PENMOUNT PCIUTILITY USERS' GUIDE FOR MICROSOFT WINDOWS

Version 1.8

2020/06/15

PenMount®
TOUCH SCREEN SOLUTIONS

Preface

Disclaimer

The information in this document is subject to change without notice. The manufacturer makes no representations or warranties regarding the contents of this manual and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, the manufacturer reserves the right to revise this publication or make changes in the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision.

Trademarks

AMT is the registered trademark of Apex Material Technology Corp. PenMount is a registered trademark of SALT International Corp. Microsoft and Windows are registered trademarks of Microsoft Corp. Other product names used in this manual are the properties of their respective owners and are acknowledged.

Copyright

This publication, including all photographs, illustrations and software, is protected under international copyright laws, with all rights reserved. Neither this manual, nor any of the material contained herein, may be reproduced without the express written consent of the manufacturer.

Copyright © 2020 All rights reserved

Revision Table

Date	Revision	Changes
31/May/2016	1.0	First release version.
03/Jun/2016	1.0A	Revised 0 Supported Devices
28/Oct/2016	1.1	(1) Added 0 Special Notice (2) Added 3.5.1 Palm Rejection Switch (3) Added 3.9.2 Edge Compensation (4) Added 0 Touch and Halt
16/Nov/2016	1.1A	Revised: 3.5.4 Disabled range near palm Revised: 3.10.2 Idle mode settings
25/Apr/2017	1.2	(1) PM1415, PM1715, and PM2204 are added to the supported device list. (2) Update for PCIUtility V1.2.x (3) Add: Section 2.4 for Environmental Reference (4) 3.9.1 Calibration, the hotkey for skipping calibration checkpoint is changed to F9.
29/Aug/2017	1.3	(1) 1.2 Support Device, PM2251 are added to the supported device list. (2) Removed 1.4.2 Application Version and 1.4.3 Open Default Export File Folder (3) 2.1 Open Short Test, added “Near Check” and “Flat Check” (4) 2.4 System Reference (SR), modified “Environmental Reference” to “System Reference”. (5) 3.1 Main Features, modified “Import Parameters” to “Import PCIUtility UI Parameters,” “Export Parameters” to “Export PCIUtility UI Parameters,” “Reload Default Parameters” to “Reload Factory Default Parameters”. (6) Removed “Baud Rate Setting” (7) 3.8.2 Enhanced Driving, added new function “Enhance

		Driving Switch". (8) 4.1 Load Firmware File, added "pfp format" support. (9) 6.5 Log Noise Information, added "Log Noise Information".
03/Jan/2018	1.4	(1) Added "Parameters CRC" to page display items (2) Added 1.5: Log UI Parameters Information Button under Information item, can read/write UI Parameter. (3) Added Auto Save Log Button to 2.1.3 Open Short Test function. (4) Added 2.4.3 Options, "Switch on the System Reference Dialog Box" and "Auto Save Log Button" selections to System Reference function. (5) Changed 3.1 Export / Import Parameters file format from PMP to INI format in the Parameters item. (6) 3.9.3 Orientation, corrected Orientation display method. (7) Added "Writer Factory Default Parameter" and "Write Host Interface/Panel Parameters" selections under 4.2 Firmware Update item.
20/Jul/2018	1.5	(1) 4.1: add description of utility supporting pfp files with virtual button settings. (2) 4.2: add description of utility automatically selects write factory default option when using PMF or PMC file.
31/Jan/2019	1.6	(1) 1.4: New options in the "Information" page. (2) 1.7: New command line options. (3) 2.4.3: New "Select Folder" option in "SR" page. (4) 3.8: New "Anti-noise" parameters. (5) 3.9.1: New Resolution Setting. (6) 4.1: New "pnc", "pnf", "pxc", and "pxf" format supported. (7) 4.3: New button to generate PFP file in the "Update"

		page. (8) 6.2: Remove “Log Noise Info” and combine the function into “Log Current Device Information”. (9) 7: New “Process Config” feature
30/Sep/2019	1.7	(1) 3.2.4: The Hid Report Parameters. (2) The customize panel settings. (3) 3.13: The customer specific parameters. (4) New section for generating golden samples. (5) : Descriptions of the new “Diagnostic Test” option.
20/May/2020	1.8	(1) Removed: Custom Panel Setting, Process List Golden Sample and Diagnostic Test. (2) Add: 1.4.3 DLL Version. (3) Add:1.4.4 Core Version. (4) Add: 1.4.6 Folder Configure. (5) Add: 2.4.3 Reset SR. (6) Modify: 3.2.4 Renamed to Digitizer and Mouse Mode, also show corresponding USB Product ID. (7) Add: 3.9.4 Edge Adjustment. (8) Modify: 3.8.1 Median Filter Renamed to Spike Filter. (9) Modify:3.8.3 Enhanced Frequency Hopping Renamed to Frequency Hopping Mode. (10) 4.1: Show CRC information of ND Base. (11) Add: 7.3 Skip Noise Level N Times Setting.

Table of Content

PREFACE ..... **I**

Disclaimer i

Trademarks i

Copyright i

REVISION TABLE ..... **II**

1. INTRODUCTION **9**

1.1 Supported Systems 9

1.2 Supported Devices 10

1.3 Summary 11

1.4 Checking Utility Information 11

1.4.1 Device Information 11

1.4.2 Application Version 12

1.4.3 DLL Version 12

1.4.4 Core Version 12

1.4.5 Open Default Export File Folder 12

1.4.6 Folder Configure 12

1.4.7 Log UI Parameters Information 13

1.4.8 Disable Data Report 13

1.4.9 Show Menu in Draw 13

1.5 Refresh and Reset Device 14

1.6 Special Notice 14

1.7 Command Line Features 15

2. DIAGNOSTIC **16**

2.1 Open Short Test 16

2.1.1 Test Items 17

2.1.2 Test Result & Analysis 18

2.1.3 Save Test Results 20
2.1.4 Test Log 20
2.2 Noise Level 22
2.2.1 Execute test 23
2.2.2 Test Analysis 24
2.3 3D Module of Touch Signal 25
2.4 System Reference (SR) 28
2.4.1 Checking Status 28
2.4.2 Rebuild the System Reference 28
2.4.3 Options 30
3. PARAMETERS 31
3.1 Main Features 32
3.2 Interface Parameters 33
3.2.1 Change Interface 34
3.2.2 I2C Address 34
3.2.3 Baud Rate 34
3.2.4 Hid Report 34
3.3 Panel Parameters 34
3.3.1 Choosing from a Preset 35
3.4 Finger Parameters 35
3.4.1 Sensitivity 36
3.4.2 Touch Points 36
3.4.3 Sample Rate Limit 36
3.4.4 Touch Debounce 37
3.4.5 De-jittering 38
3.5 Palm Parameters 40
3.5.1 Palm Rejection 40

3.5.2 Palm Size 40
3.5.3 Palm Detection on Edge 41
3.5.4 Disabled Range near Palm 41
3.5.5 Palm up De-bouncing 42
3.6 Base Parameters 42
3.6.1 Base Update Modes 43
3.6.2 Standard Base Update Configurations 43
3.6.3 Keep Base Update (KBU) Configurations 45
3.7 Water Handling 46
3.7.1 Choosing a Water Handling Solution 46
3.7.2 Tweaking Water Performance Mode 47
3.8 Anti-Noise Parameters 50
3.8.1 Configure Spike Noise Filtering 50
3.8.2 Enhanced Driving 50
3.8.3 Configuring Frequency Hopping 51
3.9 Coordinate Parameters 51
3.9.1 Calibration 51
3.9.2 Edge Compensation 54
3.9.3 Tail Orientation 54
3.9.4 Edge Adjustment 55
Figure 65 56
Please notice that click on the “Edge Adjustment” button several times might significantly decrease accuracy. To revert settings, please use the “Write Factory Default Parameter” button or do firmware update again. 56
3.10 Power Parameters 56
3.10.1 USB Interface Configurations 58
3.10.2 Idle Mode Settings 58

3.11 Miscellaneous Features 59
3.11.1 Watch Dog 60
3.11.2 Event Notify 60
3.12 Driving IC Parameters 61
3.12.1 High Voltage 61
3.13 Customer Parameters 62
4. FIELD UPDATE 63
4.1 Load Firmware File 63
4.2 Update Firmware 64
4.3 Generate New PFP File 65
5. DRAW 67
6. DEBUG 71
6.1 Reset Touch Panel 71
6.2 Log Current Device Information 71
6.3 Show Notifications for Water Detection 74
6.4 Touch and Halt 74
7. PROCESS CONFIG 75
7.1 Work Order Number 76
7.2 Load PFP File 76
7.3 Miscellaneous 76
7.4 Select Tests for Process List 77

1. Introduction

This document provides information for using PenMount PCIUtility to determine touch device settings and perform updates and tests.

1.1 Supported Systems

PenMount PCIUtility can operate in the operating systems listed below:

Types	Operating Systems
Desktop	Windows XP (32/64bit)
	Windows Vista (32/64bit)
	Windows 7 (32/64bit)
	Windows 8 / 8.1 (32/64bit)
	Windows 10 (32/64bit)[1]
Server	Windows Server 2003 / R2
	Windows Server 2008 / R2
	Windows Server 2012 / R2
	Windows Server 2016 / 2019
Embedded	Windows XP Embedded / Standard 2009
	Windows Embedded Standard / POSReady 7
	Windows Embedded 8 Standard / Industry
	Windows 10 IoT Enterprise

[1] Recommend using the Windows 10 Anniversary Edition 1607 (10.0.0.14393) and later versions.

1.2 Supported Devices

Devices and interfaces supported by PenMount PCIUtility are as follows:

Series	Control Board	USB	RS-232 / UART	I²C [2]
PenMount P2-02 Series	PM1100		v	
	PM2101		v	v
PenMount P2-03 Series	PM1200	v	v	
	PM1201A	v	v	
	PM2201	v	v	v
PenMount P2-04 Series	PM1300A	v		
	PM1302	v	v	v
	PM1302B			v
	PM1400A	v	v	
	PM1401	v	v	
	PM1401A	v		v
	PM1500	v	v	v
PenMount P2-06 Series	PM1110A		v	
	PM1210	v	v	v
	PM2103	v	v	v
	PM2203	v	v	v
	PM2203B	v	v	v
	PM2203C	v		
	PM2204	v	v	v
	PM2251	v	v	v
	PM2252	v	v	v
	PM2351	v		
	PM2300	v	v	v
PenMount P2-08 Series	PM1310	v	v	v
	PM2303	v	v	v
	PM1410	v	v	v
	PM1415	v		v

[2] The I²C, UART / RS-232 interface is supported by connecting though the PMT101 Adapter.

	PM1710	v	v	
	PM1711	v		v
	PM1715	v		v

1.3 Summary

The application version of PenMount PCIUtility is summarized in the table below.

Utility	Usage	Application Version	Notes
PCIUtility	Advanced firmware configuration	1.8.x	

The application version can be found on the “Information” page.

1.4 Checking Utility Information

The version of PCIUtility and other useful information are listed on the “Information” page.

