OptoFidelity
Touch Panel Performance Test Application
User Guide

Version 2.1

OptoFidelity Oy • info@optofidelity.com • www.optofidelity.com

Contents

1 Introduction

This document describes how to use a *TPPT (touch panel performance test)* measurement script with *OptoFidelity (OF) Touch and Test (TnT) UI*.

The measurement script and analysis are based on *Windows Hardware Certification Requirements* .

The measurement script will automatically test device according to the document requirements. Additional test cases have been added based on feedback from our customers.

2 Structure of test

Test consists of the following parts:

- Hardware initialization
- Software initialization
- Test execution
- Result analysis
- Software end actions
- Hardware end actions

3 Needed equipment

The following equipment is needed for the test.

- Robot supported by OF TnT suite. Currently script has been tested with the following robots:
  - OF-200 desktop robot
  - OF-300 desktop robot
  - OF-400 desktop robot
  - Janome Cast Pro III robot
  - OptoStandard
- Touch panel jig
- TwoFinger activation device (optional)
- Custom brass fingers
- OF-PIT (optional)
- Power supply for panels (optional)
- DUTs
- Automatic finger change equipment (optional)
- Positioning camera
- Measurement computer
  - TnT Server
  - TnT UI
  - TPPT Scripts
  - TPPT Analysis

4 Hardware initialization

4.1 TwoFinger activation device installation

TwoFinger activation device (Figure 1) can be used for 1- and multi-finger test cases.

