



PYTHON PCB ASSEMBLY TEST TOOLKIT

Getting Started

Legal Information

Copyright

© 2004–2026 National Instruments Corporation. All rights reserved.

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

NI respects the intellectual property of others, and we ask our users to do the same. NI software is protected by copyright and other intellectual property laws. Where NI software may be used to reproduce software or other materials belonging to others, you may use NI software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

Table of Contents

1	About This Manual.....	4
2	Overview.....	4
3	Application Software Support.....	4
4	Supported Hardware.....	4
5	Installation Instructions.....	4
5.1	Python Installation	5
	• Installing Python	5
	• Creating a venv	5
	• Install pip package.....	5
	• Start testing.....	5
	• Download Source code.....	6
6	Getting Started with Python PCB Assembly Test Toolkit	6
6.1	Launch Python PCB Assembly Test Toolkit	6
7	Software Components.....	7
7.1	Measurement Libraries	7
7.2	Automation Test Sequences	8
	• Execution with Simulated Hardware.....	9
7.3	Functional Test Demo Sequences.....	9
8	Related Documents.....	10
9	References	11

1 About This Manual

This manual contains information about the Python PCB Assembly Test Toolkit, getting started steps including installation and setup procedures, and steps to run example test sequences built with PCBA measurement library in simulation mode.

2 Overview

Python PCB Assembly Test Toolkit is a collection of Measurement Library, Automation Examples, Test Demo Example along with Documentation for PCB Assembly electrical functional test.

Python PCB Assembly Test Toolkit provides support for DAQmx, niDMM, niSWITCH, and compatible with PXI, PC Based DAQ, cDAQ, TestScale and high level enough to be applicable or scalable for other instruments with similar functionality later. Libraries are built in Python, designed, and structured to use readily with sequencers. Example test sequences built with the libraries are added to demonstrate basic Functional test in Python for your custom sequencers.

3 Application Software Support

Python PCB Assembly Test Toolkit requires the following dependencies.

Dependency	Version Required
NI-DAQmx	2023 Q3 and above
LabVIEW Runtime	2022 Q3 and above (64 bit)
NI-845x	2022 Q3 and above
NI-VISA	2023 Q2 and above
NI-Serial	2023 Q2 and above
NI-SWITCH	2023 Q4 and above
NI-DMM	2023 Q4 and above

Note: This table shows the recommended versions of the NI drivers required by the Python package.

All these dependencies will be installed automatically by the installer for Python PCB Assembly Test Toolkit.

4 Supported Hardware

Python PCB Assembly Test Toolkit supports all devices supported by DAQmx, NI-DMM, NI-SWITCH, I2C/SPI, Serial Drivers with a focus on:

- DMM devices compatible with the NI-DMM driver
- Switch devices compatible with the NI-SWITCH driver
- Any DAQmx devices with similar functionality and resources
- NI PC-Based DAQ
- CompactDAQ
- TestScale

5 Installation Instructions

Visit ni.com/downloads or visit [NI Package Manager](#) to install the recommended version of drivers/dependencies for Python PCB Assembly Test Toolkit, then continue and follow the steps in

[Python installation](#). If the test machine already has specific versions of the dependencies installed, you can directly follow the steps in [Python installation](#) to install python package.

5.1 Python Installation

After all the dependencies have been installed with the required or greater versions, follow the steps mentioned below to install the PyPI based nipcbatt package.

- **Installing Python**

Download and install Python 3.9 or above from [Download Python | Python.org](#).

- **Creating a venv**

Note: Skip this step to install the toolkit directly into the base Python installed in the system but use of venv is recommended.

Open terminal anywhere in your system and follow the steps mentioned below to create a Python virtual environment:

a) Create a directory for venv:

```
"mkdir .\venvs\test_env"
```

b) Use the following command to create a venv with installed Python version:

```
"<Python location>\python.exe" -m venv .\venvs\test_env"
```

One of the example commands could be:

```
"C:\Program Files\Python39\python.exe" -m venv .\venvs\test_env"
```

c) Activate the newly created venv:

```
".\venvs\test_env\Scripts\activate"
```

Follow this link to get to know more about creating python venvs: [Creation of virtual environments — Python documentation](#)

- **Install pip package**

Use the following command to install the nipcbatt package:

```
"python -m pip install nipcbatt"
```

- **Start testing**

After installation, refer to the following location to get to know more about the working of the nipcbatt package through the example automation sequences. More information on individual automation sequences can be found with Readme files associated with them.