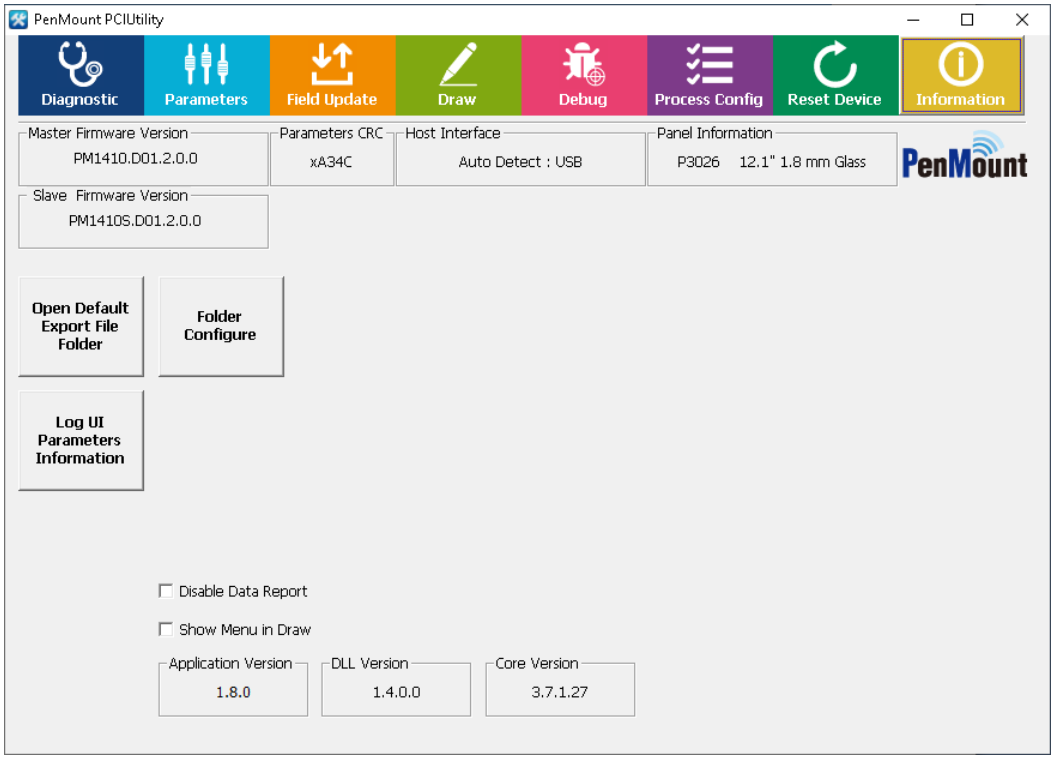


FIGURE 1

1.4.1 Device Information

Connected device information consists of several parts:

- Master Firmware / Slave Firmware / Test Version

  The firmware version of the connected device.

  If marked as “Test Version”, it means that this firmware is for testing purpose only.

- Parameters CRC

Cyclic Redundancy Check (CRC) to be used for parameter error correction

- Host Interface

  The connected interface, most P2-06 and P2-08 devices support interface auto detection.

- Panel Information

  Configuration of panel size and cover lens thickness.

1.4.2 Application Version

It is the version of the utility.

1.4.3 DLL Version

This is the version of the PM_Standard.dll file that the utility is using.

1.4.4 Core Version

It is the version of the firmware core features. Different firmware with similar core versions mean that their performance result will be very close too.

1.4.5 Open Default Export File Folder

Pressing this button will open the file manager and allow user to browse the test results, parameters, and other files generated by PCIUtility. By default, the directory is located at:

%LOCALAPPDATA%\PenMount Utility

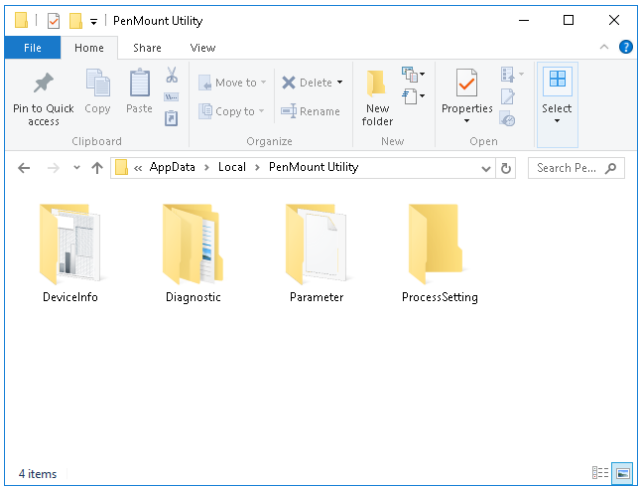


FIGURE 2

1.4.6 Folder Configure

The PCIUtility remembers the last selected directory as the default value of next operation. For example, if user loads firmware file from “D:\Firmware”, the next time user launches “Load Firmware” button, it will list firmware files in the “D:\Firmware” directory.

PCIUtility allows specifying the default folders for different actions, as shown in the below screenshot.

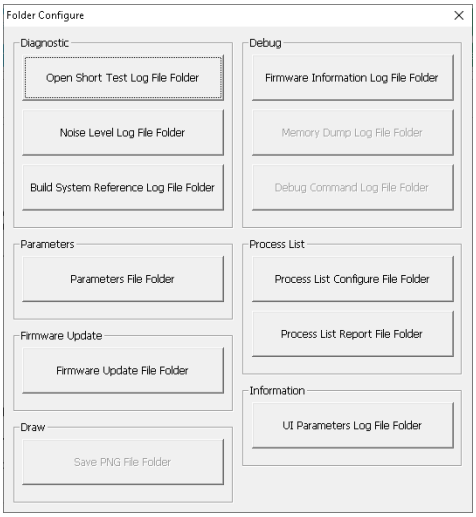


FIGURE 3

1.4.7 Log UI Parameters Information

Pressing this button will save all the parameter settings in human readable format from the connected device. This feature is useful when comparing the parameter differences of two devices with different CRC values.

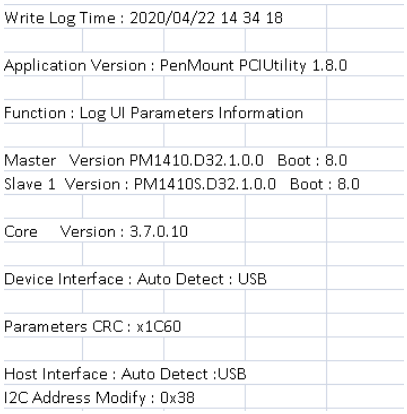


FIGURE 4

1.4.8 Disable Data Report

When this option is checked, touch will be disabled when PCIUtility is running.

1.4.9 Show Menu in Draw

When this option is checked, a menu button will appear in Draw mode. Please check chapter 5 for each option in menu.

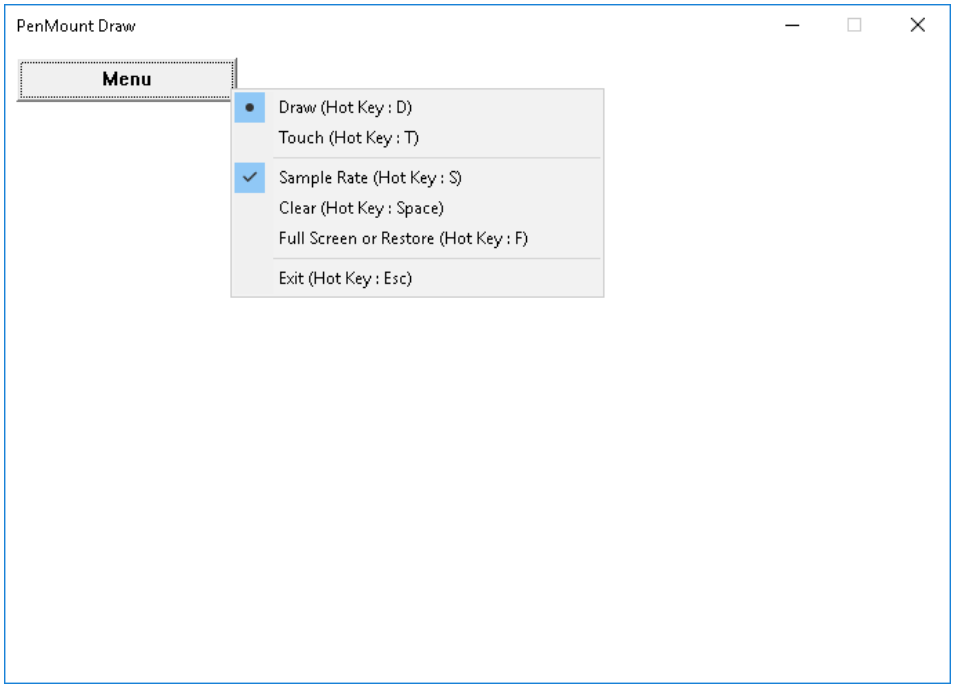


FIGURE 5

1.5 Refresh and Reset Device

If device is not connected when utility starts up, it can still automatically detect when device is attached with USB interface, or user can press the “Refresh” button for manually detecting RS-232 / UART devices.

Please notice that is no device is connected, only “Refresh”, and “Information” pages can be accessed. If the devices is connected, PCIUtility shows “Reset Device”. The user can reset device by this button.

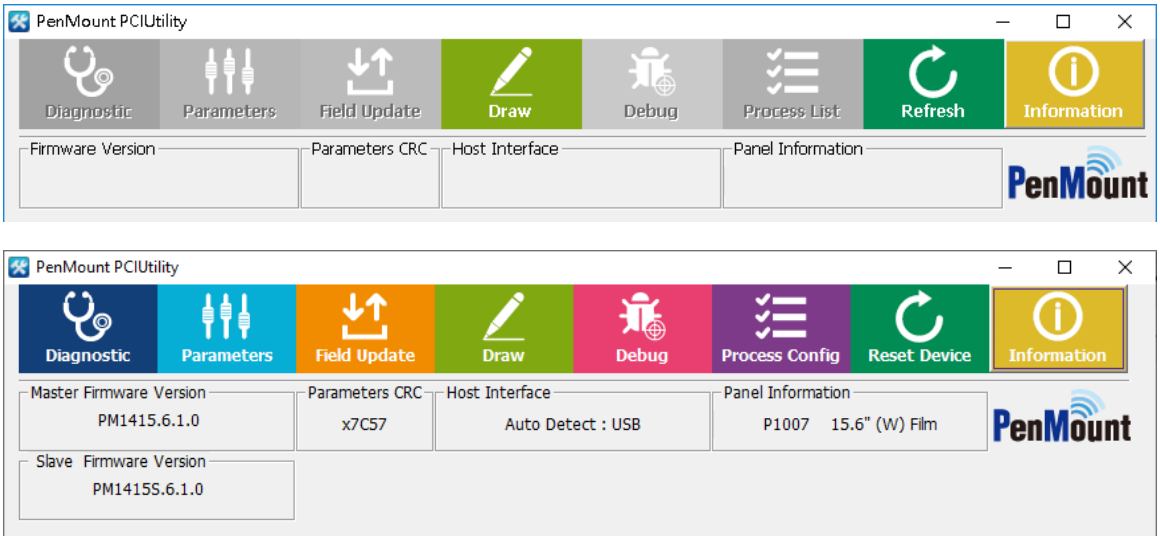


FIGURE 6

1.6 Special Notice

1. PenMount PCIUtility can only support the setting and operation of one single device at one time.

2. Not all PenMount Utilities can be opened simultaneously.

1.7 Command Line Features

This version of PCIUtility supports the following command line options. Please choose one of the following options.