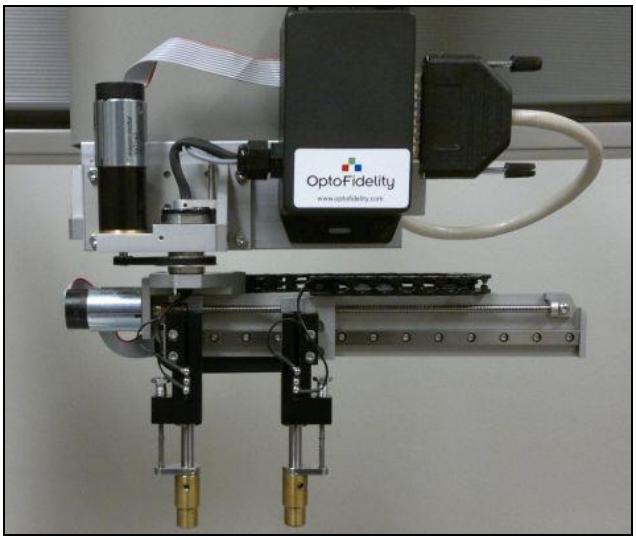


Figure 1: TwoFinger activation device

4.1.1 TwoFinger installation

1. Attach TwoFinger activation device to robot head using 4 screws (see Figure 2, Figure 3).

Figure 2: Screws for installing TwoFinger actuator

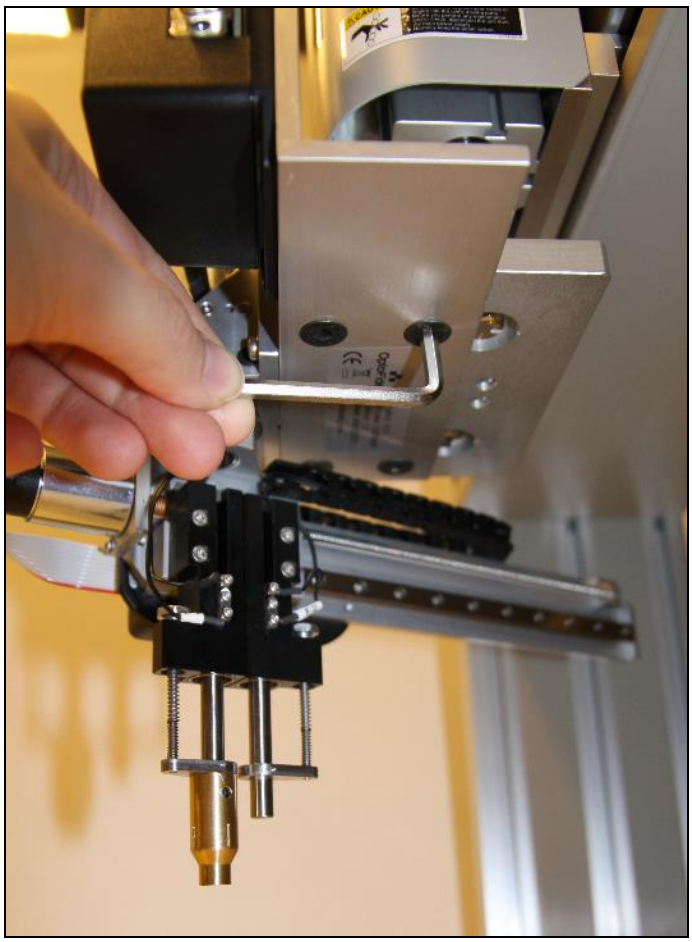


Figure 3: Using a 4 mm hex key for attaching TwoFinger actuator to robot head

2. Attach energy chain with a hex screw and two washers to the back of TwoFinger actuator (Figure 4).

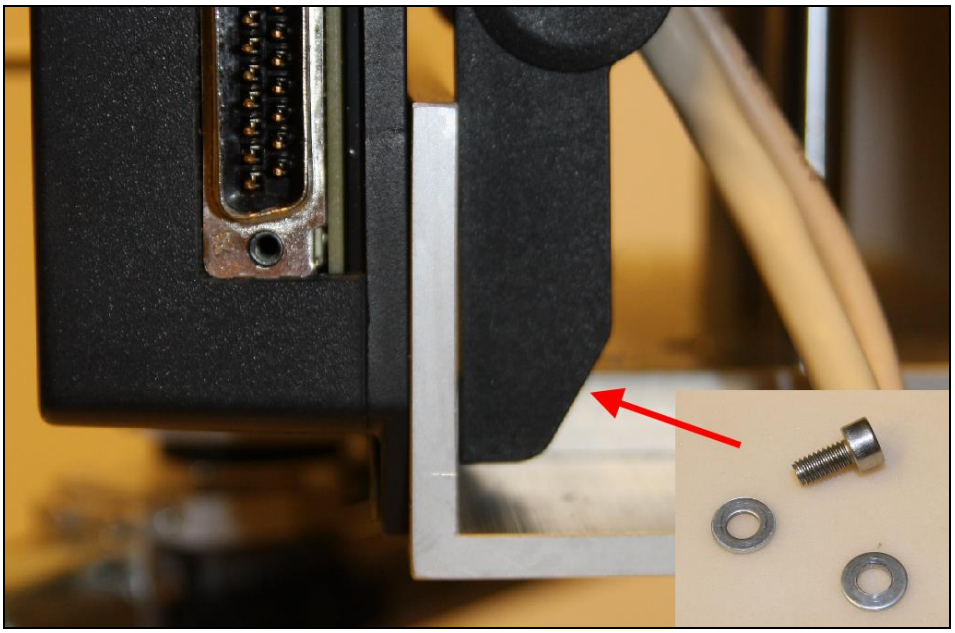


Figure 4: Energy chain attached with a hex screw and two washers

3. Connect D25 cable to TwoFinger actuator.

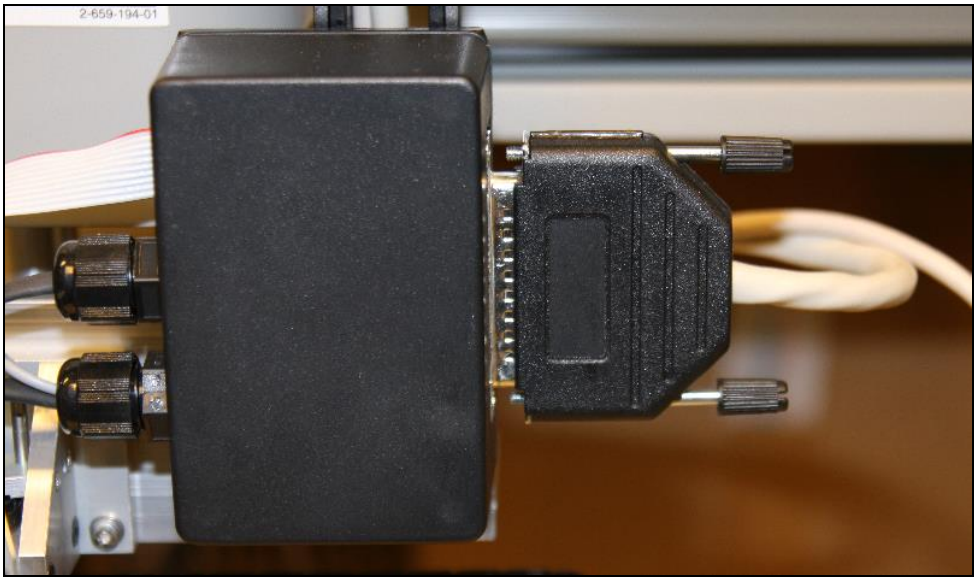


Figure 5: D25 cable connected to TwoFinger activation device

4. Connect mini USB cable to the camera mounted on the TwoFinger actuator.

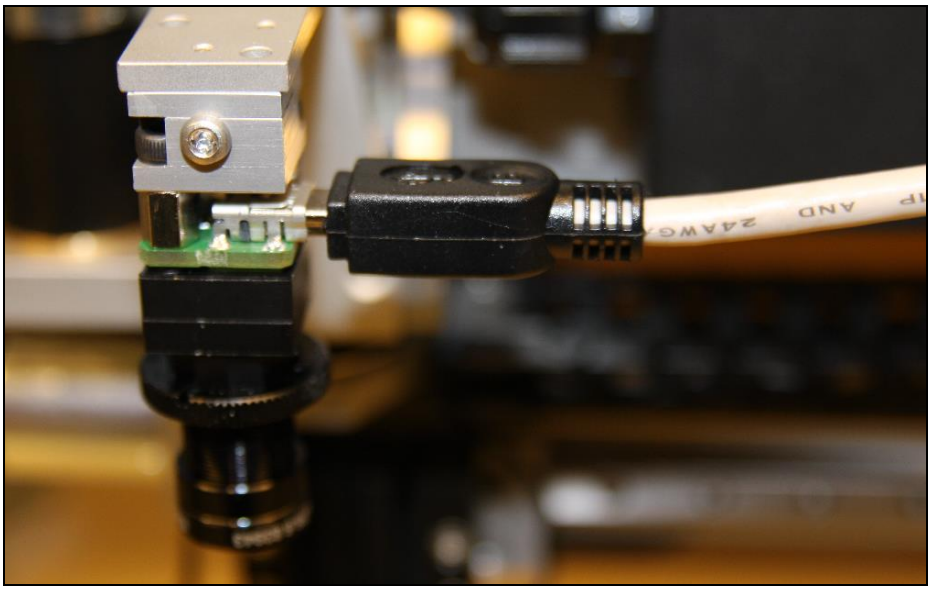


Figure 6: Positioning camera mounted to TwoFinger activation device

4.2 Initialization procedure

1. Power on TwoFinger controller and wait until the FPGA led starts blinking (Figure 7).

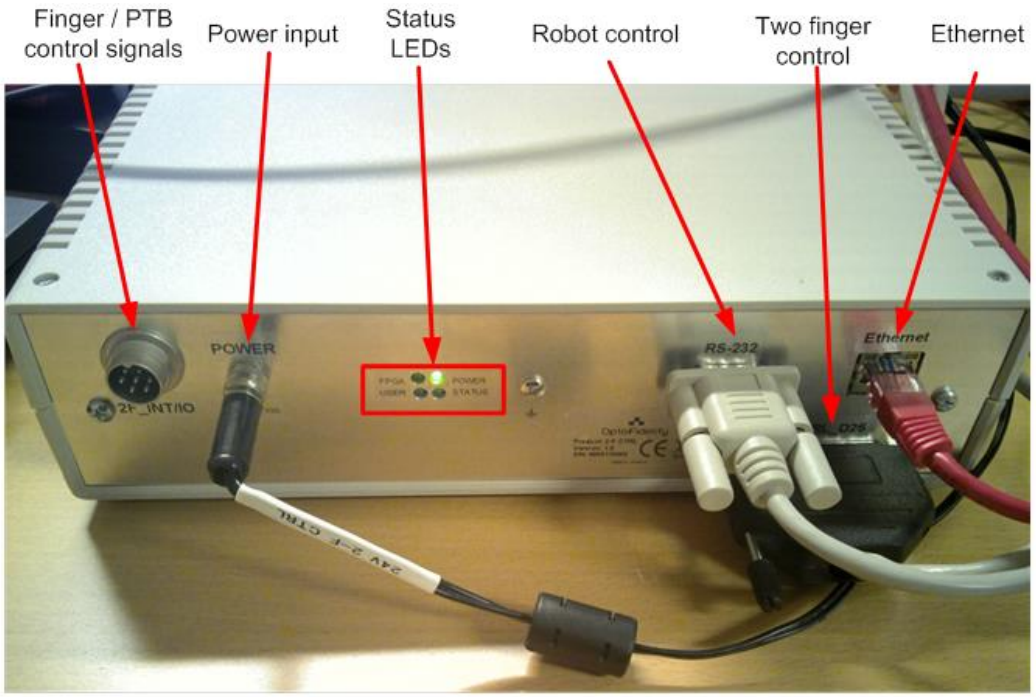


Figure 7: TwoFinger controller conenctions and status LEDs

2. Power on the robot using robot's power switch.
3. Wait until status LED starts to blink in two finger controller.
4. Move TwoFinger arm near the home position by hand. The rotation motor is powered up when arm is near the home position.

