Microphone Test Sequence

1.1 Purpose

Microphone Tests demonstrate Frequency domain measurements of captured Analog (Audio) signal generated by Analog Output (generates sine wave tones) resource with Hardware Trigger using Analog Output and Analog Input Resources. This example sequence can be executed in Python using the Measurement libraries in PCBA.

Example File Location

"\<venv>\Lib\site-packages\nipcbatt\pcbatt_automation\microphone_tests"

1.2 Highlighted Features

- Microphone Test (with Trigger)
  - Supplies analog sine tones through analog output resources and performs Frequency domain measurements on the captured Analog Input waveforms by Analog Input resources. Hardware Triggers are used to reduce the delay between Source and Measure. Libraries used in the example are "***SignalVoltageGeneration()***" and "***FrequencyDomainMeasurement()***"
- Turn Off all AO Channels
  - Powers down all analog output channels by configuring the output voltage as 0 Volts. Libraries used in the example is "***DcVoltageGeneration()***".

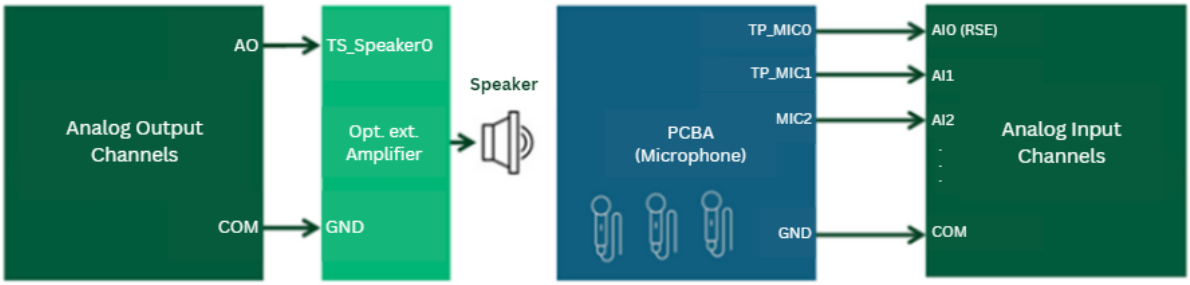
Refer this folder for more details on each Measurement library "\<venv>\Lib\site-packages\nipcbatt\pcbatt_library".

1.3 Prerequisites

- Python – 3.9 to 3.12
- DAQmx Driver – 2023 Q3 or later

1.4 Setup Diagram

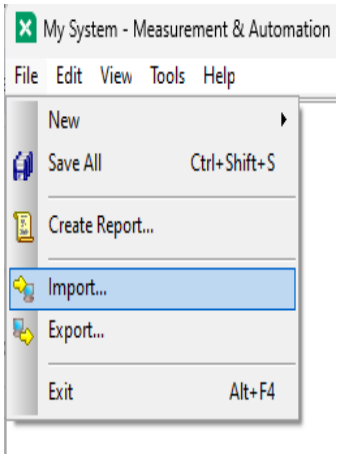
Represents the hardware setup used in this example sequence. Pin Outs of each resource is added below.



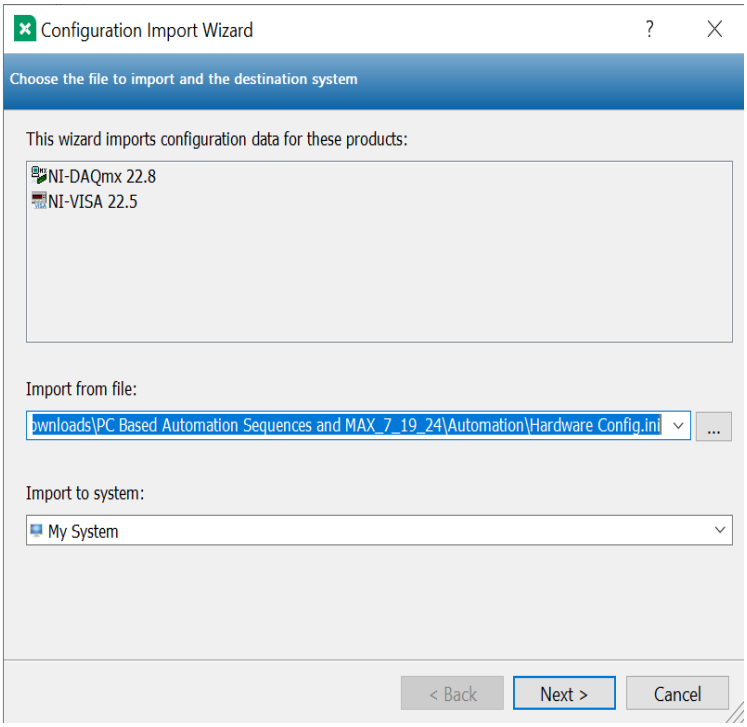
1.5 How to run this Example?