Option	Description	Feature Reference
-os	Open Short Test	2.1
-nl0 **-nl1** **-nl2** **-nl3**	Noise Level Test, using pattern “NONE” Noise Level Test, using pattern “WHITE” Noise Level Test, using pattern “BLACK” Noise Level Test, using pattern “CHECKERBOARD”	2.2
-sr	Build System Reference	2.4

2. Diagnostic

PCIUtility provides standard product testing procedures, including:

- Open-Short Test: test the electrical functions of PCI touch panels
- Noise Level: detect the size and area of the PCI touchscreen influenced by noise interference
- 3D Module: graphical view for diagnosing raw signal
- System Reference: allow user to rebuild the reference data used for adjusting environmental changes. This feature is only supported in some device firmware versions.

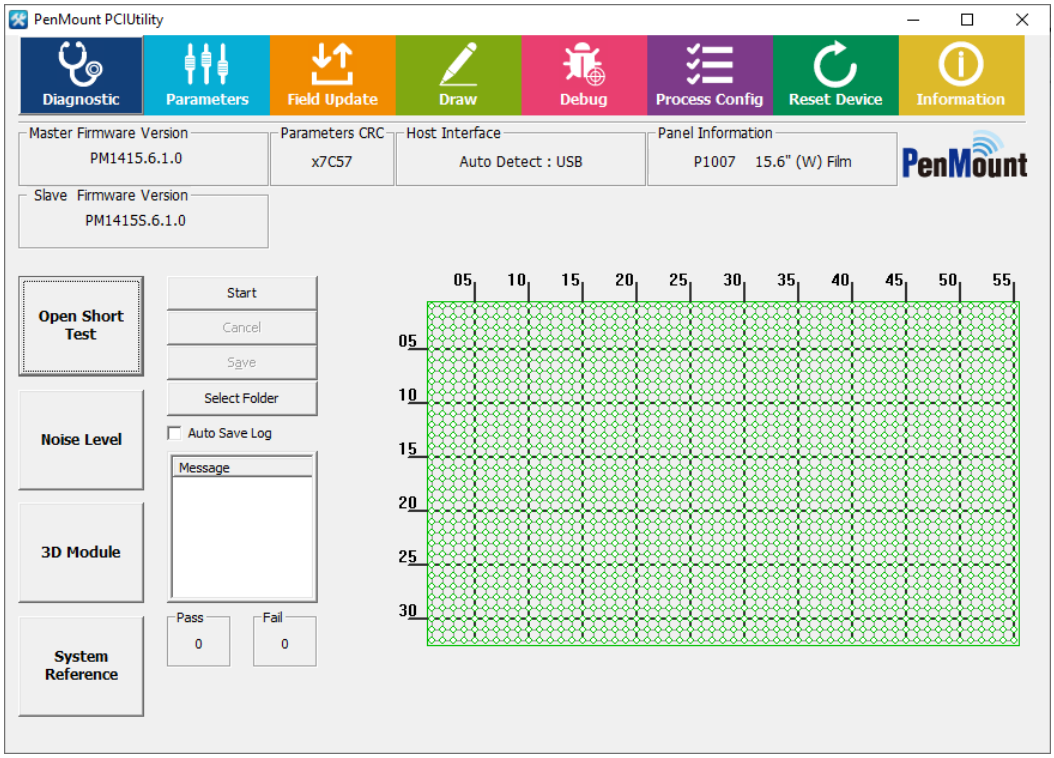


FIGURE 7

2.1 Open Short Test

The “Open Short” tests sensor and driver pins for possible open / short issues. To ensure the accuracy, please do not perform any touch action while the test is running.

Click the “Start” button; this will start electrical detection on the touchscreen. When a single test is running, status will display “START” and “progress” will rise from bottom to top.

FIGURE 8

Once detection is completed, the result will appear as the status. If the touchscreen is normal, PASS will be displayed. If it is abnormal, FAIL will be displayed. To the right side of the Status field is the “Pass & Fail” field, indicating the number of Passes or Fails after the program starts to run detection

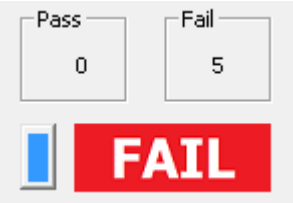


FIGURE 9

Meanwhile, the status of the touchscreen is shown at the right side of the window. Green means the node is normal and red means abnormal.

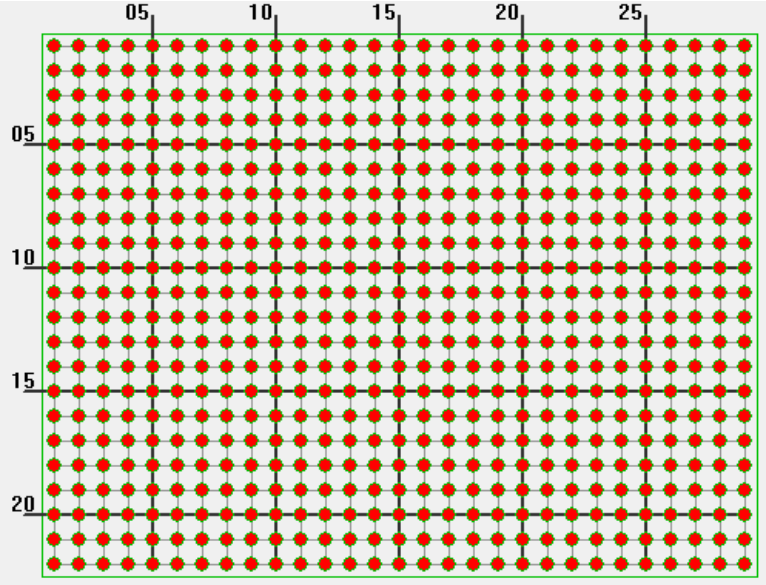


FIGURE 10

2.1.1 Test Items

When diagnosis starts, the program measures raw data from every node on touch panel. Three sets of raw data are collected for signal testing. The primary test items are as follows:

Items	Explanation
Debounce Check	If the three sets of raw data on all the nodes of the touch panels are distinctively different from one another and larger than Debounce threshold, it is determined as abnormal.

Driver Pins Abnormal Check	Test the raw data of two consecutive driving pins (use the signals for the third test) and the difference between them should not be excessively large. If the raw data of over one third of the nodes of the two neighboring driving pins is over Driver Abnormal threshold, the driving pin with higher raw data will be marked “abnormal”.
Sensor Pins Abnormal Check	Test the raw data of two consecutive sensor pins (use the signals for the third test) and the differences should not excessive. If the raw data of over one third of the nodes of the two neighboring sensor pins is larger than Sensor Abnormal threshold, the sensor pin with higher raw data will be determined as a short circuit and marked “abnormal”.
Same Raw Data Check	If the three sets of raw data on all the nodes of the touch panels are identical, it is determined as abnormal.
Illegal Data Check	Checks whether raw data is negative.
Absolute Data Check	Checks whether raw data is within valid range.
Near Check	Checks whether the raw data of two adjacent driver lines/sensor lines is similar.
Flat Check	Checks whether the base is flat.

2.1.2 Test Result & Analysis

When diagnose ends, the program displays messages for each failed test items:

FIGURE 11

The table below lists possible messages.

Message	Description
PASS	All test item passed
Debounce check fail	Debounce Check failed
Driver check fail	Driver Pins Abnormal Check failed
Sensor check fail	Sensor Pins Abnormal Check failed

Same check fail	Same Raw Data Check failed
Illogical check fail	Illegal Data Check failed
Absolute check fail	Absolute Data Check failed
Near check fail	Near line raw data check failed
Flat check fail	Base flat check failed

After the test is completed, nodes with abnormal signals will be immediately indicated. They are usually classified in three types:

1. Single point abnormal

   When the test result is “single point abnormal,” it is because the difference of the signal values for one location and the ambient nodes is too large. If the grounding for the system is not ideal, which may lead to touch instability, abnormality will be tested at different nodes during each test.

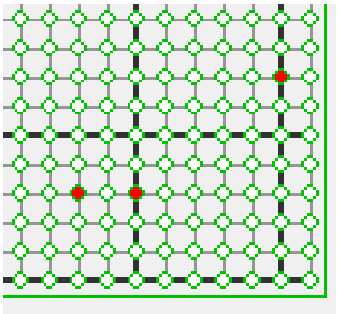


FIGURE 12

2. Whole pin abnormal

   When the test result is “Whole pin abnormal,” it is usually either due to the open state of sensor/driving pin or too many abnormal nodes on the same pin.

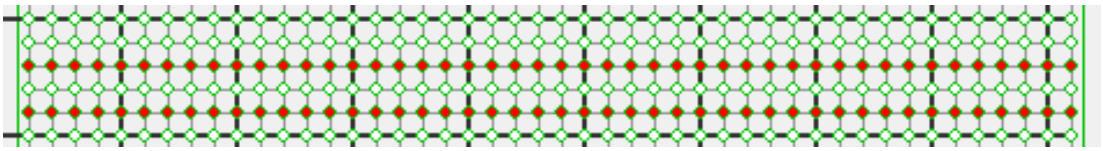


FIGURE 13

3. Whole panel abnormal

   When the test result shows whole panel abnormal, it is usually because the signals detected never change during the process. When you are connecting the touch panel with the control board, if the cables are not connected properly, this will happen.

In some circumstances, it is hard to see where the problem is from the appearance of the product. You can run PCIScan and cross reference to determine whether the source of the problem is the control board or the touch panel.

2.1.3 Save Test Results

If the touch panel shows “signal abnormal,” you can save the results into a file and provide it to PenMount for further analysis. You can click “Save” after the test is completed. The “Select Folder” button allows changing of the target folder for saving test results.

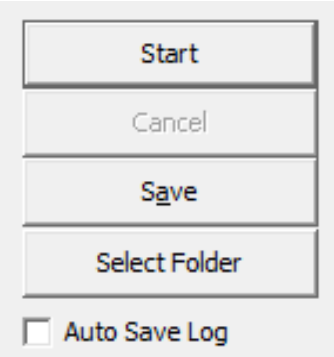


FIGURE 14

A dialogue box below will appear. The user must enter “Batch Number 1” and click OK to save. We suggest using the touch panel number as the “Batch Number” for future identification. The program will save the test results either in the “TestPass” or “TestFail” folder based on the test results under $ProductNumber_$BatchNumber_$Month$Day$Hour$Minute$Second”.csv.

- Auto Save Log

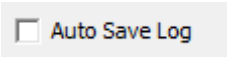


FIGURE 15

With this option checked, the “Save File” dialog will automatically appear.

2.1.4 Test Log

The program will save the detected signal in CSV file. The format of CSV is a text file, which can be opened with spreadsheet software such as Microsoft Excel.;

- Basic data

  The beginning of the CSV file records data for the corresponding touch panel and controller. You can also see the threshold value for the test.

1	
2	Write Log Time : 2020/05/19 12:06:33
3	
4	Application Version : PenMount PCIUtility 1.8.0
5	
6	DLL Version : 1.4.0.0
7	
8	Function : Open Short Test
9	
10	Master Version : PM1410.D01.2.0.0 Boot : 8.0
11	
12	Slave Version : PM1410S.D01.2.0.0 Boot : 8.0
13	
14	Core Version : 3.7.1.27
15	
16	ND Base CRC : xE33C
17	
18	Parameters CRC : xA34C

FIGURE 16

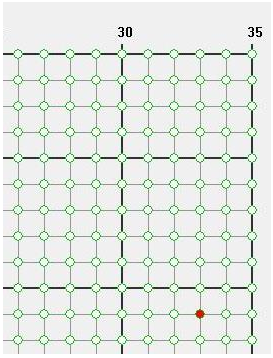
- RawData

The CSV contains several sets of raw data which are the basis for judgment for each test.