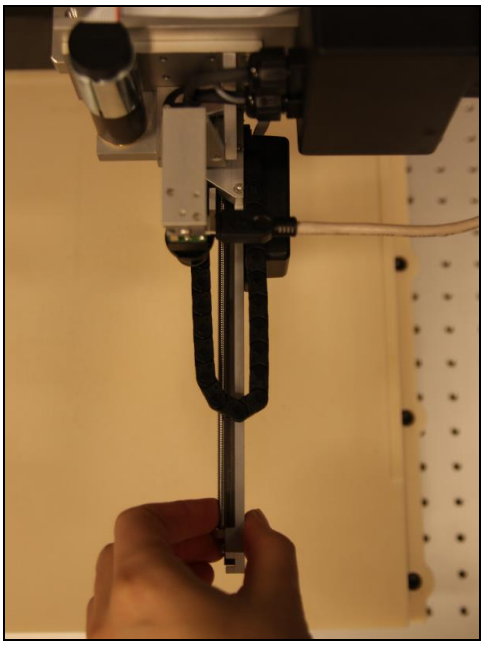


Figure 8: Moving TwoFinger arm near home position manually until the rotation motor is powered on

5. Press blinking button on the robot operation box (Figure 9) to execute the homing procedure. (Janome robot only)

Figure 9: Janome Robot operation box

6. Wait until the green LED on robot operation box is turned off. (Janome robot only)

4.3 Connecting DUTs

1. Place the DUT onto touch panel jig and fix it firmly.
    a. Tape or blue-tack can be used for panels (Figure 10).
    b. Eccentric clamps can be used for devices (Figure 11).

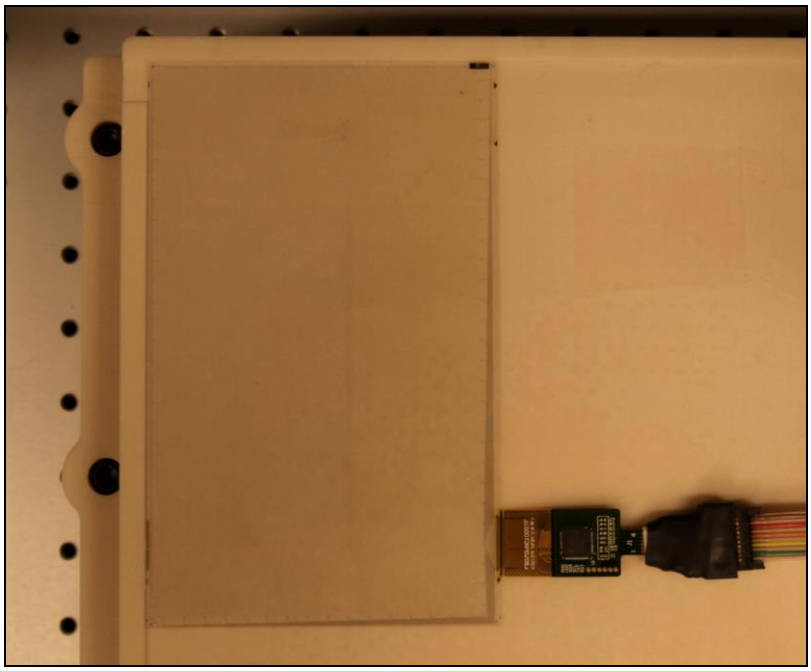


Figure 10: Touch panel on touch panel jig

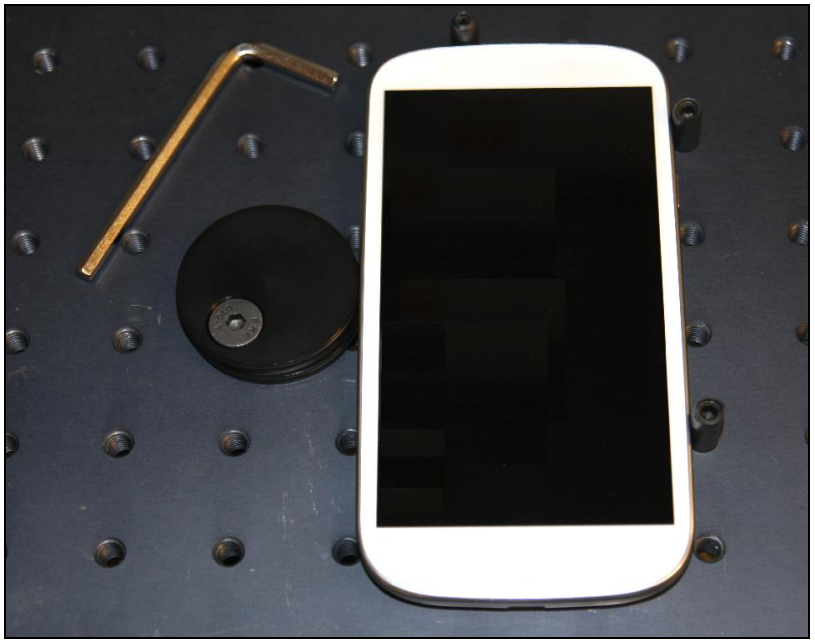


Figure 11: DUT fixed with a round eccentric clamp on perforated plate

WARNING! Make sure that top surface of the clamps are not above the tested panel surface. If clamps are above the measurement surface, robot may hit clamps while testing!

2. Connect DUTs to PIT (for panels only, connecting Android device is described in chapters 6.1 and 6.2).
    a. Make sure that adapter cables are attached between DUT adapter chips and PIT panel connectors.

