# NewTS Execution Engine User’s Manual

## Introduction

## This document aims to provide end-users – operators, engineers, and administrators with the information needed to operate the NewTS Framework Execution Engine (NewTS ExecEng).

The architecture of the NewTS Framework includes the Execution Engine and physical instruments and Soft Panels (SP).

## NewTS ExecEng

To use the NewTS ExecEng effectively and to its maximum capacity of it, it seems necessary to understand how it works – to understand its architecture.

**Ethernet**

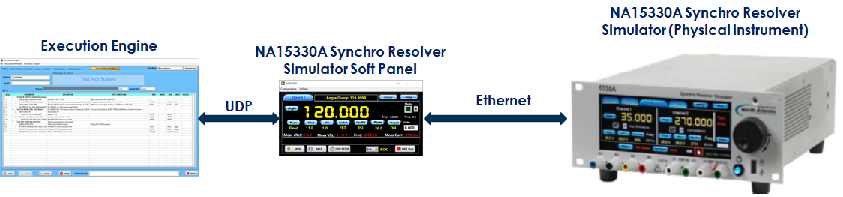
The NewTS Framework architecture is illustrated in the figure below:

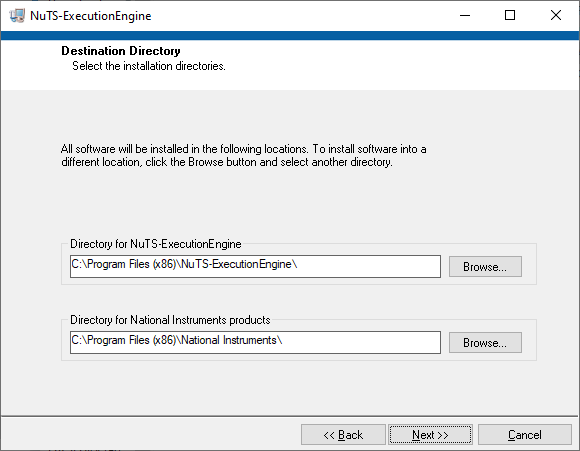
Fig. 1. NewTS Framework architecture

* The Execution Engine does not directly communicate to the physical instrument.
* The Execution Engine sends an instruction to the instrument’s SP via UDP (User Datagram Protocol).
* The Execution Engine queries measurement data from the instrument SP.
* The Execution Engine evaluates the results of measurements and generates a test report.
* The instrument’s SP maximally represents the physical instrument functionality.
* The instrument’s SP interfaces to the physical instrument via its native interface – Ethernet, USB, etc.
* The number of instruments’ SP’ is not limited.

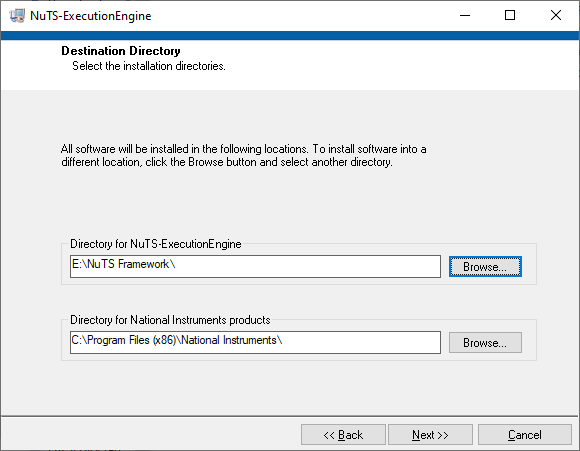
# Configuring Execution Engine

## Installing the Execution Engine software

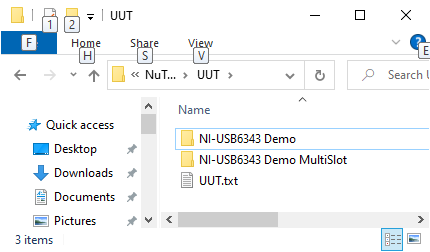
* + 1. Download the Execution Engine installer zip file from the link provided to you.
    2. Extract the Execution Engine to a directory of your choice.
    3. Create the “NewTS Framework” directory in any root directory available on your computer. As an example, “C:\NewTS Framework” or “E:\NewTS Framework”. The reason behind it is that the application periodically saves data to files, including test report data. If the applications will be installed into Program Files (x86) or Program Files, you will be facing Windows security issues.
    4. Open the folder, containing the extracted NewTS installer, go to the subfolder “Volume” and click “NewTS-ExecutionEngine\_install.exe”.
    5. After installer initialization is completed, the “Destination Directory” window will open:



* + 1. Change “Directory for NewTS-Execution Engine, which defaults to “Program Files (x86) to the path in the root:



* + 1. Then press the” Next>>” button. The execution engine will be installed and the desktop shortcut will appear on your desktop . Right-click on the Execution Engine icon and select “Open file location”
    2. Please note that folders “Instruments” and “UUT” are empty.

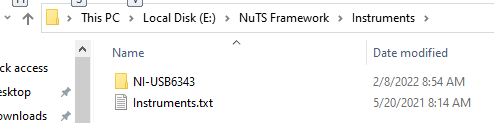
Unzip the UUT.zip file and copy the content to the NewTS\UUT folder. It should look like this: 

The NewTS installation is completed.

* + 1. The “Configuration” folder shall contain the “System Configuration.cfg” file and the “Users.cfg” file. The “System Configuration.cfg” is the text file and here stored parameters, needed for the proper functioning of the Execution Engine application.

3.1.10 The “Users.cfg” file is not human-readable and contains the users’ names, login levels, and passwords. This data is used upon launching the Execution Engine.

* + 1. The “Failed Test Logs” folder will contain the test execution failures. This data helps debug test sequences and/or software. There will be a “FailureLog.txt” dummy file, which can be deleted or ignored upon installation. This file is needed for installer configuration. The Execution Engine has the option to enable the failure logging.
    2. The “Instruments” folder is the folder where instrument Soft Panels should be installed. At the Execution Engine installation, this folder will contain the “Instruments.txt” dummy file. This file was needed for installation purposes and can be deleted or ignored. This folder will contain specific instruments’ subfolders, as an example:

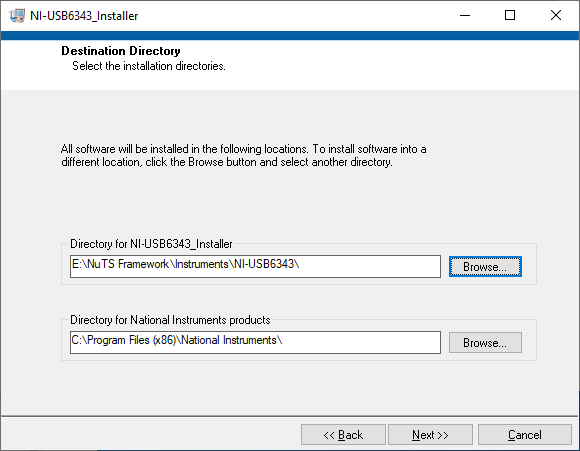


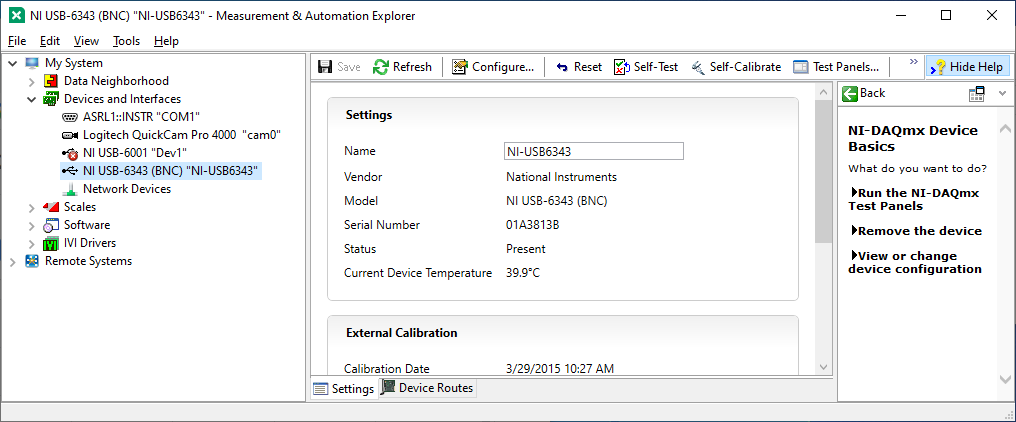
The instruments’ subfolders shall be created before Soft Panels installation. **The folders’ names MUST be the same as the soft panel executable**. As an example: for the soft panel named “NI-USB6343.exe”. In the “Instruments” folder should be created the subfolder named “NI-USB6343”. In this case, the Execution Engine will check its “Instruments” folder and recognize the instrument NI-USB6343.exe and its accompanying folders and files.

* + 1. The “UUT” folder is designated to hold specific test sequences. These test sequences are not installed during the Execution Engine installation and they should be imported or created later. Initially, there is a dummy “UUT.txt” file needed for Execution Engine installer configuration and can be deleted or ignored.

## Installing Instruments Soft Panels

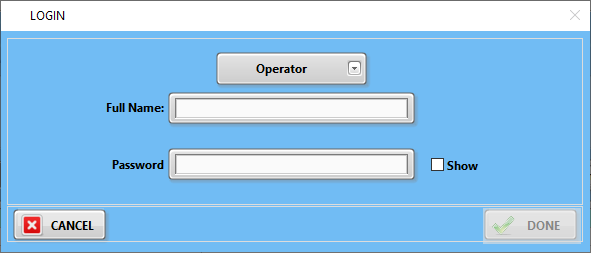
* + 1. Download the NI-USB6343 installer zip file from the link provided to you, as an example, and unzip it to a folder of your choice.
    2. Go to the “…\NewTS Framework\Instruments” folder and create the subfolder named “NI-USB6343”.
    3. Go to the NI-USB6343\_Installer folder, then “Volume” and select “NI-USB6343\_install.exe”. The installation window will appear.
    4. Using the “Browse” button navigate to the “Directory for NI-USB6343\_Installer” to “…\NewTS Framework\Instruments\ NI-USB6343\”:

1. 
   * 1. Press the “Next>>” button to complete the installation.
     2. Go to the NewTS Framework folder and verify that its “Instruments” folder contains the “NI-USB6343” subfolder: .
     3. Open NI-MAX “Devices and Interfaces”, locate your DAQ device, and rename it to “NI-USB6343”:



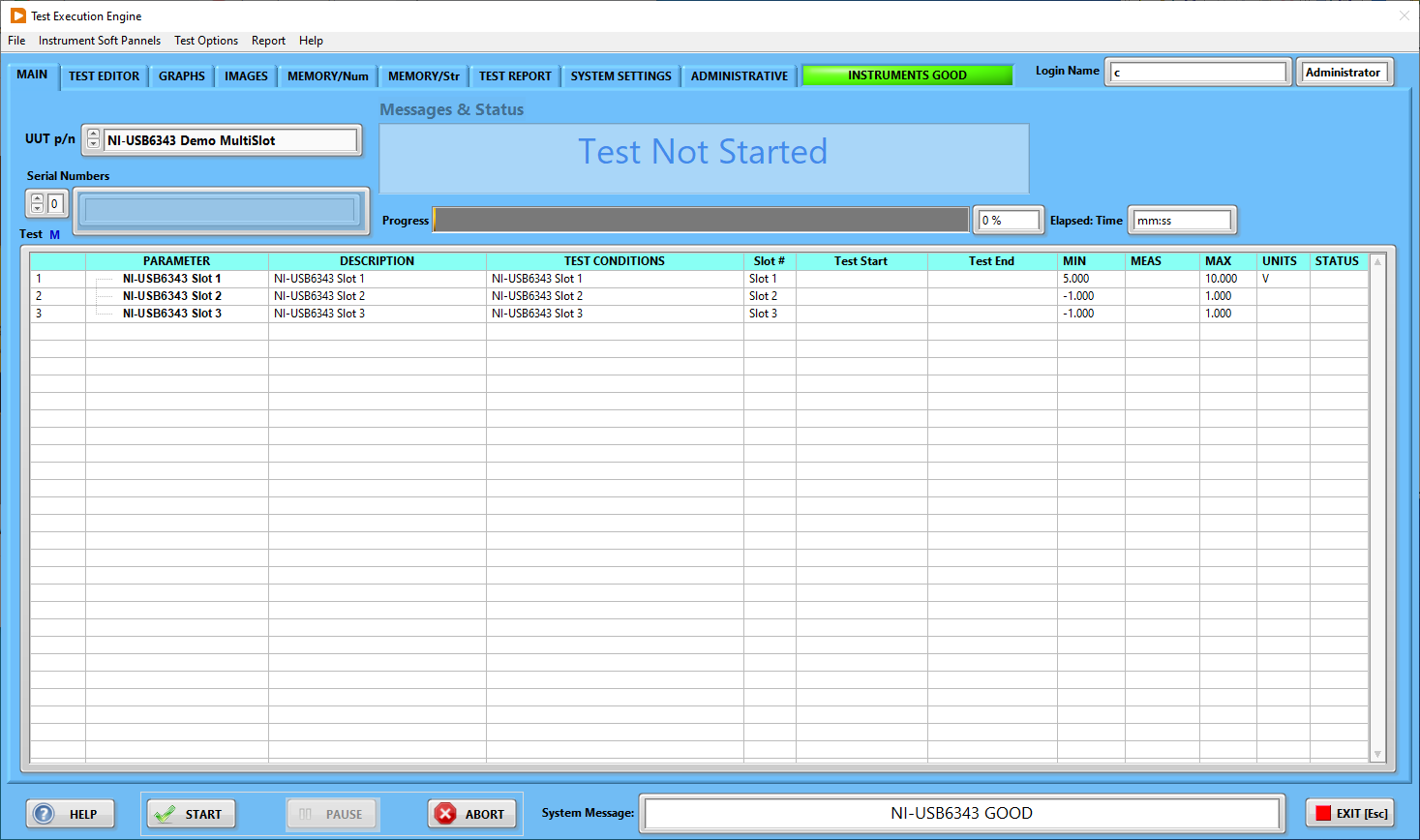
* + 1. Press the “Save” button. Close the NI-MAX.

## Starting the Execution Engine

* 1. Upon installation, the Execution Engine application will have the shortcut on the desktop. The application should be “Run as administrator”. Click on the shortcut icon: .
  2. The log-in window opens: .
  3. There are three login levels: Operator, Engineer, and Administrator. Select your login level, and enter the user’s “Full Name” and “Password”. The system will check if the log-in level corresponds to the user name and the correctness of the password: . Press the “DONE” button. As a result of the successful login, the Execution Engine will be configured.
  4. The default values:
* Operator: name – a; password – a.
* Engineer: name – b; password – b.
* Administrator: name – c; password – c.

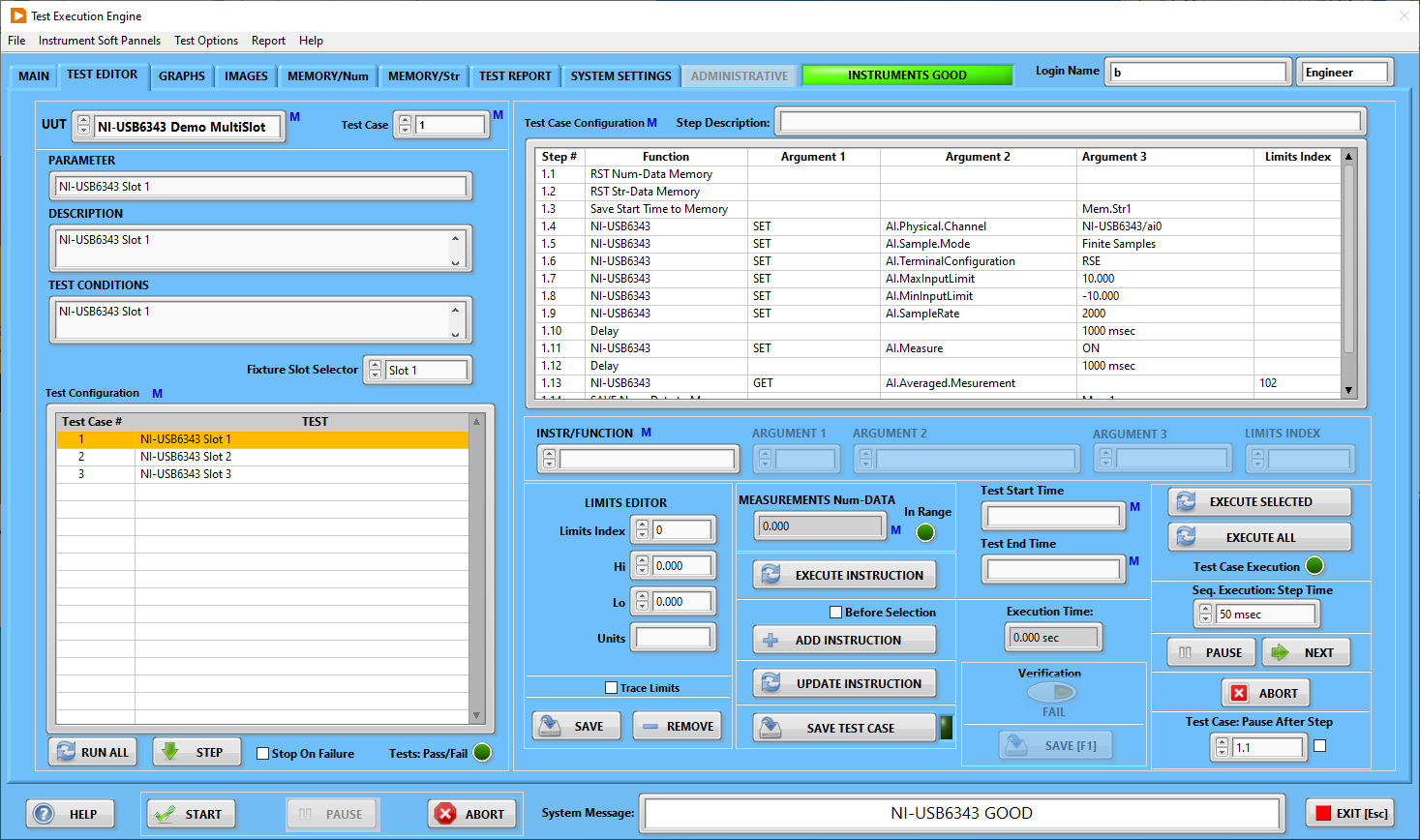
There are differences between login levels:

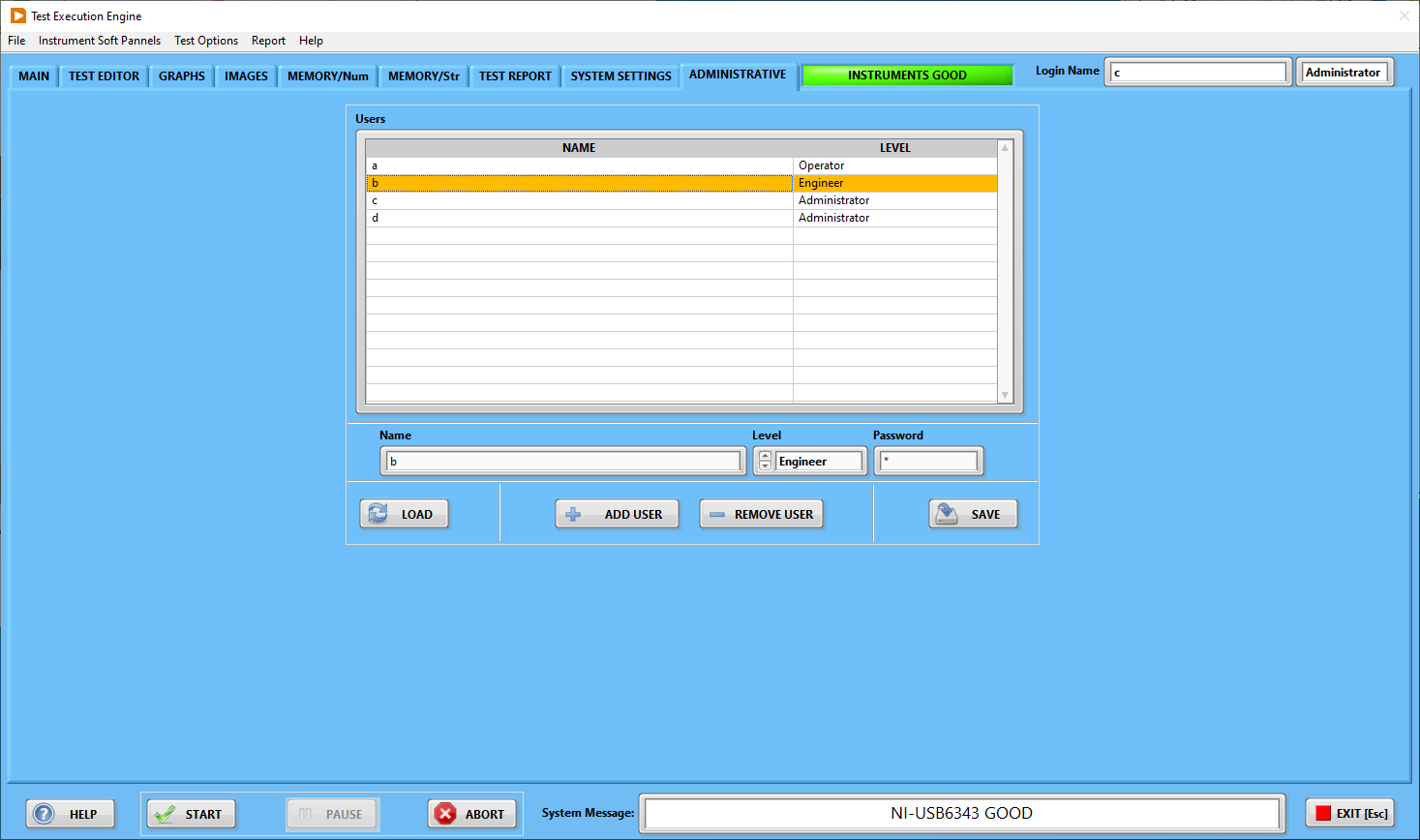
* Operator: can select UUT (Unit Under Test), enter the part’s serial number, start the test, pause, and/or abort the test. The image of the Execution Engine, configured for the operator’s login level is shown below:



Please note, that the last tested UUT sequence (procedure) is loaded automatically. In addition, the Execution Engine will perform a check of the instruments’ soft panels involved in performing that test. This rule applies to all login levels.