[RawData]							
0	1	2	3	4	5	6	7
1	493	498	499	503	501	503	509
2	511	526	525	528	527	528	528
3	514	528	528	530	531	533	533
4	508	519	523	525	526	528	529
5	501	509	510	512	519	519	519
6	516	528	534	534	531	533	533
7	517	528	528	531	532	532	539
8	512	519	523	524	521	527	526
9	518	525	527	528	530	530	533
10	520	529	529	527	530	531	538
11	519	531	535	537	535	533	538
12	514	528	525	529	529	529	534
13	516	524	527	527	530	532	539
14	525	536	537	537	538	540	553
15	534	535	540	538	542	541	558
16	523	537	534	534	534	534	536
17	528	534	536	538	540	538	545
18	530	538	540	539	541	541	546

FIGURE 17

Raw data is arranged in a two-dimensional array, which corresponds to the nodes on the sensor and the driver pin. The nodes with * are abnormal. From the raw data, you can presume the reason for the abnormality. In the example below, there is obviously less raw data coming from the nodes marked abnormal than the surrounding nodes.



AB	AC	AD	AE	AF	AG	AH	AI	AJ
27	28	29	30	31	32	33	34	35
1926	1925	1925	1926	1926	1928	1926	1925	1913
1923	1924	1922	1924	1925	1925	1923	1918	1921
1922	1923	1920	1924	1923	1922	1924	1926	1923
1926	1926	1926	1924	1926	1926	1926	1926	1926
1922	1922	1921	1921	1921	1921	1921	1924	1922
1925	1926	1926	1926	1927	1926	1926	1926	1925
1920	1922	1922	1922	1924	1924	1924	1925	1921
1924	1925	1925	1924	1926	1926	1926	1926	1925
1926	1926	1926	1926	1926	1926	1926	1926	1926
1922	1921	1922	1922	1921	1919	1922	1925	1921
1920	1922	1924	1923	1924	1925	*1910	1922	1923
1921	1923	1922	1923	1923	1921	1914	1914	1923

Figure 18

- PDData

  This is the raw data used during the “Illegal Data Check”.

- SectData

  This is the sector scan result. For reference only.

- SelfData

  This is the abnormal scan result. For reference only.

- Raw Data Near

  This is to check total delta of adjacent sensor and driver.

- Base Flat

  This checks whether base is sufficiently flat.

2.2 Noise Level

The Noise Test mainly evaluates the degree of noise interference impact on the touch panel. Usually noise will affect measurement and large fluctuations will be produced. The program will display the “noise level” interface:

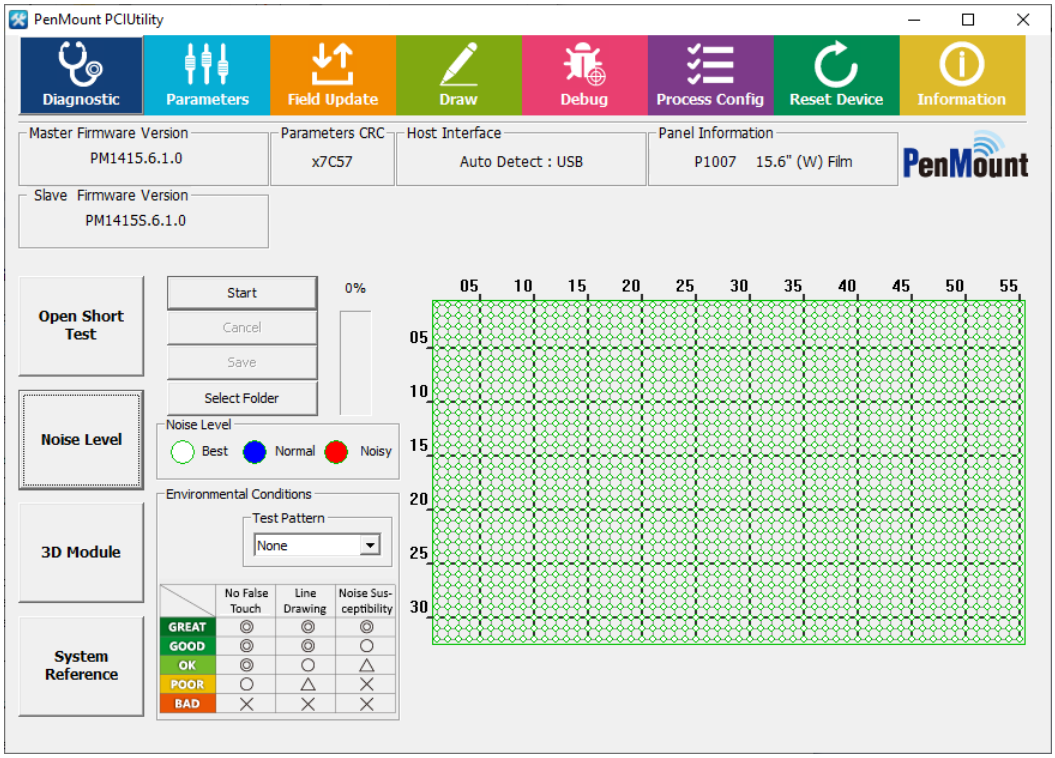


FIGURE 19

2.2.1 Execute test

Please click “Start” on the left side of the window to start running the test. To ensure test accuracy, please avoid any touch input during the test.

During this period of time, progress will be displayed in the window caption. All other utility features will be temporarily disabled during the test. Usually the test takes longer when using larger panel sizes. Please click “Cancel” if want to abort to sue other features.

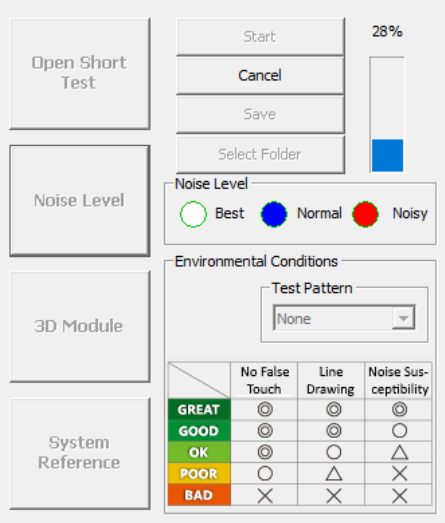


FIGURE 20

For testing LCD noise interference with specific display patterns, please change “Test Pattern”. Currently “White”, “Black”, and “Checkerboard” are supported. Please be aware that after pressing the “Start” button, the LCD will display the specified pattern in full screen mode until the test finishes.

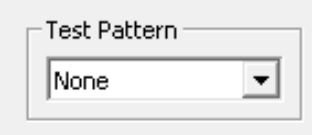


FIGURE 21

2.2.2 Test Analysis

According to the standard deviation calculated from the raw data, the program will be categorized nodes on the touch panel as “Best”, “Normal,” and “Noisy” and mark them respectively in white, blue, and red.

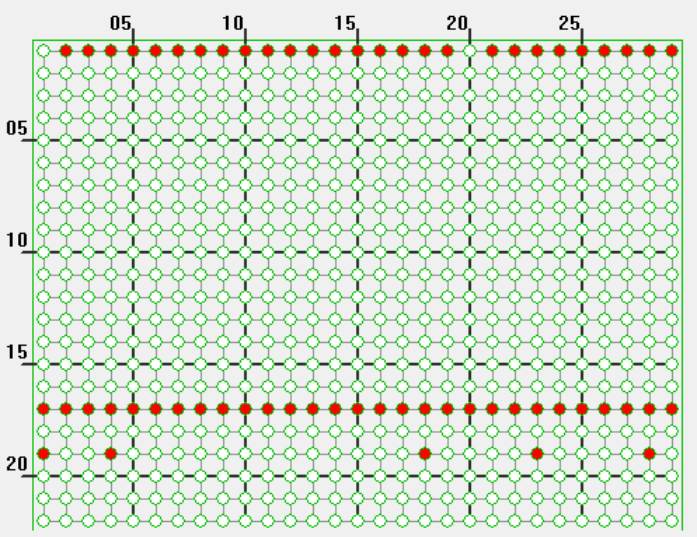


FIGURE 22

The right side of the program indicates the degree of signal changes in the nodes on the touch panel.

Nodes marked with a red circle indicate a greater change in the signal value measured and it may be unstable.

To give the user a better idea of overall signal changes, the program will analyze and show the environmental criteria, and classified in 5 levels: “GREAT”, “GOOD”, “OK”, “POOR”, and “BAD”.

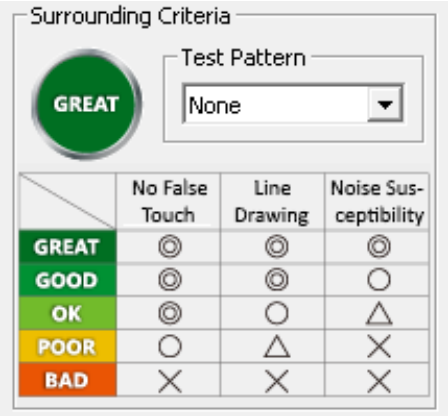


FIGURE 23

Generally speaking, please try to lower noise interference by increasing the air gap between touch panel and LCD, bring environmental conditions to a least “OK” level.

2.3 3D Module of Touch Signal

Pressing the “3D Module” button launches the 3D module for inspecting touch or noise signals. Please prepare a mouse or keyboard before launching this feature, because touch will be disabled during the entire process.

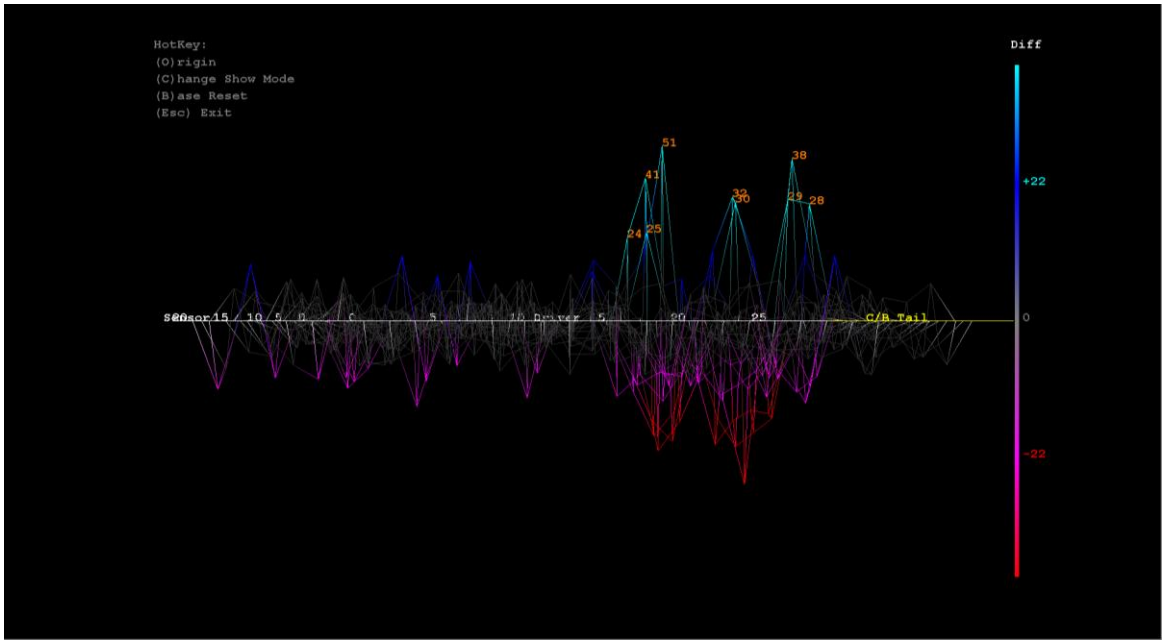


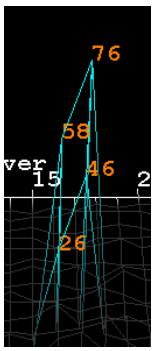
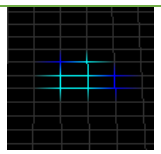
FIGURE 24

The program will use different colors to show the signal strength.

