Multiple board DDR Stress Tool Documentation

Description:

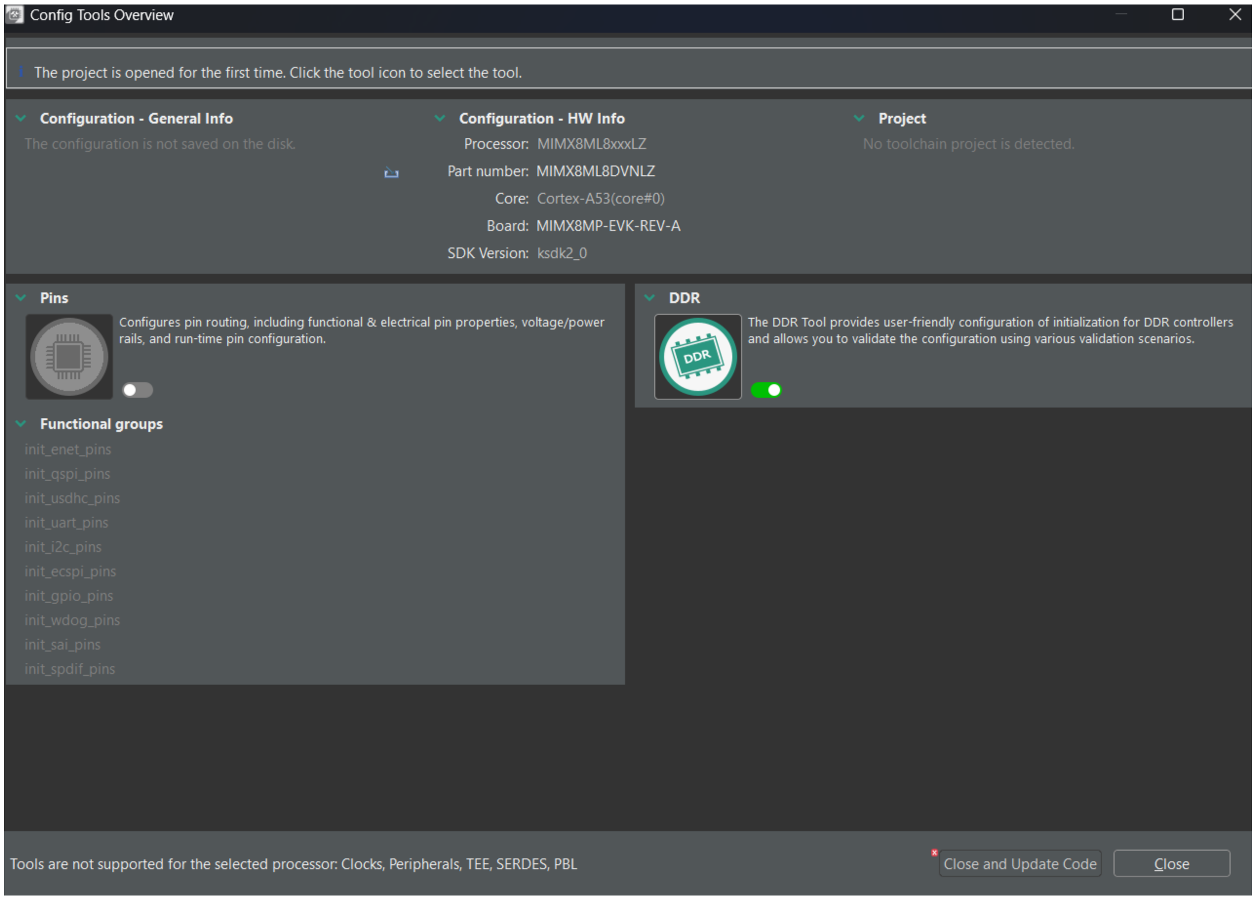
This documentation provides setup and running steps for running the i.MX Config Tools' DDR tests on multiple boards simultaneously using various DDR configurations and across multiple iterations.