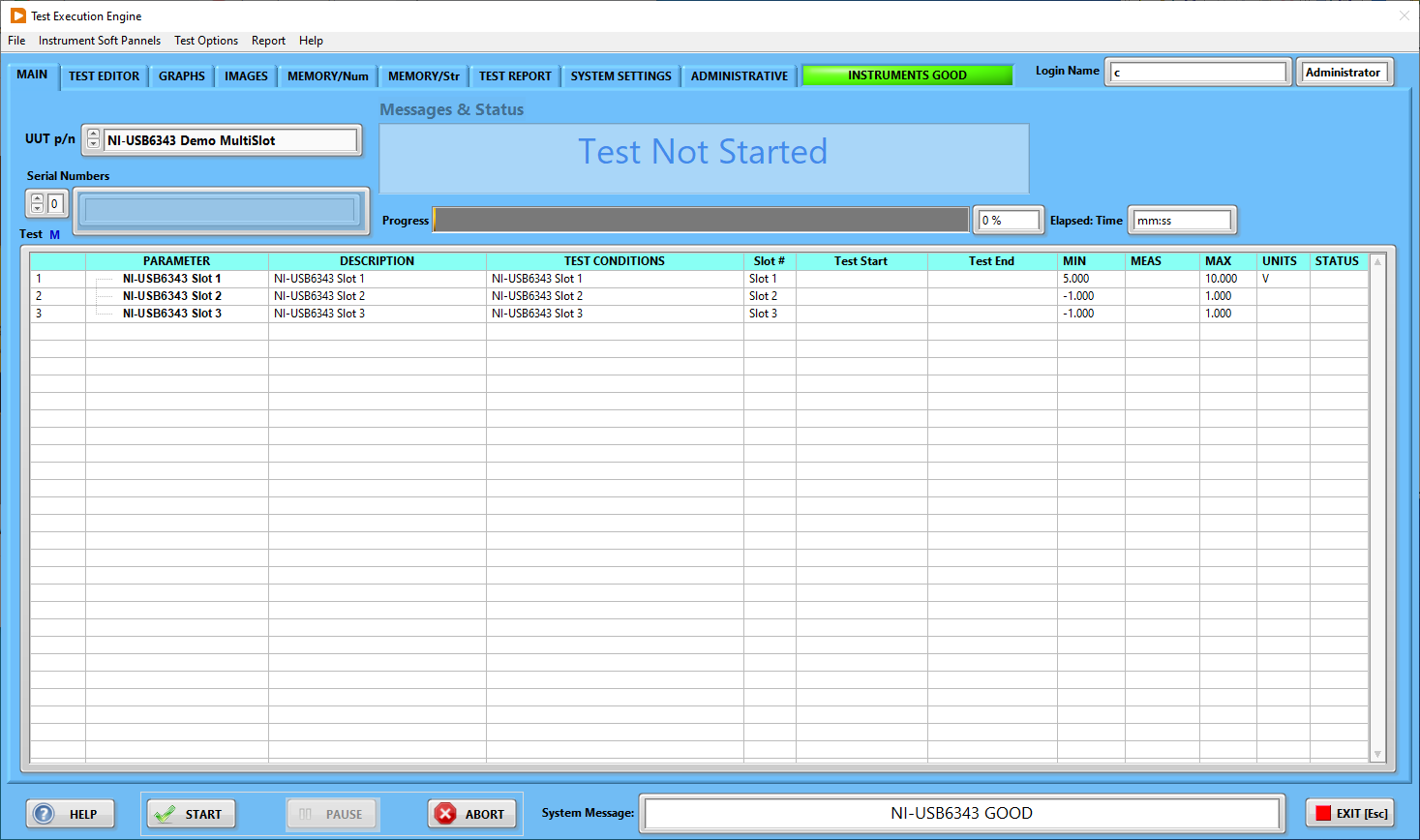
* Engineer: in addition to the features of the operator’s login provides full access to the Execution Engine functionalities, except to the “Administrative” page:



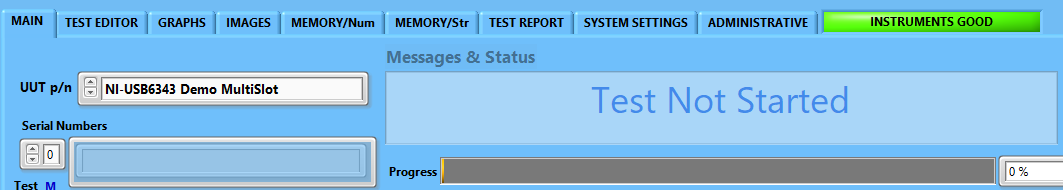
* Administrator login provides the access to the “ADMINISTRATIVE” page, where the system users are managed:
* 

## Operator’s functionalities

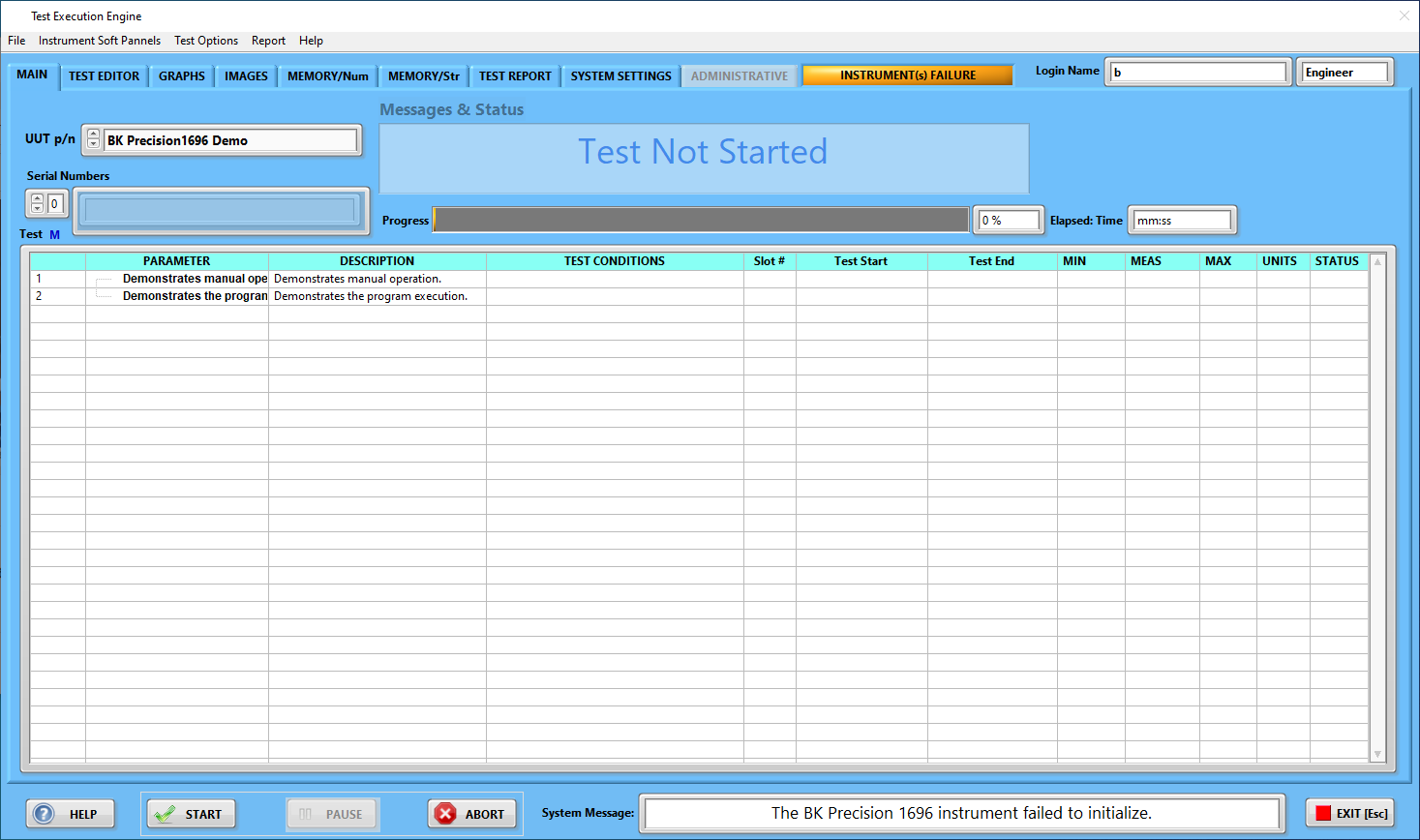
* 1. In the case the user is logged in as the operator, the Execution Engine will be configured correspondently:



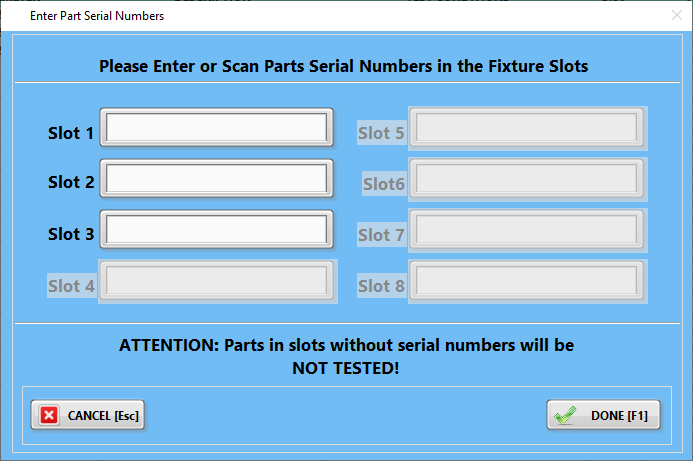
The Execution Engine application loads the last used UUT and performs the used instruments’ SP check:



In the case of any instrument’s soft panel failure to initialize, the Execution Engine will provide an indication of it:



* 1. After selection of the desired UUT and in the case of successful instrument launch the operator shall press the “START” button . The Execution Engine can handle fixtures for multiple UUTs. At the moment there are 8 slots that can be utilized. For demo purposes, the Execution Engine is configured to handle 3 slots. Upon pressing the “START” button, the window, prompting the operator to enter serial numbers will open:



Important

Note!

* 1. Enter the serial number(s) of UUT(s) and press the “DONE” button to launch the test:

Test Execution

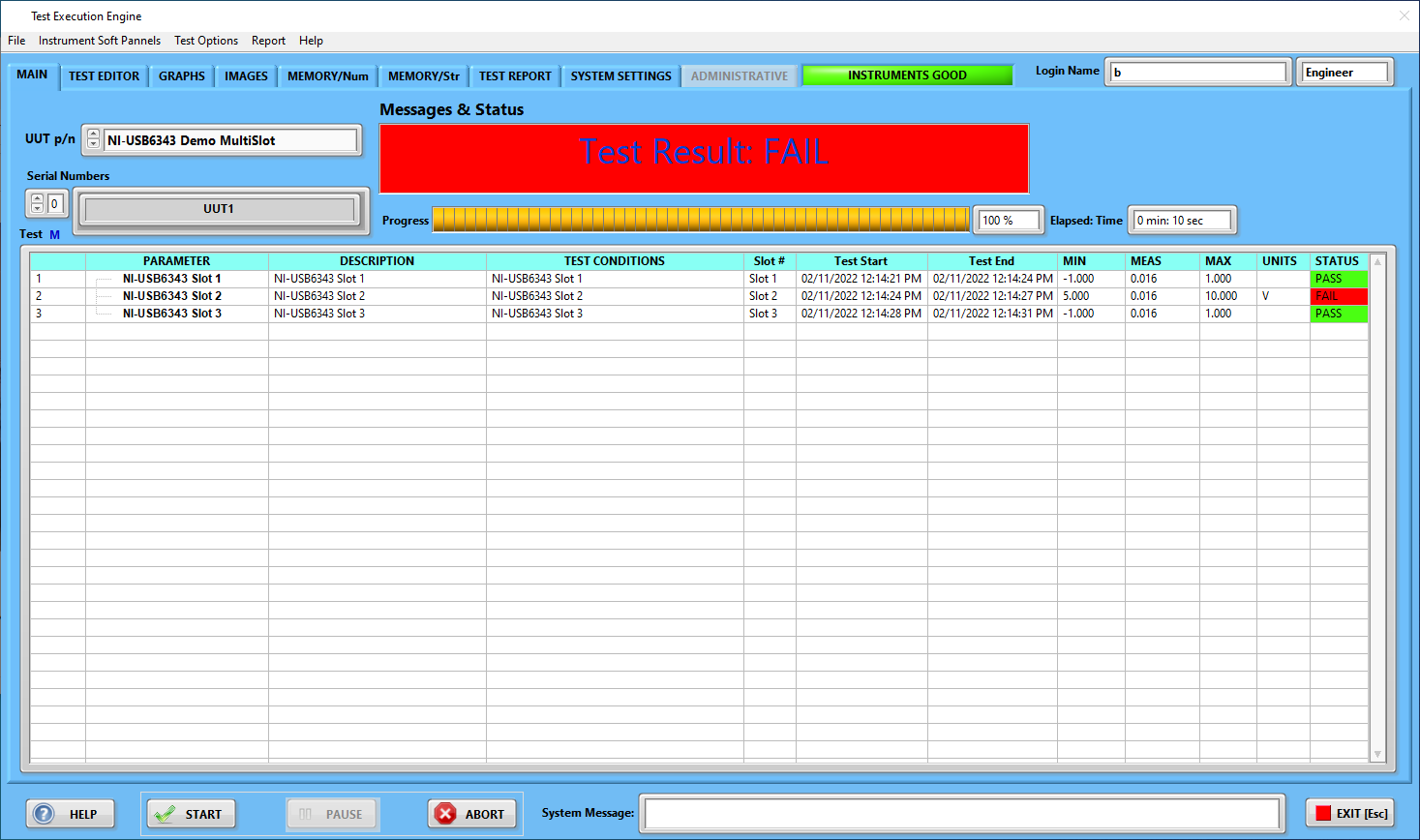
Time

Test End Time

Test Start Time

Fixture Slot

Test Case Attributes



Test Status

Measured

Value

Test Limits

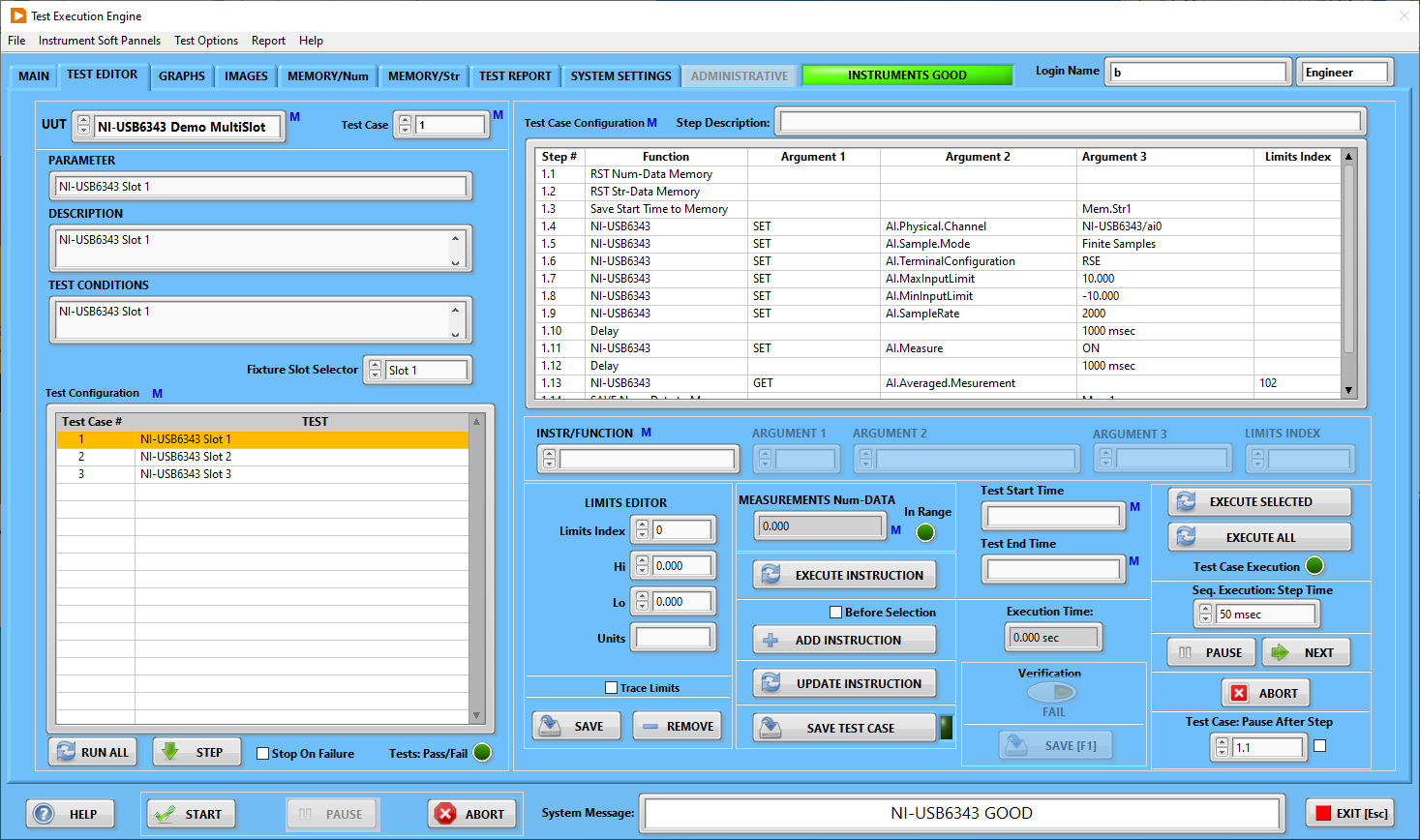
Serial Numbers

Test Case

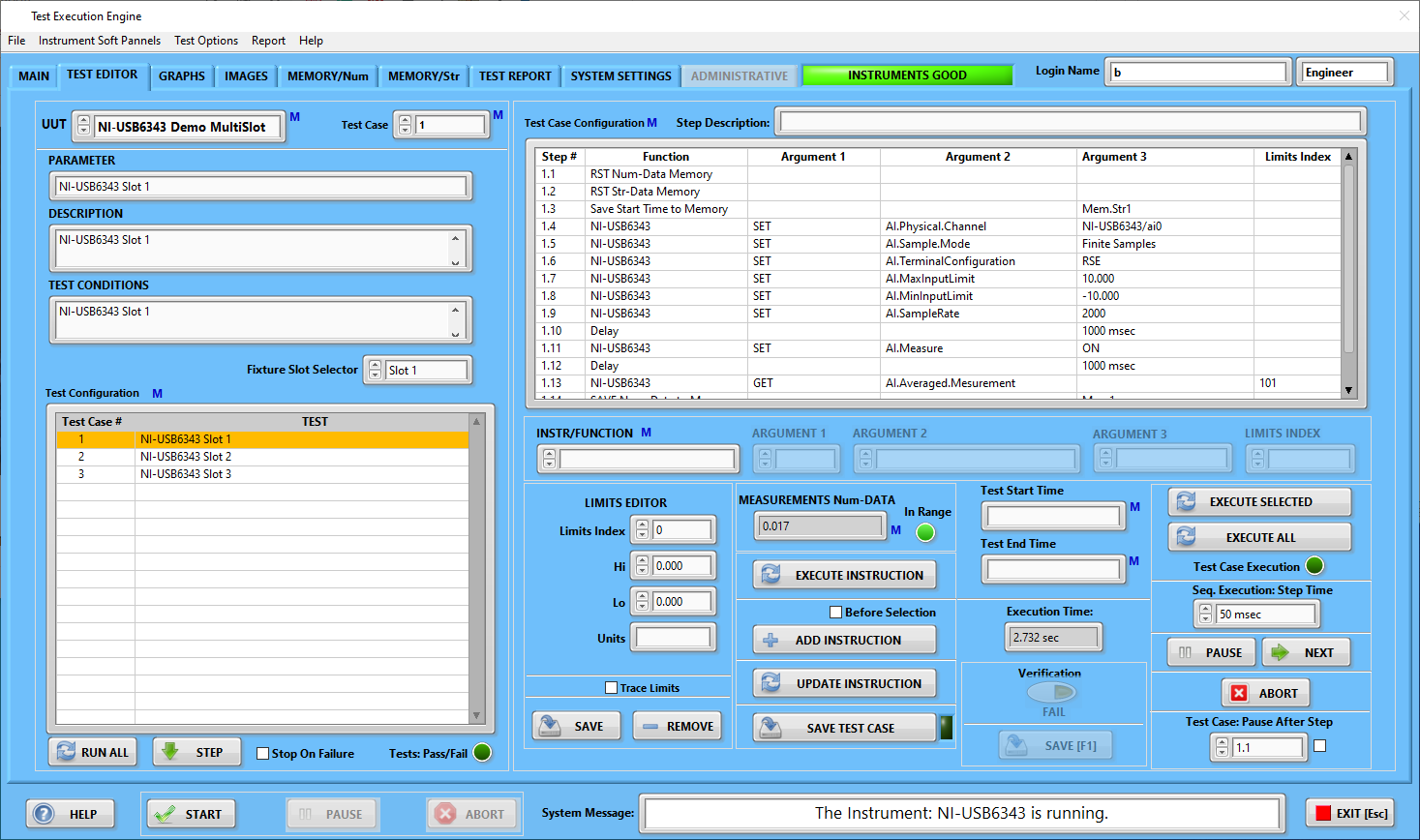
* 1. During the test execution, the operator can pause test execution by pressing the “PAUSE” button , or abort the test execution by pressing the “ABORT” button .