Colors	Signals
Cyan	Larger positive signals, possibly identified as finger touch
Blue	Smaller positive signals, usually caused by environmental noise
White	Normal signal and No touch
Magenta	Smaller negative signals, usually caused by environmental noise
Red	Larger negative signals, possibly caused by water, etc.

The 3D module supports several hot keys listed in the table below.

Colors	Actions
top	5 degree X axis counterclockwise rotation
down	5 degree X axis clockwise rotation

<table>
<tr><td>left</td><td>5 degree Y axis counterclockwise rotation</td></tr>
<tr><td>right</td><td>5 degree Y axis clockwise rotation</td></tr>
<tr><td>+</td><td>Zoom in</td></tr>
<tr><td>-</td><td>Zoom out</td></tr>
<tr><td>O</td><td>Restore to default angle: 225 degree X axis counterclockwise rotation, 270 degree Z axis clockwise rotation</td></tr>
<tr><td>B</td><td>Reinitialize the base signal</td></tr>
<tr><td>C</td><td>Change displayed mode.
<table>
<tr><th colspan="2">Mode</th></tr>
<tr><td>Mode 1</td><td> 
</td></tr>
<tr><td>Mode 2</td><td>  </td></tr>
</table></td></tr>
</table>

2.4 System Reference (SR)

The “System Reference” button is enabled if device supports this feature.

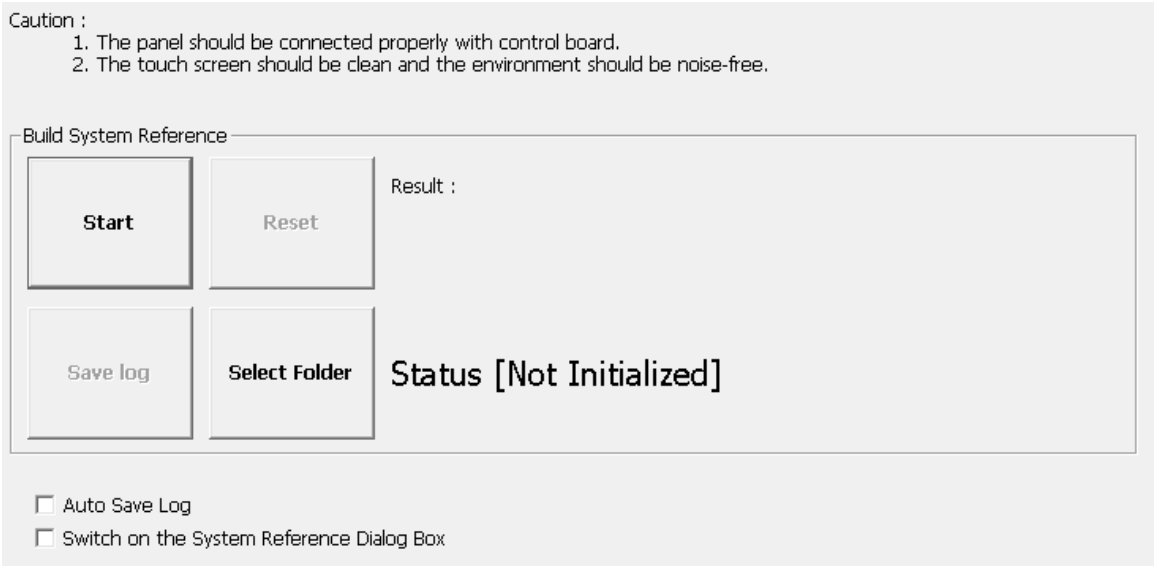


FIGURE 25

2.4.1 Checking Status

The utility shows the current status on screen, and can be one of the following:

Status	Description
Active	The system reference is built and in use
Not Initialized	The device does not have a valid system reference. Please press start to rebuild it.

In the following cases, the utility will check the reference status and prompt rebuild a new set of reference data.

- A new touch controller is connected.
- Firmware update is performed.
- The setting of touch panel size or cover lens thickness has changed.

2.4.2 Rebuild the System Reference

Before proceeding to rebuilding the system reference, please check the following:

1. The touch panel is connected to control board properly.

2. The touch screen surface is clean, and free of objects such as water, hands, fingers, etc…
3. The build environment is noise free (check using 3D module).

The rebuild process should take a few seconds after pressing the “Start” button, please wait until the progress bar proceeds to 100%.

FIGURE 26

If the touch panel is not connected properly, or having environmental interference, the utility will display a warning message after build completes. User must confirm whether to accept the build result by pressing the Yes or No button.

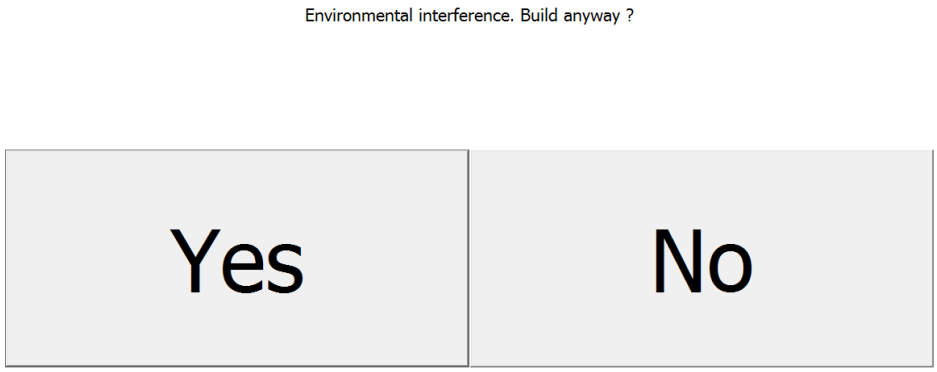


FIGURE 27

Utility will display a “FAIL” message if user presses the “No” button, and utility will roll back to previous settings.

2.4.3 Options

There are several options that user can adjust:

- Reset

Select this will reset the System Reference in device to an initialized state.

- Select Folder

Select the destination folder that log file will be saved to.

- Save Log

This option is available only after SR is built. It will save the SR status to the specified directory.

- Auto Save Log

With this option checked, the “Save File” dialog will automatically appear after the SR construction operation finishes.

- Switch on the System Reference Dialog Box

When this option is checked, the PCIUtility will notify user for SR construction when it detects that SR is not initialized.

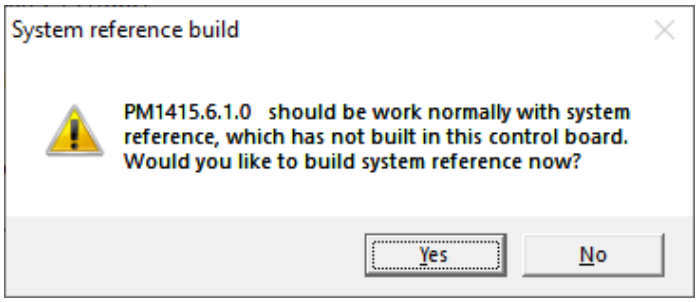


FIGURE 28

3. Parameters

Parameters provide an interface to flexibly fine tune PenMount firmware settings.

FIGURE 29

- The “Apply” Button

  Applied all the parameter changes. Any unapplied change will be lost when closing the utility.

- The “Export Parameters” Button

  Export the firmware parameter values into a text file with “.ini” extension.

- The “Import Parameters” Button

  Import the firmware parameter values from a text file with “.ini” extension. Only parameter files for the same firmware version can be imported.

- The “Write Factory Default Parameters” Button

  Reload the firmware parameters to factory default. However, the parameter of host interface and panel size will not be changed.

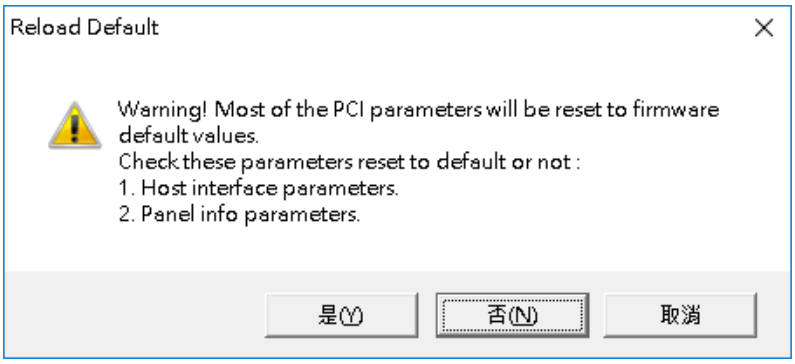


FIGURE 30

3.1 Main Features

The Parameters configuration interface includes the following features.

1. The Parameters Panel

Parameters are categorized into different tabs. For example, parameters related to anti-noise layout can be found in the “Ant-Noise” tab. When adjusting a parameter, its features and side effects will aooear for quick reference.

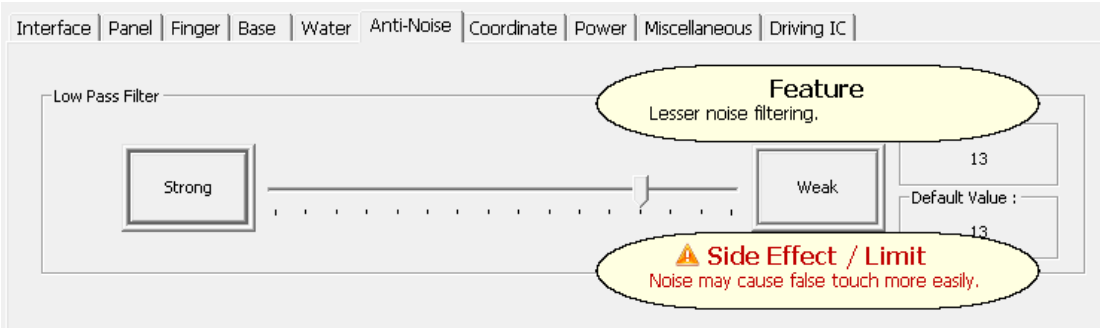


FIGURE 31

- Grayed out Options

Any parameter in gray is not adjustable. In the following example, “Change Interface” is not adjustable, because the firmware supports interface auto detection, so there is no need to manually assign the connection interface.

FIGURE 32

- Hidden Options

Any parameter that is not displayed also means that it is not adjustable. In the following example, when using older firmware versions such as PM1410 V2.1, the utility only displays fewer tabs with a few adjustable options.