Software Setup:

- Windows 10/11
- i.MX Config Tools 16.1

Setup steps:

1. Open i.MX Config Tools and start a new project for the required board. Select the DDR Tool in the following screen:



Config Tools Overview

The project is opened for the first time. Click the tool icon to select the tool.

Configuration - General Info

The configuration is not saved on the disk.

Configuration - HW Info

Processor: MIMX8ML8xxxLZ
Part number: MIMX8ML8DVNLZ
Core: Cortex-A53(core#0)
Board: MIMX8MP-EVK-REV-A
SDK Version: ksdk2_0

Project

No toolchain project is detected.

Pins

Configures pin routing, including functional & electrical pin properties, voltage/power rails, and run-time pin configuration.

Functional groups

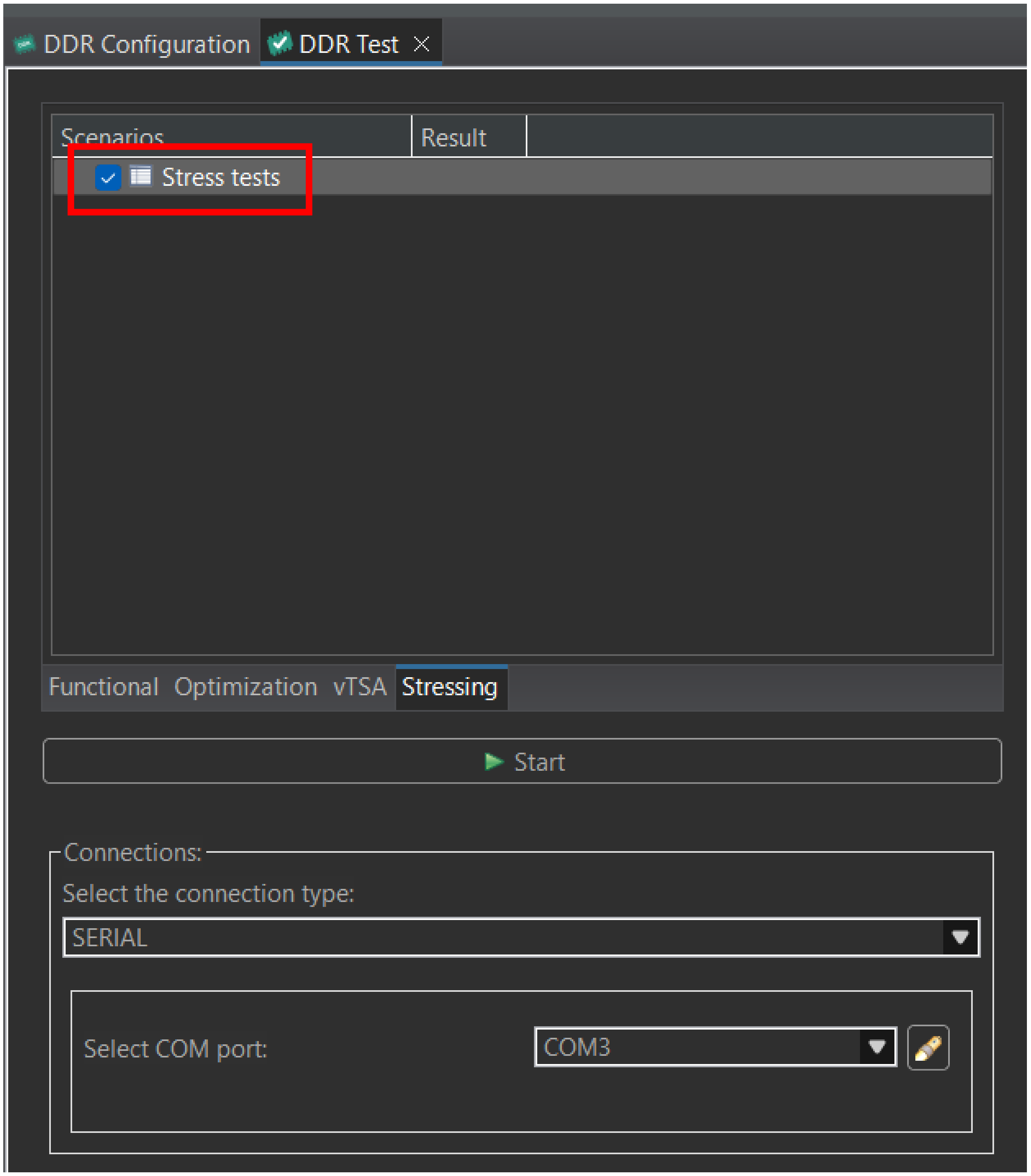
init_enet_pins
init_qspi_pins
init_usdhc_pins
init_uart_pins
init_i2c_pins
init_ecspi_pins
init_gpio_pins
init_wdog_pins
init_sai_pins
init_spdif_pins