Complete the following steps to run the sequence.

1. First we must configure NI-MAX to reflect the virtual channels which will be used by the Python script names mentioned in ***microphone_test.***
   a. A hardware configuration file for NI-MAX is required to run this example. The configuration file contains a set of predefined global channel names which are used by the nidaqmx driver to communicate with the Python scripts.
   b. To import the "Hardware Config" open NI-MAX.
   c. Click on File -> Import to open the Configuration Import Wizard



   d. In the Configuration Import Wizard window, click on the Browse (...) button and locate the *Hardware Config.ini* file in *"\<venv>\Lib\site-packages\nipcbatt\pcbatt_automation".* Then click on *Next -> Import -> Finish*



e. NI-MAX now holds the same virtual channel name references contained in the examples provided

The ***microphone_tests.py*** *f*ile will create log files in the form of a simple text (.txt) file. The default file path it will use is

"C:\\Windows\\Temp\\Signal_Generation_and_Frequency_Domain_Analysis.txt"

If you wish to create this file in a different location on your PC, change the value of the string variable *DEFAULT_FILEPATH*.

2. Open the Python scripts ***microphone_main_sequence.py*** along with ***microphone_tests.py*** in the IDE or text editor of choice. The following steps are performed within ***microphone_main_sequence.py***.
    a. **Microphone Tests (with Trigger)** - demonstrates Frequency domain measurements of captured Analog (Audio) signal generated by Analog Output (generates sine wave tones) resource with Hardware Trigger. Below are the steps included in the test.
        i. Initialize Signal Voltage Generation and Frequency Domain Measurement libraries by creating the instances of ***SignalVoltageGeneration()*** and ***FrequencyDomainMeasurement()*** classes and then using ***initialize()*** method on each object.
        ii. Configure Frequency Domain Measurement to wait for Start Trigger from Signal Voltage Generation.
        iii. Configure the Signal Voltage Generation to start sourcing Sine Tones (in the backend, Signal Voltage Generation resource sends the Trigger through the backplane during sourcing starts which in-turns starts the measurement in Analog Input resource). In this example, a Single tone sine waveform with frequency of **1KHz and amplitude of 1 Volt is generated for a duration of 0.1 seconds**.
        iv. Measure the sourced Sine Voltage Waveform with ***FrequencyDomainMeasurement()*** class instance by calling the ***configure_and_measure()*** method.
        v. Use the ***close()*** methods on both instances to stop generation and measurement.

    Refer the help/comments in the sequence for more details to know more about trigger configuration.

    b. ***Turn Off all AO Channels*** – Power downs all Analog output channels by configuring them to 0 volts. Below are the steps included in the test.
        i. Initialize the DC Voltage Generation library by creating an instance of ***DcVoltageGeneration()*** class and then using ***Initialize()*** method.
        ii. Configure the DC Voltage Generation to source 0 Volts in specified Analog Output channels by calling the ***configure_and_generate()*** method using the default parameters.
        iii. Use ***close()*** instance after setting AO channels to 0 Volts.
3. When the execution completes, **review the results** on the ***.txt*** files generated by the logger at the specified location.
    a. The report has the configurations and Measurement values captured (runs with simulated instrument by default)
    b. Verify the Measurement and data formats returned by the Measurement library.