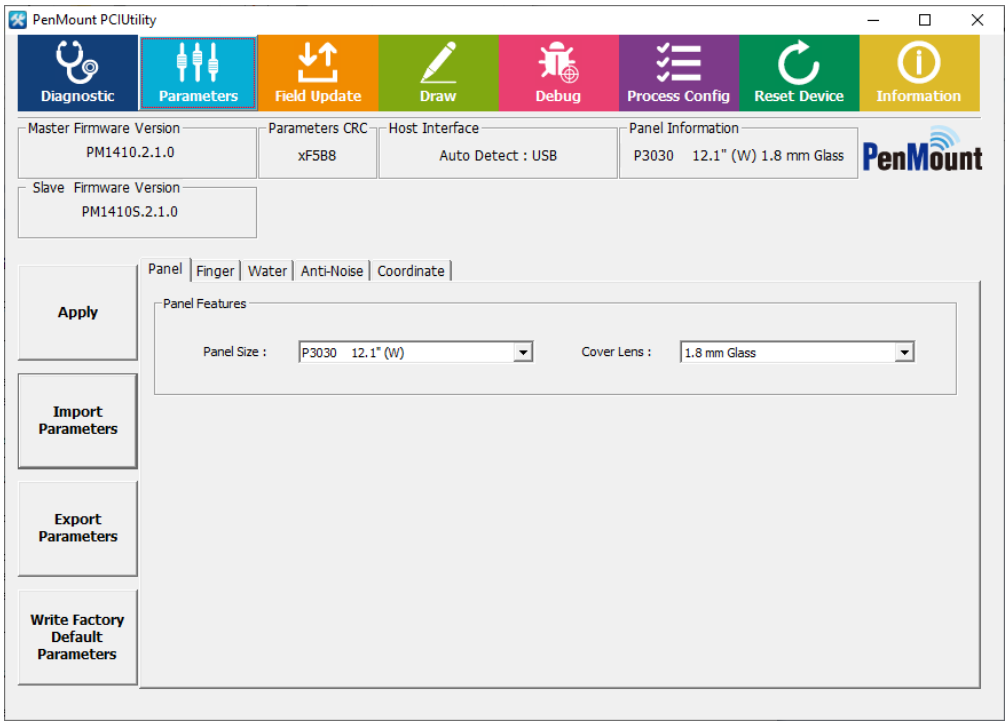


FIGURE 33

3.2 Interface Parameters

Users can change interface settings in the “Interface” tab.

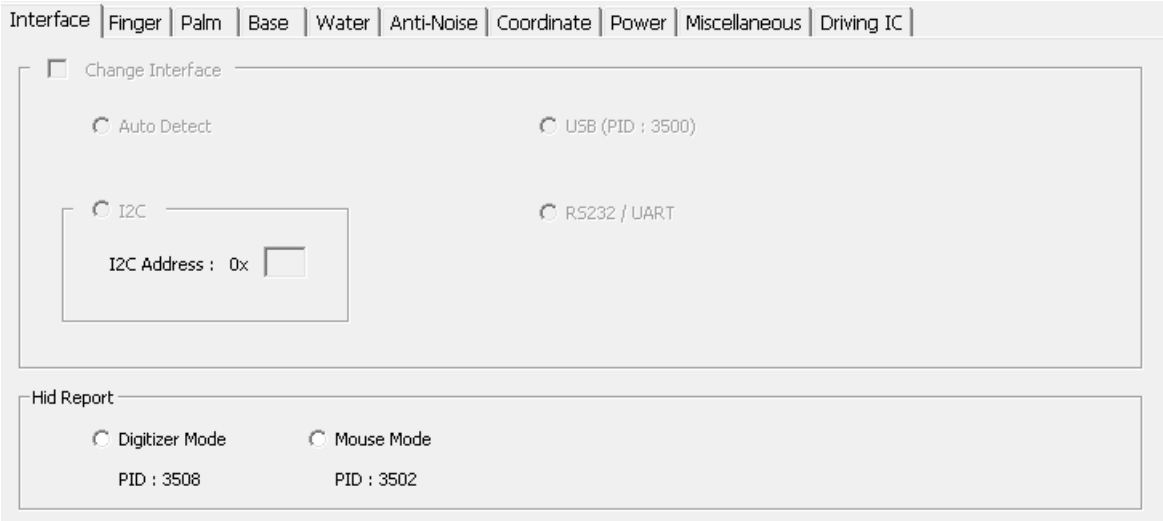


FIGURE 34

3.2.1 Change Interface

User can manually specify the connection interface when using PenMount COF products. For control boards, since the firmware can automatically detect connection interfaces, this field is not adjustable.

3.2.2 I2C Address

The slave address when using I^2C connection. Please change this value if the default slave address 0x38 conflicts with other devices.

Please note for P2-08 devices, this feature is adjustable only with boot loader version 6.4 and later.

3.2.3 Baud Rate

The transfer baud rate is used for UART/RS-232 connection.

For compatibility with PenMount device drivers, meanwhile PCIUtility does not allow changing baud rate.

3.2.4 Hid Report

Some firmware versions support using different Hid Report type. If these options are disabled, it means that the connected device firmware does not support this feature.

	Digitizer Mode	Mouse Mode
USB Product ID	0x3508	0x3502
Gesture Support	YES	NO
Windows XP	NO	YES
Windows 7 / 8 / 10	YES	YES

3.3 Panel Parameters

The panel tab allows users to specify the desired panel size, either from a preset from manually setting up the skipped driver / sensor pins.

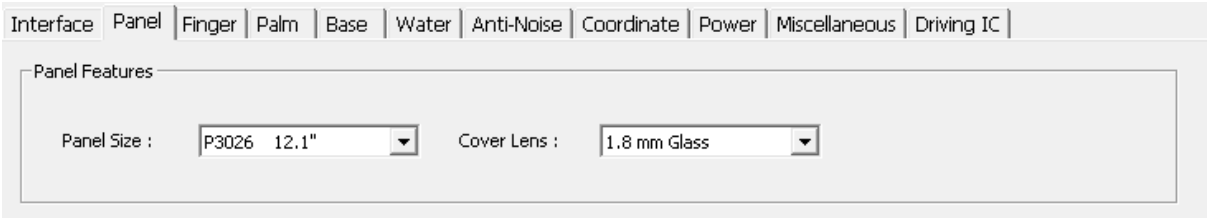


FIGURE 35

3.3.1 Choosing from a Preset

Standard PenMount firmware versions support a number of panel sizes and cover lens thicknesses. By selecting a preset, firmware will apply different configurations to parameters.

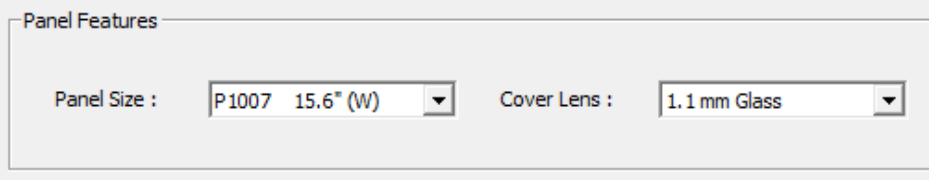


FIGURE 36

1. Panel Size

   Each PenMount firmware supports several panel size presets. For example, choosing the 'P3030 12.1" (W)' will setup firmware parameters to support the AMT P3030 12.1-inch touch panel.

2. Cover Lens

   Each PenMount firmware supports several cover lens thickness parameters, with the following standard thicknesses: 1.1mm, 1.8mm, 2.8mm, and 4.0mm. Please choose the nearest thickness from the list that most closely matches your cover lens.

3.4 Finger Parameters

The parameters in this tab are for tuning finger detection.

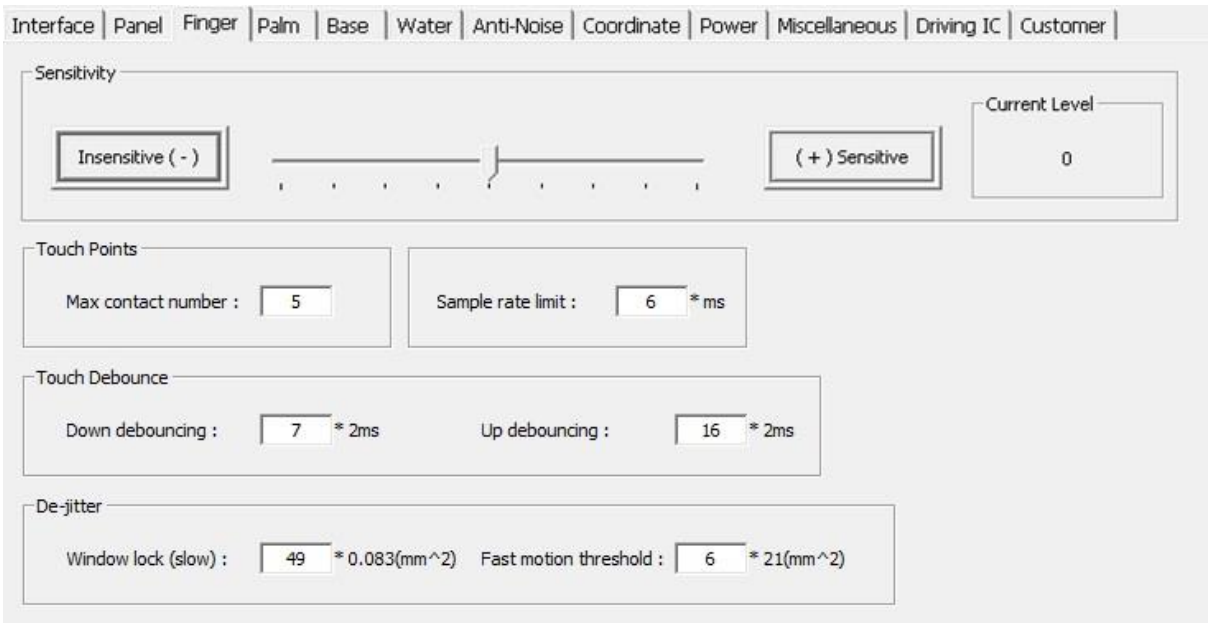


FIGURE 37

3.4.1 Sensitivity

Touch sensitivity can be adjusted with a slider. User can either use the slider or press the “Insensitive” and “Sensitive” buttons. By default, sensitivity level is 0, setting to positive values increases sensitivity and setting to negative values decreases sensitivity.

3.4.2 Touch Points

The maximum contact number limits how the number touch contacts reported to the system. For example, if the value is 1, the device supports single touch only, and other touch signals will not be sent to the system. This value will also be displayed in the Windows “System” Control Panel item.

For standard PenMount firmware, this value is adjustable. But the maximum value range is still limited by the firmware default. For example, if the firmware can support a maximum of 5 contacts, users cannot set this value to 10.

3.4.3 Sample Rate Limit

By default, the sample (or reporting) rate is above 100 points per second for each finger, which guarantees a better touch experience. However, if used on slower systems, such as ARM, the system may not be able to handle so much data at one time, and data loss may result.

In this case, users can adjust the parameter value to lower the sample rate. The parameter generally makes the firmware sample touch data a fixed rate and the reporting rate for single touch can be approximately calculated with the equation below:

$$Estimated\ Reporting\ Rate = \frac{1000}{Sample\ Rate\ Limit\ Value}$$

For example, using parameter “10” means that the firmware will send touch data every 10ms, so ideally the reporting rate is about 100 points per second.

Please note that, although the value range can be 0 to 255, we recommend keeping the value under 20. Higher values mean more overheads. Since the firmware could have completed all necessary measurements, but it still needs to wait until the specified time ends to report the result.

3.4.4 Touch Debounce

The “debounce” parameters are used to filter out sudden touch state transitions caused by noise or unstable signals.

Basically, the firmware compares raw data value with a baseline value and, if their difference rises above a predefined threshold, it indicates that the touch panel could be in an active state. However, this can also be caused by noise interference, so that the firmware needs to wait for a certain period to ensure that the signal is stable. The same situation applies when the value differences drops under the threshold.