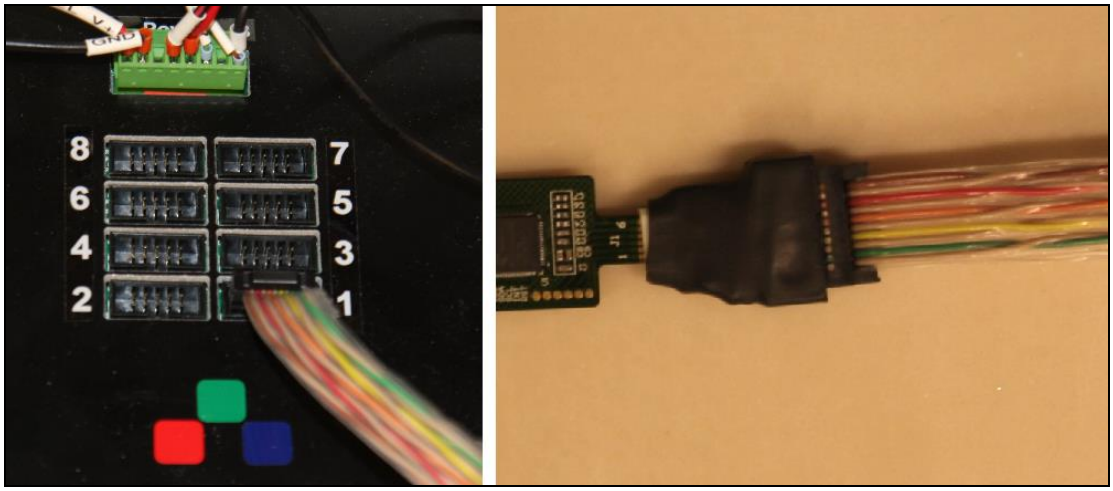


Figure 12: Flex adapter cable between DUT adapter chips and PIT panel connectors

    b. Make sure that panel is powered properly.

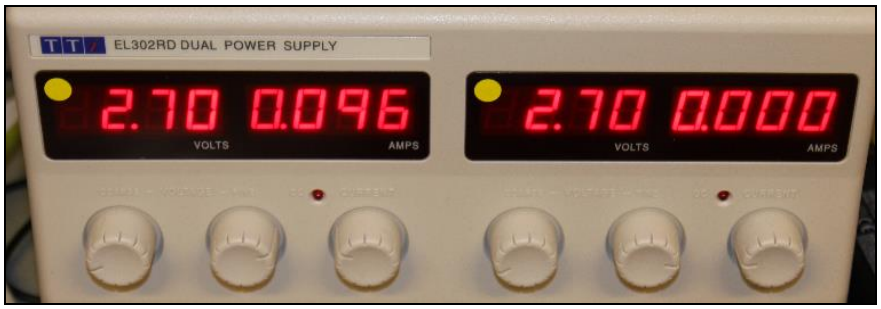


Figure 13: Typical voltage level for touch panels. Check panel specifications for recommended voltage level

3. Power on OF PIT with *OFF/ON* switch and wait until *Status* LED starts to blink.

Figure 14: OptoFidelity PIT connectors, OFF/ON switch and status LEDs

5 Software initialization

5.1 Folder structure of Touch and Test Suite

The following programs are installed under *C:\OptoFidelity* folder:

- TnT Server
- TnT UI
- TPPT Scripts
- TPPT Analysis

5.2 Folder structure of the measurement script

Measurement script for capacitive touch panel measurements is installed under *C:\OptoFidelity\TPPT* folder.

Measurement scripts have a support folder named *TPPTcommon* which includes all relevant data which is needed to perform the measurement. The support folder contains the following items:

- *TPPTcommon\Drivers* – Framework for DUT communication
- *TPPTcommon\PIT_Drivers* – Touch panel controller drivers
- *TPPTcommon\Measurement* – *Touch event measurement interface for scripts*

By default, TPPT script uses database.sqlite file for storing measurement data. It is located at *C:\OptoFidelity\TPPT* folder.

5.3 OptoFidelity TnT initialization

This chapter describes actions for the measurement session initialization.

- Start *OF TnT Server* from desktop shortcut.
- Start *OF TnT UI* from desktop shortcut.
    - Under *Configure* tab, select *Camera Calibration* from side bar. Calibrate focus height, distortion and offset according to UI instructions. Click the

question mark icon for more instructions. Most widgets also display tool tips when hovered over by cursor.

- Select *DUT position* tab from side bar. Click *Add new DUT* to create new DUT. Teach required location points: top right, top left, bottom left corners of the DUT. Use *Tip* buttons to define the corner (x,y,z) positions according to current robot tip position. Then use *Camera* buttons to refine the x and y corner coordinates from the camera image (requires that camera camera calibration was done previously).
- OptoFidelity TouchTestApp for Android, iOS, Windows Phone, Windows RT includes markers to make camera corner positioning easier.

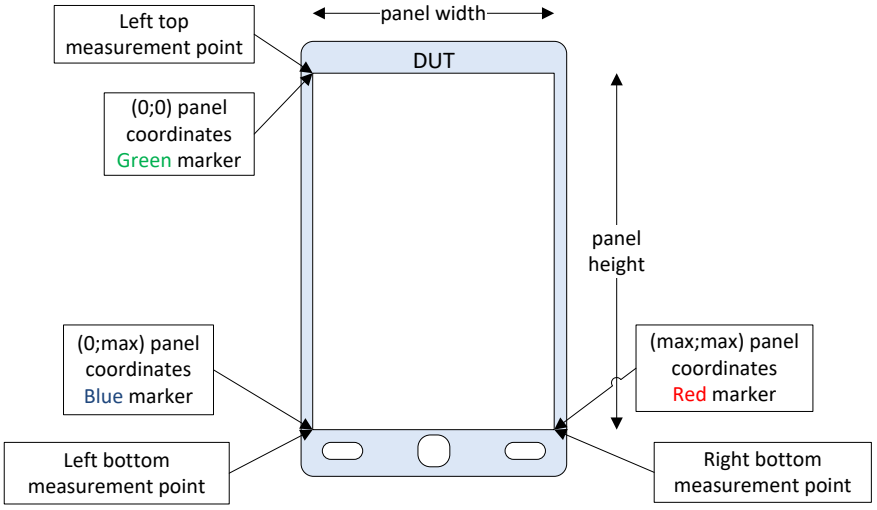


Figure 15: Panel corner points

- Select *Tips* tab from side bar. Click *Add* to create new tip. It is important to set tip length correctly as it affects the z-position where the robot moves when tip is attached. Set slot in and slot out positions if tip can be dropped to / picked from a rack.
- Select *Scripts* tab from top bar. From TPPT installation directory, open the python file that contains the Context class. After successful load, test parameters and control tree should become visible in the script page.
- Fill in information to *Program*, *Manufacturer*, *Version* and *Operator* fields in the scripts view.
- Activate at least one DUT, tip and test case to run a test sequence.
- Before starting the execution of test you can check how measurement points or measurement lines are scattered over the area by pressing *Show measurement points* button.
- Click *Run tests* to start test sequence.
- It is possible to click *Load script* again to reload test script in case some modifications were made.
- Parameters and controls from previously executed tests can be copied by selecting corresponding time stamp from the *History* drop down menu below the control tree.