## “Engineer” login functionalities

* 1. Upon login into the “Engineer” level access to most of the Execution Engine functionalities will be provided:

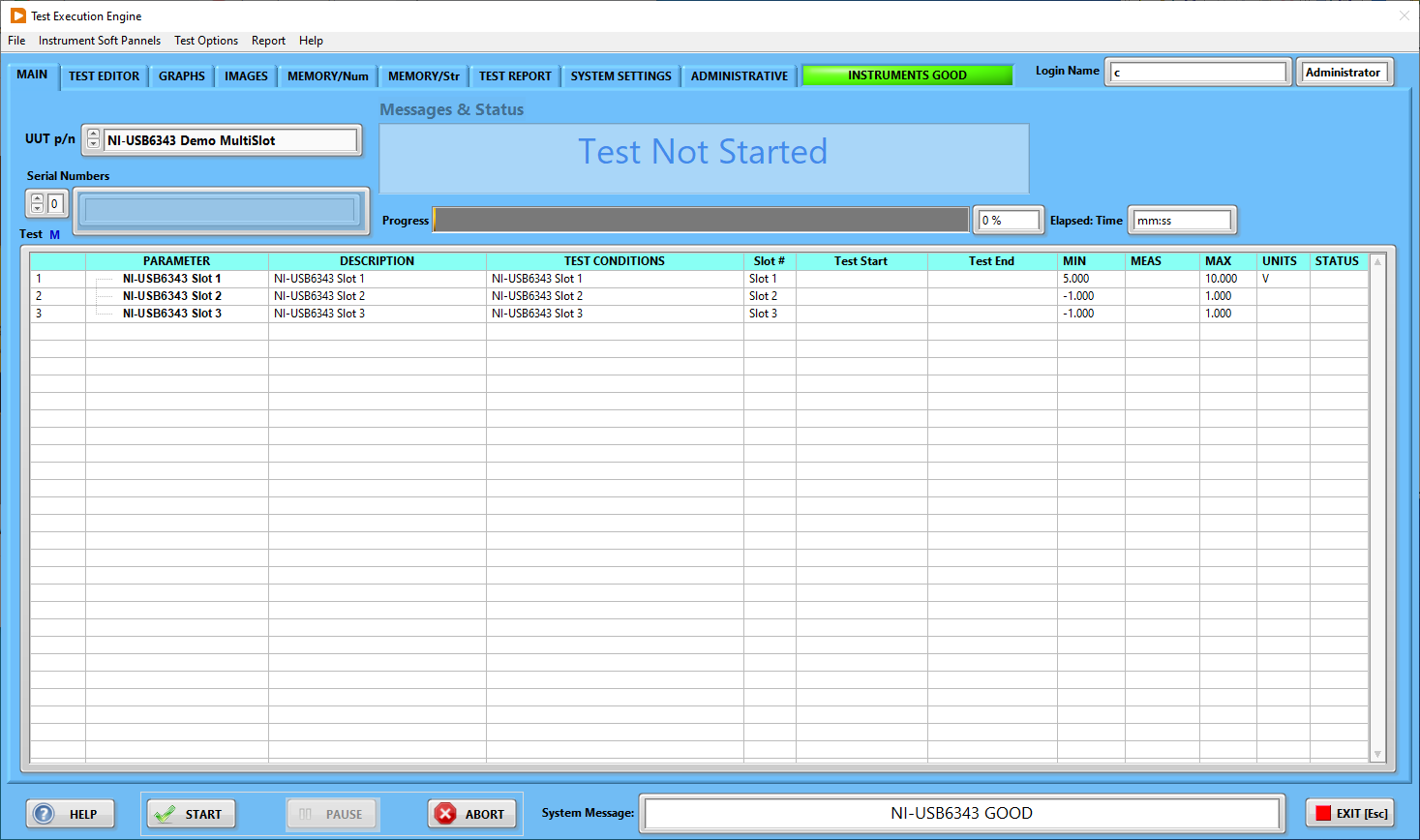


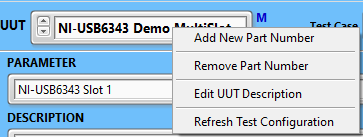
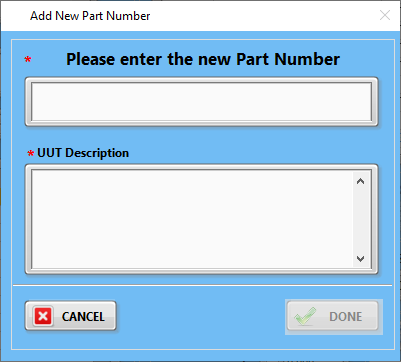
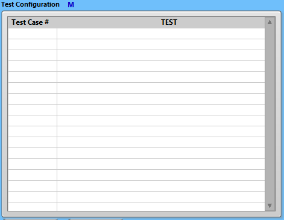
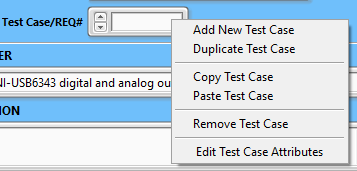
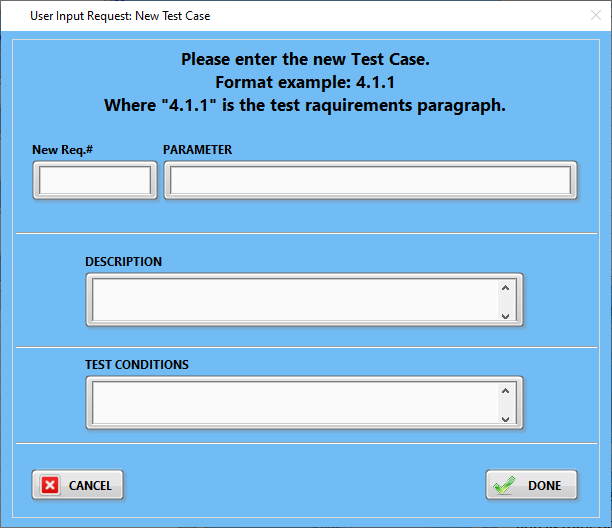
* 1. The “Main” screen is identical to the operator’s configuration, however, there are additional screens available: .
  2. The “TEST EDITOR” screen is the major resource for test sequences development:



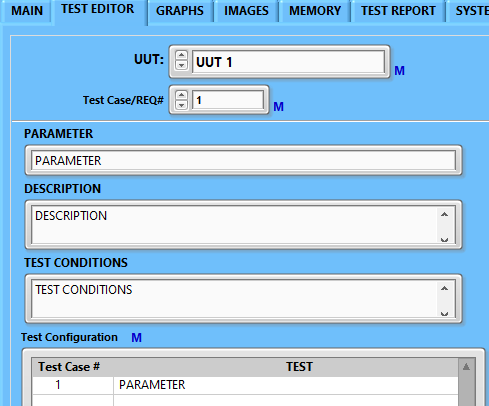
* + 1. UUT selector  is linked to the  on the MAIN screen.

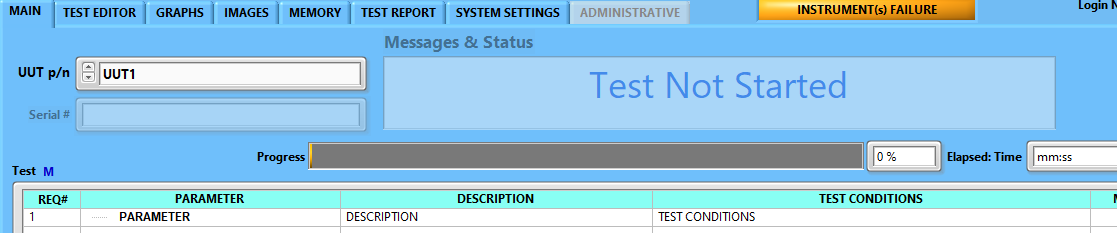
The change of either selection automatically will load the test sequences on both screens. In the MAIN screen, the test sequence is loaded in the “tree” format:

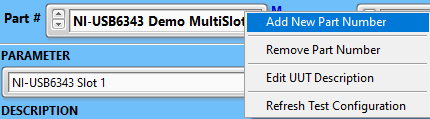


* + 1. The UUT selector  has the letter “M” which indicates that there is a “run-time menu”: 
    2. “Add New Part Number” intended for creation of the new test – composition of test cases: . Upon entering the new part number, the “Test Configuration” will be emptied: because the new part does not have any test sequences/test cases - .
    3. Please note the “Test Case” selector also has the letter “M” indicating that there is the run-time menu:. Select the “Add New Test Case” menu item to open the user interface: . It seems reasonable to use the test requirements number as the test case number. However, it is just recommended to make test validation.

The test “PARAMETER” is the title of the specific test. The “DESCRIPTION” field is self-explained. The “TEST CONDITIONS” is the statement of pass/fail conditions. These fields will be displayed in on MAIN page in the test tree:

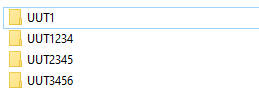
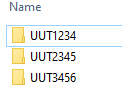




* + 1. The “Part #” selector has four menu items:
    2. The “Edit UUT Description” and “Remove Part Number”, are self-explanatory.

“Remove Part Number” deletes the corresponding folder from the Execution Engine>UUT folder:

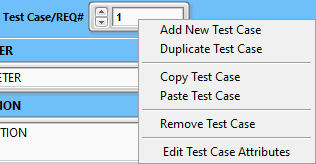
UUT1 Deleted

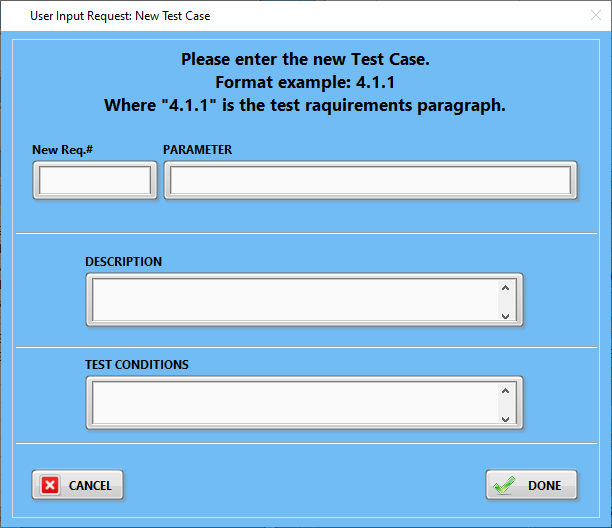
6.3.7 The “Refresh Test Configuration” re-loads the part# test sequence.

* 1. “Test Case” 

The “Test Case” is the set of instructions needed for the execution of the specific test. It seems reasonable to associate the test case with corresponding test requirements. This way it will be easier to conduct the test verification. The “Test Case” selector has its own “run-time” menu:

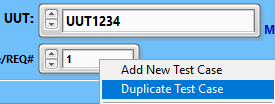


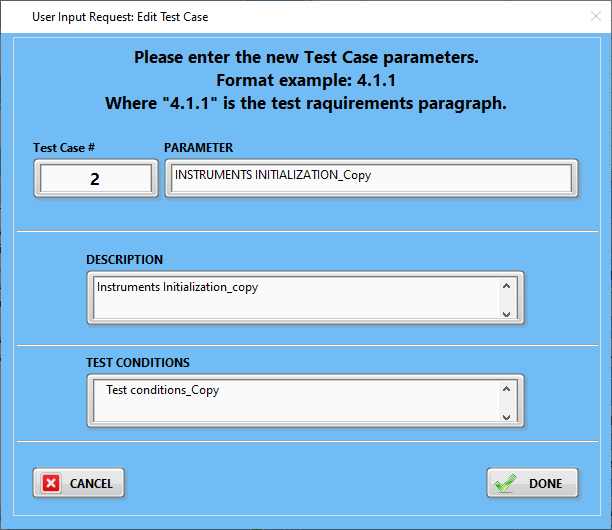
* + 1. “Add New Test Case” allows for the creation of a new set of test instructions. Upon selection of this function, the user will be prompted to enter the test case parameter, description, and test conditions:



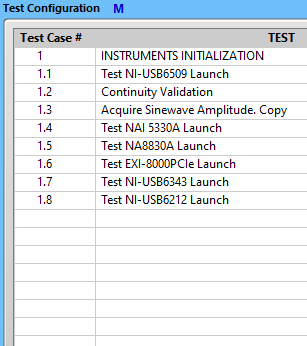
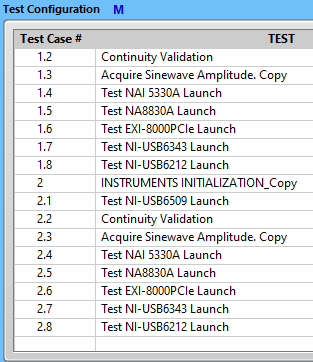
The purpose of these parameters is described in section 6.3.4.

* + 1. “Duplicate Test Case” function allows the user to create a new test case with its instruction set and lower-level test cases. As an example, we will duplicate test case #1 for UUT1234:

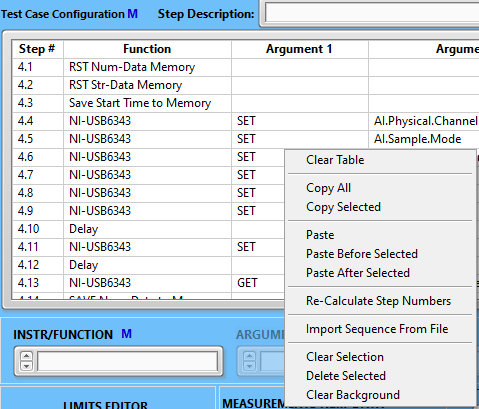
1. Select the test case to be duplicated: . The user will be prompted to enter new test case parameters, descriptions, and conditions:



1. Select the “DONE” button:

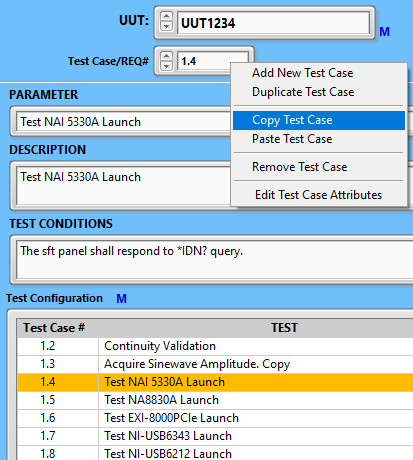
 

1. Please note that “Step#” will be the same as in the source test case. To fix this problem, please, go to the “Test Case Configuration” table, right-click on it to open its menu, and select the item “Re-Calculate Step Numbers”:



* + 1. “Copy Test Case” function allows the user to copy test cases from the selected UUT or different UUTs.

1. Select desired test case:

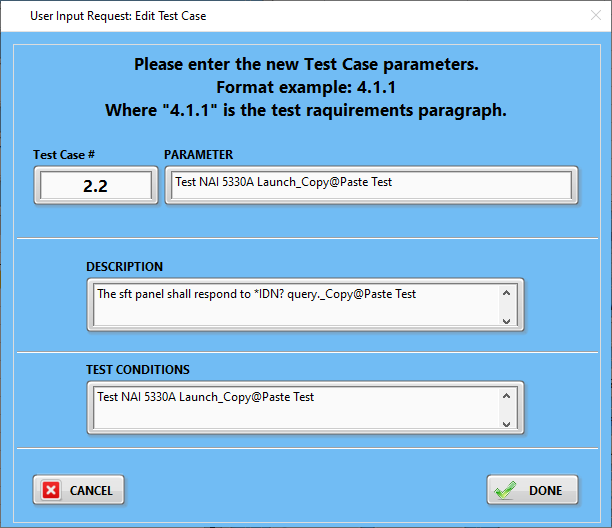
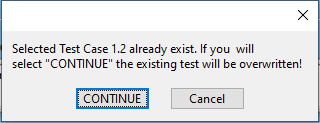


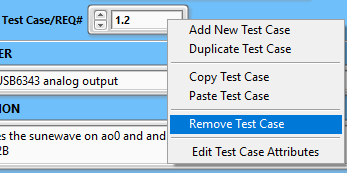
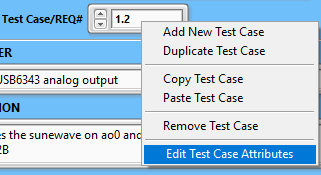
1. Select “Copy Test Case”:

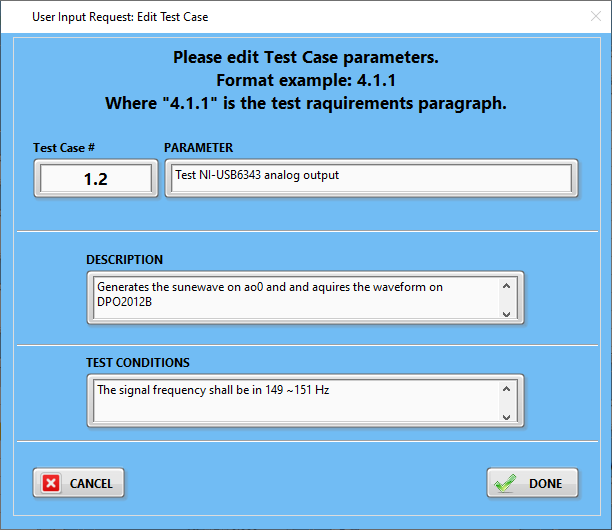
The test case and its instruction set will be saved into the buffer.

* + 1. Select “Paste Test Case”: 

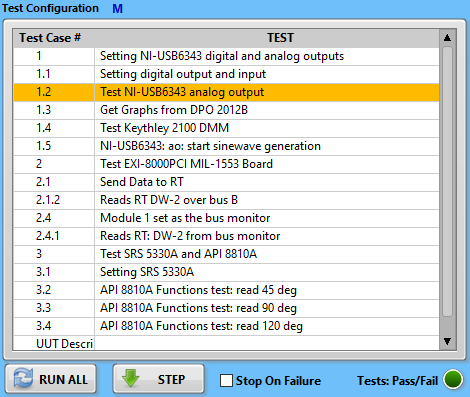
The new test case editing window will open:

 The user shall enter the new test case number at the minimum. If the new test case number is already used, there will be the warning message: .

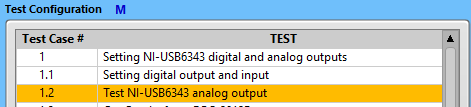
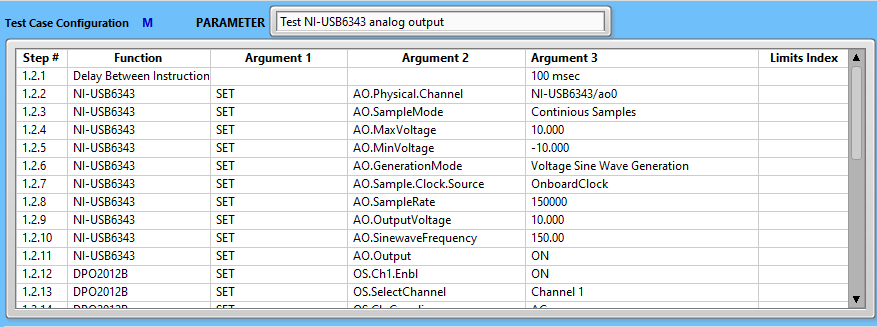
* + 1. “Remove Test Case” function deletes the selected test case: .
    2. “Edit Test Case Attributes”  opens the editing window:



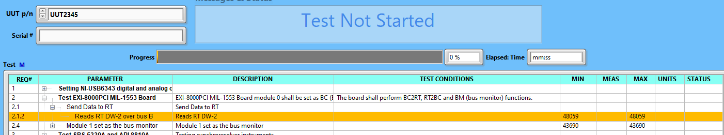
* 1. “Test Configuration” table:

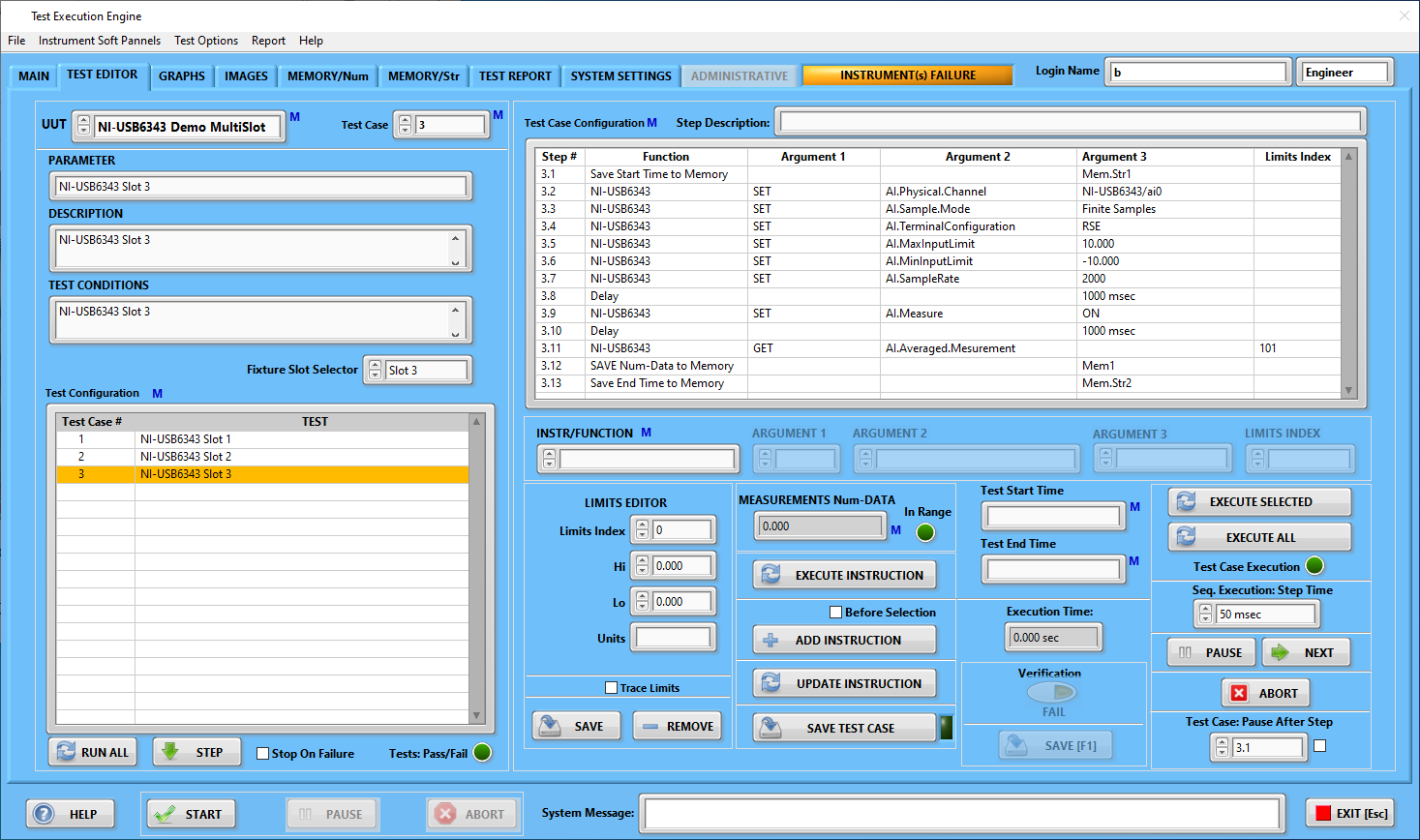


The Test Configuration Table displays the test case numbers and their parameters. By selecting the specific test case, the “Test Case Configuration” table will display the complete set of instruction associated with it:

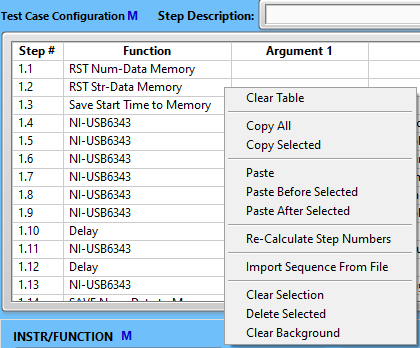
 

* + 1. There is a link between the test tree (MAIN page), Test Configuration, and Test Case Configuration. When the user will select any test case in the tree, it will be automatically reflected in the Test Case, Test Configuration, and to the Test Case Configuration:

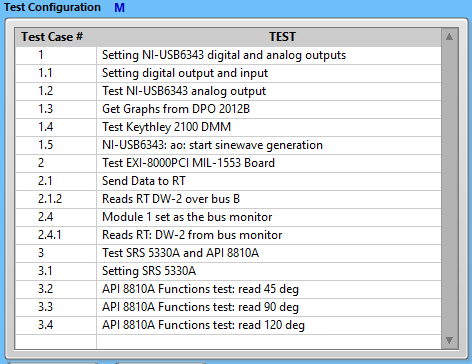
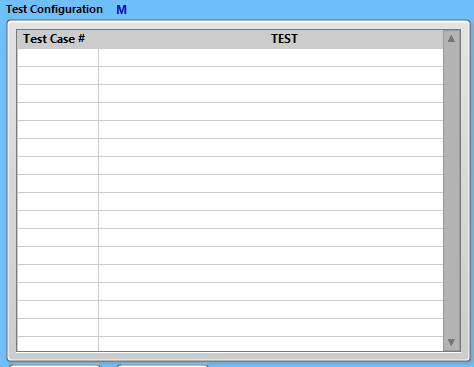
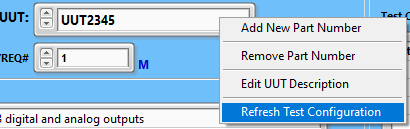


