Communication Test Sequence

1.1 Purpose

Communication Tests demonstrate data read and write operations through various communication protocols like I2C, SPI and Serial Port Interactions by using NI Hardware's like USB-845x and USB-232. This example sequence can be executed in a custom Python sequence script using the measurement libraries written in Python.

Example File Location

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

1.2 Highlighted Features

- I2C Comm Test
  - Demonstrates simple read and write data operations through I2C protocol communication using NI 845x Device. Library used in this example is "***I2cReadCommunication()"***.
- SPI Comm Test
  - Demonstrates simple read and write data operations through SPI protocol communication using NI 845x Device. Library used in this example is "***SpiReadCommunication()"***.
- Serial Comm Test
  - Demonstrates simple read and write data operations through serial port like RS232 through USB connectors like USB-232. Library used in this example is "***SerialCommunication()"***.

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

1.3 Prerequisites

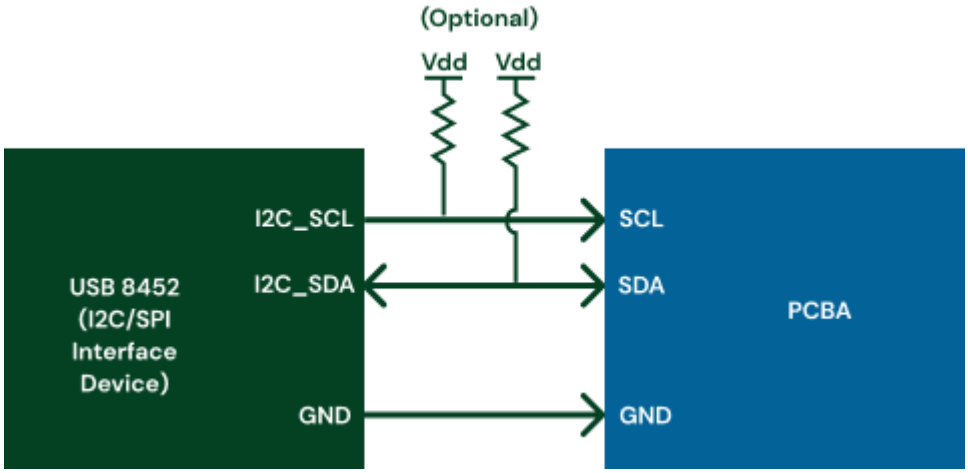
- Python – 3.9 to 3.12
- DAQmx Driver – 2023 Q3 or later
- NI 845x Driver – 2022 or later
- NI Serial Driver – 2023 or later
- NI VISA Driver 2023 or later

© National Instruments Corporation

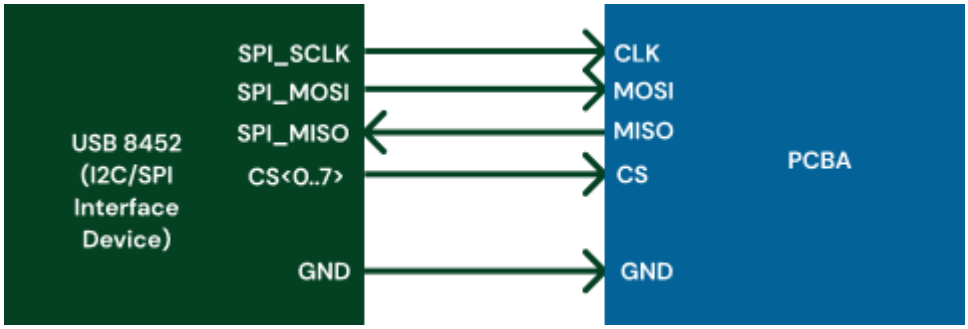
1.4 Setup Diagram

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

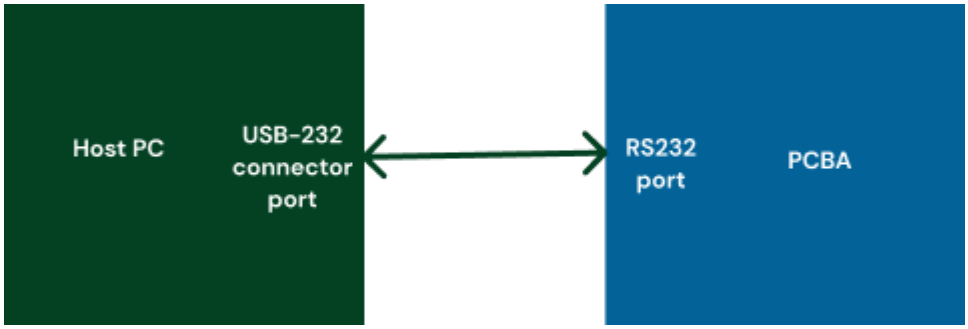
a. I2C Communication Test



b. SPI Communication Test



c. Serial Communication Test



© National Instruments Corporation

1.5 How to run with Hardware?

Communication Test Sequence **does not support simulated mode Run**. Once the hardware setup is available, you can do the below changes to enable running the test with the hardware.