DDR

The DDR Tool provides user-friendly configuration of initialization for DDR controllers and allows you to validate the configuration using various validation scenarios.

Tools are not supported for the selected processor: Clocks, Peripherals, TEE, SERDES, PBL

2. Configure your tests as usual using DDR Tool. For this example, we can choose stress test in the DDR test tab:



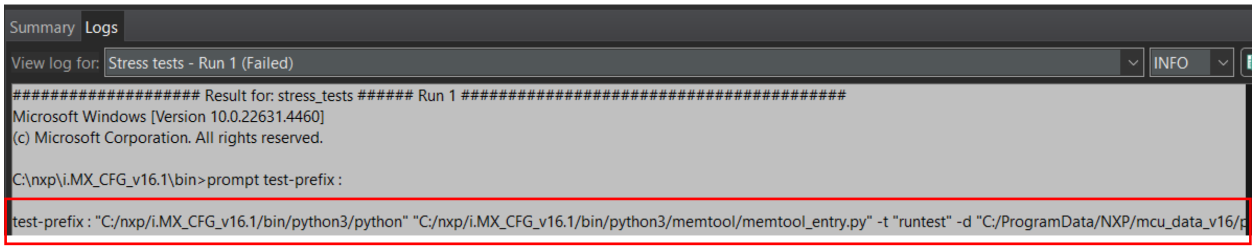
✓ Stress tests

Select the connection type: SERIAL

Select COM port: COM3

3. Connect **a single** board to the PC for the next steps
4. Select SERIAL Connection and the proper COM port (some try and error may be necessary to find the right COM port) for the A-core of the connected board. Please write the port down as you will need it later for the tool configuration.

5. Press Start to start the test and look in the logs for the following line at the start:

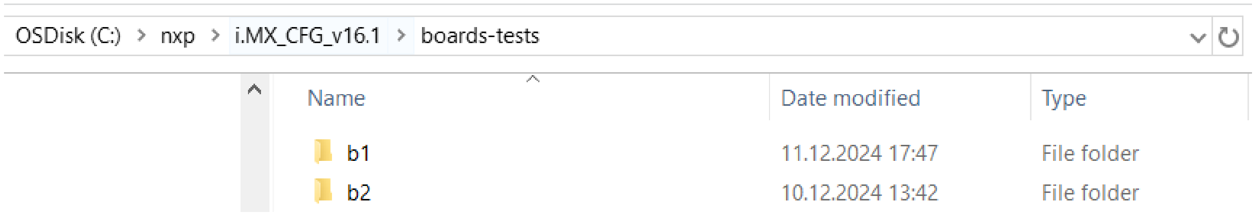


```
################### Result for: stress_tests ###### Run 1 #########################################
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\nxp\i.MX_CFG_v16.1\bin>prompt test-prefix :

test-prefix : "C:/nxp/i.MX_CFG_v16.1/bin/python3/python" "C:/nxp/i.MX_CFG_v16.1/bin/python3/memtool/memtool_entry.py" -t "runtest" -d "C:/ProgramData/NXP/mcu_data_v16/p
```

Keep the complete command line in a word pad. It will be the command to be executed fron the command line (e.g. PowerShell). Looks like:

Test-prefix : "C:/nxp/i.MX_CFG_v16.1/bin/python3/python"
"C:/nxp/i.MX_CFG_v16.1/bin/python3/memtool/memtool_entry.py" -t "runtest" -d
"C:/ProgramData/NXP/mcu_data_v16/processors/MIMX8ML8xxxLZ/ksdk2_0/mem_validation/ddrc" -p
"C:/Users/**USER-NAME** /AppData/Local/Temp/2/mem_validation/phy_training_stress_tests_0_0_.log" -l INFO
"C:/Users/**USER-NAME** /AppData/Local/Temp/2/mem_validation/connect.json"
"C:/Users/**USER-NAME**/AppData/Local/Temp/2/mem_validation/test.json"
"C:/Users/**USER-NAME** /AppData/Local/Temp/2/mem_validation/phy.json"
"C:/Users/**USER-NAME** /AppData/Local/Temp/2/mem_validation/ddrc_registers.json" "C:/Users/**USER-NAME** /AppData/Local/Temp/2/mem_validation/ddrc_config.json" "C:/Users/**USER-NAME** /AppData/Local/Temp/2/mem_validation/ddrc_config_in.json"

6. Copy that line in a text editor. Its format is as follows:
   <python executable> <**memtool_entry**> -t "runtest" -d **<ddrc_folder>** -p <phy_log_file> -l <log_level> **<list_of_configuration_files>**
7. You now need to create the following folder structure (keep the format, b[n] is the board number, t the configuration, you may have multiple ones) depending on the number of boards and tests you want to do for each board. Base_dir can be anywhere on the PC. For example, if you want to test with 3 boards and have 2 tests, create the following structure:

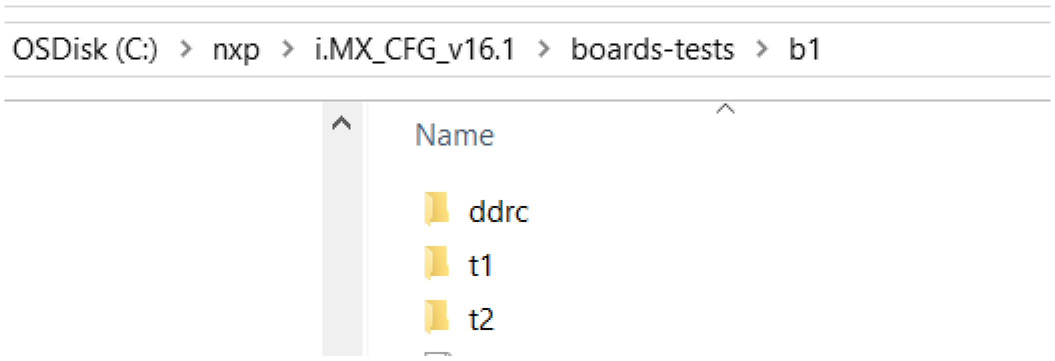


Name	Date modified	Type
b1	11.12.2024 17:47	File folder
b2	10.12.2024 13:42	File folder

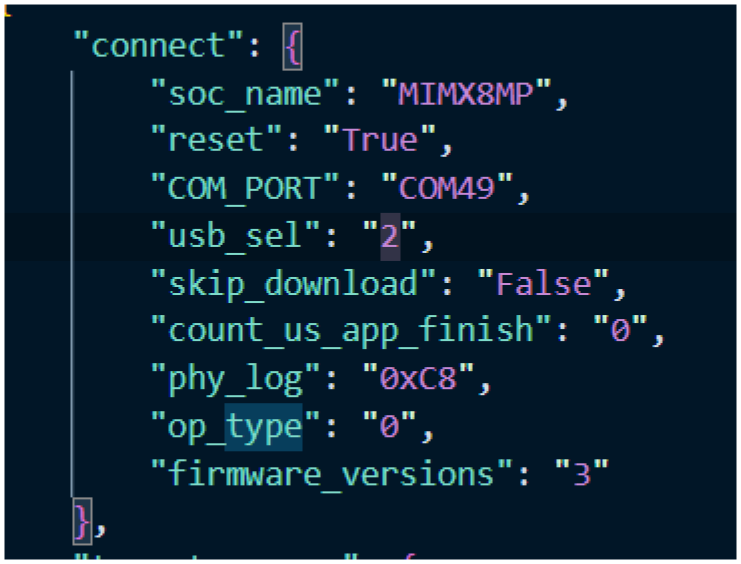
-**base_dir**
|--b1
|--|--t1
|--|--t2
|--|--ddrc
|--b2
|--|--t1
|--|--t2
|--|--ddrc