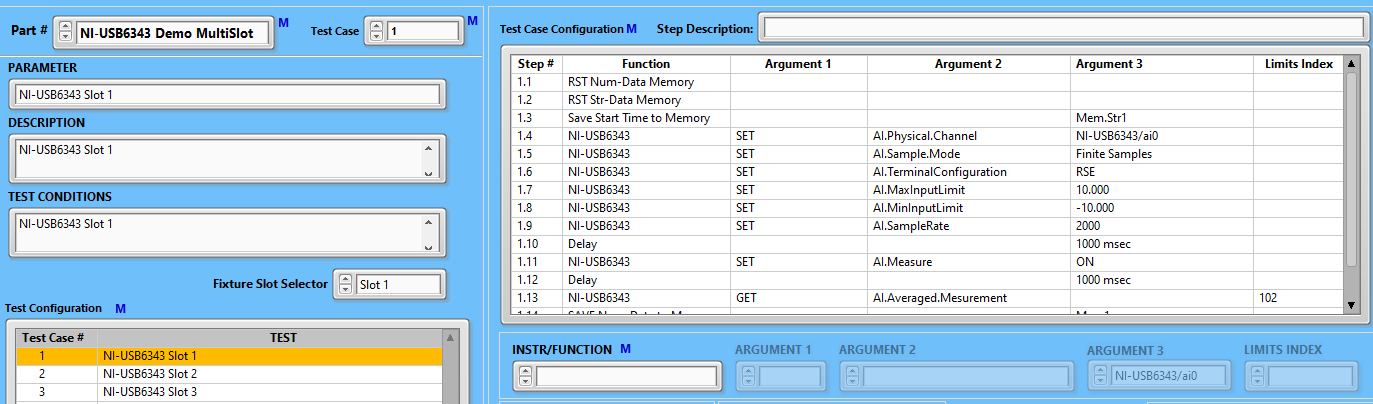


* + 1. The Test Configuration has its own run-time menu:



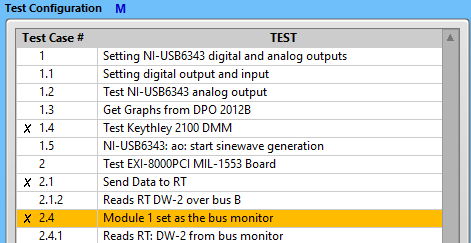
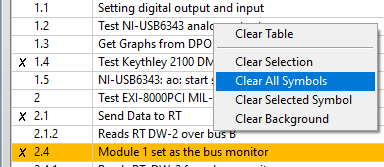
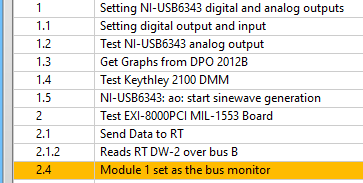
* + - 1. “Clear Table” function removes all contents of the Test Configuration table. Please note, it does not affect any test configuration, it is only clearing display:

 . To display the content of the test, select Part#>Refresh Test Configuration:  or re-select the test case of interest from the “Test Configuration” table:



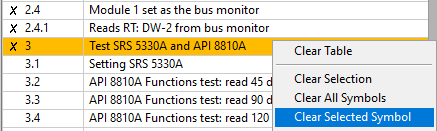
* + - 1. “Clear Selection” function deselects the line in the table:. The Test Configuration table line selection has two functions:

1. Displays the test case instruction set in the Test Case Configuration, and
2. Will run the test starting from the selected line if the  button is selected. Therefore, to execute the complete test, any selections in the Test Configuration table shall be removed.
   * + 1. The “Clear All Symbols” function removes  symbols from the whole Test Configuration table:

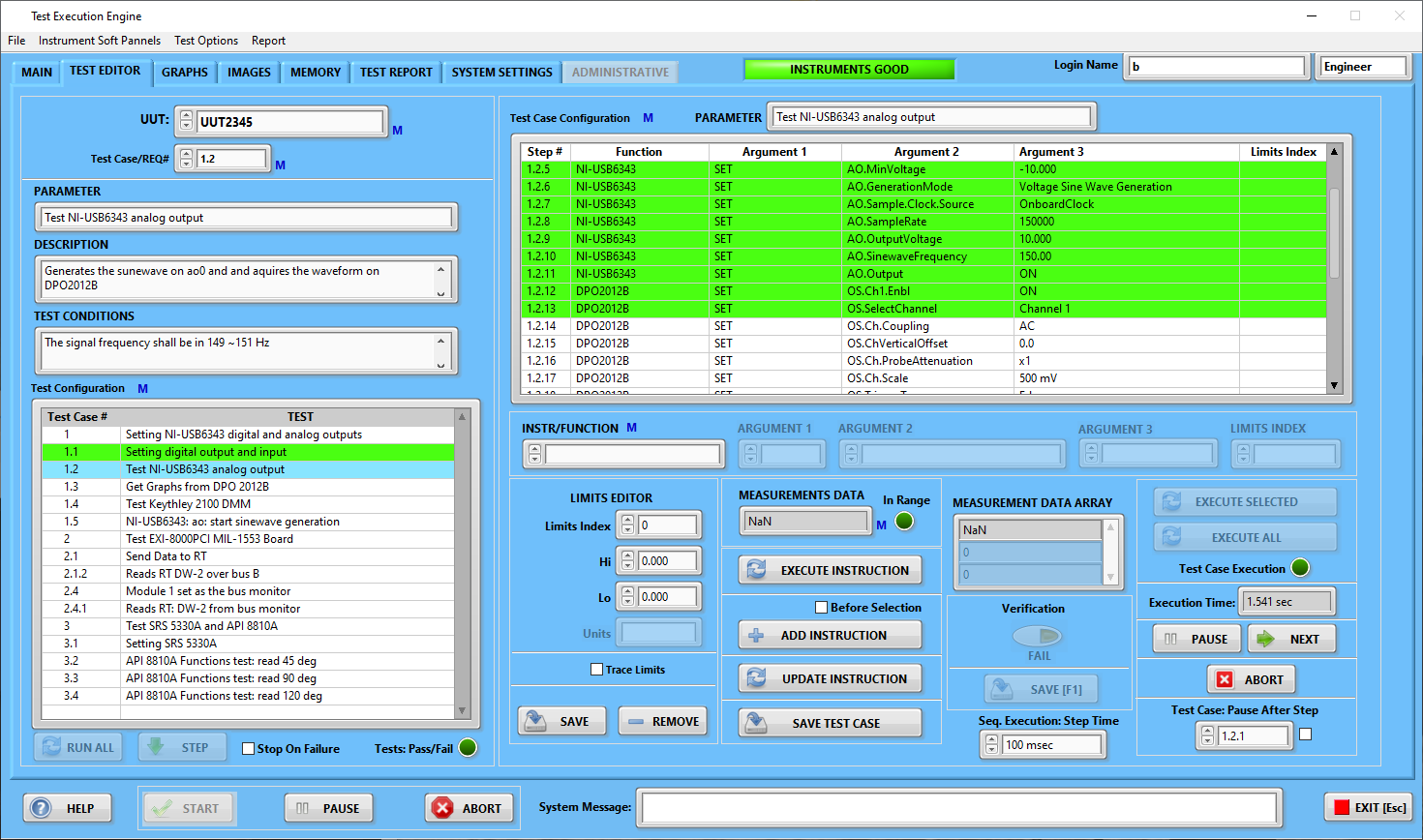
If any of the lines have  symbol, that line(s) will be ignored during test execution. To insert or remove the  symbol, double-click on the selected line.

* + - 1. The “Clear Selected Symbol” function  removes the symbol from the selected line:

or double-click on the selected line.

* + - 1. The “RUN ALL”  button will execute a complete test if no selection is made. During execution, the instruction sub-test line will be highlighted:



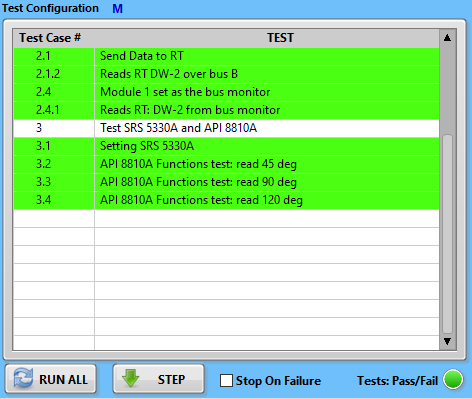
Test case instruction set

progress and status

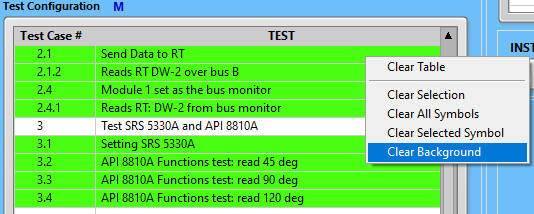
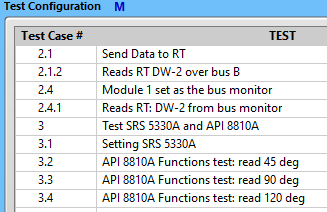
Test case in progress

Test case Pass

Upon finishing test execution, the Test Configuration table will be highlighted:



The “Clear Background” menu item  will remove the Test Configuration table highlighting:

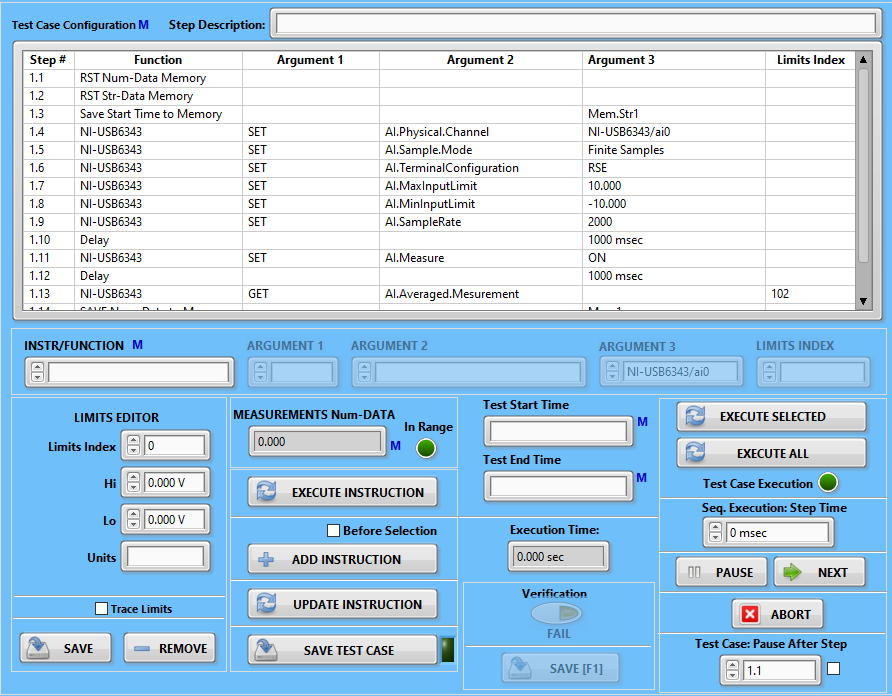
 

* + - 1. The “STEP”  button will step through the test sequences.
      2. The “Stop On Failure” checkbox, if checked, will pause the “RUN ALL” function in the case of test failure.

### Test Development Tools

Test case Configuration Table

Test step description



Limits Editor

Execute whole instruction set

Execute selected instruction

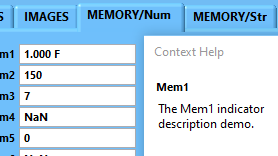
Instruction/Function selector

* + 1. “Step Description” is optional and intended to help the end-user to store information about the meaning of this instruction effect. However, if the step is “SAVE Num-Data to Memory”



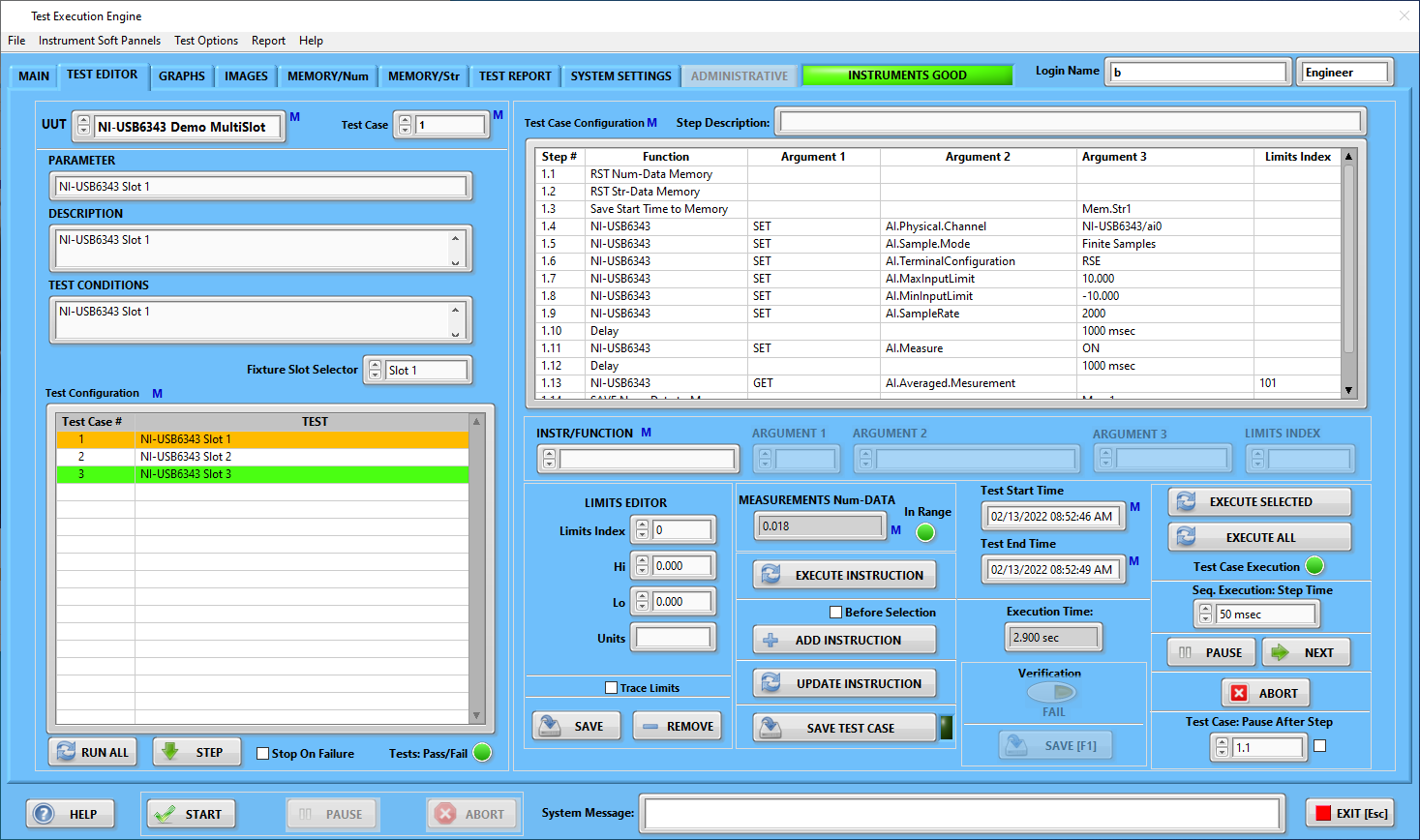
or “SAVE Str-Data to Memory”, the “Step Description” data will be transferred as the corresponding memory indicator’s description. To do this, follow the steps below:

1. Double-click the line with the instruction:” SAVE Num-Data to Memory” .
2. Enter the step description  and .
3. Press the  button.
4. Go to the  page and place the cursor over the “Mem1” indicator .
5. The description of this indicator should display “The Mem1 indicator description demo.”:

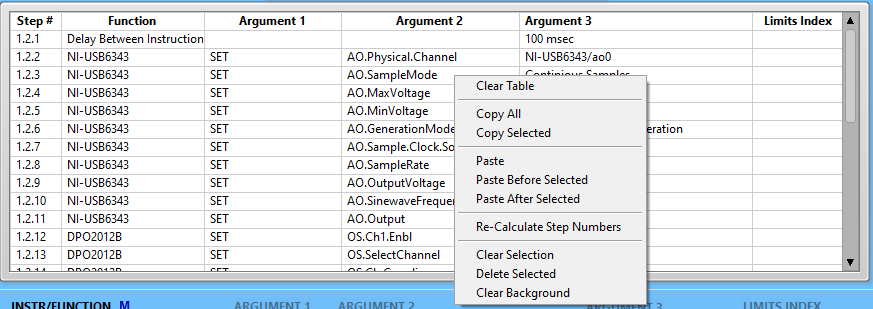


This feature is very useful for test report template creation.

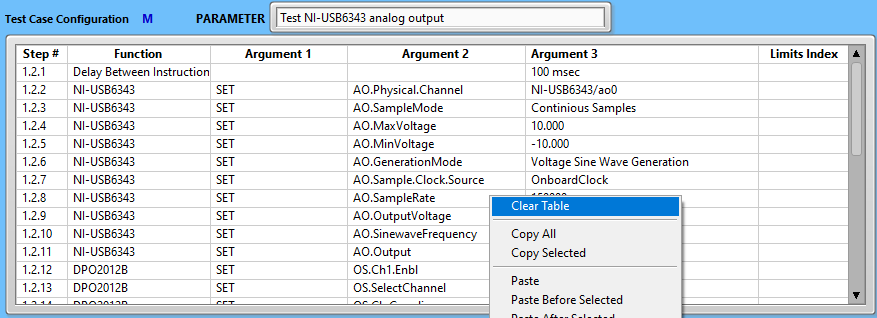
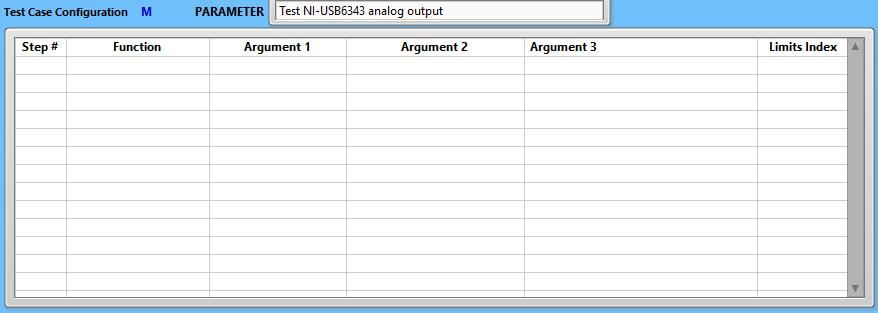
* + 1. The “Test Case Configuration” table lists all instructions for the specific test case:



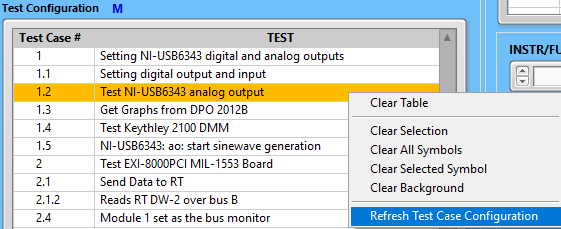
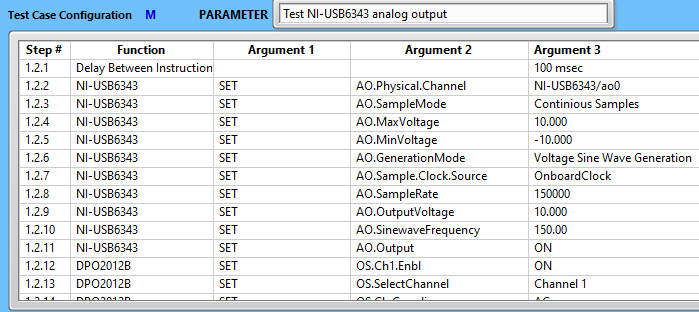
* + - 1. The “test Case Configuration” table has a run-time menu:



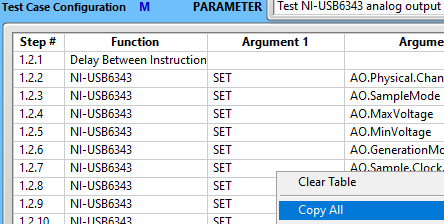
1. “Clear Table” removes all table content. Please note, it does not change the instructions set – just does not display it:

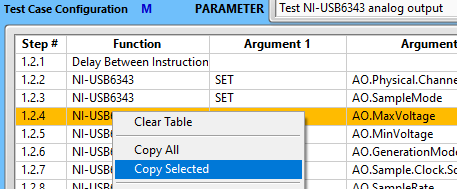
To restore displayed instruction, a select  menu item in the Test Configuration table:

1. “Copy All” function performs copying complete instruction set to the buffer:

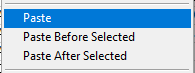
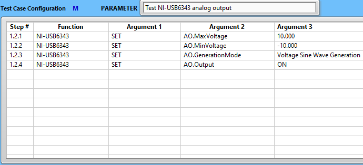


1. “Copy Selected” will copy the selected instructions to the buffer:

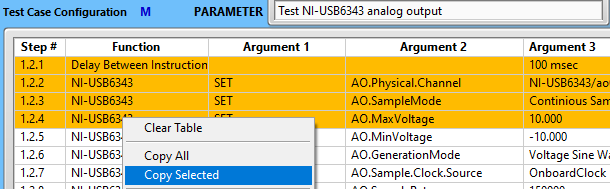
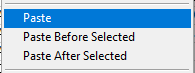
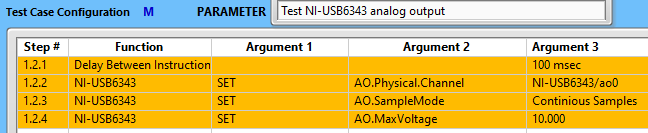
 or 

1. “Paste” will paste previously copied instructions to the Test Case Configuration table:

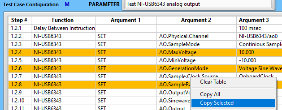
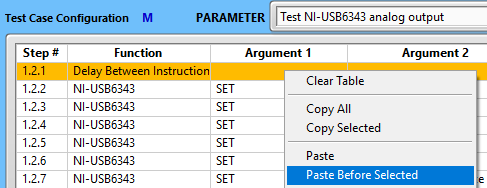
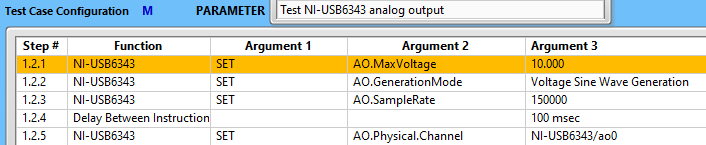
Paste to the empty table:

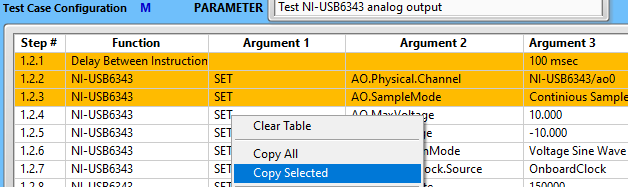
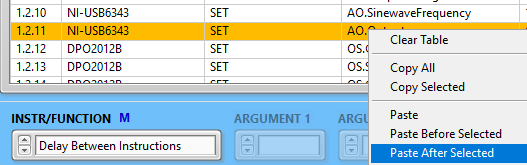
1. “Paste” to the not empty table:

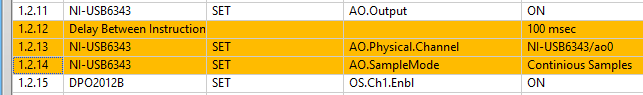
  

1. “Paste Before Selected” to the not empty table:

1. “Paste After Selected” to the not empty table:

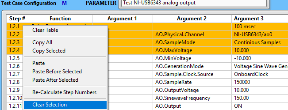
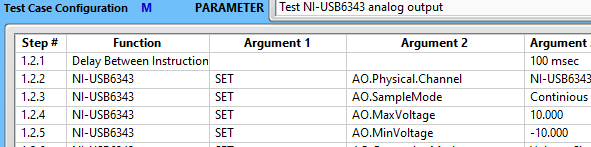
 



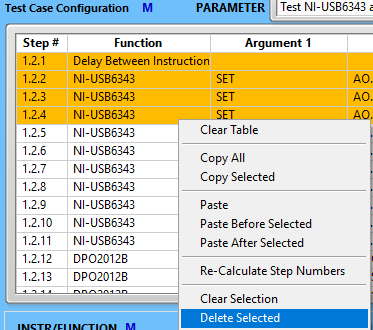
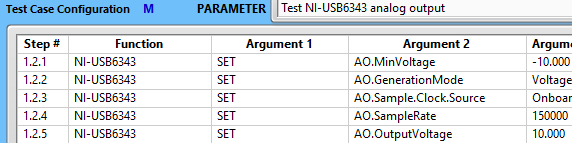
1. “Re-calculate Step Numbers” function is self-explanatory:



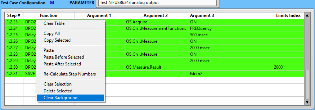
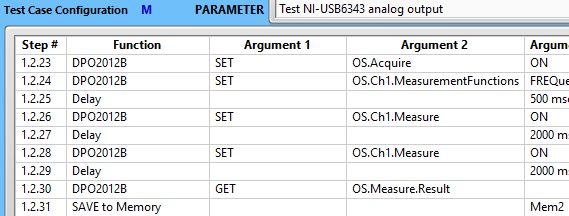
1. “Clear Selection” performs de-selection:

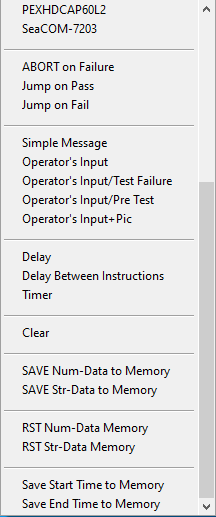
1. “Delete Selected”:

1. “Clear Background”. During test execution, the Test Case Configuration table has highlighted rows - green for successfully executed instructions and red for failed ones. The “Clear Background” function removes that highlights:

* + 1. The “INSTR(uments)/FUNCTION  selector lists installed instrument soft panels and Execution Engine specific functions:

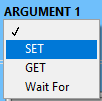


Exec Engine Instructions

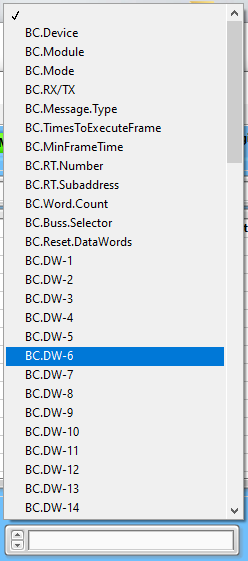
Instrument Soft Panels

* + - 1. Instrument soft panels’ selection provides access to the corresponding instrument functions. There are associated selectors – arguments:



1. “ARGUMENT 1”  has three values:. Upon selection, the “ARGUMENT 2” is populated.

As an example, in the picture above, the Excalibur MIL-1553 board EXI-8000PCIe was selected. As soon, the Argument 1 value was selected as “SET”, the values of the ARGUMENT 2 were populated with this particular instrument functions:



Upon ARGUMENT 2 selection, the proper type of ARGUMENT 3 is displayed. The ARGUMENT 3 can be of three types: numeric, “text ring” and string. The numeric display format is adopted to the corresponding ARGUMENT 2 type:



16 Bit integer



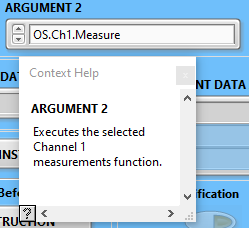
Floating point

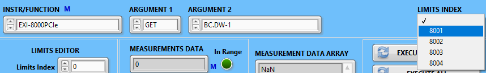


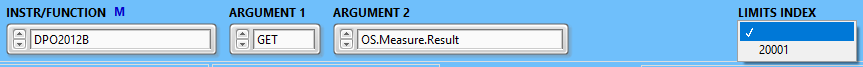
Text ring



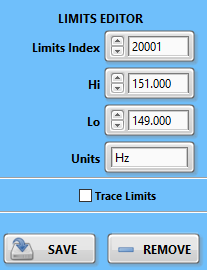
String

1. The Execution Engine application has an online help function: . Pressing this button  will open the help window: . Putting the cursor over the “ARGUMENT 2”, will display the help for the selected function.
2. ARGUMENT 1 “GET” function  intended to receive data from an instrument. Please note that ARGUMENT 3 is disabled and grayed out  because this control is not applicable. However, “LIMITS INDEX” becomes enabled . The “GET” function, in the essence, is used for the evaluation of measurements in terms of “test pass or fail”. In order to do this, the limits shall be declared. It should be noted, that for user convenience, limit indices are associated with the specific instrument:

**** or



To define test limits, the Execution Engine application provides a facility for limits configuration:



Saves the limit’s data to the file

If checked, the controls will be updated during test execution.

Units

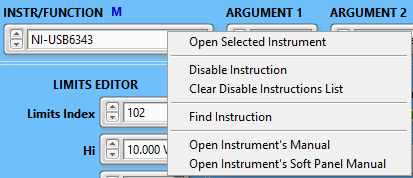
Low Limit

High Limit

Limits index

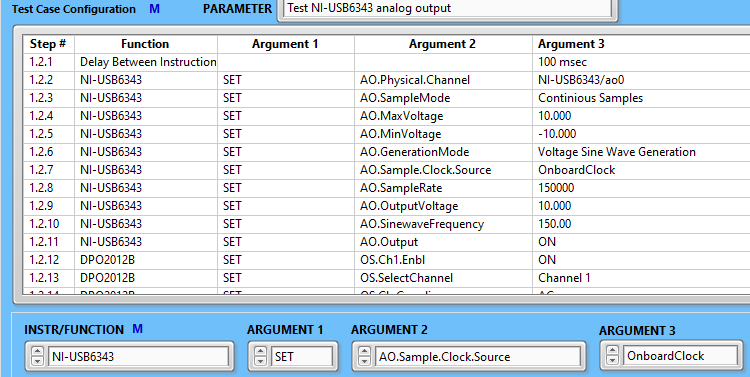
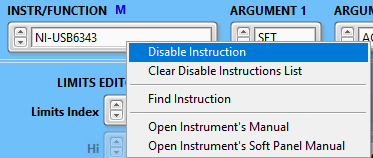
Deletes the limit’s section from the file.

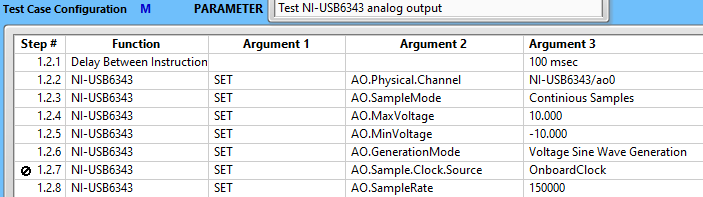
1. The “INSTR(uments)/FUNCTION  selector has the run-time menu (denoted by letter M):

****

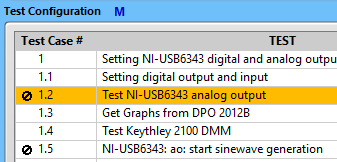
1. The “Open Selected Instrument “will open the instrument’s soft panel.

The “Disable Instruction” function will mark the selected instruction in the Test Case Configuration table:

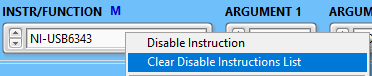
 

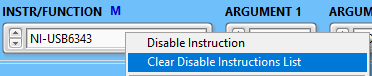


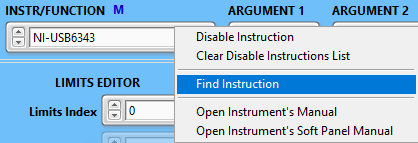
and, in the Test Configuration table:

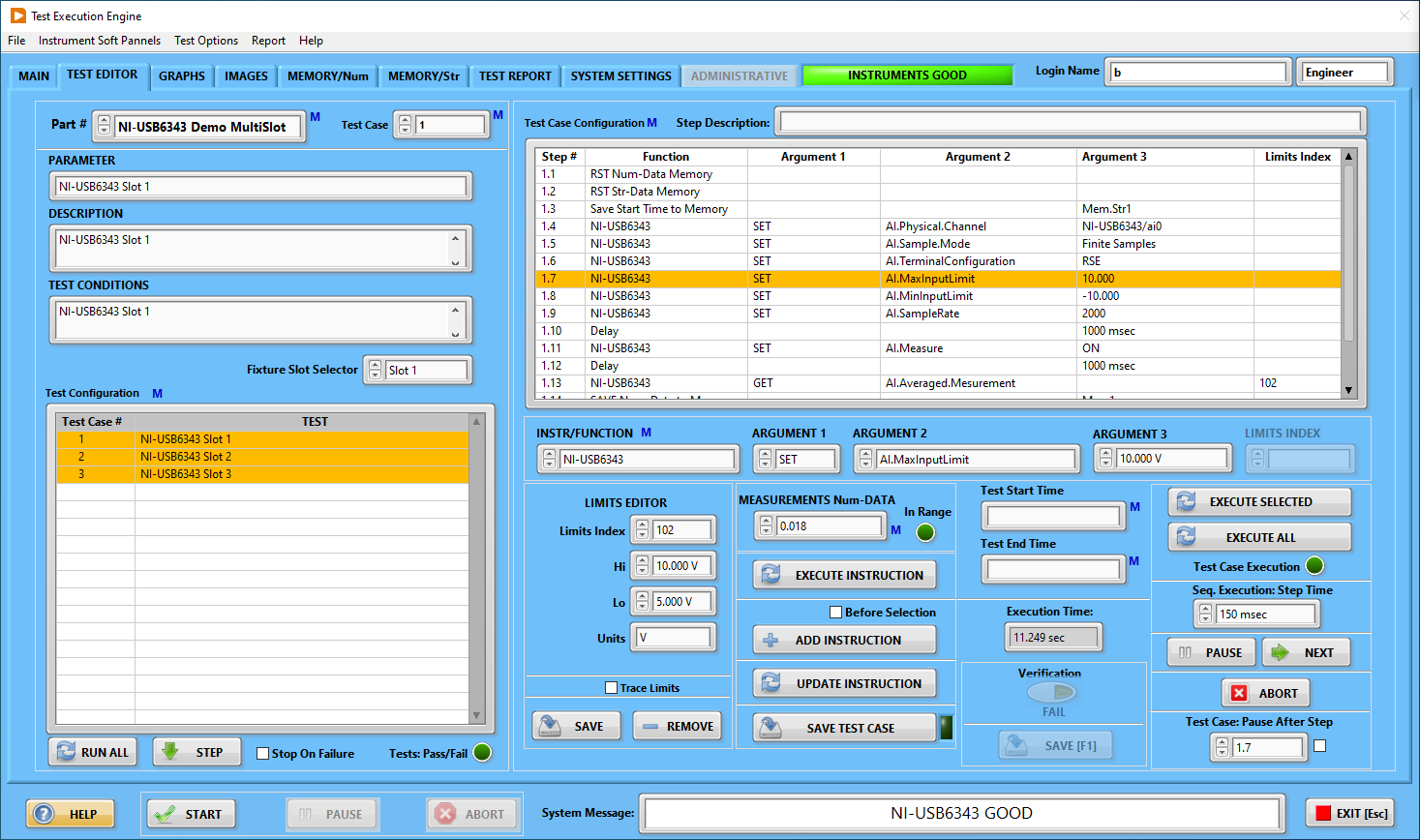


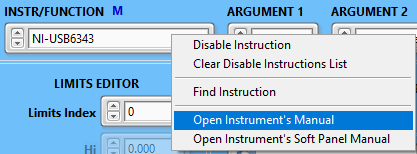
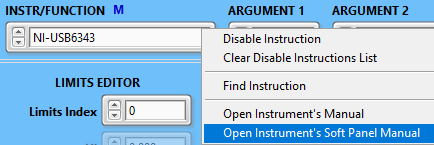
In this case, the marked instruction(s) will be excluded from execution while pressing the “EXECUTE ALL”  or “RUN ALL”  buttons. At the same time, disabling instruction(s) will not affect the test execution from the operator’s page: .

1. “Clear Disable Instructions List”:

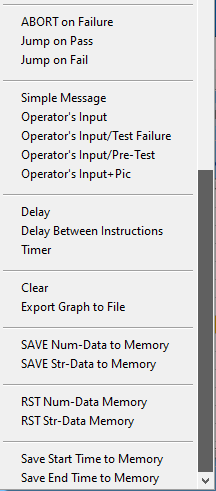
  

1. “Find Instruction”  will find the specified instruction in the Test Case Configuration and Test Configuration tables:



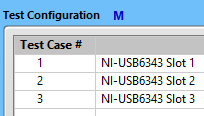
1. The “Open Instrument’s Manual”  function will open the instrument’s manufacturer user manual in the ADOBE Acrobat Reader. This is done for the user’s convenience.
2. The “Open Instrument’s Soft Panel Manual”  function opens the instrument soft panel user manual.

### 6.6.4 Execution Engine Instructions

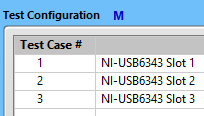


The Execution Engine has a set of functions not related to the instruments, but rather to the process of instructions execution.

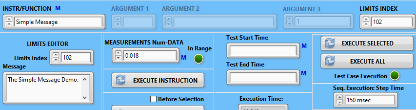
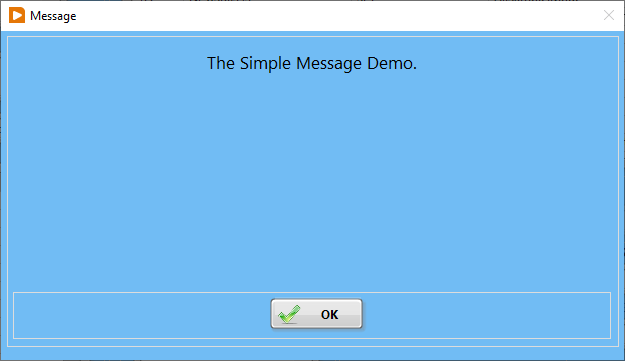
1. “ABORT on Failure” will terminate the test case execution in the case of test failure.
2. “Jump on Pass” will skip the execution of instructions below it and continue with the test case specified:

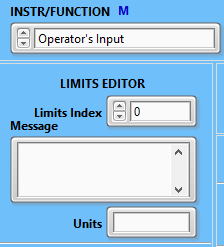
1. “Jump on Fail” will skip the execution of instructions below it and continue with the test case specified:

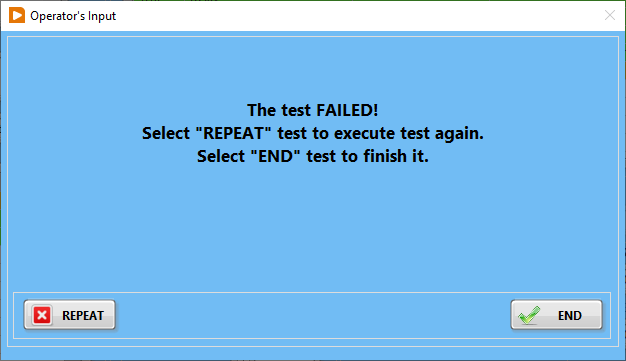
1. “Simple Message” allows to display instructions to the operator:

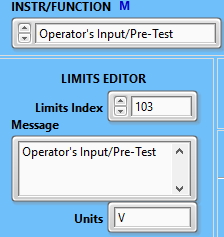
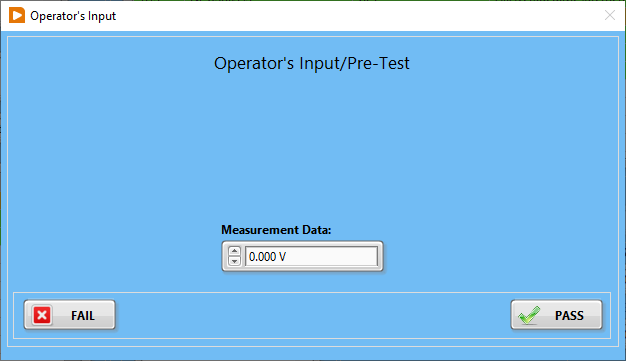
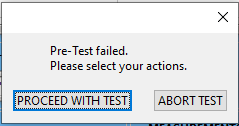
1. “Operator’s Input”: before inserting this function into the “Test Case Configuration” table, the “Operator’s Input” shall be configured:

 . This function should be used when the operator has to make manual measurements.

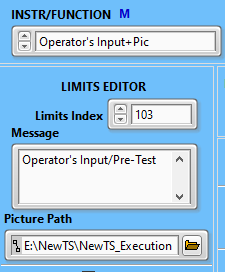
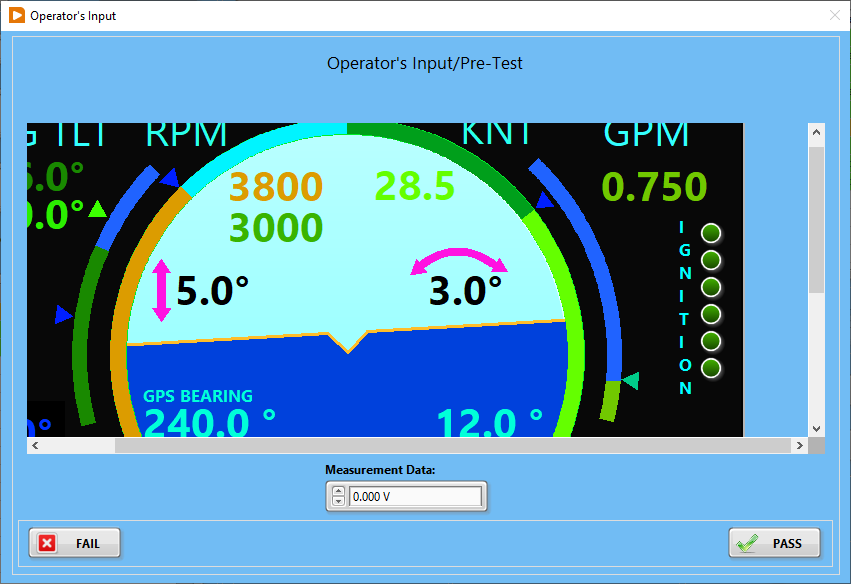
1. The “Operator’s Input/Test Failure” is used to prompt the operator to decide in the case of the test failure:



1. The “Operator’s Input/Pre-Test” function prompts the operator to enter manually obtained measurement data and decide if the test passed or failed:

  In the case of failure, the operator is provided with options: 

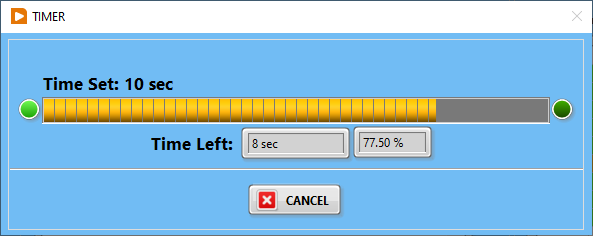
1. The “Operator’s Input + Pic” allows displaying visual instructions to the operator:

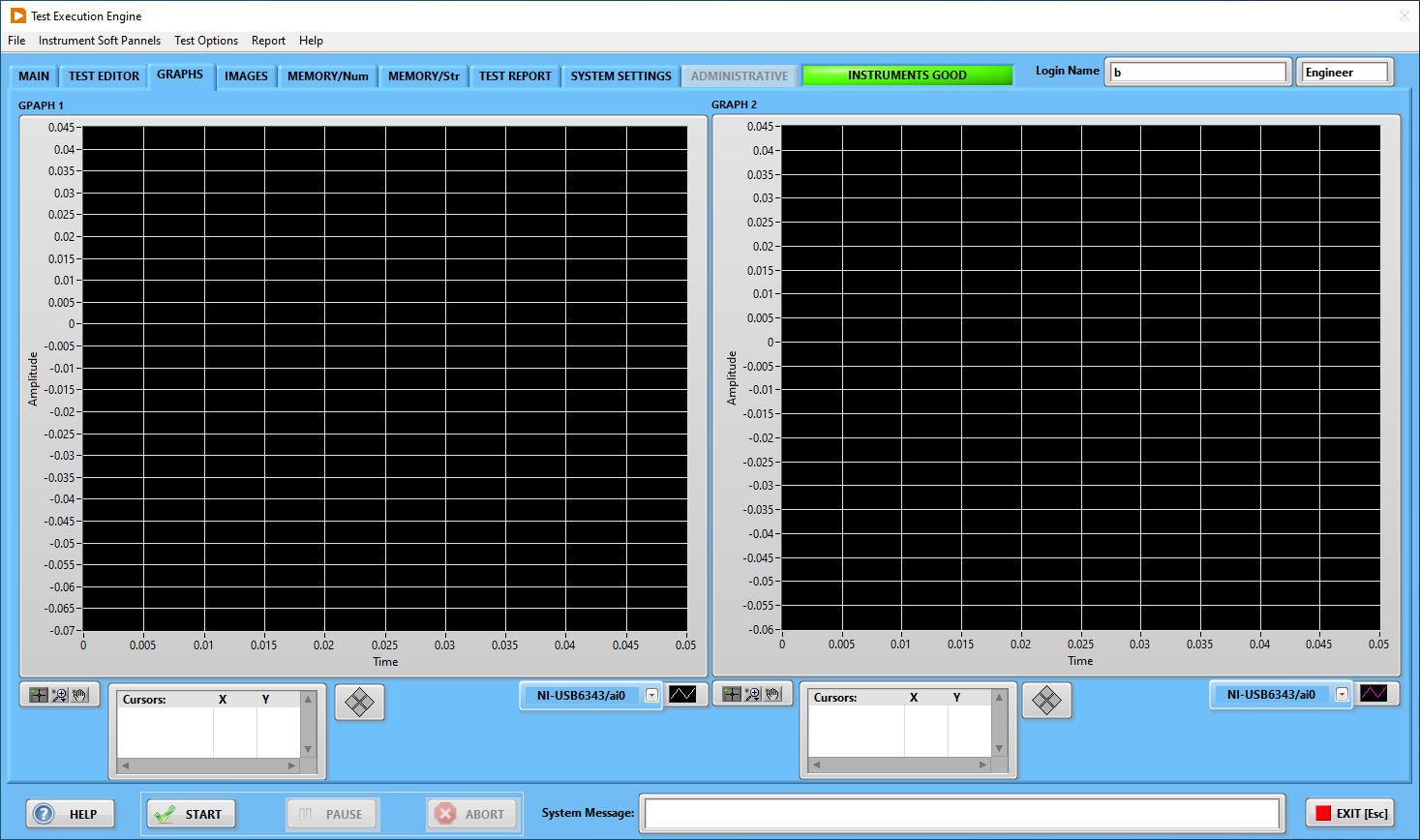
1. “Delay” sets the time delay before executing the next instruction:

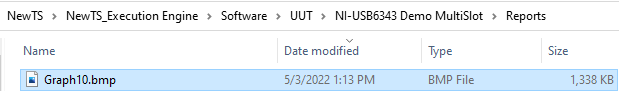


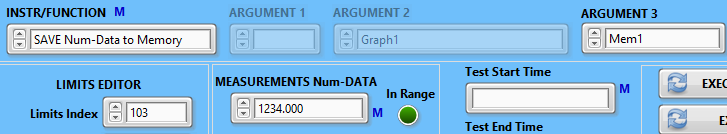
1. The “Delay Between Instructions” sets the time delay between the execution of the instructions in the sequence. Some instruments require a long time to execute an instruction: . Once it is set, it will propagate through all test cases. For this reason, this parameter should be used carefully and re-set if needed.
2. The “Timer” function is the similar to “Delay” function but provides the visual representation. It is useful when there is a need to stabilize the temperature in the environmental chamber:

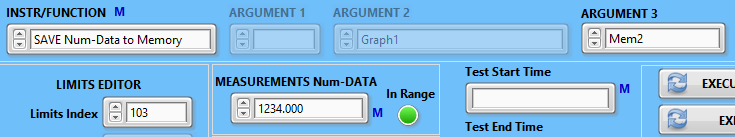
1. The “Clear” “Graph1” or “Graph2” deletes data from the graphs found on the “GRAPS” page:



1. The “Export Graph to File” saves the graph image to the “Reports” folder as a .bmp file: .
2. The “SAVE Num-Data to Memory” function performs the transfer of the measurement data to the specified “memory” location:

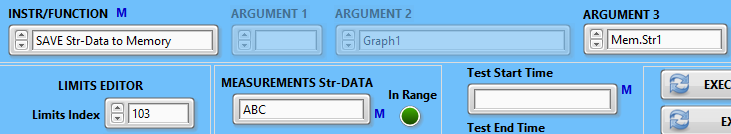
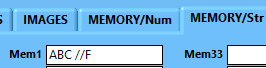
 

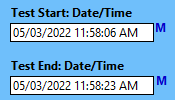




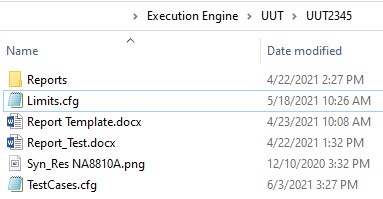
1. The “SAVE Str-Data to Memory” function performs the transfer of the measurement string data to the specified “memory” location:

1. The “RST Num-Data Memory” and “RST Str-Data Memory” functions remove previously saved data from numeric and string data.
2. The “Save Start Time to Memory”  and the “Save End Time to Memory”  functions save the corresponding data to the “memory” locations:. This data may be needed for a test report.