Warning! When operator presses Stop test button, test will not stop immediately. It stops to next breakpoint that has been implemented to the test case. In case of emergency, please press emergency stop button that is located near the robot to minimize possible damage to the system!

Warning! *DUT Dimensions [x;y, mm]* control parameter is filled automatically using data set via TnT UI Configure page when script is loaded. If control parameters are copied from a previously executed test run, *DUT Dimensions [x;y, mm]* field value will be copied from the previously executed test run and it might not correspond to the actual DUT dimensions any more.

Fill test parameters for capacitive touch panel test in the tree list.

- Active grid
  - Defines parameter to calculate test grid for certain test cases.
- Line drawing speed [mm/s]
  - Defines movement speed of the robot while executing swipe tests. NOTE: Robot can move with a different speed during test cases when it is not swiping the DUT.
- Effectors for one finger tests
  - Defines effectors which are used during one finger tests.
- Effectors for multifinger tests
  - Defines effectors which are used during multifinger tests.
- Effectors for separation test
  - Defines effectors which are used during separation tests.
- DUT items
  - Define if tests are executed for selected DUTs
- DUT handler

  - TCP Socket → For device level testing (e.g. Android phones/tablets, Windows phones, iOS devices etc.). NOTE: Requires WiFi connection to measurement computer and a specific test application installed to DUT.
  - PIT → For module level testing. (e.g. touch panel modules that support I2C or SPI protocol) NOTE: Requires OptoFidelity PIT connected to measurement computer)
  - Dummy DUT → DUT simulator for test case development. No hardware needed.
- PIT Driver
  - Defines the driver used to communicate with touch panel module.
  - Normally named after chip manufacturer (e.g. PIT_Atmel_mxt.py)
  - Drivers are in support folder under the subfolder *PIT_Drivers.*
  - Devices over TCP socket does not use driver, but control can't be empty.
- PIT Slot
  - Defines hardware slot for DUT if connected to PIT
  - Possible selections are from slot 1 to slot 8
- DUT dimensions [x;y, mm]
  - Defines DUT dimensions in millimeters in x- and y-direction. Dimensions set by TnT UI are retrieved automatically when script is loaded and used as default values.
- DUT resolution [x;y, p.c.]
  - Touch panel max coordinates. Example: "720 x 1280"
- Display resolution [x;y, p.c.]
  - Display resolution. Example: "720 x 1280"

6 Testing with devices

Operating system specific application can be used for measuring touch panel performance on devices using TCP socket connection. This chapter describes how to set up TCP socket connection between measurement PC and Android device. Test steps are same when using other operating system than Android.

6.1 Setting up network

- Power on WLAN access point (if not already on)
- Check that measurement computer is connected to WLAN access point
  - SSID = TCPSocket
  - WPA2 PSK password = touchtester

- Get PC IP: Open Windows Command Prompt and write *ipconfig*. IPv4 Address under Wireless LAN is the one you are looking for (normally 192.168.1.5).

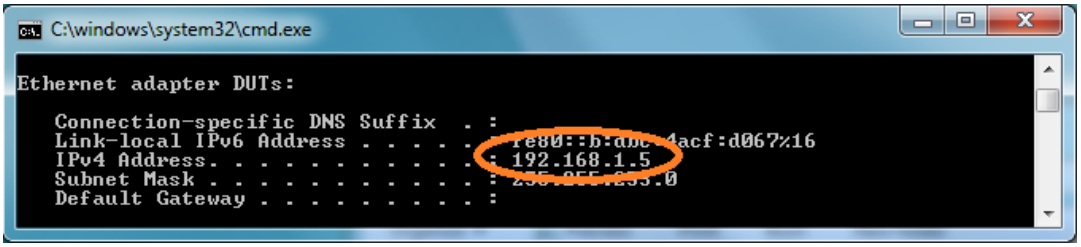
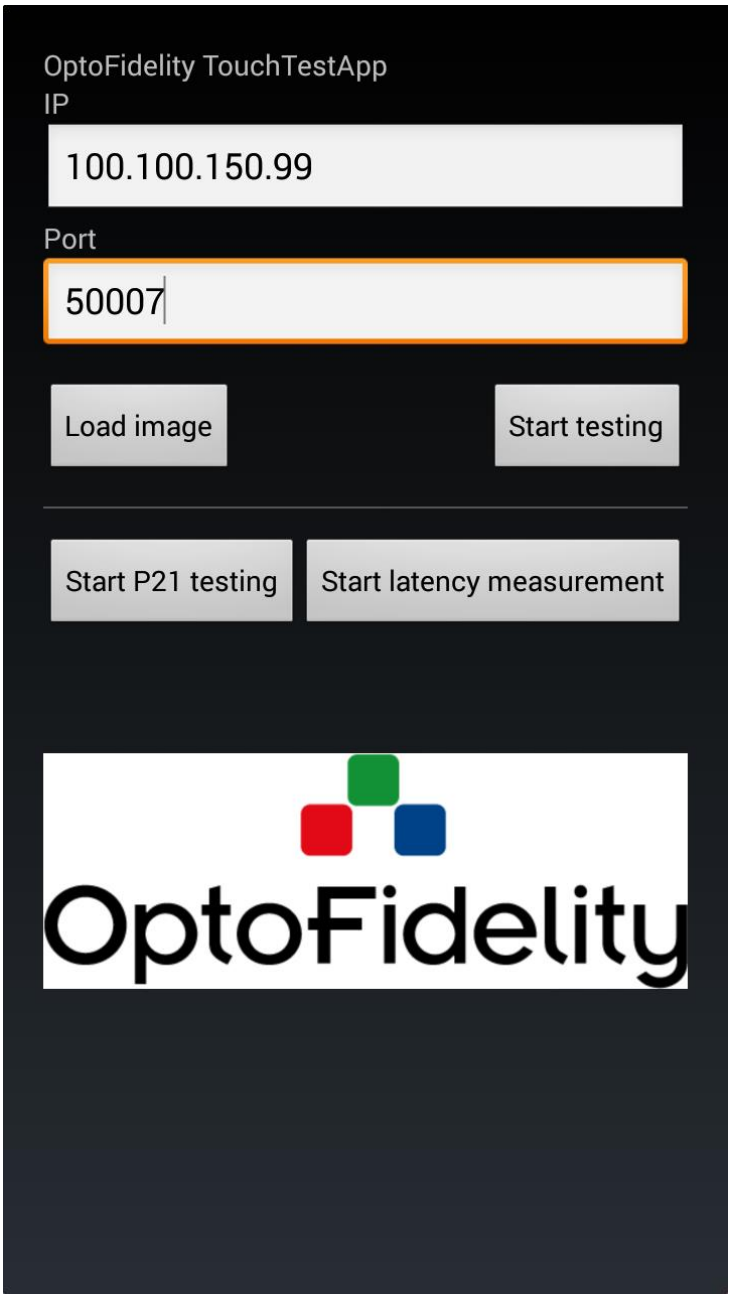


