Select **"BuildATE"** from the *Switch Executive Virtual Device* ring control.

   Now that you have selected the BuildATE virtual device, all of its routes and route groups are available for you to use. To test ONLY the filter on DUT0, the route group that you want to use is **FilterTest_DUT0**. You could simply select that route as the *Route(s) to Connect* and this test will work for fine for one DUT.

   But ultimately this sequence will need to be able to test multiple DUTs, and selecting a "static" route doesn't offer the scalability that you need when you want to use this same sequence to test multiple DUTs. Therefore, with the end result in mind, you will add this route using an expression so that the route used during the step will change for the DUT being tested.

e. Click on the **Expression Browser**, *f(x)* button  , next to *Route(s) to Connect*, and type the following expression.

```
"FilterTest_DUT" + str(RunState.TestSockets.MyIndex)
```

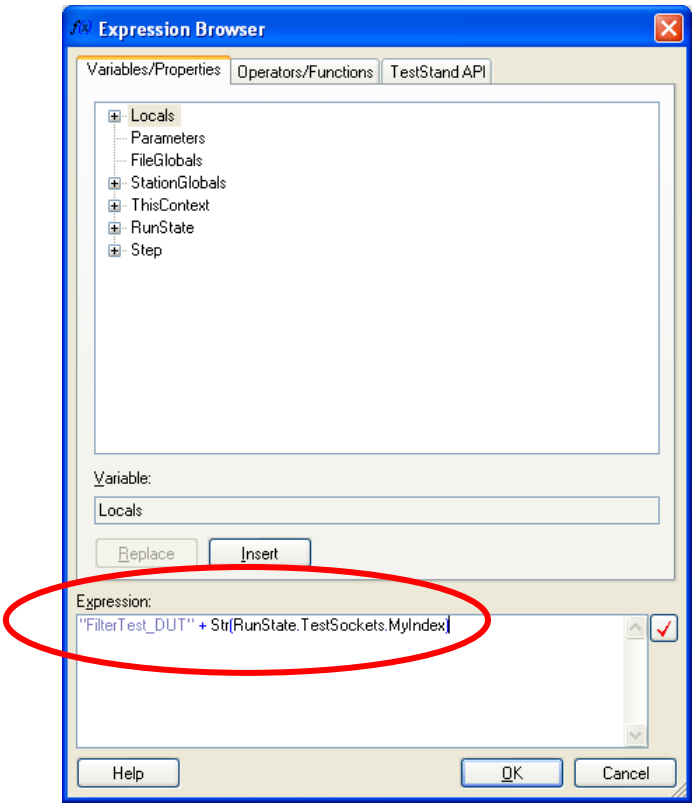


Figure 8. Expression for Filter Test switch route in Expression Browser Dialog Box

This expression is used because the switch route now depends on one thing: the DUT being tested. All of the possible path combinations have already been pre-defined in the virtual device: FilterTest_DUT0, FilterTest_DUT1, FilterTest_DUT2, and FilterTest_DUT3.

`RunState.TestSockets.MyIndex` is the index of the current execution in TestStand which equates to the DUT to be tested. Remember, this expression is used in anticipation of the next exercise in which multiple DUTs will be tested.

f. Click on the Check Expression for Errors button to confirm that the expression syntax is correct. If you receive an Evaluation or Syntax Error, determine and correct the problem before continuing.

g. Click **OK** to return to the Sequence Editor. Change the *Connection Lifetime* to **Step** so that the route will disconnect when the step completes. The Properties dialog box should now match the next figure.

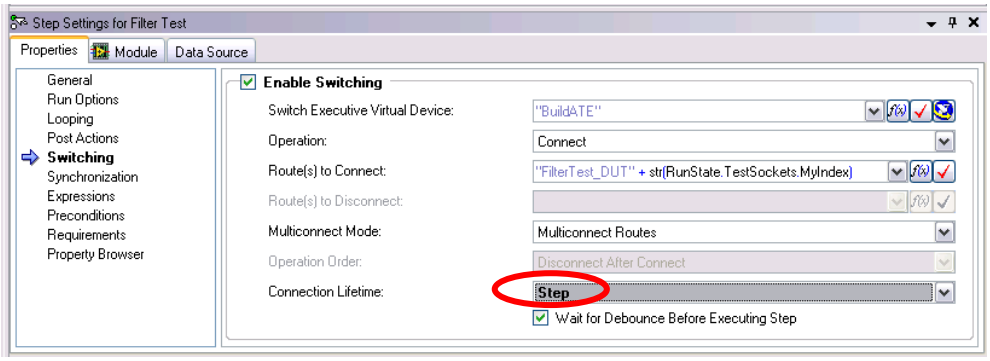


Figure 9. Filter Test Properties for Switching

3. You have successfully added switching to the Filter Test. Now continue onto Exercise 4B.

Exercise 4B: Add switching to the LED Test step

In the previous exercise you tested one LED on the DUT, but there are actually two LEDs that need to be tested on each DUT. Both LEDs can be tested in the same way, using the DMM Express VI, with the only difference being where the DMM is connected on the DUT. There are a couple ways to add this test.

One way to add this test is to add a second LED Test step and add a different switch route that only changes on one criteria: the DUT count (similar to the Filter Test step). As more LEDs are added to the DUT, this approach requires more steps to added as well.

Another way to add this test do this is to use the same test step to test both LEDs, and configure the switch route for the step so that it changes based on two criteria: which DUT is being tested **AND** which LED is being tested. This solution is more scalable than the first in that for every new LED that is added to the DUT, only a new switch route and another loop iteration is needed in order to test it.

For this exercise, you will use the second method. This method also allows you to use the For Loop condition in TestStand and build and even more flexible switch expression.

1. To begin, you will need the LED Test to execute twice. TestStand offers several methods to loop on the execution of a step based on different criteria, including using the Properties of the step. For this exercise, you will use the **For** flow control step type.

    a. Right-click on *LED Test* and select **Insert Step » Flow Control » For**. Inserting a For loop will test both of the LEDs provided the step has the correct switching properties based on the loop index.

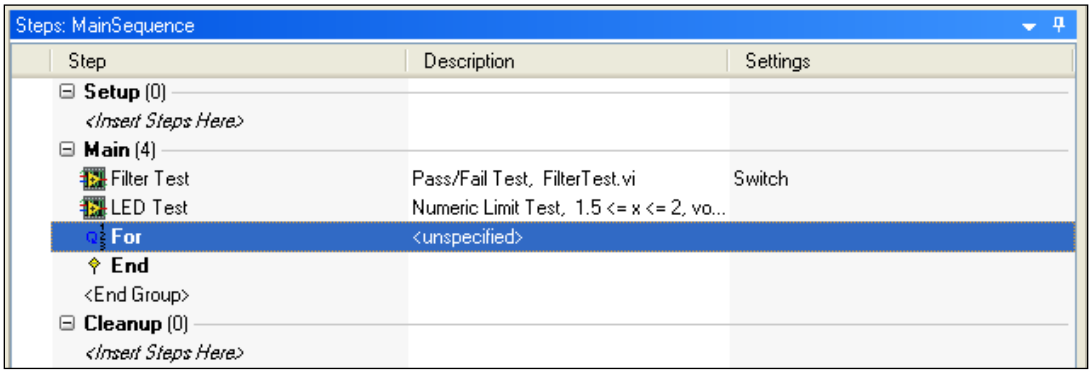


Figure 10. Insert For Loop

b. Select and drag the **LED Test** step after the **For** step and before the **End** step. If done correctly, the **LED Test** will be placed in the middle of the **For Loop**. Your sequence should match the following figure.

