HyperNova CapTouch Test System

1. Prerequisite

For the code to work. Python 3.8 is required. Other dependencies can be found in requirements.txt under the “HyperNova\_Captouch\_Test\_System” folder. After installing python in the command window add project folder/2DTouchGeture into the path so the program can invoke TouchCoordinatesGesture.exe correctly.

1. Folder Structure

The “main.py” in this project is the main file that needs to be run if you want the stdout output to the Dialog window. Otherwise, you can run “CapTouchTestSystem.py” if you only need GUI. Once the motor is connected to the USB. Please make a note of the com port the motor is using. Change the MOTOR\_COM = ‘XXX’ in Constants.py to the port the windows is using.

The raw file captured will be stored under the data folder of the project. Make sure you have the folder created beforehand. The ui folder has the GUI files and resource files used in the project, you can edit them using Qt Creator or Qt Designer.

|  |
| --- |
| Fig.1 Project folder structure |

1. Stella CapTouch Test System Data Flow

The system consists of two parts, motor and ADB. Both of them are controlled by the software. Fig.2 shows the test setup.

|  |
| --- |
| Fig.2 System Diagram of the Cap Touch Test System |

Fig.3 is the high level block diagram of the software. The CalibrationLogic and CapTouchLogic are the main classes that control the worker threads TestFixture. The worker thread itself is metex and thread protected so only one thread is actively controlling the TestCases at one time. The test case will have the actual tests and it controls the ADB bridge and motor directly. All logics are using shared memory and signals for communication and paramization.

|  |
| --- |
| Fig3. Block diagram of the software |

1. Stella Captouch Test System User Guide

Once GUI is executed, the dialog box in Fig.4 will show up.

|  |
| --- |
| Fig.4 Main window |

**User needs to calibrate the test system before making measurements.**

Mount the glass so the glass temple is evenly and at 90 angles with the probe at both hinge side and earpiece side. Follow the instruction to perform the calibration.

|  |
| --- |
| Fig.5 Test System Setup |

|  |
| --- |
| Fig.6 Calibration Menu |

1. Noise Measurement

Users can perform noise measurement of the 4 pads. Number of samples for noise measurement can be set at NUM\_OF\_SAMPLES\_PER\_MEASUREMENT\_NOISE\_FULL\_SCAN = 50

in Constant.py

After running the test. You can use the Preliminary\_Report.py and load the processed csv file to get the results.

|  |
| --- |
| Fig.6 Example of Noise result from Preliminary\_Report.py |

6.Linearity/Jitter Measurement

Jitter measurement measures the position reported by the glass when the probe scanning through the glass arm(x axis). Fig.7 shows the setup, the probe step along the x axis can be configured at MOTOR\_X\_STEP\_SIZE\_IN\_MOTOR\_UNIT in Constants.py. Number of samples to captured per step can be configured at NUM\_OF\_SAMPLES\_PER\_MEASUREMENT\_JITTER\_LINEARITY in Constants.py

|  |
| --- |
| Fig.7 Test setup for Linearity/Jitter measurement |

|  |
| --- |
| Fig.8 Linearity/Jitter measurement report from Preliminary\_Report.py |

7. Full Scan Measurement

Full Scan Measurement measures the position reported by the glass when the probe scans through the glass arm(X axis). Fig.9 shows the setup, the probe scans along the x axis back and forth and retracts in Y direction exponentially(can be configured as linearly) every time it reaches one end.

Configurations in Constants.py:

The step size in X direction:

MOTOR\_X\_STEP\_SIZE\_IN\_MOTOR\_UNIT

The step size in Y direction:

MOTOR\_Y\_STEP\_SIZE\_IN\_MOTOR\_UNIT

Linearly retract on Y direction MOTOR\_Y\_STEP\_SIZE\_INCREMENT\_BASE\_MOTOR\_UNIT=1

Exponentially retract on Y direction MOTOR\_Y\_STEP\_SIZE\_INCREMENT\_BASE\_MOTOR\_UNIT=2

Number of samples to capture per step: NUM\_OF\_SAMPLES\_PER\_MEASUREMENT\_NOISE\_FULL\_SCAN

|  |
| --- |
| Fig.9 Full Scan Measurement probe movement direction |

|  |
| --- |
| Fig.10 Example report from Preliminary\_Report.py |

8.Arbitrary X hover

Similarly to Full Scan Measurement but the probe only moves in Y direction at particularly X point.

|  |
| --- |
| Fig.11 Example report from Preliminary\_Report |

9.System Tests

Perform system tests and give out report