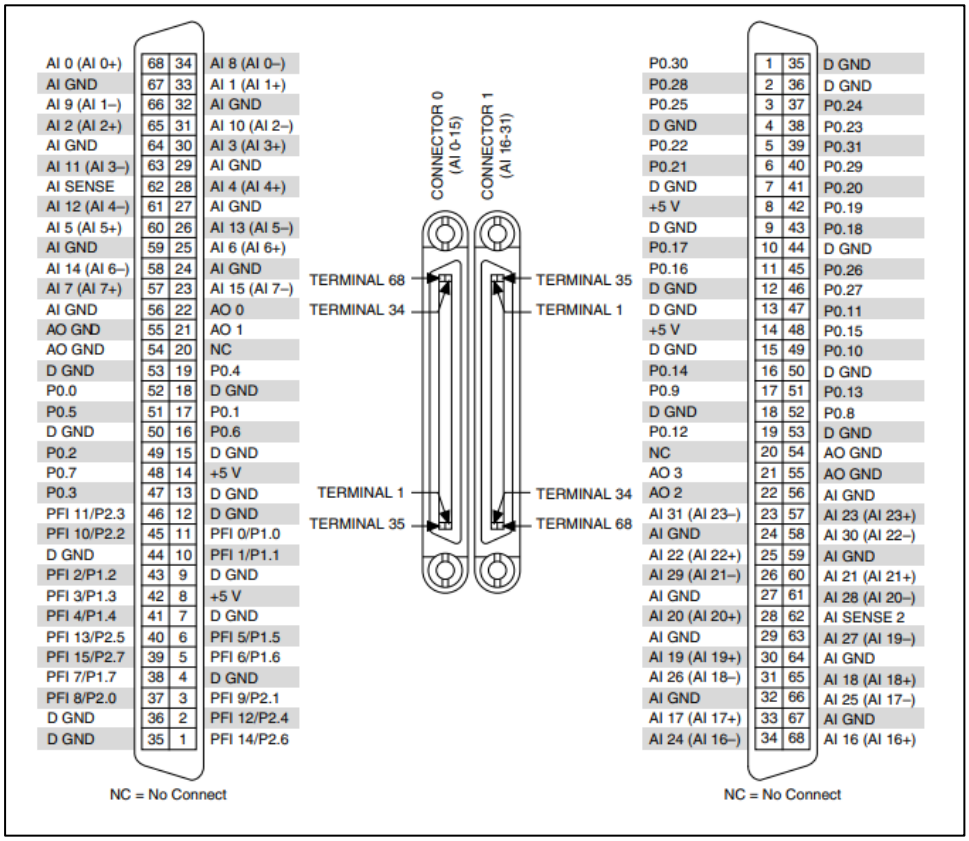
1.6 How to enable the Hardware?

Microphone Test sequence runs with simulated hardware by default. Once the hardware setup is available, you can do the below changes to enable running the test with the hardware.

Note : In this example, physical and global virtual channels are used to configure the terminal or pin to perform the instrument actions. Global Virtual Channels are software entities that encapsulate the physical channel along with other channel specific information—range, terminal configuration, and custom scaling. Global Channels can be created in NI-MAX and called in Measurement Libraries.

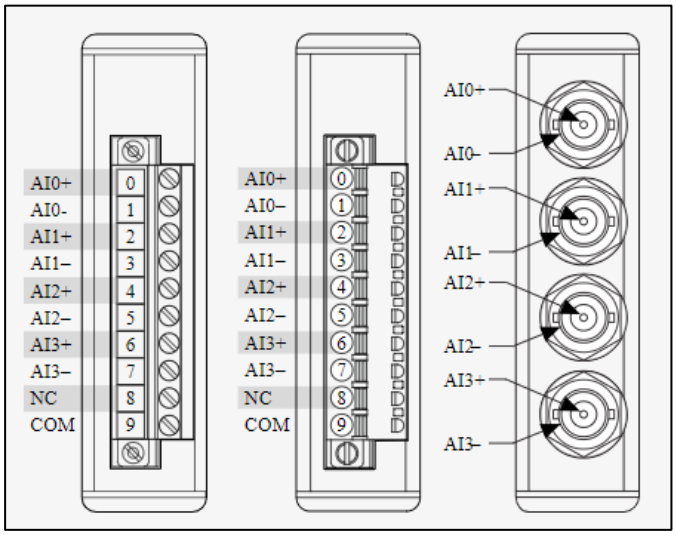
1. Skip the first step "Import Hardware Config" in **section 1.5.** This step imports the Simulated Hardware and creates Global Virtual Channels with Simulated instrument in NI-MAX.
2. Follow the below steps for each sequence. Refer "**Note to run with Hardware**" labels in the sequence.
    i. **Microphone Tests**
        1. Step into the "**microphone_tests.py**" file.
        2. Open NI-MAX and update the physical Channel linked to the Global Channels – **TS_Speaker0, TP_MIC0, TP_MIC1, TP_MIC2** (called in the initialize step of Signal Voltage Generation and Frequency Domain Measurement).
        3. Update the global channel names in *"FREQ_DOMAIN_MEAS_CHANNEL"* and "SIGNAL_VOLTAGE_GEN_CHANNEL".
        4. Update the "Digital Trigger Source" based on NI-MAX Hardware in the "Frequency Domain Measurement".
        5. Review the Configurations of Analog Output and Analog Input Pins for the intended use case.
    ii. ***Turn Off all AO Channels***
        1. Step into the "***turn_off_all_ao_channels.py***" example.
        2. Update the physical channels input with Analog Output Channel in the initialize step of ***Turn_off_all_ao_channels.***
        3. Review the Configurations of Analog Output for the intended use case