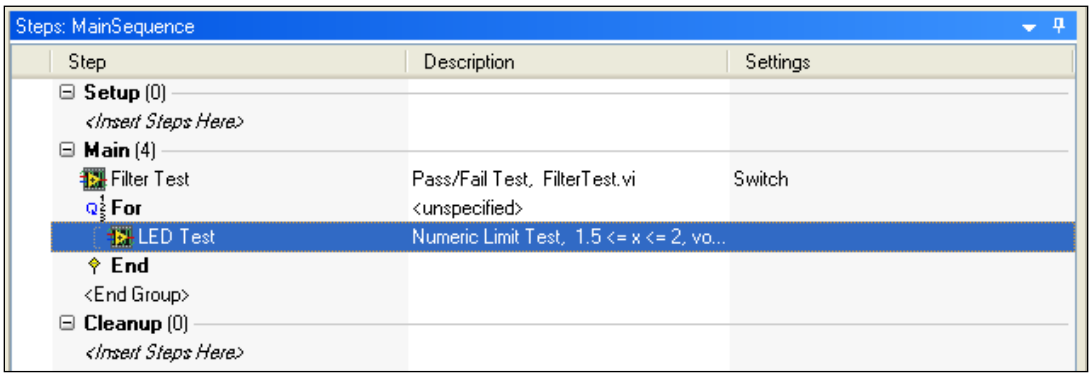


Figure 11. LED Test inside For Loop

2. Now you will configure the For Loop to loop two times. In order to do that, first you will create a local variable.

   a. Ensuring that the LED Test step still selected, Right-click in the variables side window, and select **Insert Local » Number**, as shown in Figure 7

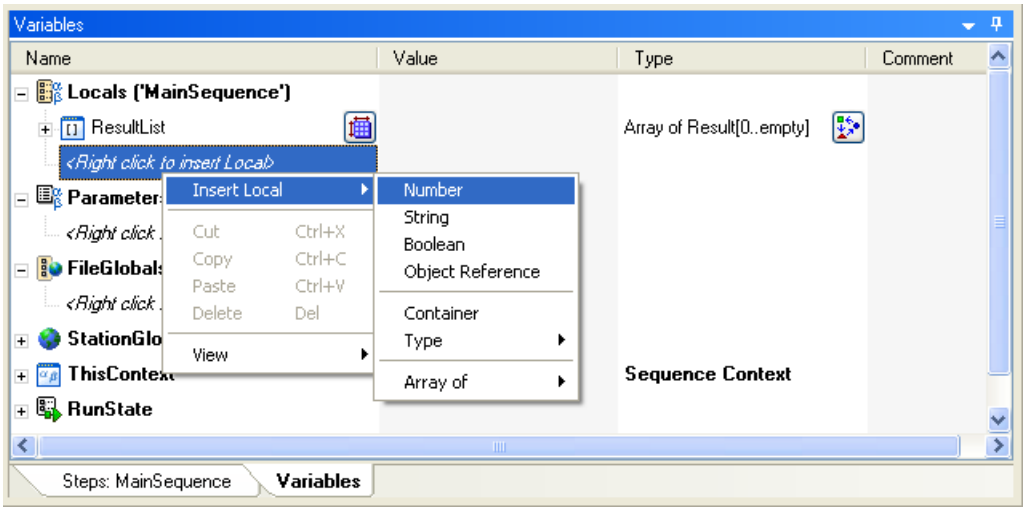


Figure 7 Add new local variable

b. Name the variable **LED**. This variable will be used to initialize the For Loop.

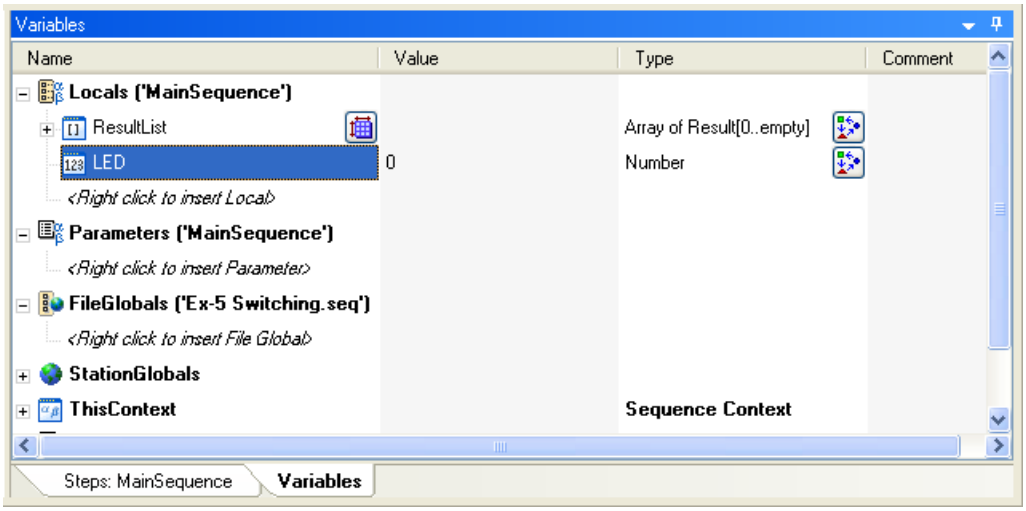


Figure 8 LED local variable

3. Now you will use the LED local variable that you just created to store the value of the loop iterations.
   a. Click on the **Steps: MainSequence** tab of the sequence editor.
   b. Select the **For Loop** tab in the Step Settings for For window.

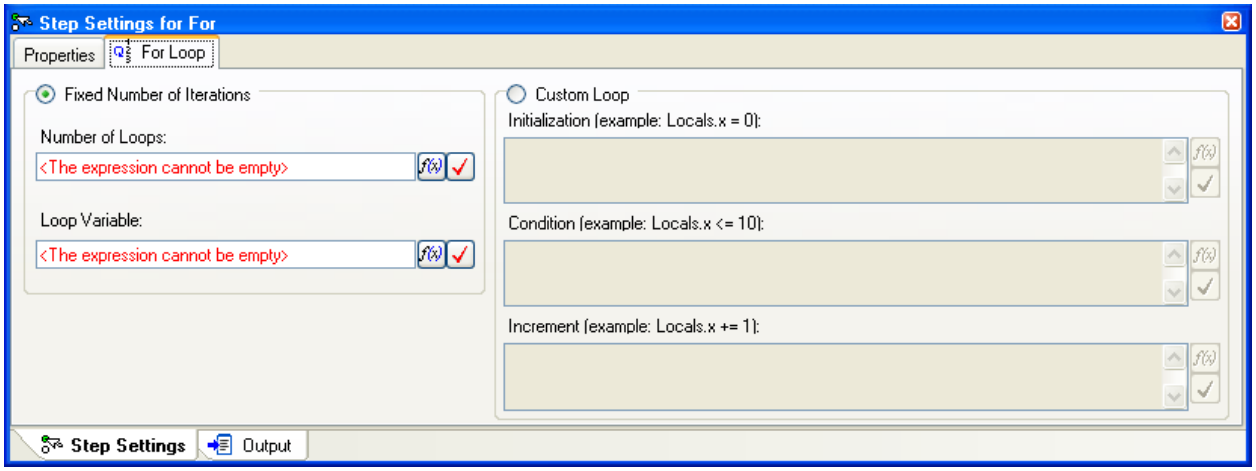


Figure 9 Configure For Loop

   c. Select **Custom Loop** to configure the loop properties using `Locals.LED`.
      i. For the **Initialization** of the Loop, type `Locals.LED = 1`
      ii. For the **Condition** of the Loop, type `Locals.LED <= 2`
      iii. For the **Increment** of the Loop, type `Locals.LED += 1`

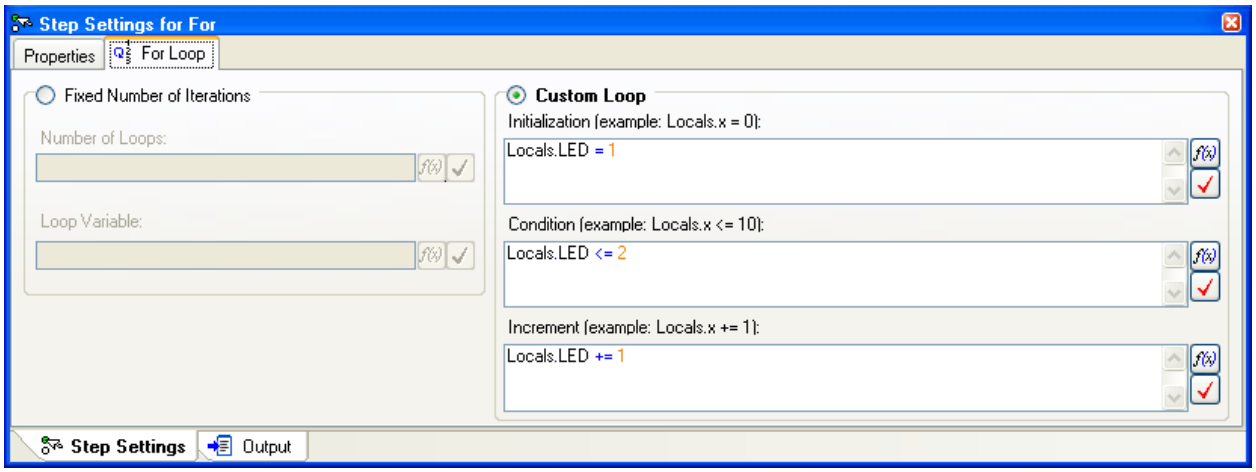


Figure 12 Configure For Loop Dialog Box

d. Click on each of the Check Expression for Errors buttons to confirm that the expression syntaxes are correct. If you receive an Evaluation or Syntax Error, determine and correct the problem before continuing.