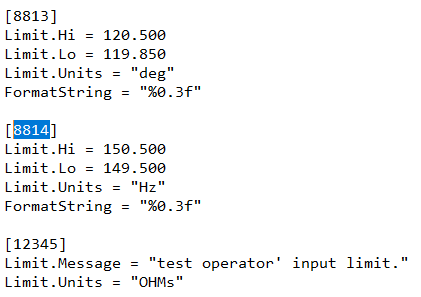
### 6.6.5 The test limits configuration

The test limits, used for the selected part number (UUT) are located in the “Limits.cfg” file in the corresponding folder:

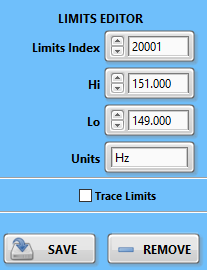


Limits File

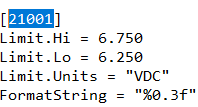
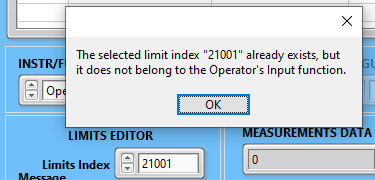
The “Limits.cfg” file is the configuration file and it is a text file:



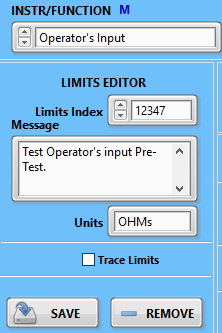
Please note, that it is highly unadvisable to edit this file directly. Please use the Execution Engine application limits editor facility:



While entering the “Limits Index”  value, the Execution Engine application automatically checks if this index already exists in the “Limits.cfg” file to prevent unintentional limits overwrite:

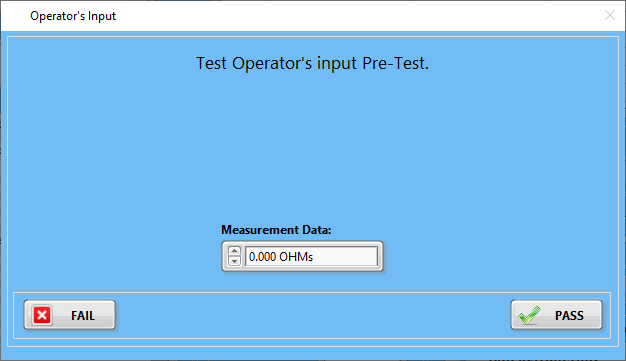
 

To configure the operator’s input limit, as an example, the user shall define the Limits Index, message, and units:

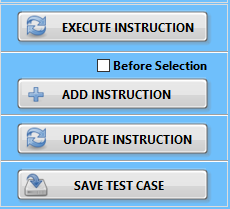


After entering the required data, the user shall save this data.

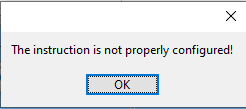
To view the operator’s input in actions, press the “EXECUTE INSTRUCTION” button:

### 6.6.6 Test Case creation functions

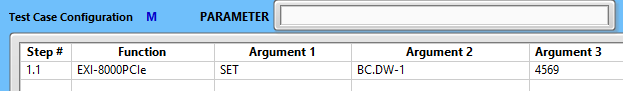


1. The “EXECUTE INSTRUCTION” function  sends the configured above instruction: .

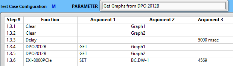
Please note, that all instruction fields MUST be filled, otherwise, the error message will be generated: .

1. “ADD INSTRUCTION”  function inserts the properly configured instruction into Test Case Configuration table. There are three possible cases:

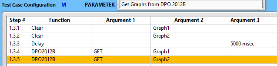
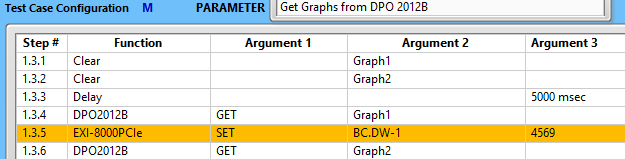
The table is empty or there is no selection made:

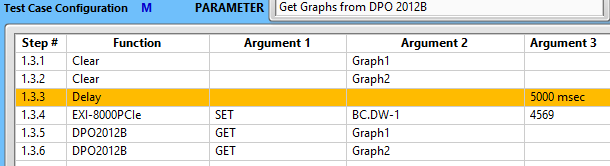
  

If the table is not empty and there is no selection made, the new instruction will be added to the end of the table:

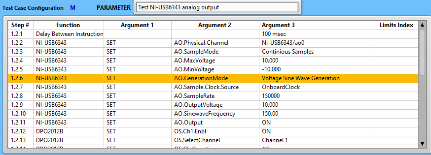
  

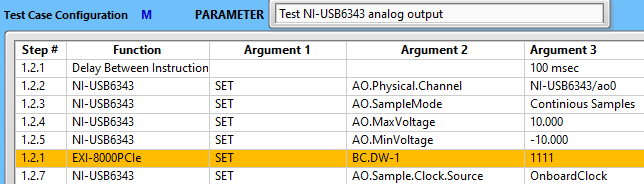
If the Test Case Configuration table is not empty and the instruction line is selected, the new instruction will be added according to the check box:

   or

1. “UPDATE INSTRUCTION”  function performs the update of the selected instruction from the Test Case Configuration table. To use this function, the instruction line in the Test Case Configuration table must be selected:

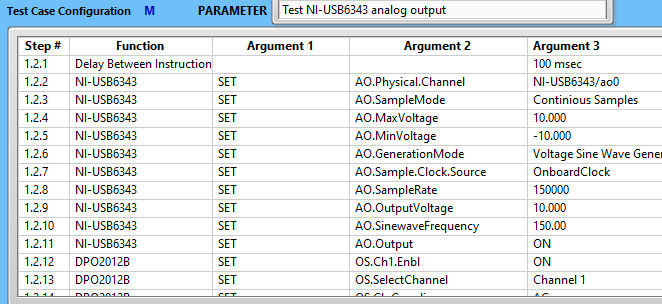
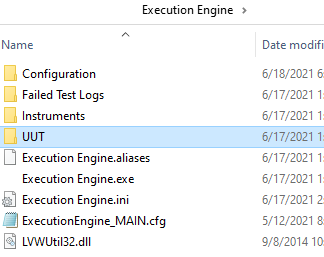
  

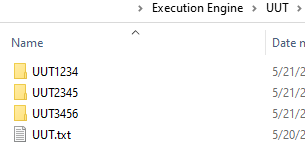


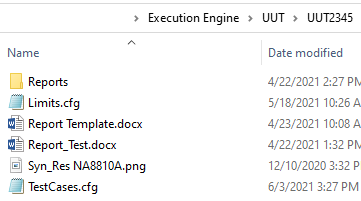
As can be seen from this example, the selected instruction in the Test Case Configuration table was replaced with a new instruction. However, intend of the “UPDATE INSTRUCTION” function is to edit existing instructions. The workflow of this process is as follows:

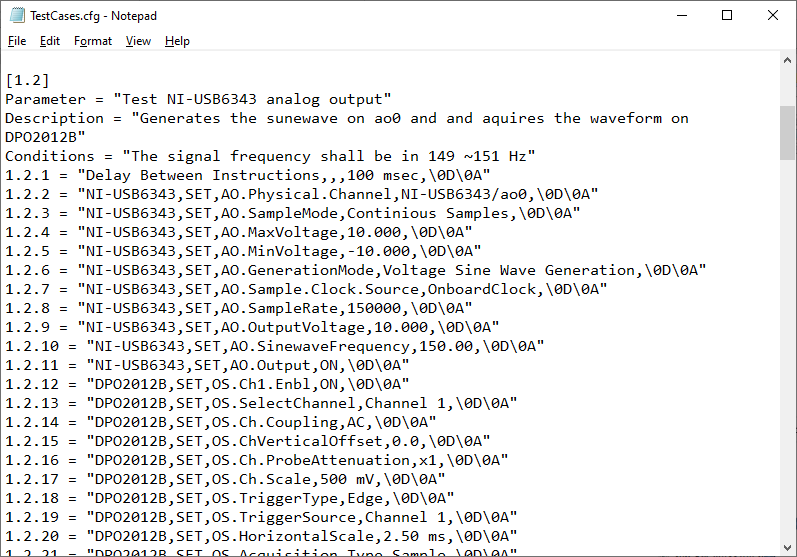
* Double click on the desired instruction in the Test Configuration Table. As the result, the instruction configuration field will be populated with values from the table.
* Change arguments or limits index as needed.
* Press the “UPDATE INSTRUCTION” button.

1. The “SAVE TEST CASE”  function performs saving of the test case sequence to the “TestCases.cfg” file:







Please note, that if the test save function is executed correctly, the indicator next to the “SAVE TEST CASE” will turn green:. If any changes to the test case were made and it was not saved, this indicator will be dark green (OFF):.

### 6.7 Evaluation of the test results

Any test, as the process, consists of two segments:

1. Set up test conditions. It is executed by the function “SET”:

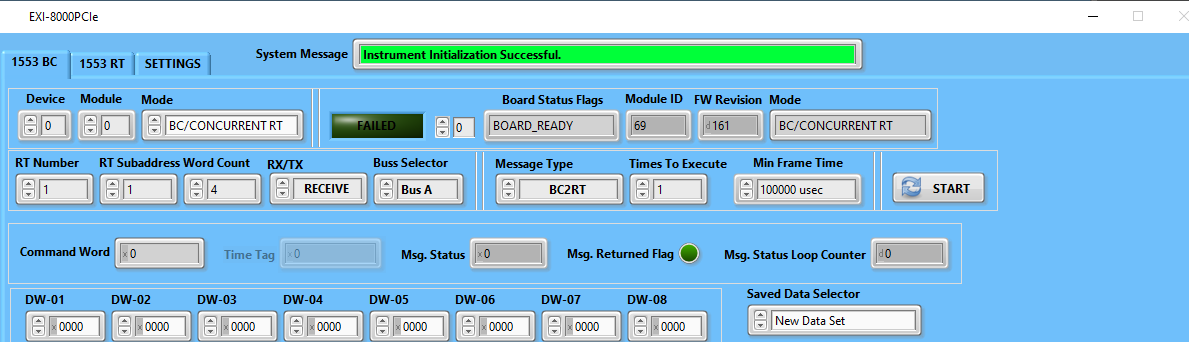


1. Read the instrument’s measurement, which is performed by the function “GET”:

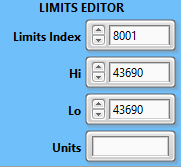


In this section, we will be talking about the “GET” function.

Execution of this function includes a query of the instrument for the specified parameter. In the example above, the instrument “EXI-8000PCIe” (Excalibur MIL-1553) will be queried for the value of “BC.DW-1”:

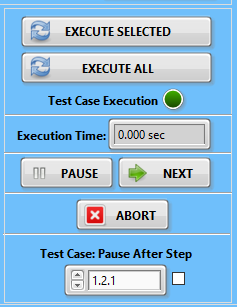


The obtained result will be displayed in the “MEASUREMENTS DATA” indicator.

The measured data is compared with data specified by the “LIMITS INDEX” , which was previously configured by the “LIMITS EDITOR”: and saved to the “Limits.cfg” file.

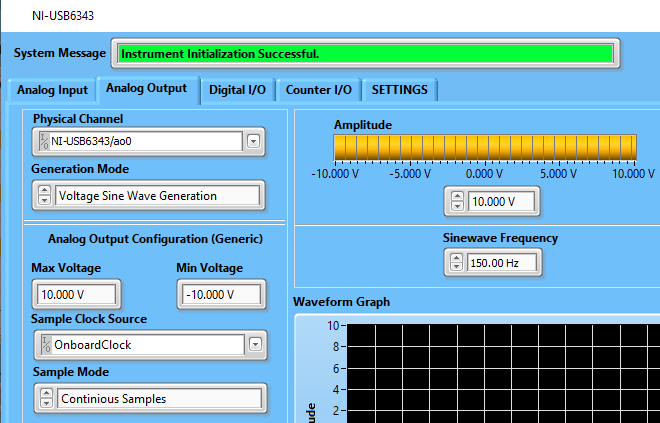
The results of data evaluation are reflected by the “In Range” indicator .

### 6.8 Execution of the test case

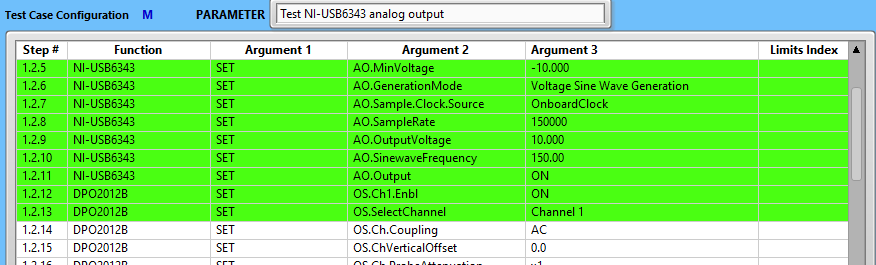
After completion of the test case configuration, the user can execute not just single instruction but the whole test case using the facility provided:

* + 1. The “EXECUTE SELECTED”  function performs the execution of the selected instruction lines:



6.8.2 The “EXECUTE ALL”  function performs the execution of the whole test case from the Test Case Configuration table. During execution, the instruction line, which was executed is highlighted. If the instruction is executed successfully, the highlight color is green, in the case of failure, the highlight color will be red. Please remember, these highlights indicate just the success or failure of the communication of the Execution Engine to the corresponding instrument’s soft panel, not to a physical instrument.



The results of instruction execution (NOT THE TEST RESULTS!) are reflected by the “Test Case Execution”  indicator. The test case execution time is displayed by the “Execution Time”  indicator.

The test case execution can be paused . If the “PAUSE” button  was pressed before pressing the “EXECUTE ALL” , the test will be paused before the start of execution. If the button “PAUSE” is pressed during the test case execution, it will be paused after the instruction is executed.

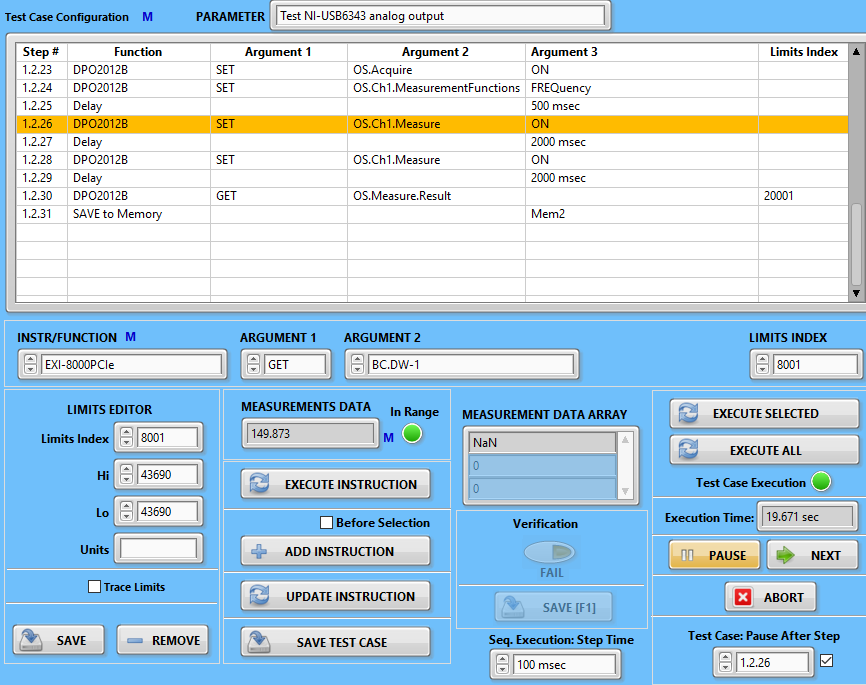
To continue the test case execution, the user can depress the “PAUSE” button:

The user can step through the instruction set, the user can use the “NEXT”  button.

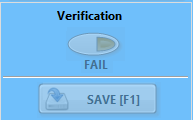
The user can terminate the test by using the “ABORT”  button.

If, during test case development, the need to pause the test at a specific instruction line, the user can select the instruction line before desired and set the check box:

 Check box

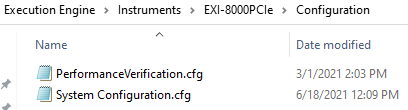
To set the “Case: Pause After Step”  selector is automatically updated when the instruction line in the “Test Case Configuration” table is selected. On the other side, this selector contains all instruction numbers, thus it can be selected manually as well.

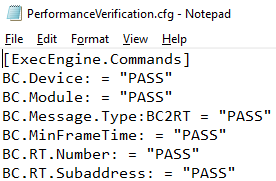
* + 1. The “Verification” section is intended for the instrument’s soft panels verification and implemented for developers’ convenience:



Some instrument’s soft panels have large numbers of the “ARGUMENT 2” items and it may be difficult to verify their performance. The “Verification” controls can be enabled by checking the box on the “SYSTEM SETTINGS” page: .