Figure 16: PC IPv4 Address

- Connect Android device to WLAN access point
  - SSID = TCPSocket
  - WPA2 PSK password = touchtester

6.2 Using OptoFidelity TouchTestApp Android application

- TouchTestApp application must be installed on the Android device.
- When measurement script is loaded to TnT UI Script page, open the application and insert following settings:


OptoFidelity TouchTestApp
IP
100.100.150.99
Port
50007
Load image
Start testing
Start P21 testing
Start latency measurement
OptoFidelity

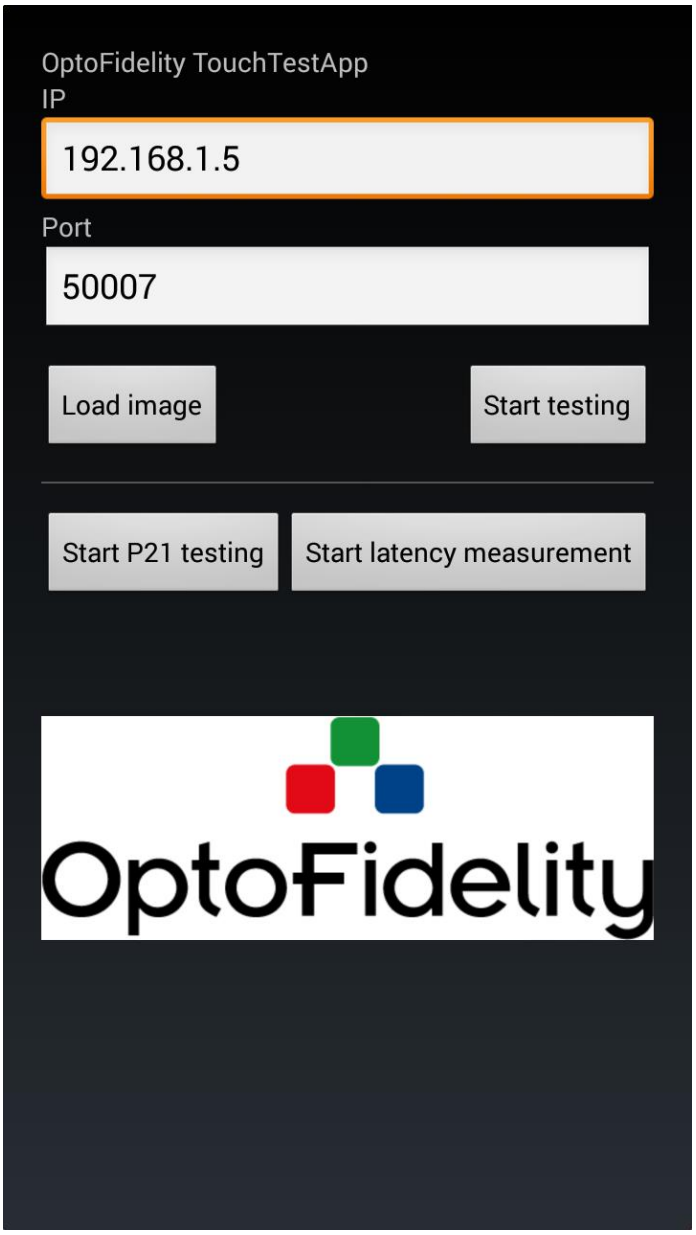


Figure 17: OptoFidelity TouchTestApp Android application

- IP: IP of measurement computer WLAN card (see chapter 6.1)
  - normally 192.168.1.15
- Port : 50007

- Application remembers these settings so user only has to input these values once.
- To change background image, press ***Load Image*** button and select background image from image gallery.
- After all parameters are typed in, press ***Start testing*** button
  - Check that text ***Connecting*** changes to ***Connected*** on the screen to verify that connection is working properly

7 Results

During the measurement sequence, test results from a touch panel are written to a script specific database (*database.sqlite*). This is a primary storage for the measurement data.

Test results are visualized and analyzed with *TPPT Analysis software*. Start *Analysis software* by double-clicking ***TPPT Analysis*** shortcut on the desktop. This opens command line prompt that shows the status of the application and web browser where Analysis software is shown. Web browser will show the main page of Analysis SW (Figure 18).