4. Now that the step is configured to execute twice, you will need to define which route group is used for each of the two iterations of the loop.

   a. Select the *LED Test* step and click on the **Properties** tab in the Step Settings for the LED Test window.

   b. Select **Switching**.

   c. Select **Enable Switching**.

   d. Select **"BuildATE"** from the *Switch Executive Virtual Device* ring control.

   To test ONLY the two LEDs on DUT0, the two route groups that you want to use for this test are **LED1Test_DUT0** and **LED2Test_DUT0**. If you weren't concerned with testing multiple DUTs later, you could create an expression to use route LED1Test_DUT0 on the first loop iteration and LED2Test_DUT0 on the second.

   But ultimately this sequence will need to be able to test multiple DUTs. Therefore, you will add this route using an expression so that the route used during the step will change for both the LED and the DUT being tested. Using an expression help to keep your switching scalable as your test needs change.

e. Click on the **Expression Browser**, *f(x)* button  , next to *Route(s) to Connect*, and type the following expression.

```
"LED" + str(Locals.LED) + "Test_DUT" + str(RunState.TestSockets.MyIndex)
```

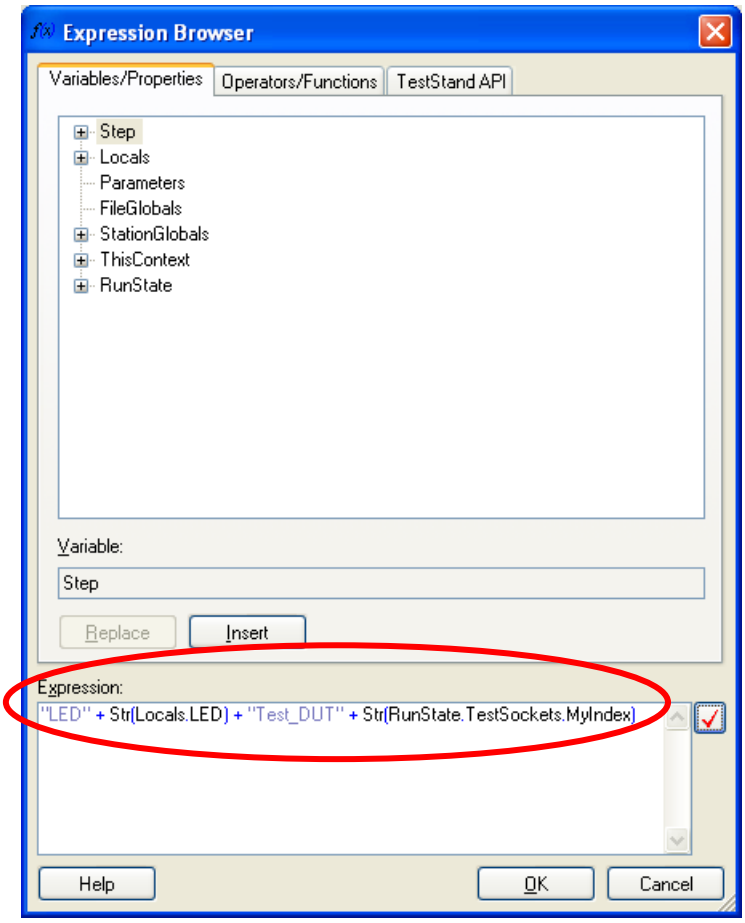


Figure 13 Expression for LED Test switch route in Expression Browser Dialog Box

This expression is used because the switch route depends on 2 things: the LED being tested and the DUT being tested. All of the possible path combinations have already been pre-defined in the virtual device: LED1Test_DUT0, LED2Test_DUT0, LEDTest1_DUT1, . . . , LEDTest2_DUT3.

"LED" stands for the LED in the switch route and `Locals.LED` is the LED number. The variable `Locals.LED` was created in the step and was used as the index for the For Loop. Hence, the loop is used to manage which LED is being tested.

`RunState.TestSockets.MyIndex` is the index of the current execution in TestStand which equates to the DUT to be tested. Remember, this expression is used in anticipation of the next exercise in which multiple DUTs will be tested.

f. Click on the Check Expression for Errors button  to confirm that the expression syntax is correct. If you receive an Evaluation or Syntax Error, determine and correct the problem before continuing.

5. The MainSequence should now look like the following figure.

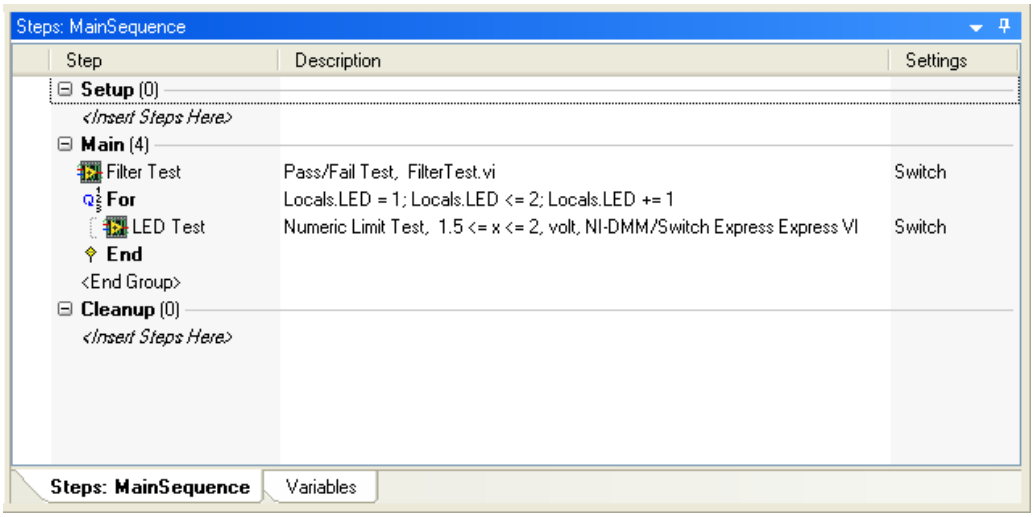


Figure 14 Completed Exercise 5 MainSequence

6. Select **File » Save**.
7. Run your sequence by selecting **Execute » Single Pass**. The sequence will now use switching for each step. The LED Test will execute twice and test both LEDs on DUT0.
8. When the test has finished, examine the report. Notice the status of the tests and the result of both LED Test measurements are displayed.
9. To run another test, click **Execute » Restart**.
10. Moving the jumper on DUT0 to a different position will make either the filter or LEDs fail. Move the jumper to **C4** and run the sequence again using **Execute » Restart**. View the test results to see that the Filter Test failed.

Tip: Placing the jumper on the **C4** or **R2** jumper pins will cause the Filter Test to fail. Placing the jumper on the **LED** jumper pin will cause the LED Test to fail.

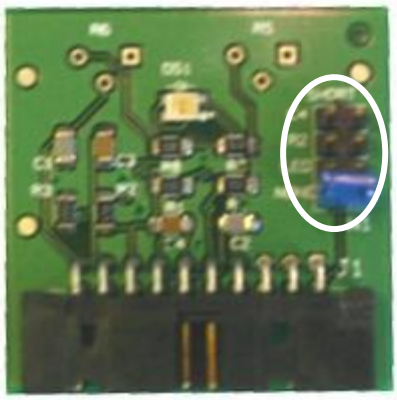


Figure 15. DUT jumpers

11. Place the jumper back on the **NONE** jumper pin when finished.
12. Close the Single Pass window by pressing the Close button on the window or selecting **Window » Close Completed Execution Displays** when you are finished.