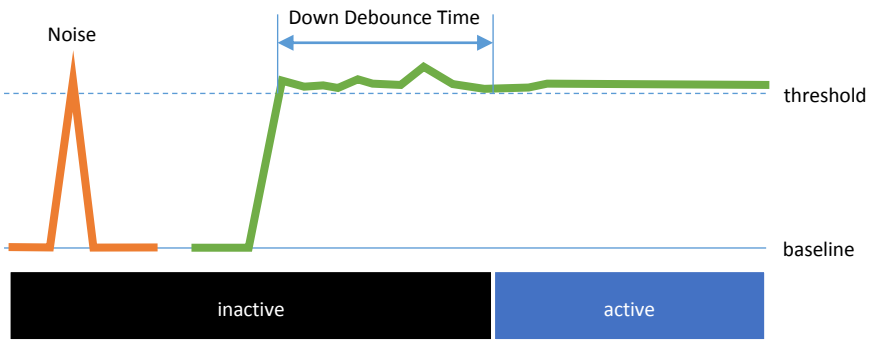


FIGURE 38

In some situations, the “Up Debounce” parameter could have a side effect that drops finger up events. This could happen when double clicking quickly, when the finger up period is so short that the second finger down event comes before the “Up Debounce Time” ends. In this case, users can lower the “Up debounce” parameter to optimize for double click operations.

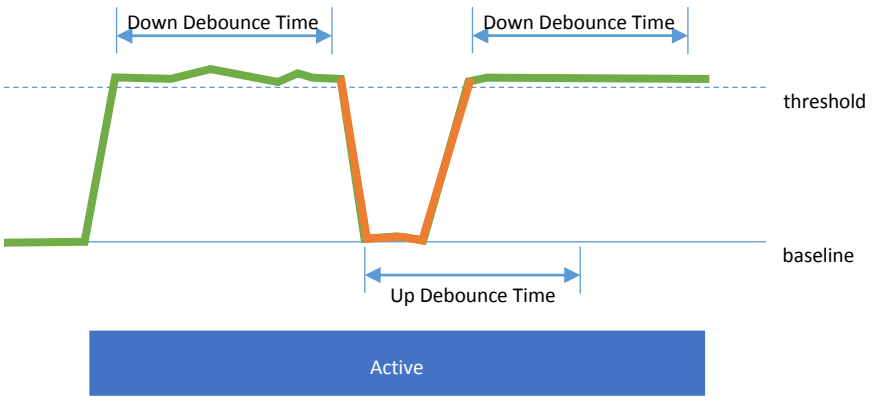


FIGURE 39

3.4.5 De-jittering

The PenMount firmware performs several smoothing processes which transforms jittery raw data into a smoothed result.

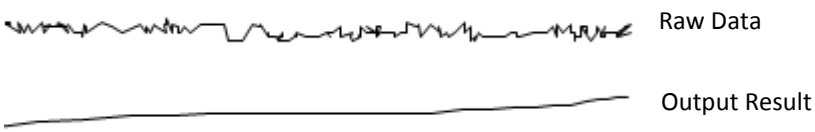


FIGURE 40

The “Window Lock” parameters are adjustable to control output smoothness.

1. Window Lock (slow)

This parameter is used to avoid jitter within small movements. Any movement that is inside the range will be *locked* to the previous touch position. The parameter value is calculated as follows:

$$\text{Parameter Value} = (Lock\ Range)^2$$

For example, if parameter value is 49, it means that any movement less than 7 will be locked. The movement distance is calculated as:

$$dist^2 = dx^2 + dy^2$$

In the formula above, dx and dy each represents the offset on the x and y axis, and the resolution is the “pitch resolution” * X lines and “pitch resolution” * Y lines respectively[3].

[3] The“pitch resolution” can be 16 for PenMount PM171x controller, and 32 for other P2-06 or P2-08 devices.

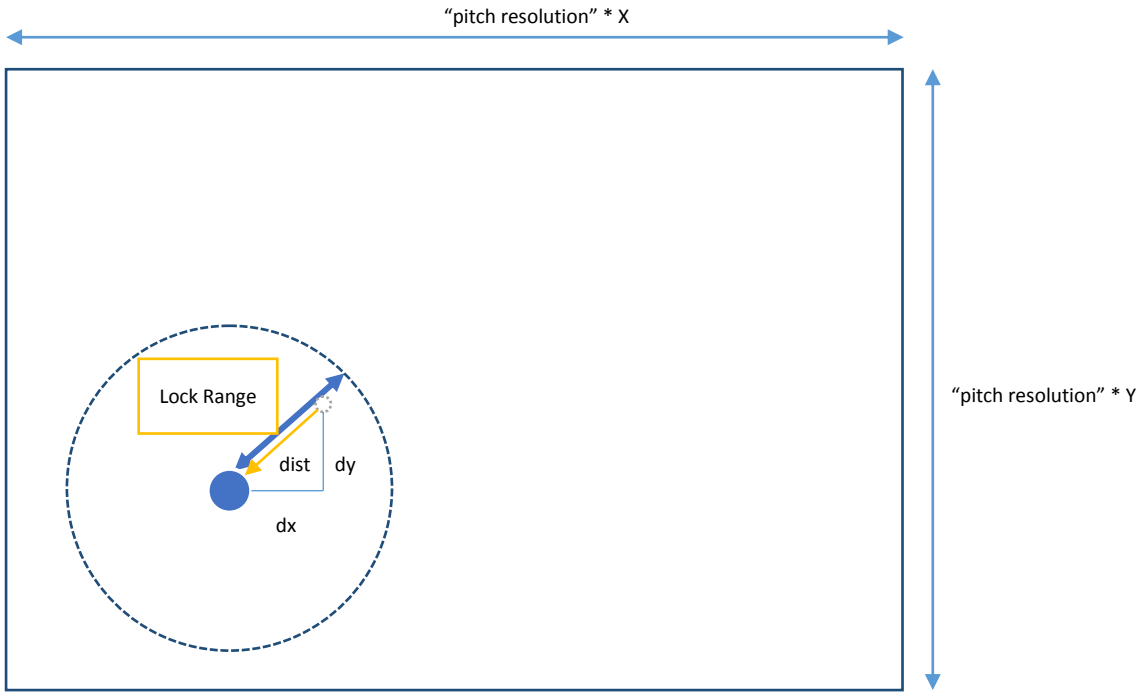


FIGURE 41

The parameter value can be set from 0 to 255. Larger values help avoid jitter in noisier environment but will also make it less easy to draw small objects.

2. Fast Motion Threshold

This is used to help adjust the line smoothing algorithm when a finger is moving swiftly.

For PenMount firmware, it should reference several previous points to calculate the current point position and make the output line as smooth as possible. In practice, it will need to reference more points when a finger is moving slowly and less points when a finger moving fast. The table below lists some presets used by PenMount:

Movement	Range	Points used for smoothing
Slow	<= Window Lock (Slow)	Window Locked
Normal	> Window Lock (Slow) <= Fast Motion Threshold	10
Fast	> Fast Motion Threshold	4

The more points used for smoothing, the smoother the output line can be. However, the output points may be some distance away from the actual touch position, especially during fast movement. To balance these two considerations, users can adjust this parameter to control when the PenMount firmware will treat a movement as “fast” or “normal”.

The parameter value is calculated as below:

$$\text{Parameter Value} * 256 = (\ Fast\ Range\ )^2$$

The DIP_RESO[4] value defines the unit. If the parameter value is 9, when moving a distance larger than 48 the firmware will treat it as a fast movement and use 4 points to perform line smoothing.

3.5 Palm Parameters

If the PenMount firmware supports palm rejection, there will be a “Palm” tab in the PCI Parameter utility. Palm rejection is used to detect and ignore large contact areas.

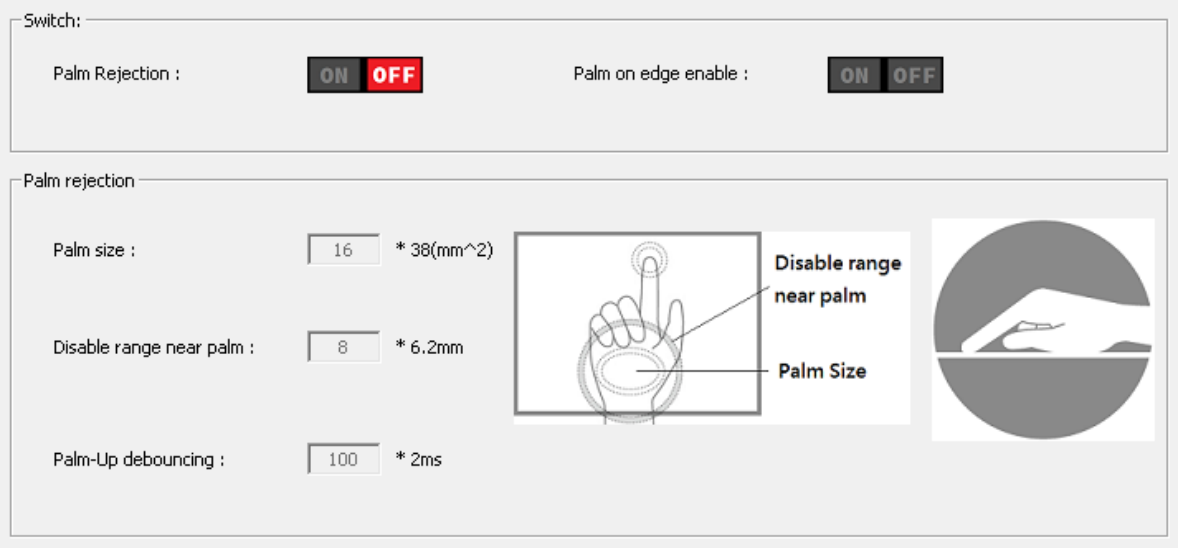


FIGURE 42

3.5.1 Palm Rejection

This is the overall switch for the firmware palm rejection feature. When this is turned OFF, all other palm features will be grayed out and a palm on the touch panel will be scanned and reported as touch event.

3.5.2 Palm Size

The palm “size” is the number of nodes on the touch panel are covered by a palm. The PenMount firmware will ignore signals within a “palm” object.

---

[4] The DIP_RESO value can be 4094 when using firmware V6.1 USB interface, or 2047 for any other case.

-5	-6	-11	-11	-12	-13	-18	-14	-12	-11	-13	-8	-14	-6
-4	-6	-7	-9	-13	-13	-9	-9	-12	-5	-7	-10	-7	-5
-11	-4	-7	-13	-15	-13	-13	-11	-6	4	3	-7	-9	-8
-6	-5	-2	-10	-7	-9	-20	-3	35	57	42	-3	-8	-6
-6	-3	14	28	7	-4	-10	25	55	62	53	12	-4	-24
-18	-5	48	74	63	39	43	43	26	23	28	23	-13	-14
-2	-5	3	27	41	52	58	64	36	23	34	17	-14	-6
0	-17	-5	-18	-11	39	73	57	52	48	48	5	-18	-15
-18	-8	-20	-24	-26	-14	25	48	55	51	52	7	-15	-12
-15	-17	-16	-30	-26	-17	-21	-12	18	54	59	17	-10	-13
-14	-14	-21	-19	-18	-19	-22	-19	-14	-5	-3	3	-19	-18
-10	-12	-11	-16	-18	-17	-17	-24	-23	-24	-17	-6	-5	-14

FIGURE 43

To tune palm size, please count the nodes that will be covered by palm-approximately. Using a value smaller than the palm size can cause firmware detect palm objects faster. However, the lower this value, the easier an object is treated as a “palm.” This will also increase the chance of misjudgment when touching the panel with number of “thick fingers”.