1.7 Pinouts of PCIe-6323

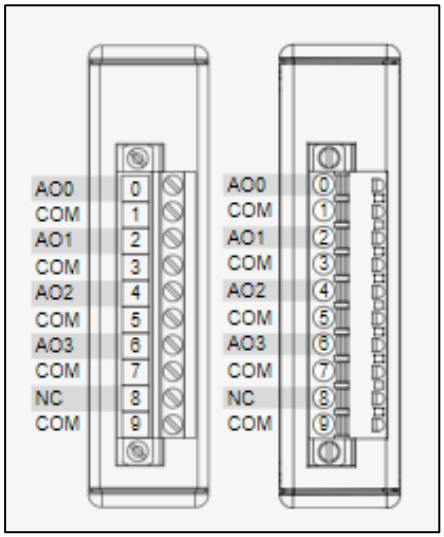


1.8 Pinouts of cDAQ Modules

1. **Analog Input Module (NI-9215)**

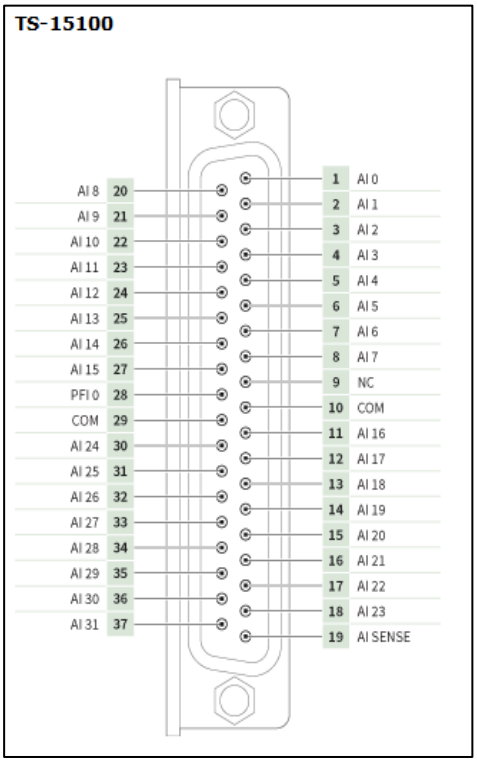


2. **Analog Output Module (NI-9263)**

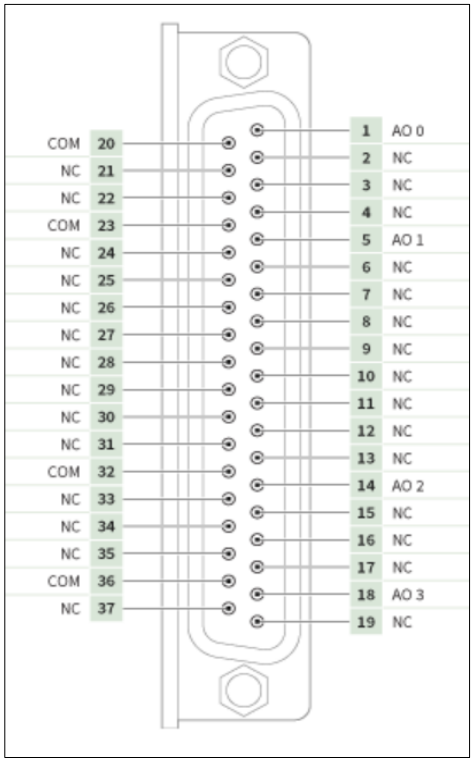


1.9 Pinouts of TestScale Modules

1. **Analog Input Module (TS-15100)**



2. **Analog Output Module (TS-15110)**



1.10 How to create/Modify Global Virtual Channels?

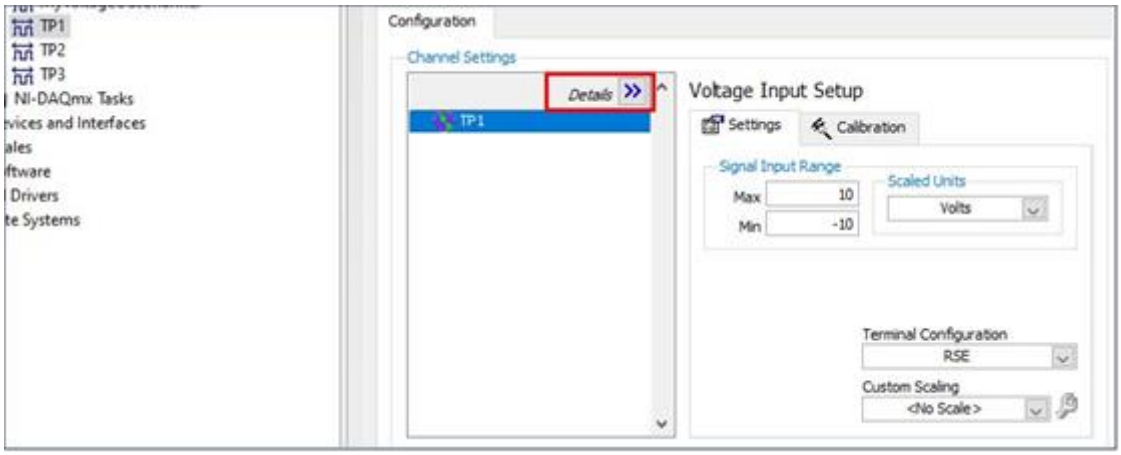
A virtual channel is a collection of settings such as a name, a physical channel, input terminal connections, the type of measurement or generation, and can include scaling information. A virtual channel created outside a task is a Global Virtual Channel.

Follow the below steps to **create Global Virtual Channel** in NI-MAX.

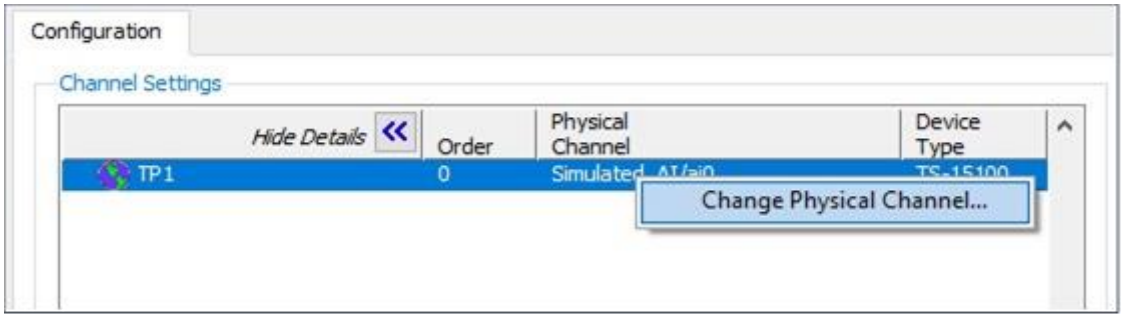
1. Launch NI-MAX
2. In NI-MAX, right-click **Data Neighbourhood** and select **Create New**
3. In the Create New window, select **NI-DAQmx Global Virtual Channel** and click **Next**. The DAQ Assistant opens.
4. Select an I/O type, such as analog input
5. Select the physical channel of Hardware 6
6. Type the global virtual channel name. Click **Finish**
7. Save your configuration.

Follow the below steps to **modify the existing Global Virtual Channel** in NI-MAX.

1. Launch NI-MAX
2. In NI-MAX, expand **Data Neighbourhood** > **NI-DAQmx Global Virtual Channel**
3. Select the Global Channel to modify. Configuration window opens.



4. Click on “Details >>” as highlighted above to view the Physical Channel
5. Right click and **Change Physical Channel** to update the Physical Channel. Select the Physical Channel from Hardware as per the connection and Click “Ok”



6. **Save** your configuration