End of Exercise 4

Exercise 5: Perform Auto Schedule Parallel Testing

Objective: To test the filter and LEDs on all four DUTs in parallel, using the TestStand Batch Process Model and Auto Scheduling. Figure 1 shows the results of this exercise.

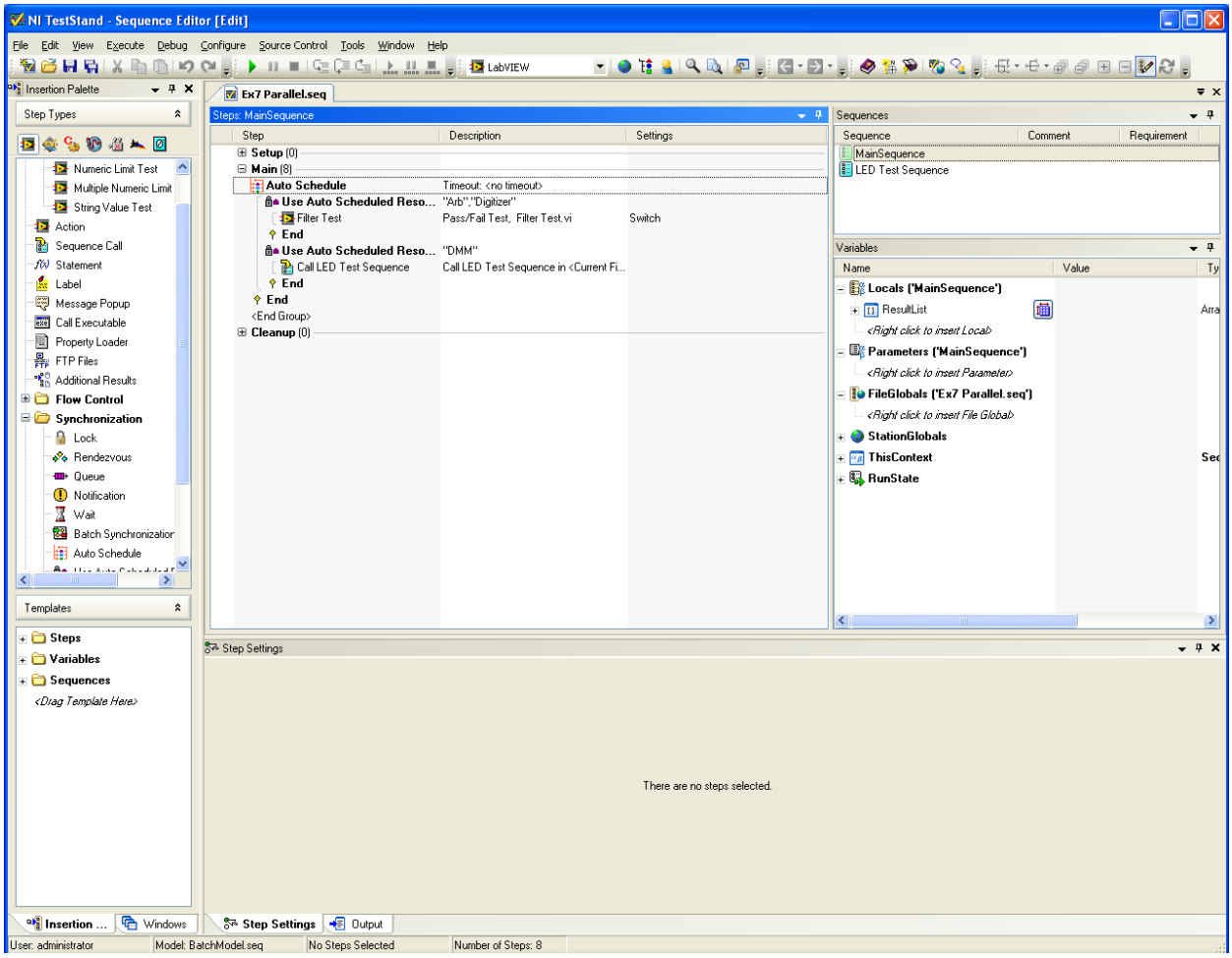


Figure 16. Completed exercise

Exercise 5A: Add subsequence to the test sequence

In this exercise, you will be adding a subsequence. Subsequences allow you to modularize portions of a sequence into smaller sequences that are easier to manage and maintain. Breaking up a sequence file into manageable segments is similar to how a program developer breaks up sections of a program into separate functions. Each smaller segment has a very specific purpose and contributes some basic functionality to the calling sequence. Like functions, subsequences can be called by any sequence. This reusability and modularity make subsequences very handy.

1. Follow these steps to make a working copy of the sequence that you created in the previous exercise:
    a. Open `Ex4 Switching.seq` in your Exercises folder or Solutions folder.
    b. Select **File » Save As…**.
    c. Name the sequence as `Ex5 Parallel.seq` and click **Save**.
2. Now you will create a subsequence to contain the LED Test steps. Adding a subsequence to this exercise isn't necessary to test multiple DUTs, however, it is included to provide you with a better understanding of how subsequences can help you create modular, reusable sequences.
    a. Select **View » Sequence File » Sequences** to be sure to display the *Sequences* view. Notice that there is only one sequence, named MainSequence, in this sequence file.
    b. Create another sequence by right-clicking in the *Sequences* view and selecting **Insert Sequence** from the context menu.

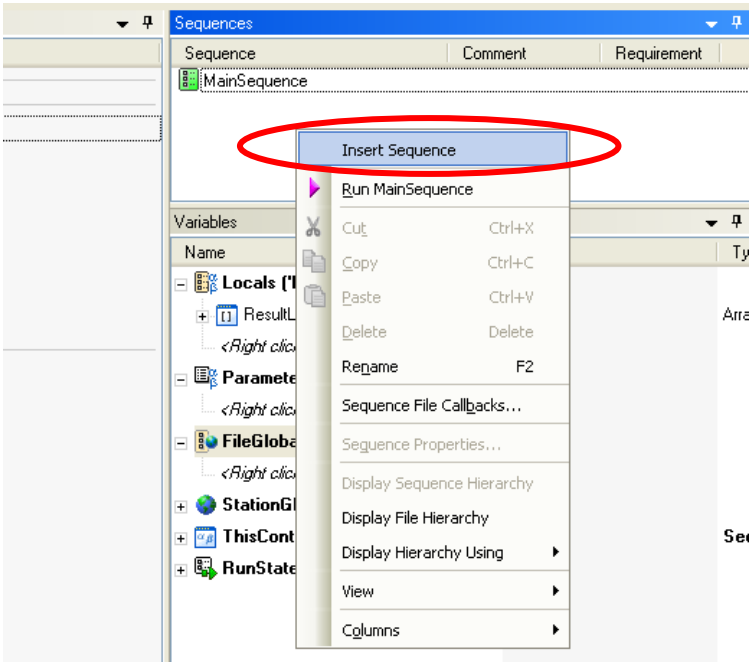


Figure 17. Insert Sequence

    c. Name this sequence **LED Test Sequence**. Now that you have created the subsequence, you will need to move the LED Test steps from the MainSequence into the subsequence. Remember that the LED Test steps include not only the LED Test step and the For Loop, but also include the LED local variable. Each need to be moved into the subsequence.

d. Select **MainSequence**.

e. To move the **For, LED Test** and the first **End** steps to the subsequence, you will click on the **For** step, hold <Shift> and then select the **End** step. Right-click on the highlighted group and select **Cut**.