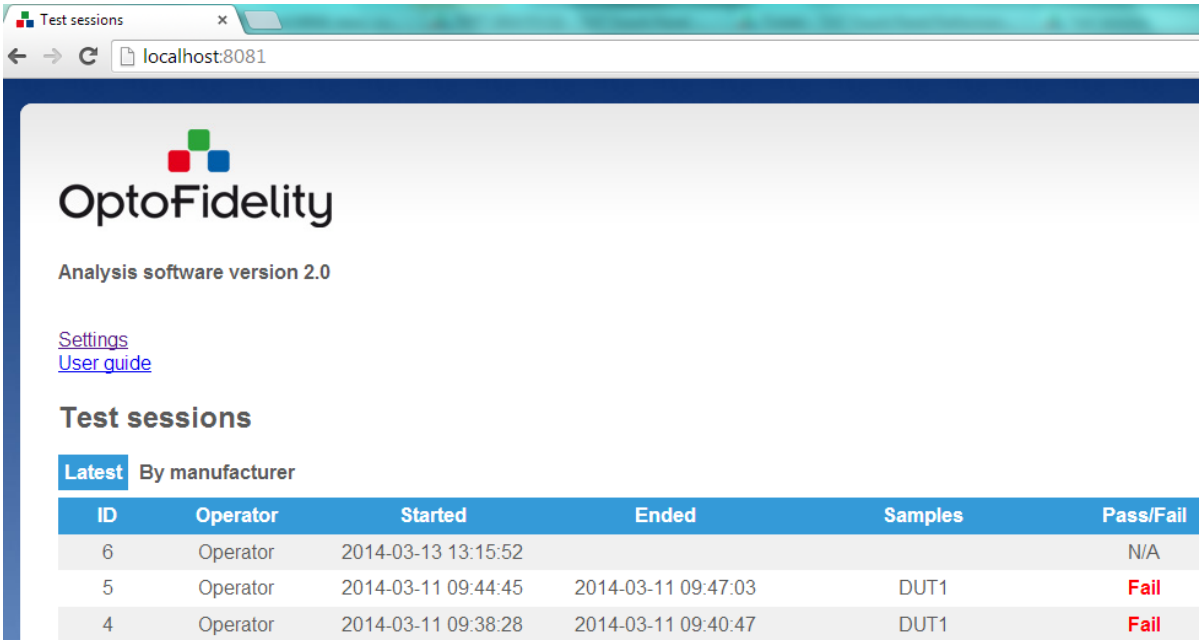


Figure 18: Analysis SW main page

For instructions on how to use analysis SW, open *Analysis Software User Guide* by clicking ***User guide*** link on the main page. Example results plots are shown below.

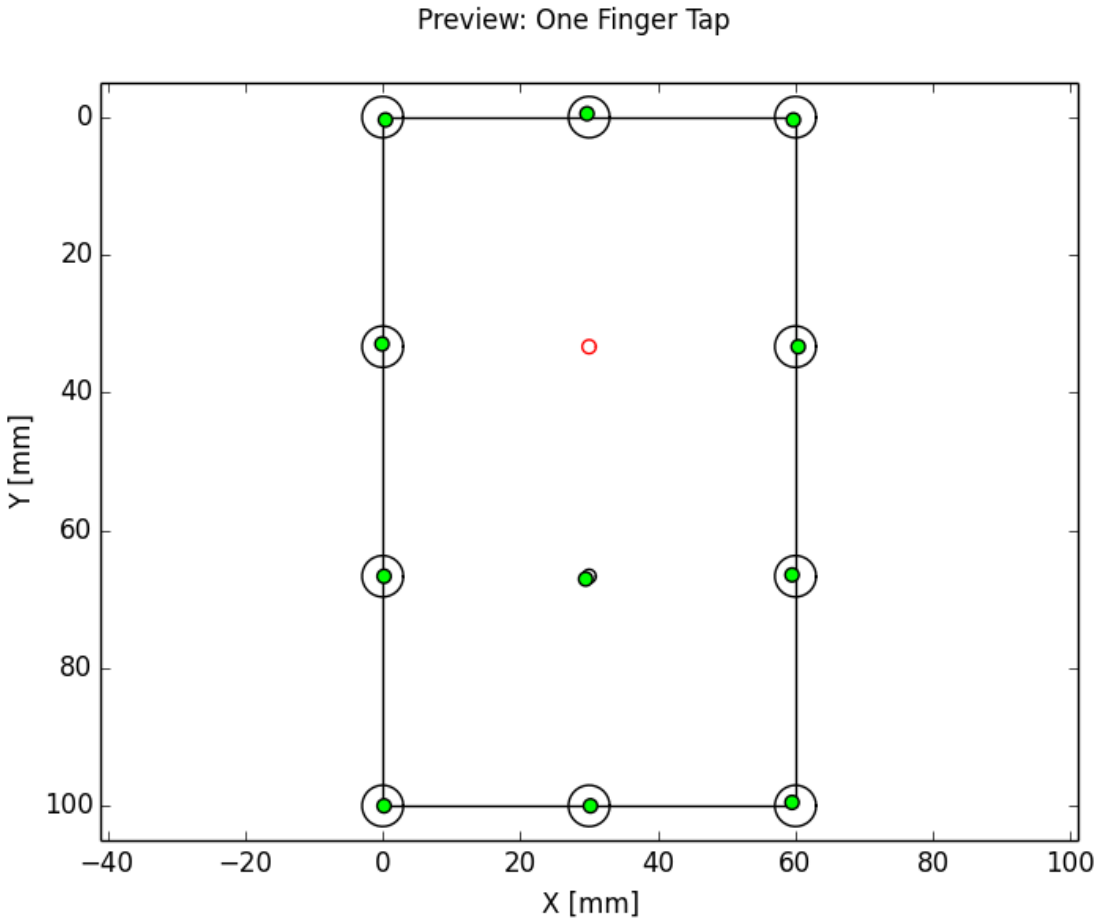


Figure 19: Example result plot, One finger tap test. Green markers indicate that results are within error limits. Red circles indicate that no cordinate was reported during tap. Notice that edge area and center area has different limits.