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

→ Example location:

```
"C:\Test Workspace\venvs\test_env\Lib\site-packages\nipcbatt\pcbatt_automation"
```

- **Download Source code**

Source code for the package can be downloaded from GitHub: <https://github.com/ni/nipcbatt> website.

Extract the downloaded zip file “nipcbatt-main” and refer to this path from the extracted files to go to the automation sequences:

“\nipcbatt-main\src\nipcbatt\pcbatt_automation”

Source code for the package can also be downloaded from the pypi.org/project/nipcbatt/#files website.

Extract the downloaded “nipcbatt-1.x.tar.gz” and refer to this path from the extracted files to go to the automation sequences:

“\nipcbatt-1.x\src\nipcbatt\pcbatt_automation”

When everything is ready, to check the installation and dependencies, follow instructions on [Execution with Simulated Hardware](#) Section to run all test sequences. They should pass without any errors.

6 Getting Started with Python PCB Assembly Test Toolkit

6.1 Launch Python PCB Assembly Test Toolkit

Start testing with Python PCB Assembly Test Toolkit with the steps below:

1. To open the source folder which includes Library and tests, refer to:

“\<venv>\Lib\site-packages\nipcbatt\pcbatt_library”

If have downloaded source code from GitHub, refer to this path:

“\nipcbatt-main \src\nipcbatt\pcbatt_library”

If have downloaded source code from PyPI website, refer to this path:

“\nipcbatt-2.x\src\nipcbatt\pcbatt_library”

Refer the [software components of PCBA library](#) for more information on individual libraries.

2. Go to the following location to copy the individual and pair examples into your workspace:

“\<venv>\Lib\site-packages\nipcbatt\pcbatt_validation_examples”

OR

“\nipcbatt-main\src\nipcbatt\pcbatt_validation_examples”

OR

“\nipcbatt-2.x\src\nipcbatt\pcbatt_validation_examples”

3. Go to the following location to refer to the example automation sequences:

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

OR

`"\nipcbatt-main\src\nipcbatt\pcbatt_automation"`

OR

`"\nipcbatt-2.x\src\nipcbatt\pcbatt_automation"`

Refer to the [Automation Test Sequences](#) section to obtain more information on running the automation sequences.

7 Software Components

7.1 Measurement Libraries

Measurements are added based on the common actions performed in PCBA electrical functional test. Python PCBA Measurement Libraries are present in the folder:

`"\<venv>\Lib\site-packages\nipcbatt\pcbatt_library"`

This folder contains all the measurement libraries and common reusables. Below are the measurement directories present,

- common
- communications
- daq
- dmm
- dmm_scan
- switch

Inside these directories you will find individual measurement libraries:

- communications
 - common
 - i2c_communications
 - spi_communications
 - serial_communications
- daq
 - common
 - power_supply_source_and_measurements
 - dc_voltage_generations
 - dc_rms_voltage_measurements
 - dc_rms_current_measurements
 - signal_voltage_generations
 - time_domain_measurements
 - frequency_domain_measurements
 - static_digital_state_generations
 - static_digital_state_measurements
 - dynamic_digital_pattern_measurements
 - dynamic_digital_pattern_generations

- digital_clock_generations
- digital_pulse_generations
- digital_frequency_measurements
- digital_pwm_measurements
- digital_edge_count_measurements
- synchronizations
- temperature_measurements
- dmm
 - common
 - dc_rms_current_measurements
 - dc_rms_voltage_measurements
 - mixed_measurements
 - resistance_measurements
- dmm_scan
 - dmm_scan_pmps_16V_15C
- switch
 - static_digital_path_generations

**NOTE:**

Detailed Help on how to use the above measurement libraries can be found in “\nipcbatt-main\src\nipcbatt\docs\Python PCB Assembly Test Toolkit - User Manual.