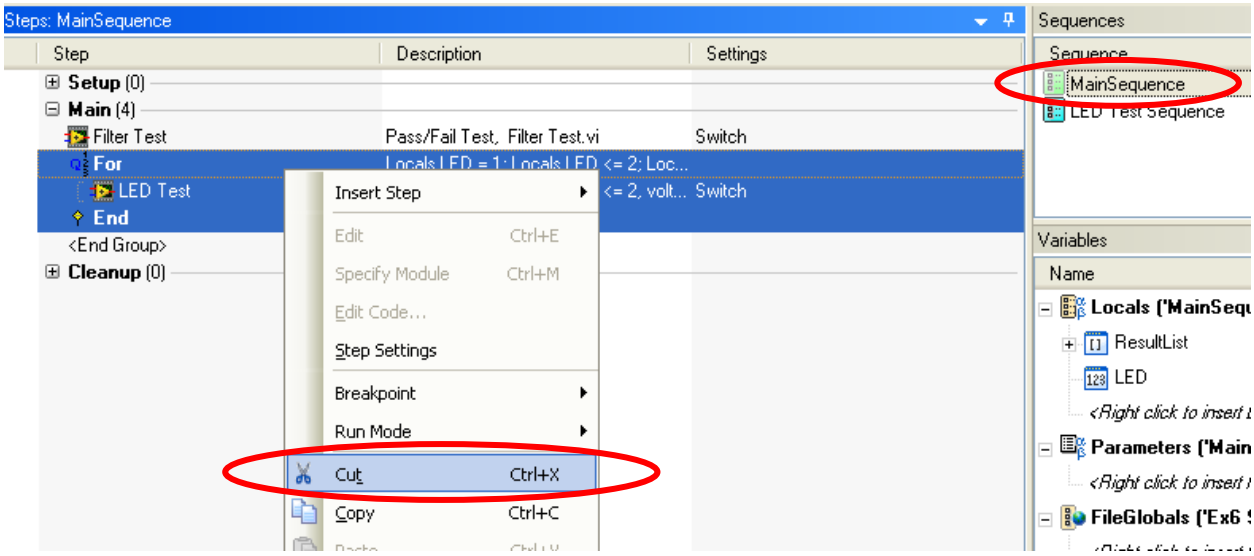


Figure 18. Cut LED Test Steps from MainSequence

f. Select **LED Test Sequence**.

g. Right-click on *<Insert Steps Here>* and select **Paste** from the context menu. When this subsequence is called by other sequences, it will test the voltages of the LEDs on the DUTs.

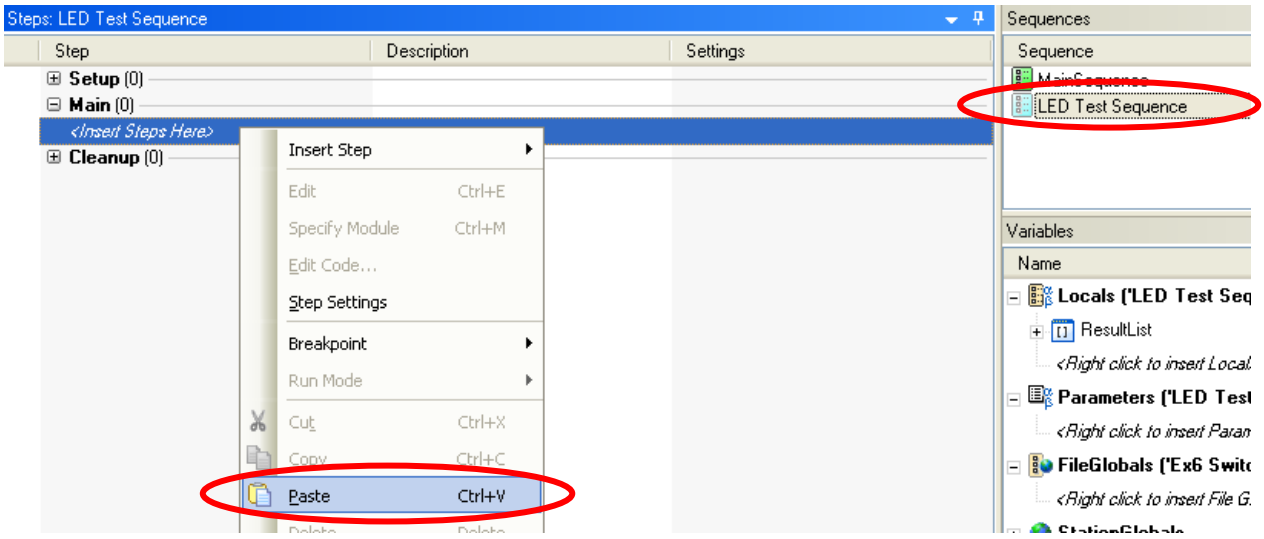


Figure 19. Paste LED Test Steps in subsequence

h. Select **MainSequence**.

i. Select **Variables** tab, right-click on the *LED* local variable and select **Cut**.

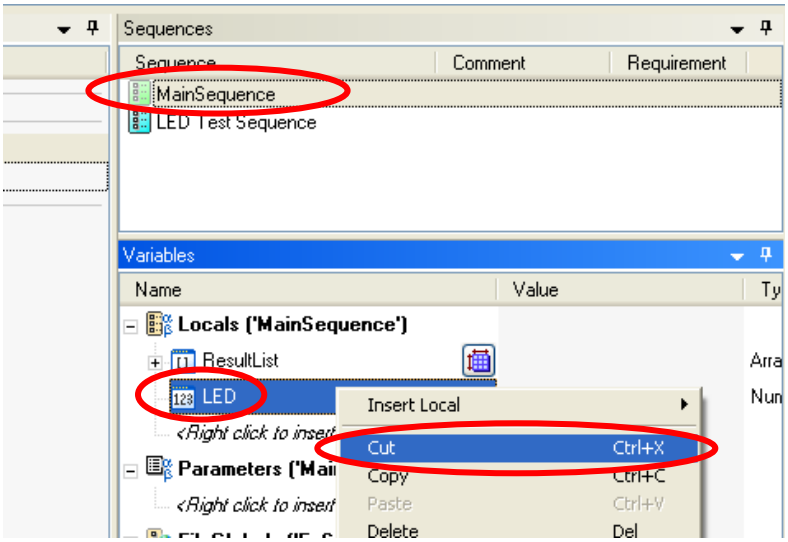


Figure 20. Cut LED Local Variable from MainSequence

j. Select **LED Test Sequence**

k. Right-click on <*Right click to insert Parameter*> and select **Paste** from the context menu.

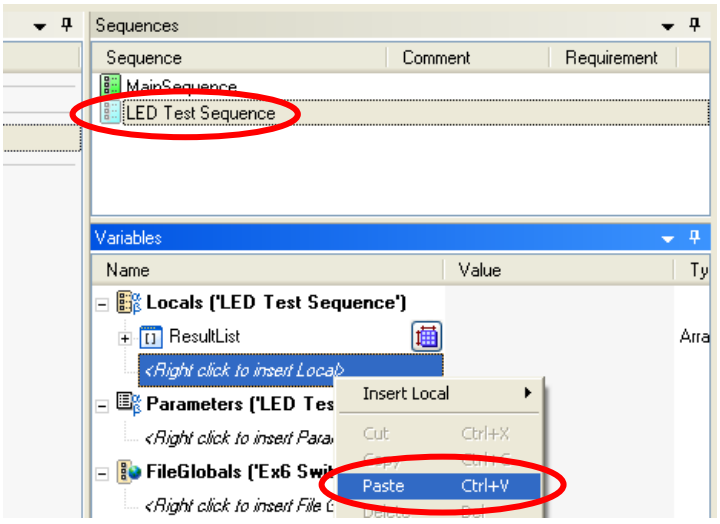


Figure 21. Paste LED Local Variable into subsequence

3. Now that the LED Test Sequence is finished, you will need to call this new subsequence from the MainSequence.

   a. Return to **MainSequence** and select the **Steps: MainSequence** tab.

b. Drag and drop a Sequence Call step after the *Filter Test* step and name the newly inserted step **Call LED Test Sequence**.

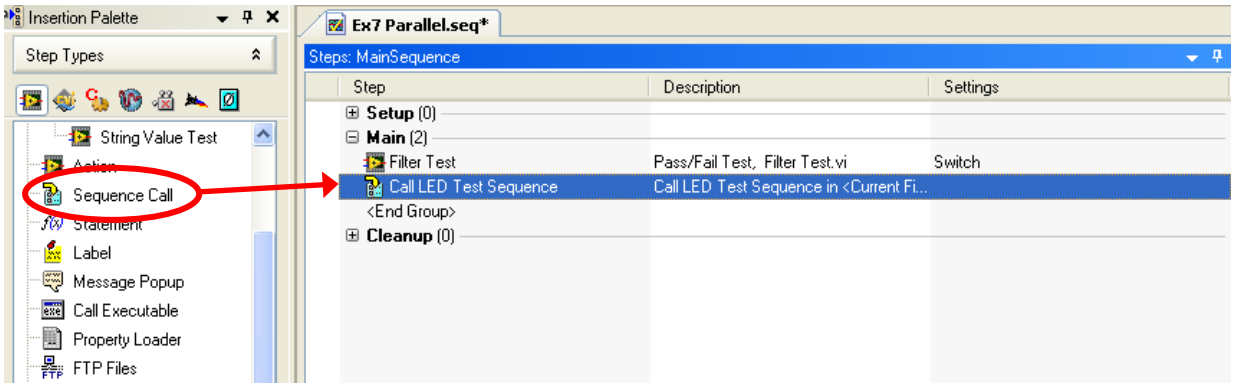


Figure 22. Selecting Sequence Call

c. Select the *Call LED Test Sequence*, and select the **Module** tab.

d. Enable the **Use Current File** checkbox and change the *Sequence* to **LED Test Sequence**.