1. Follow the below steps,
    a. Open the "**communication_test_main_sequence.py**" along with "***i2c_comm_test.py***", "***serial_comm_test.py***" and "***spi_comm_test.py***" in your IDE or text editor of your choice.

2. Follow the below steps for each sequence. Refer "**Note to run with Hardware**" labels in the sequence.
    a. **I2C Comm Test**
        i. Step into the "***i2c_comm_test.py***" sequence.
        ii. Initialize the i2c communication library by creating the instances of ***I2cReadCommunication()*** and ***I2cWriteCommunication()***" using ***initialize()*** method on the objects.

        Note: It is recommended to copy this step along with the variables during custom sequence creation.

        iii. Configure the setting to read data by calling the ***configure_and_read_data(self, configuration: I2cReadCommunicationConfiguration)*** method with ***device_parameters, communication_parameters*** and ***read_parameters.***
        iv. Configure settings to write data by calling the ***configure_and_write_data(self, configuration: I2cWriteCommunicationConfiguration)*** method with ***device_parameters, communication_parameters*** and ***write_parameters.***
        v. Review the Configurations of 845x I2C Pins for the intended use case.
        vi. Use the ***close()*** methods on both instances to close all tasks and release resources allocation.
    b. **SPI Comm Test**
        i. Step into the "***spi_comm_test.py***" sequence.
        ii. Initialize the i2c communication library by creating the instances of ***SpiReadCommunication()*** and ***SpiWriteCommunication()*** using ***initialize()*** method on the objects.

        Note: It is recommended to copy this step along with the variables during custom sequence creation

        iii. Configure settings by calling the ***configure_and_read_data(self, configuration: SpiReadCommunicationConfiguration,)*** method with ***device_parameters, communication_parameters*** and ***read_parameters.***
        iv. Configure settings by calling the ***configure_and_write_data(self, configuration: SpiWriteCommunicationConfiguration)*** method with ***device_parameters, communication_parameters*** and ***write_parameters.***
        v. Review the Configurations of 845x I2C Pins for the intended use case.
        vi. Use the ***close()*** methods on both instances to close all tasks and release resources allocation.

© National Instruments Corporation

c. **Serial Comm Test**
    i. Step into the "***serial_comm_test.py***" sequence.
    ii. Initialize the Serial communication library by creating the instance of ***SerialCommunication()*** using ***initialize()*** method on the object.

    Note: It is recommended to copy this step along with the variables during custom sequence creation

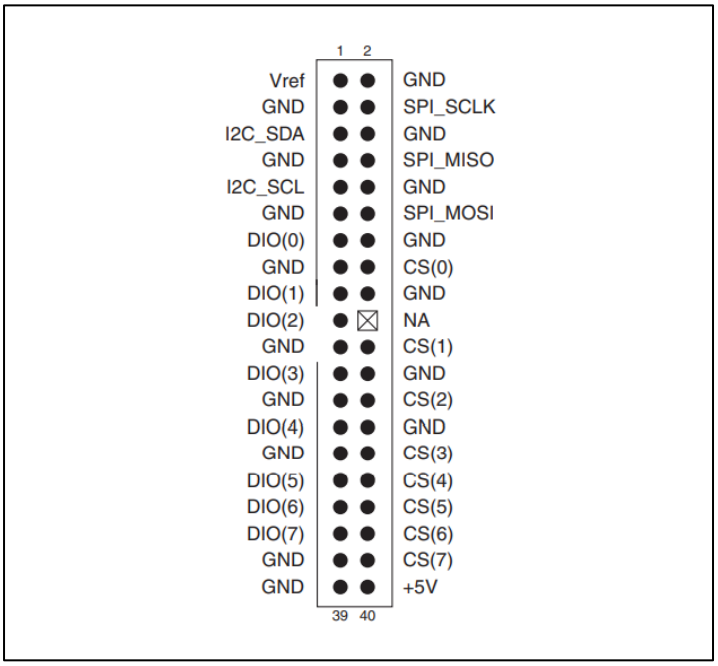
    iii. Review and update step settings for "***config***" step based on the Serial Port Specifications. These settings will be used to perform read/write operation through serial COM port.
    iv. To setup communication with serial instruments refer the NI Document here - https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000x1jtCAA&l=en-IN
    v. Review the connections between the serial com ports based on the intended use case.
    vi. Use the ***close()*** methods on both instances to close all tasks and release resources allocation.

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

© National Instruments Corporation

1.6 Pinouts of Devices

1. **I2C/SPI Interface Device (USB-8452)**



For more details refer – NI-845x Hardware and Driver Software Getting Started Guide.

1.7 References

1. NI 845x Software and Hardware Installing Procedure - NI-845x Software and Hardware Installatrion Guide - National Instruments
2. NI 845x Example location - https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000wyG5CAI&l=en-IN
3. NI USB-232 Serial Getting Started Guide - NI cRIO-9035 Getting Started Guide - National Instruments
4. Set Up Communication with Serial Interface - Set Up Communication with Serial Instruments in LabVIEW using NI-VISA - NI

© National Instruments Corporation