|--b3
|--|--t1
|--|--t2
|--|--ddrc

8. From the <ddrc_folder> location at step 6, copy the contents to the ddrc folder in step 7 for each board you want to test on, like shown below. If you use the same board model for multiple board folders, you can copy the same ddrc folder to the respective board folders.



9. From the <list_of_configuration_files> , get the directory of the files. It should be located at:
C:\Users\USER-NAME\AppData\Local\Temp\2\mem_validation\
Those files are the effective test configuration. Copy all contents from that folder to one of the test folders (t1 or t2 etc depending on the number of test). You can either copy the same files to all the boards to run the same test on all or repeat steps 2-5 to generate a new test.
10. You can now connect all the boards to the PC. Mind that you need to connect both the serial port and the Serial Download port for each.
11. In the connect.json of each test, you should see the following:

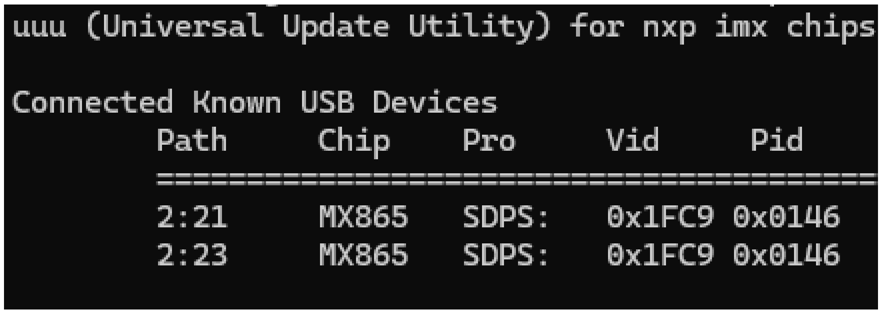


```
    "connect": {
        "soc_name": "MIMX8MP",
        "reset": "True",
        "COM_PORT": "COM49",
        "usb_sel": "2",
        "skip_download": "False",
        "count_us_app_finish": "0",
        "phy_log": "0xC8",
        "op_type": "0",
        "firmware_versions": "3"
    },
```

Write down the correct usb_sel and COM for each board's A core. There's not easy way to match the usb_sel and COM ports, so a bit of trial and error is required as mentioned above on step 4.

Recommended procedure:

1. Connect all the boards you want to test with both the Serial and Download ports to the host PC and switch them on in Download Mode.
2. Use uuu to list the connected boards in Download Mode: *uuu -lsusb*



```
uuu (Universal Update Utility) for nxp imx chips

Connected Known USB Devices
        Path      Chip     Pro      Vid     Pid
        ==========================================
        2:21      MX865    SDPS:    0x1FC9 0x0146
        2:23      MX865    SDPS:    0x1FC9 0x0146
```