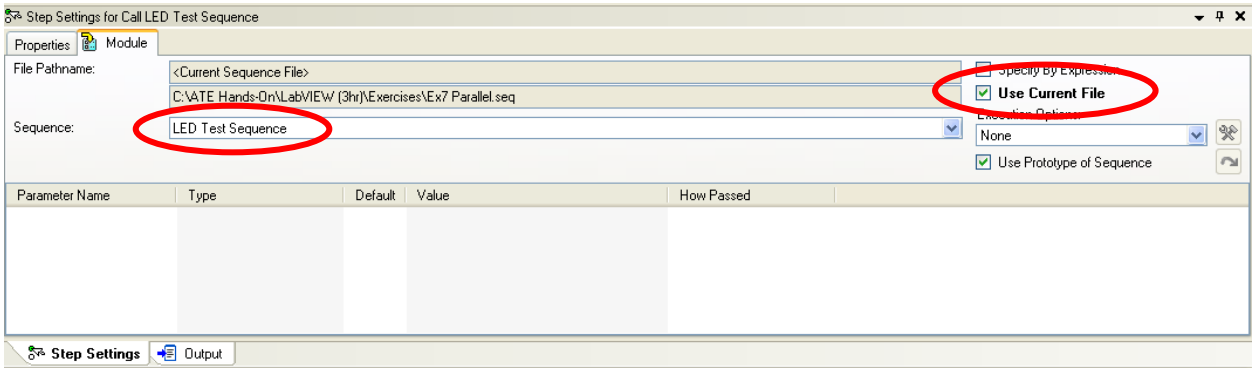


Figure 23. Edit Sequence Call

4. You've successfully added a subsequence for the LED Tests to your main test sequence.

Exercise 5B: Add Auto Scheduling and configure the Batch Process Model

You will add the Auto Schedule step type to the MainSequence. When you add the Auto Schedule step type, it includes an Auto Schedule Resource. An Auto Schedule Resource performs two functions. First, it provides a lock around the included steps. This means only one thread can execute the steps at a time. Second, it identifies a block of steps that can be reordered with any other block of steps contained within an Auto Schedule Resource within the same Auto Schedule group.

1. Configure the Arb and Digitizer as Auto Schedule Resources for the Filter Test.

    a. Drag and drop an Auto Schedule step (located in the Synchronization folder) after the *Filter Test* step.

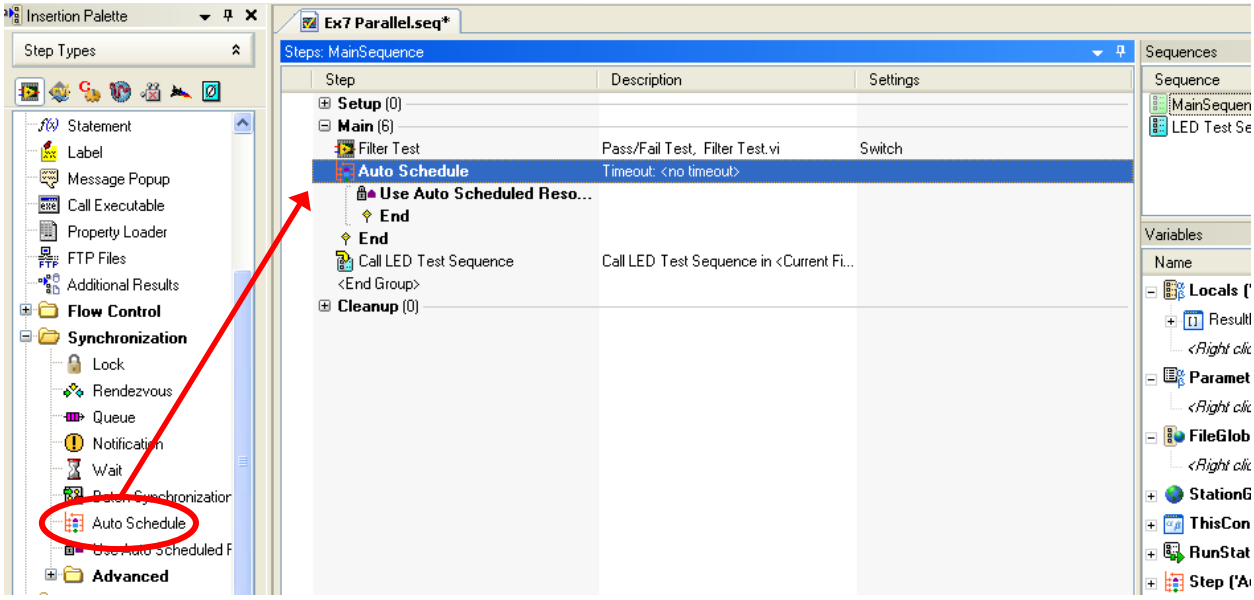


Figure 24. Move Filter Test into Auto Schedule Resource

    b. Drag and drop the *Filter Test* step after the Use Auto Scheduled Resource step.

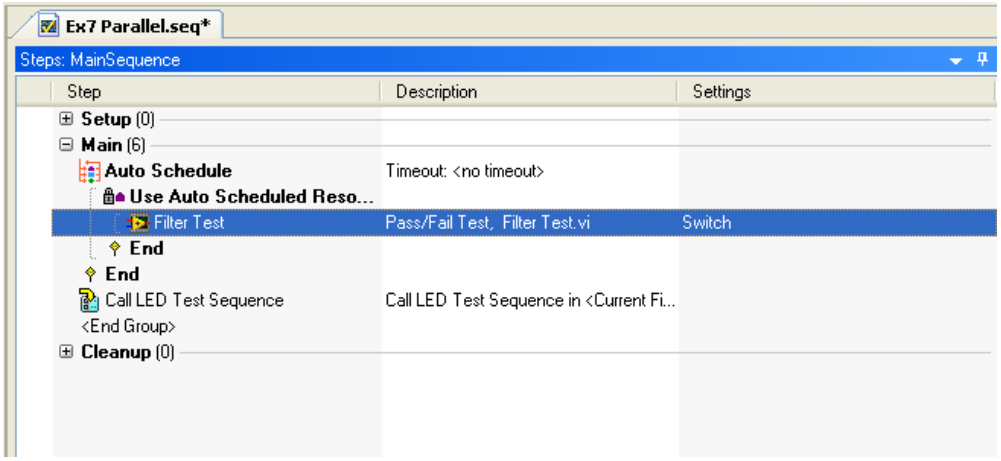


Figure 25. Move Filter Test step in Auto Scheduled Resource

c. Click on the *Use Auto Scheduled Resource* step and ensure the **Auto Scheduled Resource Settings** tab is selected in the lower half of the screen.

d. Left click on the field that currently says *<The Expression cannot be empty>* and enter the following string including quotes and the comma: **"Arb", "Digitizer"**

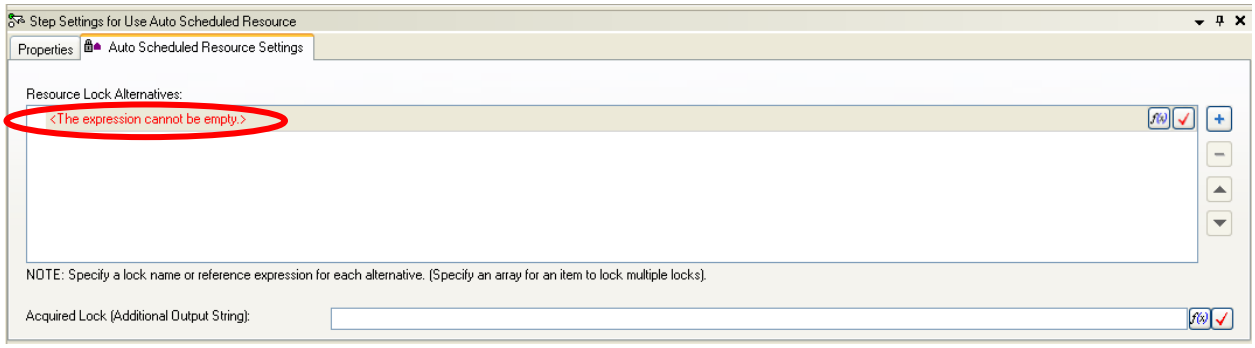


Figure 26. Configure Filter Test Resource

e. Press the **Enter** key to return to the sequence.