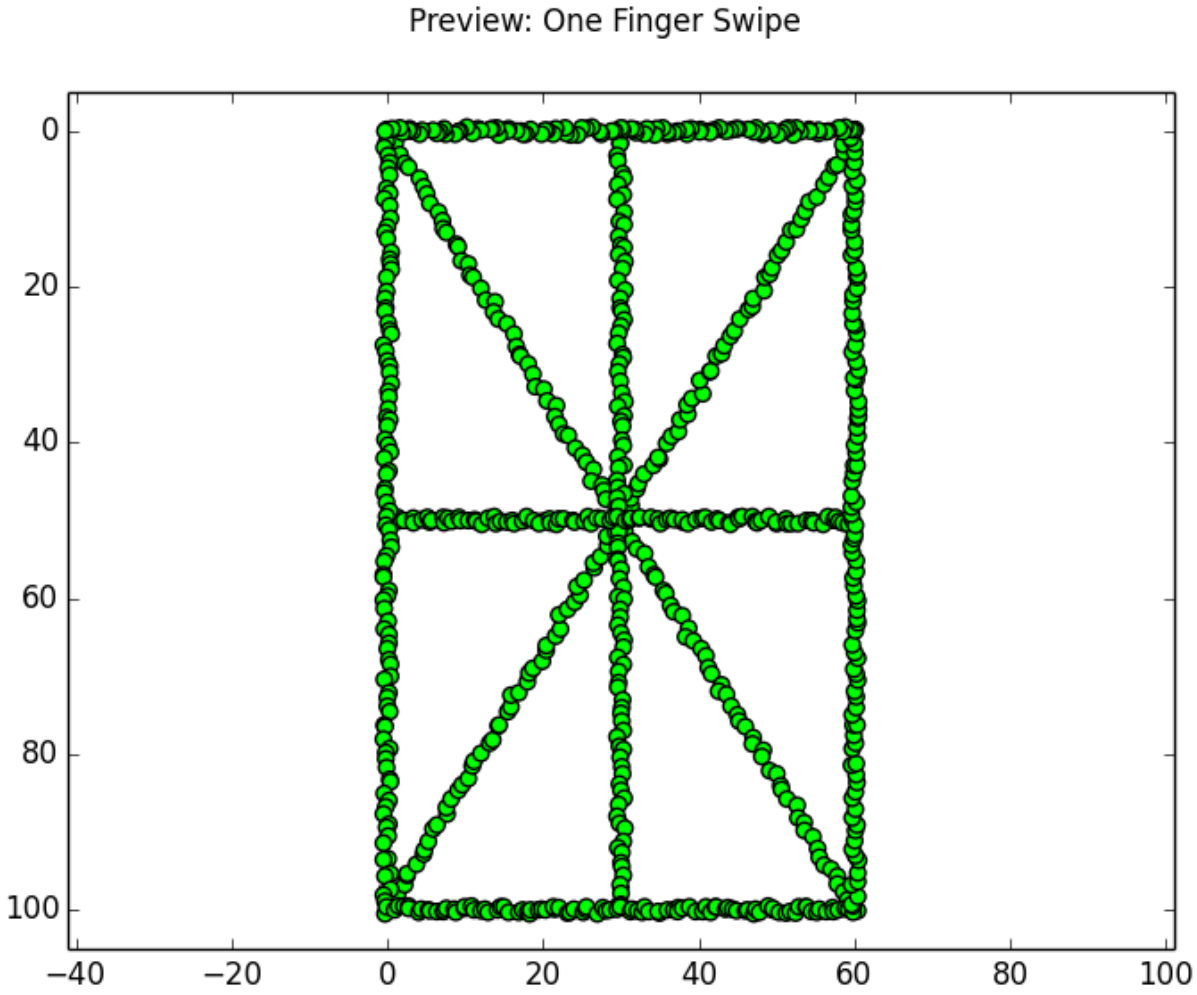


Figure 20: Example result plot, One finger swipe test

8 Software end actions

Shutdown software if you are going to shut down the whole system.

9 Hardware end actions

None.

10 Test cases

10.1 First Contact latency

First contact latency test is used to measure delay between actual finger touch and when touch panel controller sends interrupt signal indicating that it has detected touch.

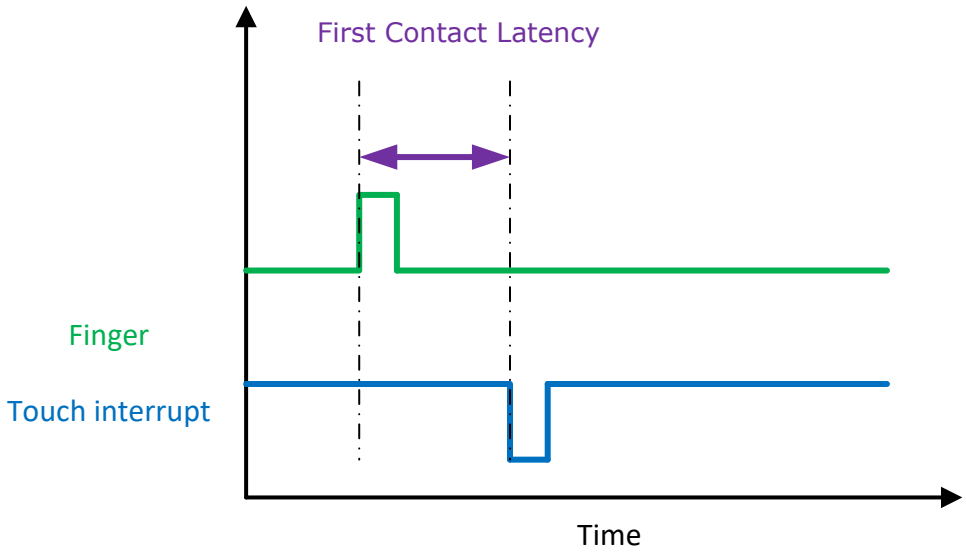


Figure 21: First contact latency

NOTE: This measurement can be done only for Touch panel modules.

10.1.1 Parameters

- Amount of points per test
  - Defines how many points are tested. By default these points are in random locations.
- Calibrate system latency
  - If selected, system will try to calibrate itself using **delay calibration target**
  - Without proper calibration, delay between actual touch and finger switch is not removed from final report.

10.2 Non Stationary Reporting Rate

Non stationary reporting rate test is used to measure reporting rate of DUT when finger is swiping screen with constant speed.

10.2.1 Parameters

None

10.3 Repeatability

Repeatability test is used to measure repeatability of touch panel when tap gesture is repeated to the same location on the screen.

10.3.1 Parameters

- Amount of points per test
    - Defines how many points are tested. By default these points are in random locations.
- Amount of taps per point
    - Defines how many tap gestures are done at each test point.

10.4 Stationary Jitter

Stationary jitter test is used to measure jitter when finger is not moving and finger is touching the screen.

10.4.1 Parameters

- Amount of points per test
    - Defines how many points are tested. By default these points are in random locations.

10.5 Stationary Reporting Rate

Non stationary reporting rate test is used to measure reporting rate of DUT when finger is not moving and finger is touching the screen.

10.5.1 Parameters

- Amount of points per test
    - Defines how many points are tested. By default these points are in random locations.

10.6 Swipe

Swipe test is used to measure linearity and jitter of DUT when finger is swiping screen with constant speed.

10.6.1 Parameters

- Worst case lines
    - Defines if this test pattern is used during testing
- Vertical/Horizontal lines
    - Defines if this test pattern is used during testing
- Diagonal lines

- Defines if this test pattern is used during testing

- Active grid (global parameter)
  - Defines distance between measurement lines for vertical, horizontal and diagonal lines.

NOTE: Each test pattern will create a new test to analysis report.

10.7 Tap

Tap test is used to measure accuracy of DUT when tap gesture is used across the screen.

10.7.1 Parameters

- Active grid (global parameter)
  - Defines distance between individual taps

10.8 Multifinger Tap

Tap test is used to measure accuracy of DUT when tap gesture is used across the screen. Tap gesture is made with 3, 4 and 5 fingers.

10.8.1 Parameters

None

10.9 Multifinger Swipe

Swipe test is used to measure linearity and jitter of DUT when multiple fingers are swiping screen with constant speed.

10.9.1 Parameters

None

10.10 Separation

Separation test is used to measure minimum finger separation where DUT can recognize two individual fingers.

10.10.1 Parameters

None

11 Change history

Ver.	Status	Date	Author	Remarks
2.0	Final	07.05.2014	JTU	
2.1	Draft	02.10.2018	JM	Updated to match new platform