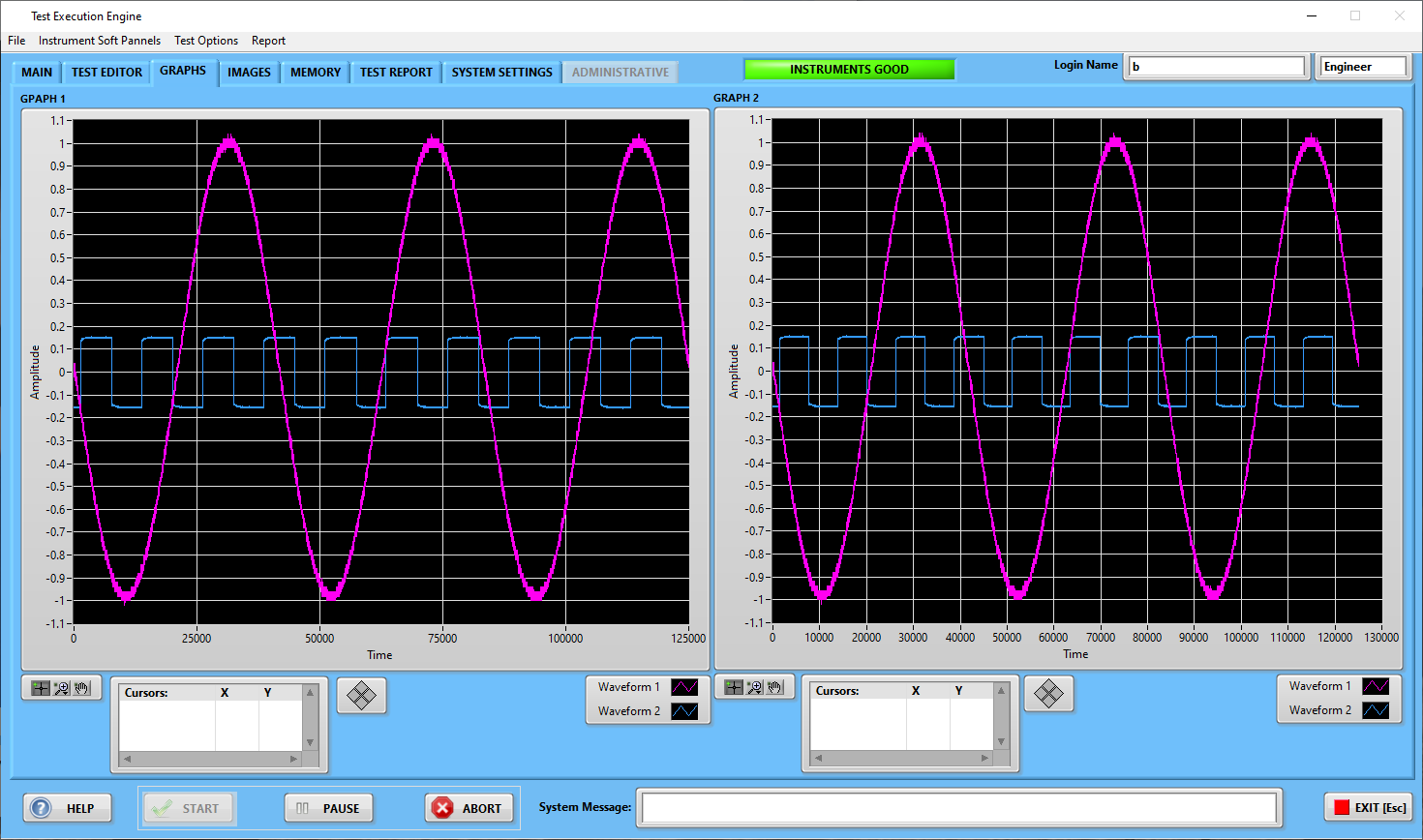
When this function is enabled, the developer can save to file the current instruction execution status:





## “GRAPHS” page

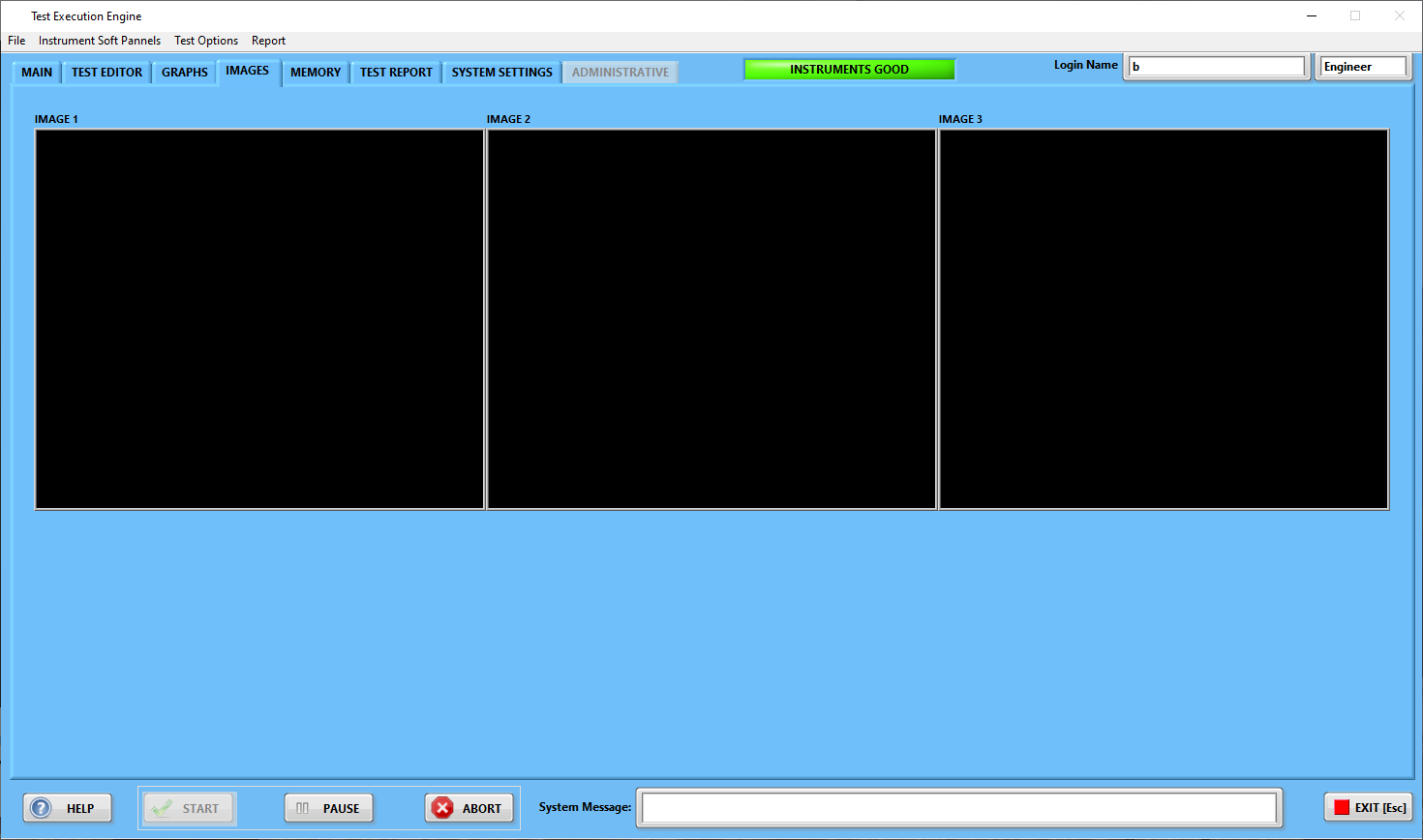
The graphs page has two waveform graphs:



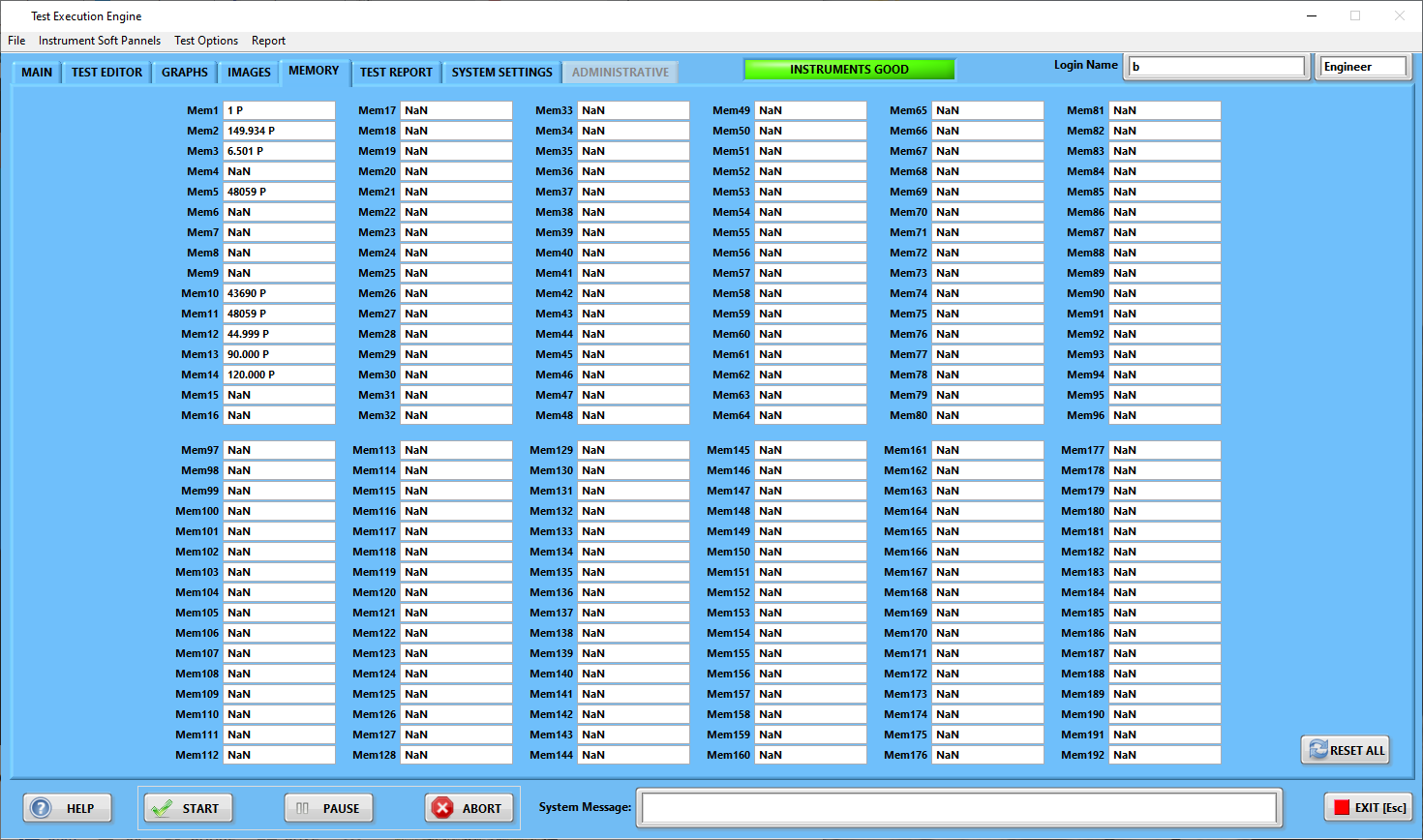
In this case, the data for these graphs was acquired from the oscilloscope.

## “IMAGES” page

There are three image indicators:



## “MEMORY/num” page:



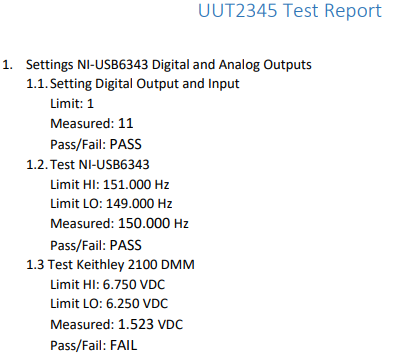
The memory page contains 192 digital controls as the data placeholders. During test execution, the measurement results can be placed to these controls using instructions:



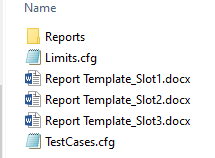
This function intends to transfer measurement results to the test reports. The test data formatted to reflect Pass/Fail:



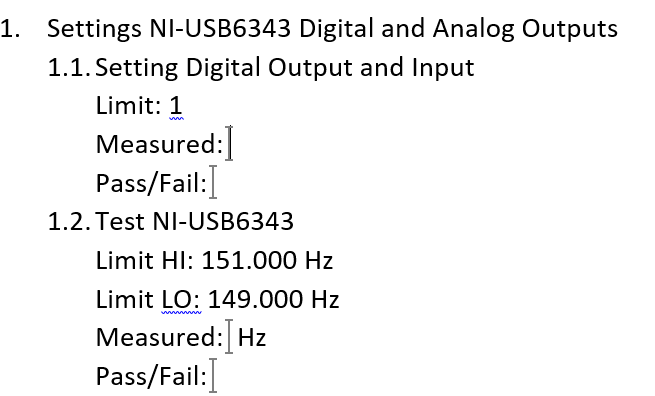
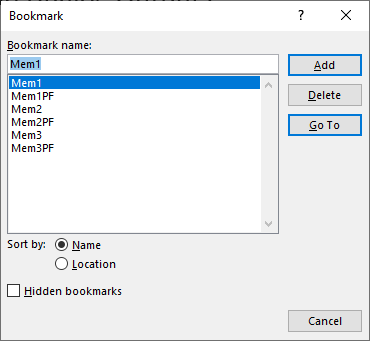
Below is the fragment of the Test Report Template:

Since the Execution Engine can handle fixtures with multiple slots, each part number (UUT) has the report template file assigned to fixture slots:



The Report Template File has bookmarks where test data will be inserted:

During test report formatting, the memory data will be divided into two parts – numeric and text:



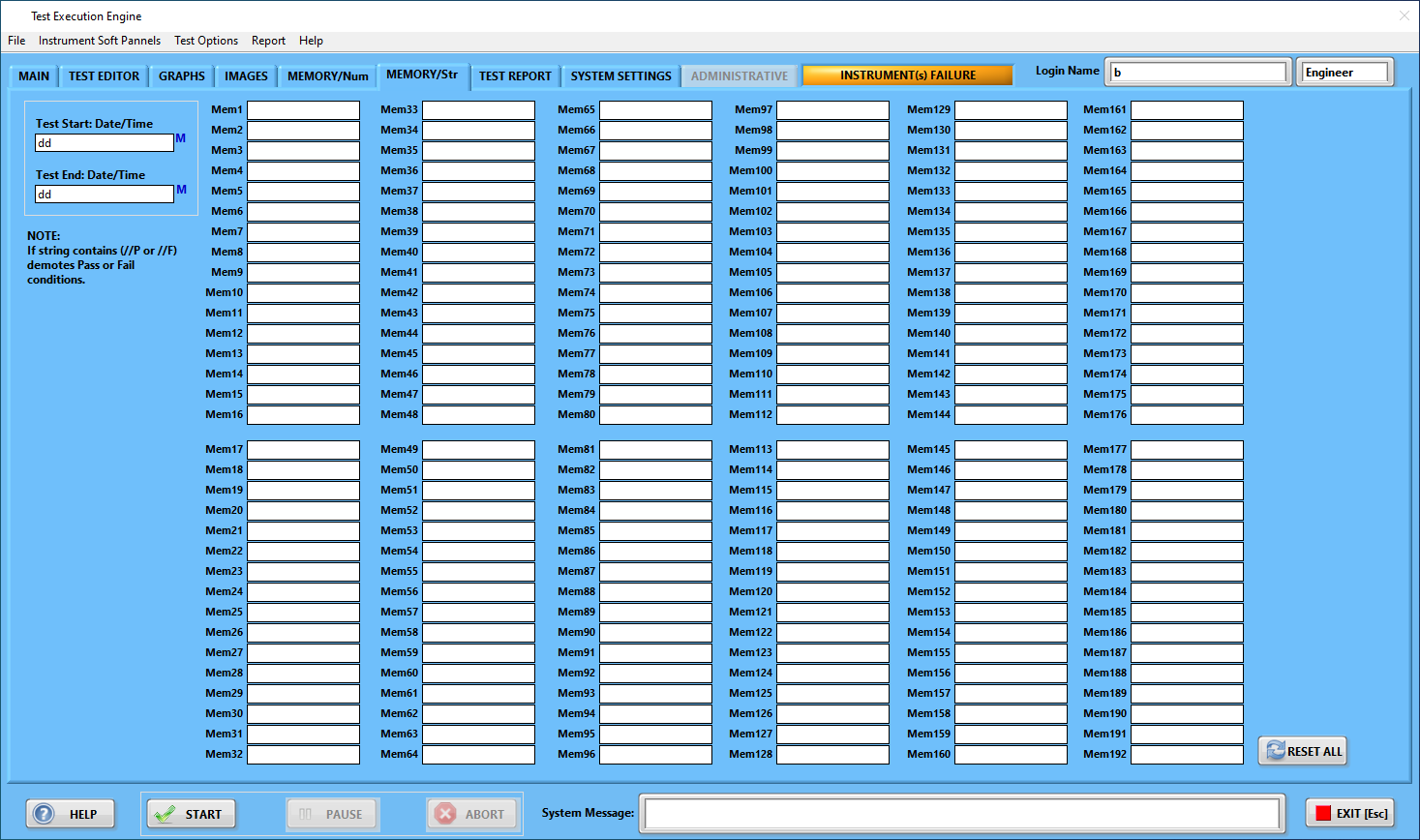
PASS/FAIL

Numeric

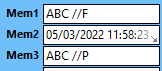
## “MEMORY/str” page

* 1. The memory page contains 192 digital controls as the data placeholders:

During test execution, the measurement results can be placed to these controls using instructions:



* 1. Strings can be saved using instruction .
  2. This function intends to transfer measurement results, “Test Start Time” or/and “Test End Time” to the test reports. The test data is formatted to reflect Pass/Fail by adding “//P or //F” to the end of the test results string:



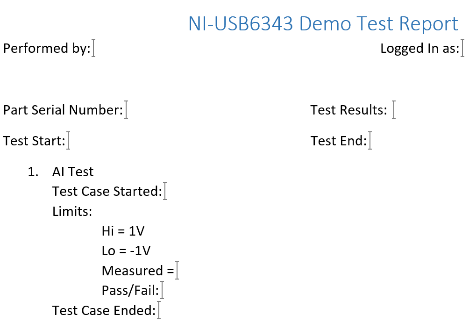
PASS

FAIL

## TEST REPORT GENERATION

The test report is the result of the UUT testing. Most companies have their own unique test report format. With this consideration, the NewTS framework provides two ways to format the custom test reports – “bookmark” based and “tag” based. To select the test report format, go to the “SYSTEM SETTINGS” page and use the “Report Template Type” check box: or . The report templates are created in MS Word.

* 1. The “bookmark-based example is illustrated in the picture below:



10

9

6

5

4

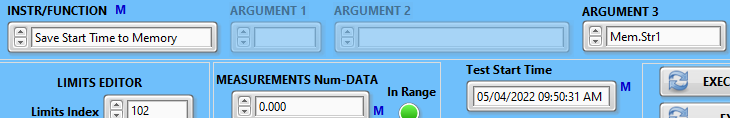
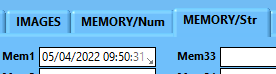
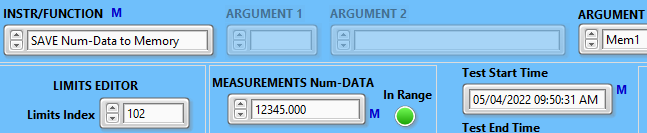
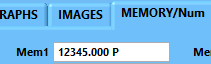
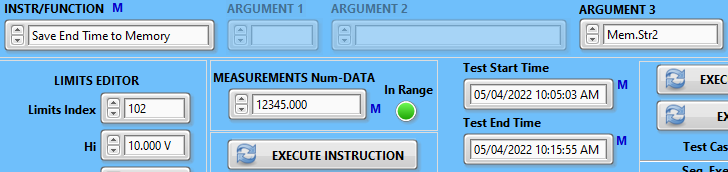
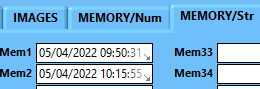
3

1

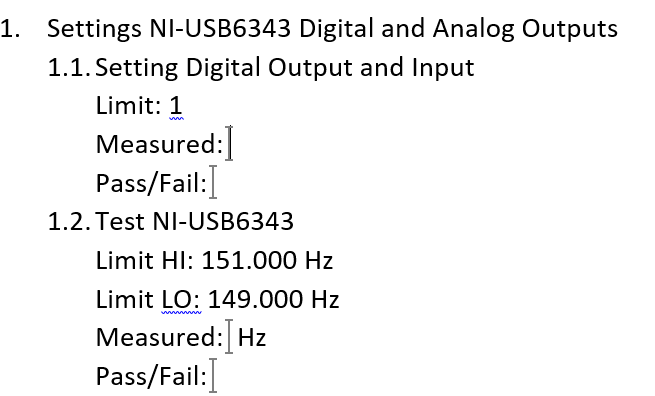
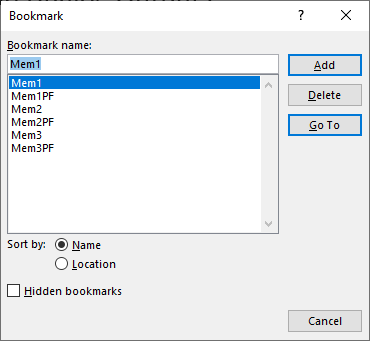
2

7

8

1. “Performed by”: bookmark = *LoginName*
2. “Logged In as”: bookmark = *LoginLevel*
3. “Part Serial Number”: bookmark = *SerialNumber*
4. “Test Start” date/time: bookmark = *TestStartTime*
5. “Test Case Started” date/time: bookmark = *Mem.Str#* (where “#” is the number of the string memory:  ).
6. “Measured” value: bookmark = *Mem#* (where “#” is the numeric memory location  ).
7. The test case “Pass/Fail”: bookmark = *Mem#PF*
8. “Test Case Ended” date/time: bookmark = *Mem.Str#* (where “#” is the number of the string memory:   ).
9. Overall “Test results”: bookmark = *TestResultsPassFail*
10. “Test End” date/time: bookmark = *TestEndTime*

The Report Template File bookmarks where test data will be inserted example:

During test report formatting, the numeric memory data will be divided into two parts – numeric and text:

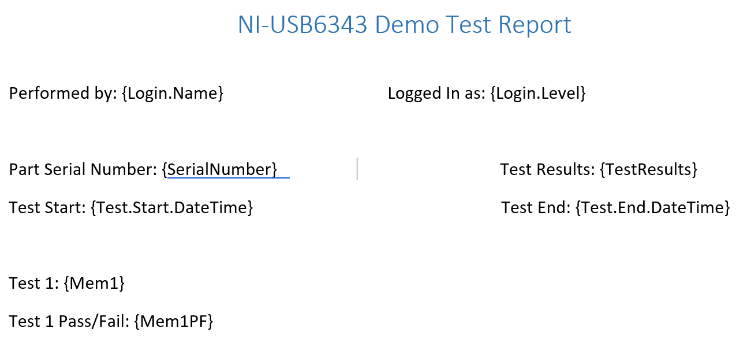


PASS/FAIL

Numeric

The “bookmark” approach has a serious disadvantage – there is no visibility of their value. It makes the report template development difficult and time-consuming. Considering this, the NewTS framework offers the “tag” based test report template.

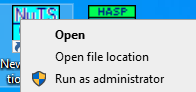
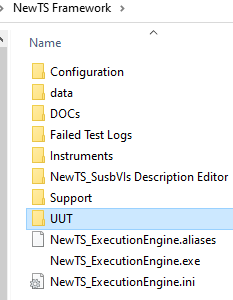
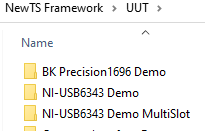
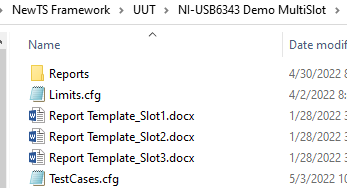
* 1. “Tag” based test report template illustrated on the picture below:



Tags ({Login.Name}, {Login.Level}, {SerialNumber}, {TestResults}, {Test.Start.DateTime}, {Test.End.DateTime}, {Mem1}, and {Mem1PF}) are self-explanatory.

## Test report template development workflow

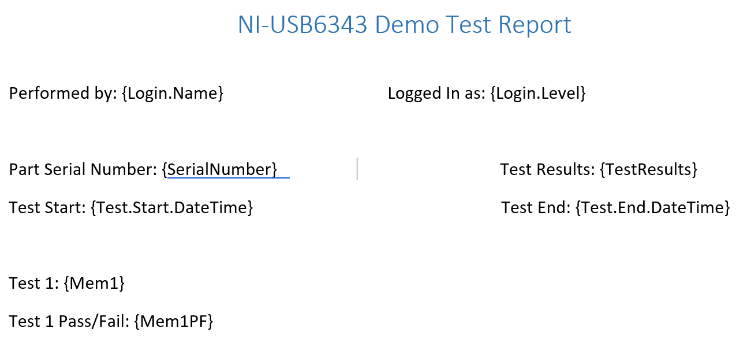
To develop the test report template there is no need to run the test. What is necessary is to have test cases configured. The “MEASUREMENTS Num-DATA”, “MEASUREMENTS Str-DATA”, “MEMORY/Num”, and “MEMORY/Str” objects are controls. This is done to enable the developer to enter values manually specifically for the test report template developments. Below are the steps to develop the “tag” based test report template.

12.1 Go to the NewTS Execution Engine icon , right-click on it and select the “Open File Location” item . In the opened folder, go to the “DOCs” subfolder . In the DOCs subfolder, there is the “Report Template\_Slot1.docx” template file. Copy this file to the   .

Your Test Folder

12.2 Open the “Report Template\_Slot1.docx” file:

Rename the test report title



You can rename text (Performed by:, etc., but DO NOT change any tags ({Login.Name}, etc. – they are “hard-coded”. You may delete them if they are not used.