2. Configure the DMM as an Auto Scheduled Resource for the LED Test.

a. Drag and drop a *Use Auto Schedule Resource* step (located in the Synchronization folder) after the existing Use *Auto Schedule Resource* step as seen in figure 12.

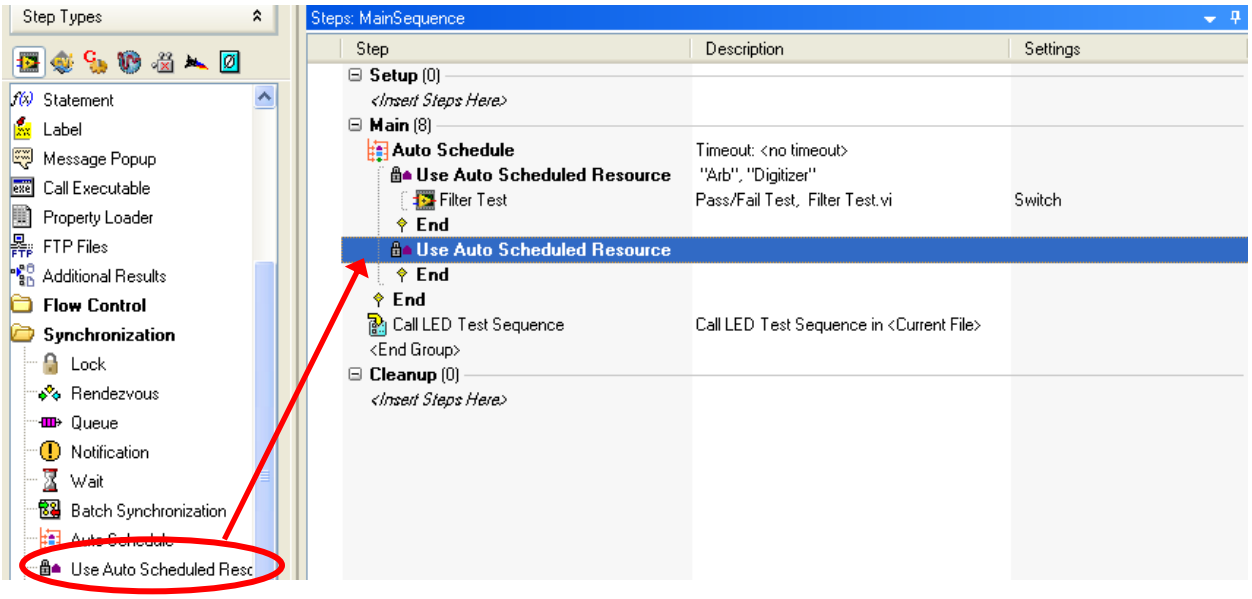


Figure 27. Add a second Use Auto Scheduled Resource step

b. Drag and drop the *Call LED Test Sequence* step after into the Use Auto Scheduled Resource group as seen in Figure 13.

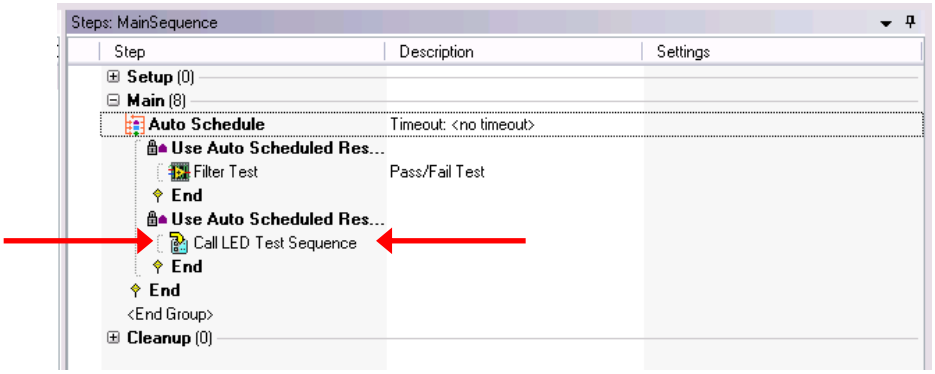


Figure 28. Insert Call LED Test Sequence into Auto Schedule Resource

c. Left-click on the second *Use Auto Scheduled Resource* step to view the *Step Settings for Use Auto Schedule Resource*.

d. Left-click on *<The expression cannot be empty>* as shown in Figure 14 and then enter the following string including quotes: **"DMM"**

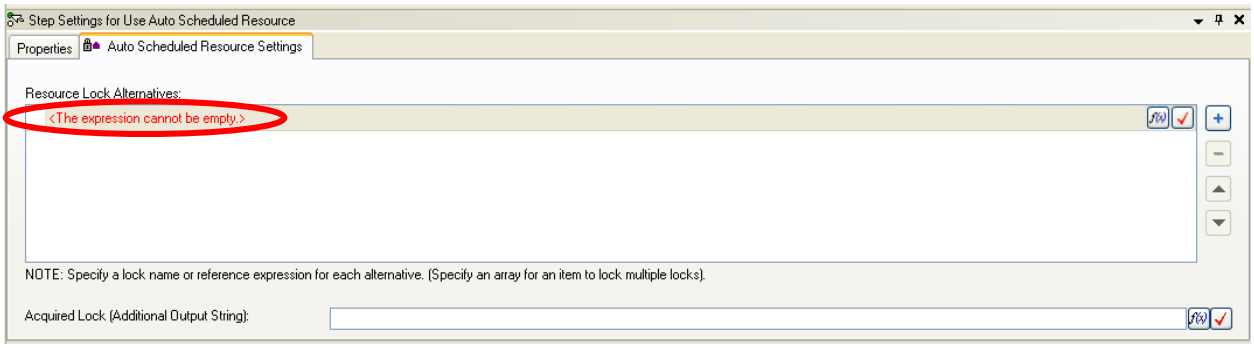


Figure 29. Configure LED Test Resource

e. Press the Return key to exit the text field.

3. The result should match the following figure.

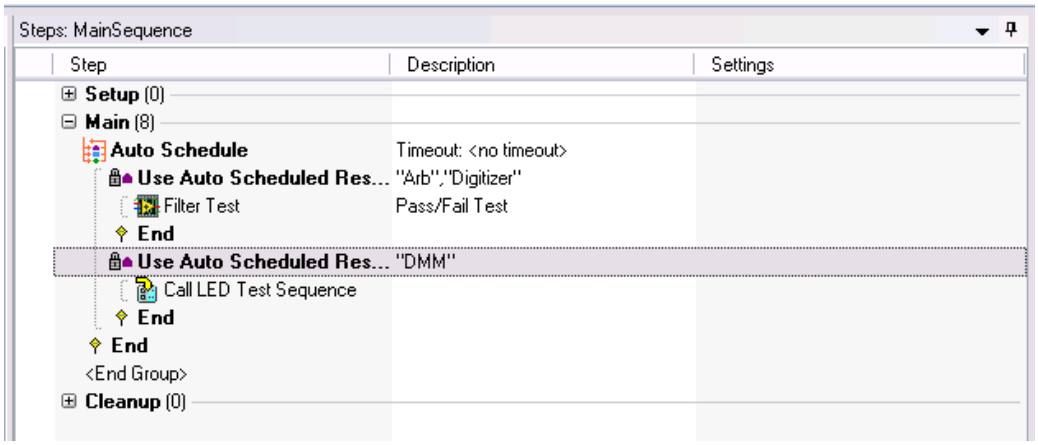


Figure 30. Completed Exercise 4 MainSequence

4. Now you will change the process model to the Batch process model.
    c. Select **Configure » Station Options** and select the **Model** tab.

d. Use the *Station Model* drop-down to select **BatchModel.seq** or click the browse button, to go to `C:\Program Files\National Instruments\TestStand x.x\Components\NI\Models\TestStand Models` and select **BatchModel.seq**.