7.2 Automation Test Sequences

Test Sequences built in Python using PCBA measurement library demonstrate basic electrical functional test which can be reused or modified based on the procedure. Test Sequences are placed in the location:

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

OR

`"\nipcbatt-main\src\nipcbatt\pcbatt_automation"`

OR

`"\nipcbatt-2.x\src\nipcbatt\pcbatt_automation"`

Open the python sequence files in any IDE of your choice to view all the tests and Libraries. Below are the example tests added,

- action_button_tests
- audio_tests
- digital_io_tests
- led_tests
- microphone_tests
- power_supply_tests

- communication_tests
- synchronization_tests
- sensor_tests



NOTE

Detailed Help on how to use the above test sequences can be found in **ReadMe.pdf** placed parallel to the test sequence.

• Execution with Simulated Hardware

Follow below steps to run the automation sequences using simulated hardware:

- Import NI MAX configuration file to get the simulated hardware. The configuration file is located at:

```
"\<venv>\Lib\site-packages\nipcbatt\pcbatt_automation\Hardware Config.ini"
```

OR

```
"\nipcbatt-main\src\nipcbatt\pcbatt_automation\Hardware Config.ini"
```

OR

```
"\nipcbatt-2.x\src\nipcbatt\pcbatt_automation\Hardware Config.ini"
```

- Open the validation examples/sequences in any IDE of your choice.
- Run the main sequence file present in folder of any sequence with a VENV having nipcbatt library installed.
- Results can be obtained from the .txt files created by the logger. Edit the file path in the sequences to save the results in any location of your choice.



NOTE

To run with hardware, Refer **ReadMe.pdf** placed parallel to the test sequence.

7.3 Functional Test Demo Sequences

Python PCB Assembly Test Toolkit also comes with demo test sequence which covers basic functional test for PCBA. This demonstrates how to use Python PCB Assembly Test Toolkit, how to extract test data and determine test result as pass/fail.

Demo Sequences can be located at

```
"\<venv>\Lib\site-packages\nipcbatt\pcbatt_ft_demo_test_sequence"
```

It consists of the following Sequences:

- Power_Diagnostics

- Reset and Self-Test
- Animation_and_Sound_User_Input_Test
- Audio_Filter_Test
- Turn Off all AO Channels
- Power Down Supply

8 Related Documents

- **User Manual** for Python PCB Assembly Test Toolkit is available at “\nipcbatt-main\src\docs\Python PCB Assembly Test Toolkit - User Manual” or “\nipcbatt-2.x\src\docs\Python PCB Assembly Test Toolkit - User Manual”.
- Detailed Help on how to use the example test sequences can be found in **ReadMe.pdf** placed inside respective test folders.
- Detailed Help on how to use the demo test sequence can be found in **ReadMe.pdf** that comes along with the demo sequence.

9 References

- TestScale Help - [What is TestScale? - NI](#)
- NI DMM Overview - [NI-DMM Overview - NI](#)
- PXI-4065 Specifications - [PXI-4065 Specifications - NI](#)
- PXI-2527 Overview - [PXI-2527 Overview - NI](#)
- PXI-2527 Specifications - [PXI-2527 Specifications - NI](#)
- PXI-2568 Overview - [PXI-2568 Overview - NI](#)
- PXI-2568 Specifications - [PXI-2568 Specifications - NI](#)
- TestScale Module TS-15050 Overview - [TS-15050 Overview - NI](#)
- TestScale Module TS-15050 Specifications - [TS-15050 Specifications - NI](#)
- TestScale Module TS-15100 Overview - [TS-15100 Overview - NI](#)
- TestScale Module TS-15100 Specifications - [TS-15100 Specifications - NI](#)
- TestScale Module TS-15110 Overview - [TS-15110 Overview - NI](#)
- TestScale Module TS-15110 Specifications - [TS-15110 Specifications - NI](#)
- TestScale Module TS-15120 Overview - [TS-15120 Overview - NI](#)
- TestScale Module TS-15120 Specifications - [TS-15120 Specifications - NI](#)
- TestScale Module TS-15130 Overview - [TS-15130 Overview - NI](#)
- TestScale Module TS-15130 Specifications - [TS-15130 Specifications - NI](#)
- TestScale Module TS-15200 Overview - [TS-15200 Overview - NI](#)
- TestScale Module TS-15200 Specifications - [TS-15200 Specifications - NI](#)
- DAQmx Help - [NI-DAQmx Help - NI](#)