Rename the test case measured results memory locations.

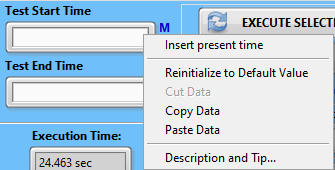
Rename the test case titles.

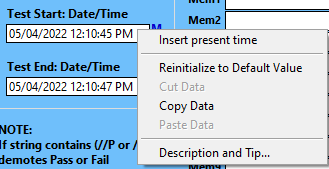
For the numeric values the tag format: *{Mem#}*. For the test case results, the tag format: *{Mem#PF}*. Where “*#*” denotes the numeric memory number.

For the string values the tag format: *{MemStr#}*. For the test case results, the tag format: *{MemStr#PF}*. Where “*#*” denotes the string memory number.

Create test cases as it is required by your project and populate it with corresponding tags.

12.3 Open the NewTS Execution Engine application and log-in as an “Engineer”.

12.4 In the TEST EDITOR page, go to the “Test Start Time” and the “Test End Time”, right cleek on each of them, and select the “Insert Present Time” item .

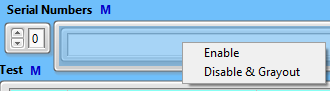
12.5 In the MEMORY/Str page, go to the “Test Start: Date/Time” and the “Test End: Date/Time”, right cleek on each of them and select the “Insert Present Time” item .

12.6 Go to the TEST EDITOR page and for numeric data type, set “INSTR/FUNCTION”. Select the proper numeric memory number, where data will be transferred.

12.7 Enter any numeric values to the “MEASUREMENTS Num-Data” control and set the “In Range” control to the desired state. Press the “EXECUTE INSTRUCTION” button . Repeat this step for all numeric data to be inserted into the test report.

12.8 To use a string data, set “INSTR/FUNCTION”. Select the proper numeric memory number, where data will be transferred.

12.9 Enter any string values to the “MEASUREMENTS Str-Data” control and set the “In Range” control to the desired state. Press the “EXECUTE INSTRUCTION” button . Repeat this step for all string data to be inserted into the test report.

12.10 Go to the “MAIN” page, right-click on the “Serial Numbers” array and select the “Enable” item.

12.11 Enter one serial number (anything).

12.12 Go to the “SYSTEM SETTINGS” page, set the “Report Template Type” to “Tag”  and press the “REPORT TEST” button.

12.13 Go to the “TEST REPORT” page

## 

Opens the report file

with ADOBE Acrobat

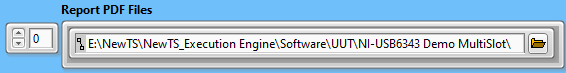
Displays the report file

content

Clears the text

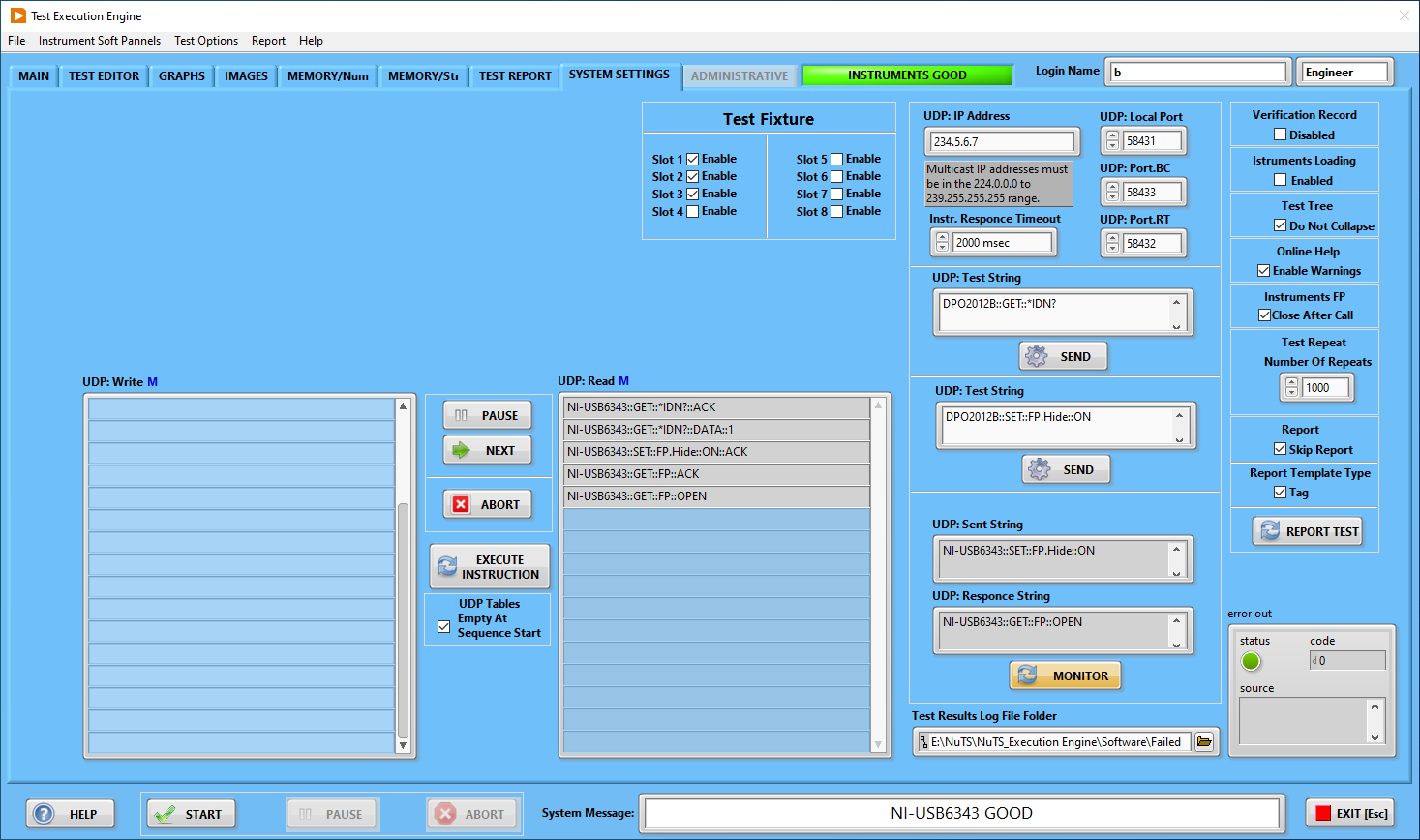
Report file content

List of report files

and set the “Report PDF Files” index to “0”.The path to the created test report will be displayed. Press the “CLEAR” button and, then, the “DISPLAY” button. The report file content will be displayed. If you want to open this file with ADOBE Acrobat Reader, press the “OPEN with ACROBAT” button.

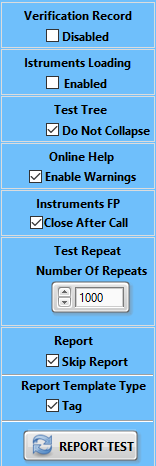
## The “SYSTEM SETTINGS” page

On this page, the engineers or test developers have a set of tools:



* 1. Test execution options:

Enables instruction verification



Generates the report for testing purposes

Defines the report type

Allows skipping the test report generation

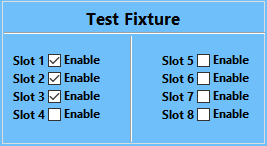
If the test is set to be repeated, this is the number of repeats

Enables soft panel close after execution of the related instruction

Enables expansion/contraction of the test tree

Enables instrument’s soft panels loading

### Test Fixture

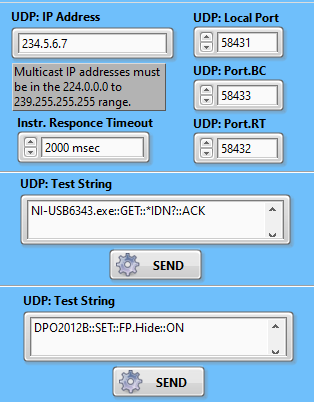


Each test fixture shall have a dedicated test report file.

### UDP setup

The UDP (User Datagram Protocol) is a communications protocol that is primarily used for establishing low-latency and loss-tolerating connections between applications on the internet. It speeds up transmissions by enabling the transfer of data before an agreement is provided by the receiving party.

The UDP is the mean the Executive Engine application communicates to the instrument’s soft panels. Without proper configuration of all applications (the Execution Engine and soft panels), the whole system will not function properly. The UDP configuration is located on the “SYSTEM Settings” page:



### UDP Interface Monitor

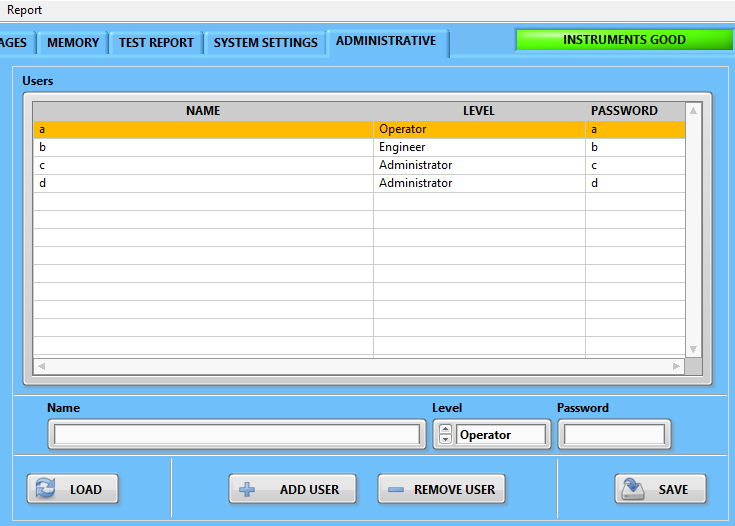
These functions are used to interface to the instrument’s soft panel debugging and verification:



In order to see communication between the Execution Engine and Soft Panels, the “MONITOR” button shall be “ON”.

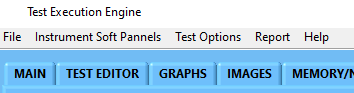
## “ADMINISTRATIVE” page

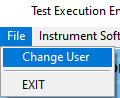
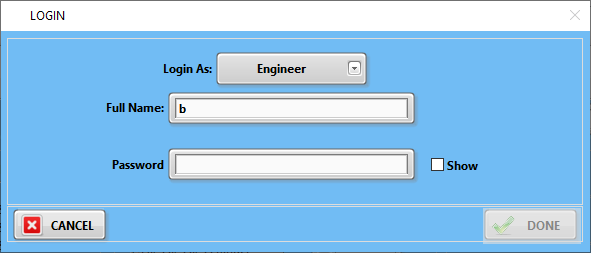
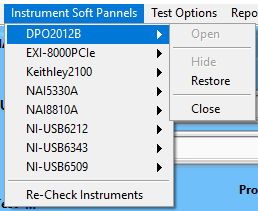
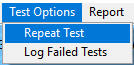
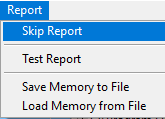
This page will be available upon login as “Administrator”:



The administrator can manage users in the terms of the Execution Engine functionalities access.

## Execution Engine menu

The execution Engine has a menu bar:.

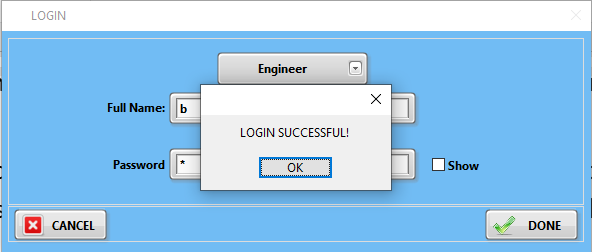
1. “File”:  allows to change user  or exit the Execution Engine application.
2. “Instrument Soft Panel”  allows access to the instrument soft panels.
3. “Test Options”  allows the user to run the test multiple times per set, and to log failed test instructions. This feature is intended for thorough test verification.
4. “Report” contains options related to the test report generation. The user can skip reports, test report generation, save memory to a file, and load memory from the file. These features are intended for the part (UUT) test generation.
5. The “Help” item  allows opening the NewTS Execution Engine User manual with ADOBE ACROBAT.

## Test Case creation workflow

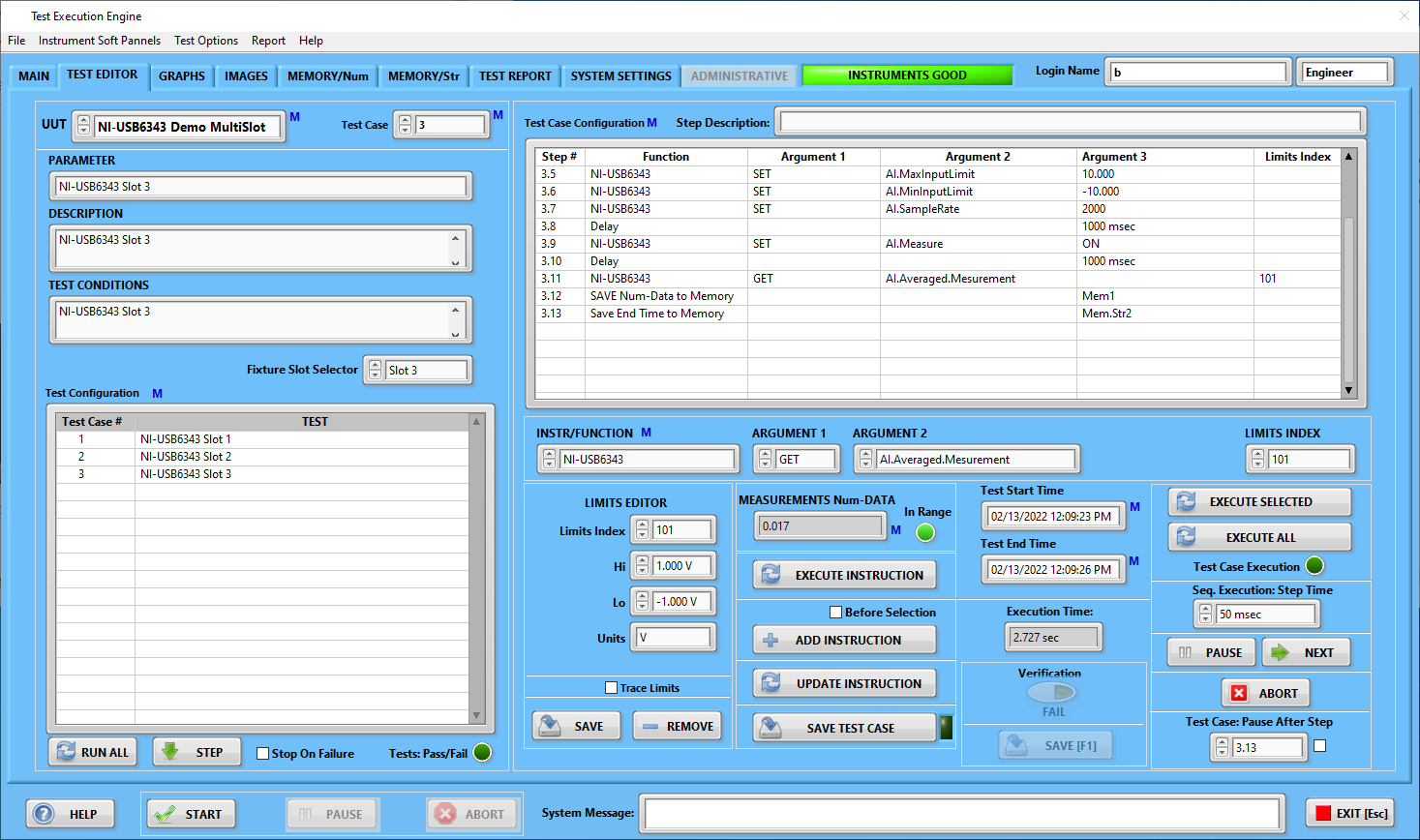
The following steps explain how to create a UUT test using the Execution Engine application.

It is highly recommended, but not required to have physical instruments present in the system.

16.1 Log in to the Execution Engine application on at least the “Engineer” level:



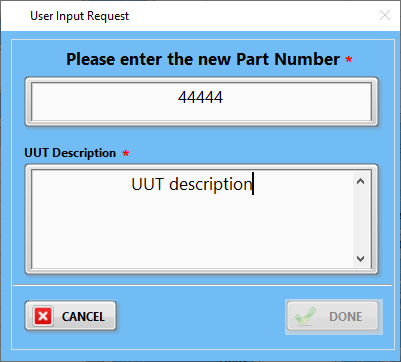
16.2 Go to the “TEST EDITOR” page:



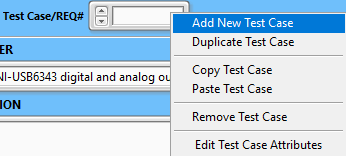
16.3 Right-click on the “UUT” selector:



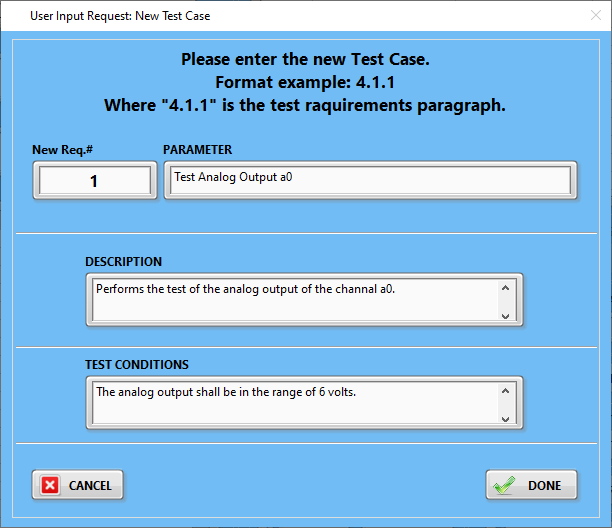
and select “Add New Part Number”. The dialog window will open:

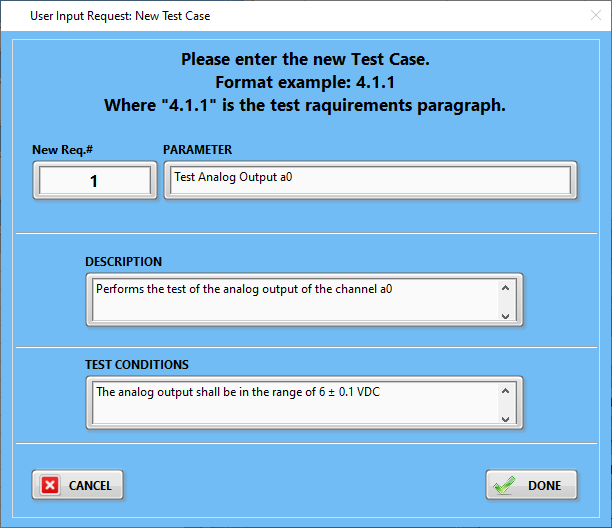


The user MUST enter the new part number and its description. When it will be done, the button “DONE” will be enabled.

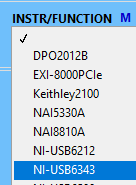
16.4 After the part number (UUT) was created, right-click on the Test Case/REQ# and select “Add New Test Case”: . The user input window will be open:

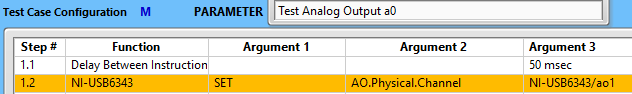


The user must enter the required parameters and click the button “DONE”: 

As the result, a set of indicators will be populated: .

At this moment, the test case does not have any instructions set. Left-click on the test case line #1:

16.5 Adding instructions to the test case: In the “INSTR/FUNCTION” selector, select  NI-USB6343 DAQ device. The “ARGUMENT 1” will be enabled, select “SET” and “ARGUMENT 2” select “AO.Physical.Channel”, select “ARGUMENT 3” NI-USB6343/ao0:

1. Press the “ADD INSTRUCTION” button. The instruction will be added to the test configuration table: 
2. Highlight the step number, select “delay Between Instructions” and add instructions:



1. Perform the following selections, followed by “ADD INSTRUCTION” and “SAVE TEST CASE”:













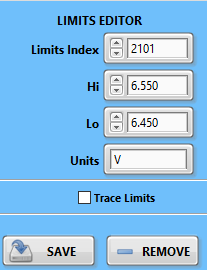








1. Go to “LIMITS EDITOR” and perform the new limits configuration:

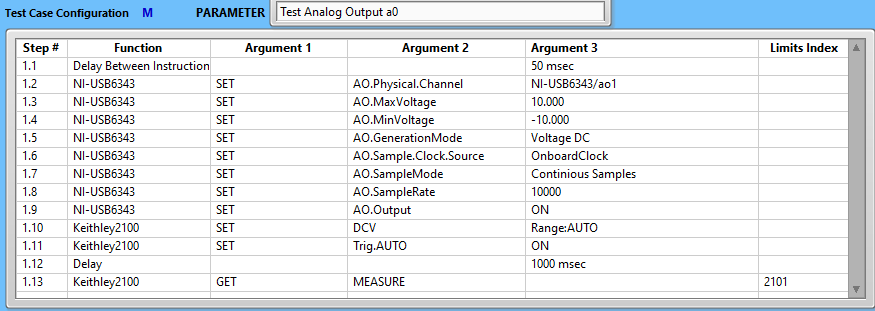


Press the “SAVE” button.

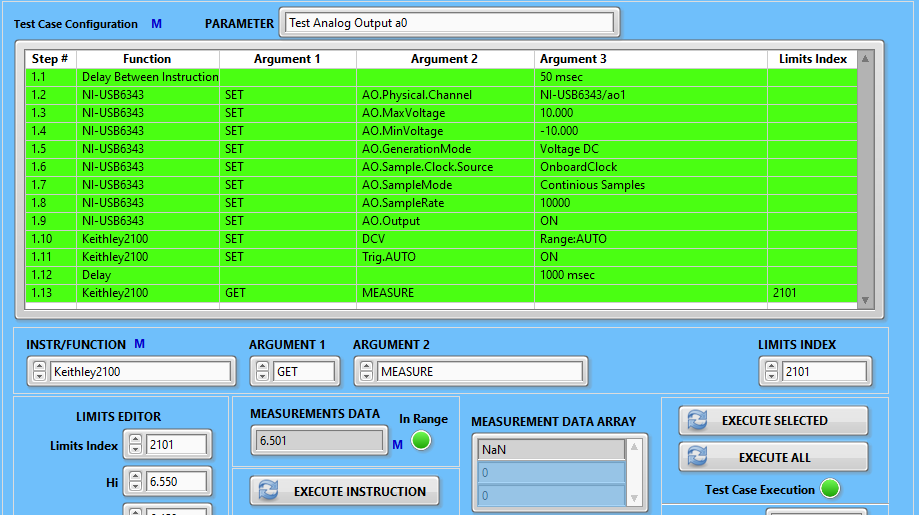
1. Configure DMM measurement, add instruction, and save test case:



1. The “Test Case Configuration” table will look like that:



1. Press “EXECUTE ALL” and observe execution:



THE TEST CASE CREATION IS COMPLETED.