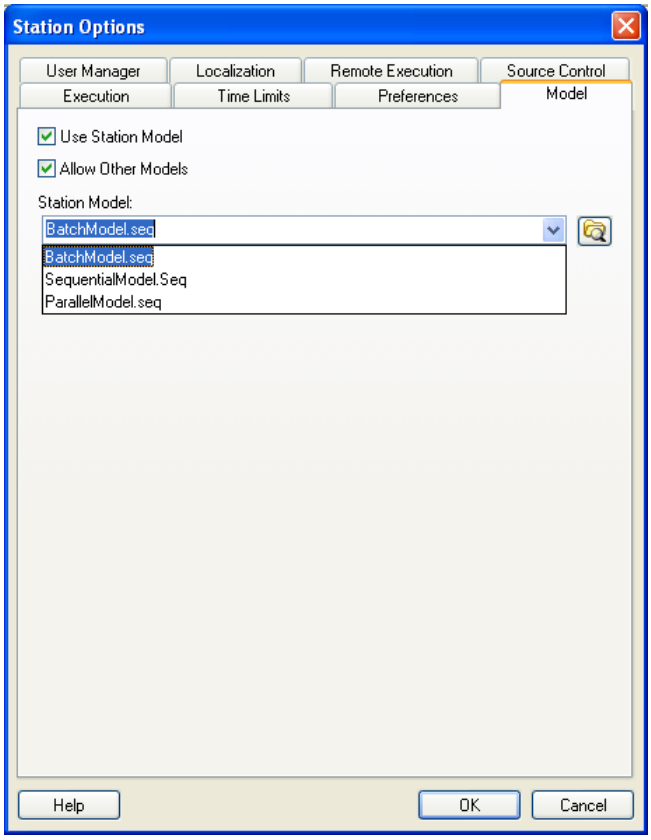


Figure 31. Select Batch Process Model

e. Select **OK** to return to the sequence file.

5. Now that you have selected the Batch Process model, you will need to specify the total number of test executions that will be performed in parallel on multiple sockets as a group.

   a. Select **Configure » Model Options**.

b. Select the **Number of Test Sockets** as 4.

Note: Later you can try 1, 2, or 3 Test Sockets to see the results.

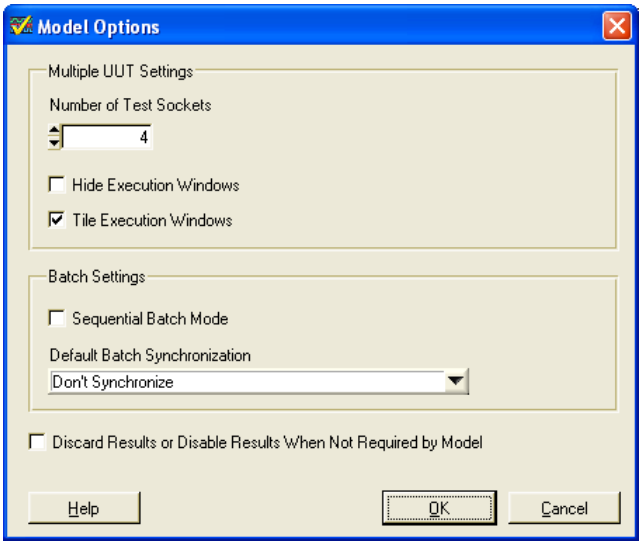


Figure 32. Model Options Dialog Box

c. Select **OK**.

6. Select **File » Save**.
7. Maximize the TestStand Sequence Editor window.

8. Select **Execute»Single Pass** to run the sequence file.

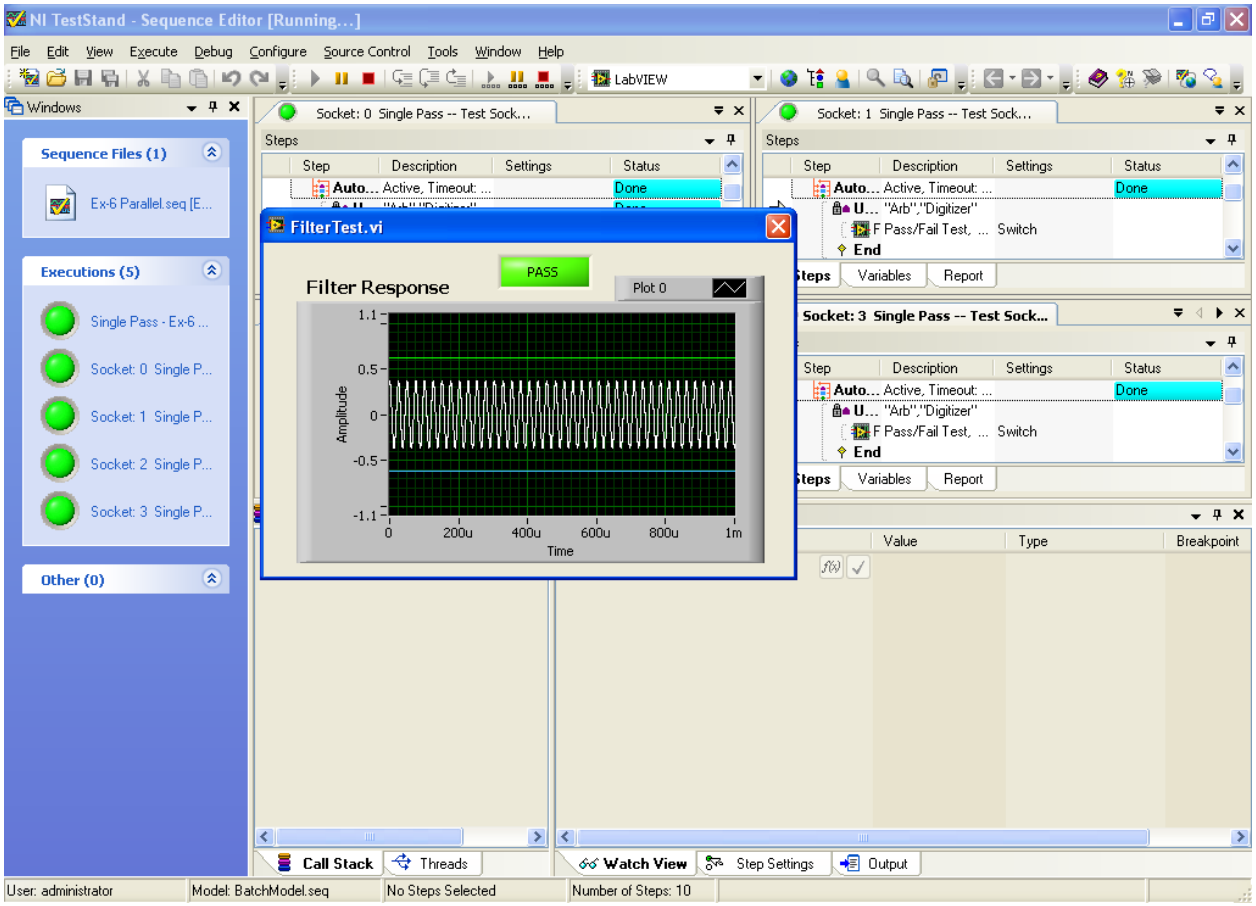


Figure 33. Executing sequence

9. You will see four executions appear on the left side of the window. TestStand will test the sockets in parallel. Each socket tests a different DUT (DUTs 0-3).
10. When the test has finished, examine the Batch Report. To run another test, click **Execute » Restart**.
11. Moving the jumper on any of the DUTs to a different position will make either the filter or LEDs fail. Move a jumper(s) and run the sequence again using **Execute » Restart**. View the test results to see that the appropriate test(s) fails.
12. Place the jumper(s) back on the **NONE** jumper pin when finished.
13. Close the Single Pass window by pressing the Close button on the window or selecting **Window » Close Completed Execution** when you are finished.

End of Exercise 5