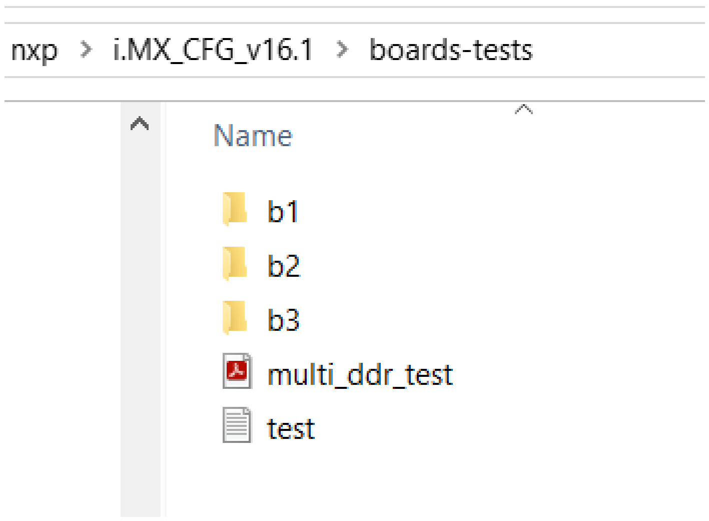
3. Take note of the order of the boards in the output as that will be the usb_sel order, starting from 0. They will always be in the same order as long as they are connected to the same USB ports, no matter the order they are connected in.
4. Power off and disconnect one of the boards from both the Serial and Download ports.
5. Run *uuu -lsusb* again*.* Take note of the COM port that disappeared and the missing entry from uuu's index. Use the two values to populate the connect.json of one of the board folders.

Tip: It's recommended to populate the board folders in ascending order of usb_sel (so b1 will be the board with usb_sel = 0, b2 with usb_sel = 1 etc)

12. In the <memtool_entry> folder from step 6, apply *memtool.patch*
    a. This modifies few files located in C:\nxp\i.MX_CFG_v16.1\bin\python3\memtool\common where the base_test.py is located
    b. C:\nxp\i.MX_CFG_v16.1\bin\python3\memtool where the s.py is located

How to run:

1. You should now be ready to run the tests using multi_ddr_test.py from the install directory where all board repositories reside:



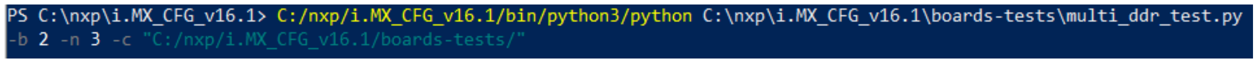
nxp > i.MX_CFG_v16.1 > boards-tests

Name

b1
b2
b3
multi_ddr_test
test

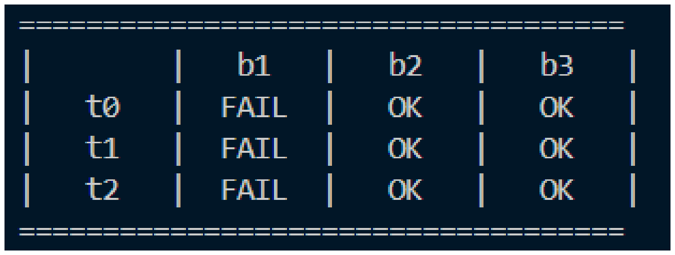
2. From PowerShell
<python_executable> .\multi_ddr_test.py -b <num_boards> -n <num_tests> -c <base_dir> [-r]
Where:
<python_executable> **is the same as in step 6**
<num_boards> - number of boards to run on
<num_tests> - number of tests to run
<base_dir> - folder created on step 7
-r - if provided, wil repeat the first test for each board num_tests times

e.g.:



```
PS C:\nxp\i.MX_CFG_v16.1> C:/nxp/i.MX_CFG_v16.1/bin/python3/python C:\nxp\i.MX_CFG_v16.1\boards-tests\multi_ddr_test.py
-b 2 -n 3 -c "C:/nxp/i.MX_CFG_v16.1/boards-tests/"
```

3. After running the tests, you should get a report like this at stdout:



```
=====================================
|        |   b1   |   b2   |   b3   |
|   t0   |  FAIL  |   OK   |   OK   |
|   t1   |  FAIL  |   OK   |   OK   |
|   t2   |  FAIL  |   OK   |   OK   |
=====================================
```

4. The resulting log for each run can be seen In tool.log for each test test folder for every board if **-r** was not provided, or in toolX.log, where X is the iteration number, in the t1 folder for each board if **-r** was provided.

Current limitations:

- In case of errors (such as disconnected cables, improper configuration, sporadic errors), the test script may hang, requiring the user to stop and restart the test suite.
- When running a heterogenous setup with different board models, the test suite may hang if there are missing test folders for some of the boards.