3.5.3 Palm Detection on Edge

PenMount firmware additionally supports ignoring unwanted touch signals generated by palms holding onto the edges of a touch panel.

After enabling “Palm on edge enable”, the firmware will be checking any newly detected objects on the panel edge and will ignored it if its size is larger than a predefined threshold.

The threshold value used to detect large objects on edge is different from the “palm size” parameter in 3.5.1 because the object can be either a palm or finger. Currently the PCI Parameter does not support changing the threshold value.

0	-1	2	0	2	-1	4
-1	-2	2	8	9	6	1
1	4	2	26	67	24	7
4	-2	-1	52	104	97	36
0	4	3	26	92	107	85
1	0	2	2	18	37	68
-2	0	-1	3	2	8	11
0	2	2	-1	0	1	4

FIGURE 44

3.5.4 Disabled Range near Palm

To help avoid false activation by signals originating from a palm, PenMount firmware also ignores any nearby finger touches.

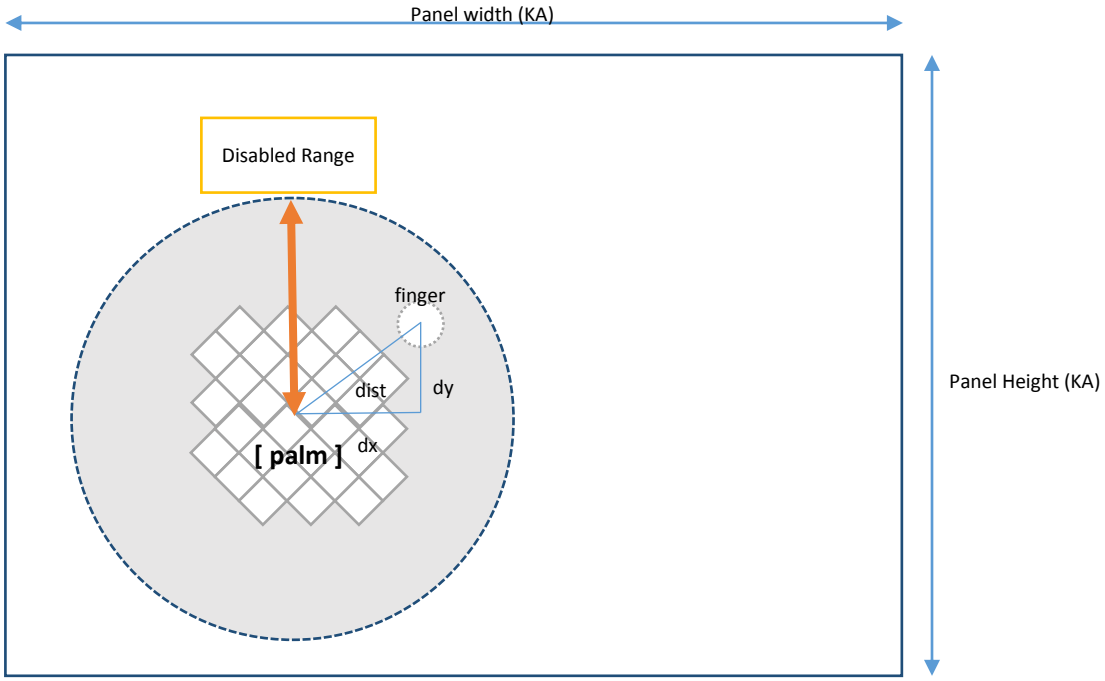


FIGURE 45

The “disabled range” can be calculated as follows:

$$Disabled\ range = \text{Parameter Value} * \text{Sensor Pitch}$$

Any finger at a distance from the center of the palm that is less than this range will be ignored.

3.5.5 Palm up De-bouncing

When a “palm” object is about to be removed from the touch panel surface, its decreasing size could also make it small enough to look like a normal finger touch. Without setting up a de-bouncing time for this, the firmware could send some unwanted touch data.

The “up-debouncing” time used for palms is different from the one introduced in section 3.4.4, since it will need more time to ensure that the palm object is completely removed. Please keep in mind, do not set this value too high because the palm area will be unresponsive to finger touches until the up de-bouncing ends.

3.6 Base Parameters

Base signals are the most important data for correctly detecting finger contacts. The PenMount firmware retrieves the base signal for the whole touch panel during initialization and updates the base value from time to time. The update should be configured to be frequent enough to catch up to environment changes but not be too frequent to affect the reporting rate.

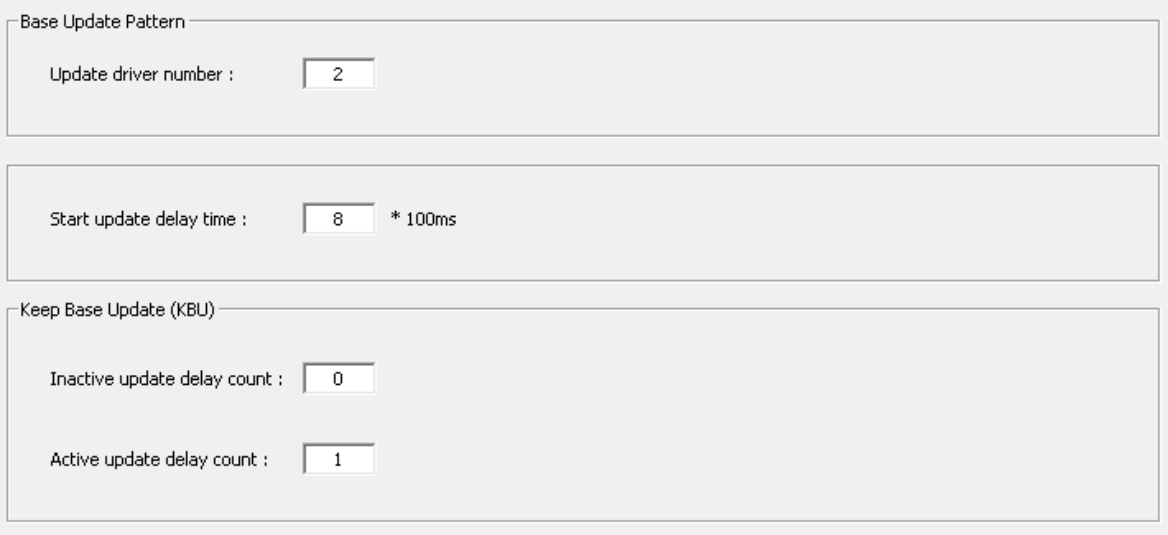


FIGURE 46

3.6.1 Base Update Modes

The PenMount firmware will use one of the two supported update modes: standard mode or keep base update mode, depending on whether the water handling feature is enabled.

Water Handling Feature	Base Update Mode Used
Enabled	Standard Update Mode
Disabled	Keep Base Update (KBU) Mode

3.6.2 Standard Base Update Configurations

When the PenMount firmware uses standard base update mode, the base update process is shown in the flowchart below.

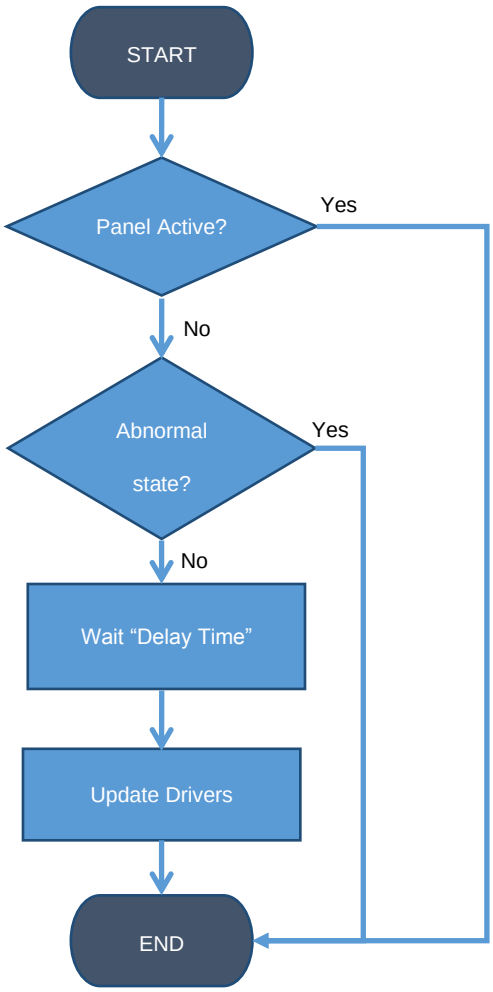


FIGURE 47

Water can be one of the causes of “abnormal state” and base update will not be performed until it is removed.

1. Start Update Delay Time

   The delay time (in 100ms unit) for PenMount firmware to perform base update after all finger contacts are removed and touch panel becomes inactive.

2. Update Driver Number

   By design, the PenMount firmware performs base update incrementally, only updating nodes on several driver pins each time. The number of driver pins to be updated is configurable from 1 to 16.

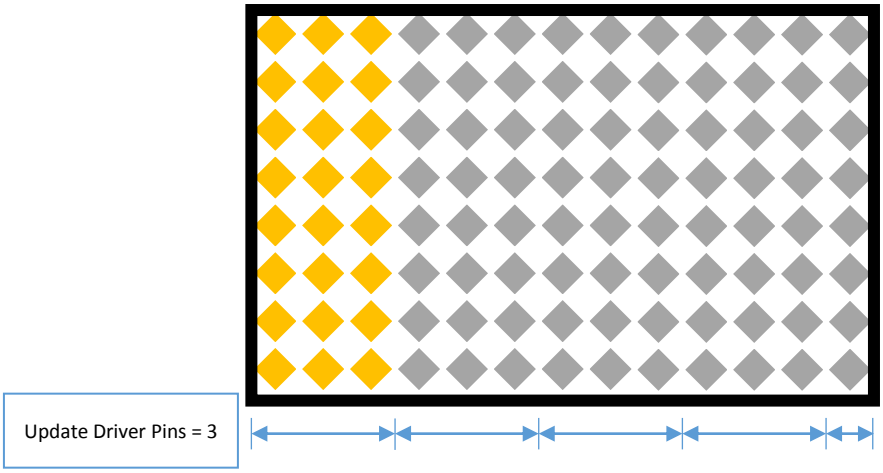


FIGURE 48

Updating more driver pins at one time may facilitate water handling or noise suppression. However, since updating more nodes means taking more time to finish, it might affect finger down response latency. In this case, please consider using smaller values for better response time.

3.6.3 Keep Base Update (KBU) Configurations

If water features are disabled, the PenMount firmware will enter “keep base update” mode.

In this mode, base signals are updated both when touch panel is active and inactive. The PenMount firmware uses “low pass filter” (described in 3.8.1) to prevent base signals from changing rapidly, so it will take several rounds for the base signal to be completely updated.

The parameters used are summarized in the table below:

Touch Panel Status	Updated driver pins each time	Delay Count
Inactive	The “Update Driver Number” parameter	Inactive Update Delay Count
Active	1	Active Update Delay Count

The “Delay Count” parameters control the update frequencies when touch panel is active or inactive. Since the firmware code structure is basically an infinite loop that keeps doing the same check routines, more delay counts mean more loops before performing one base update operation.

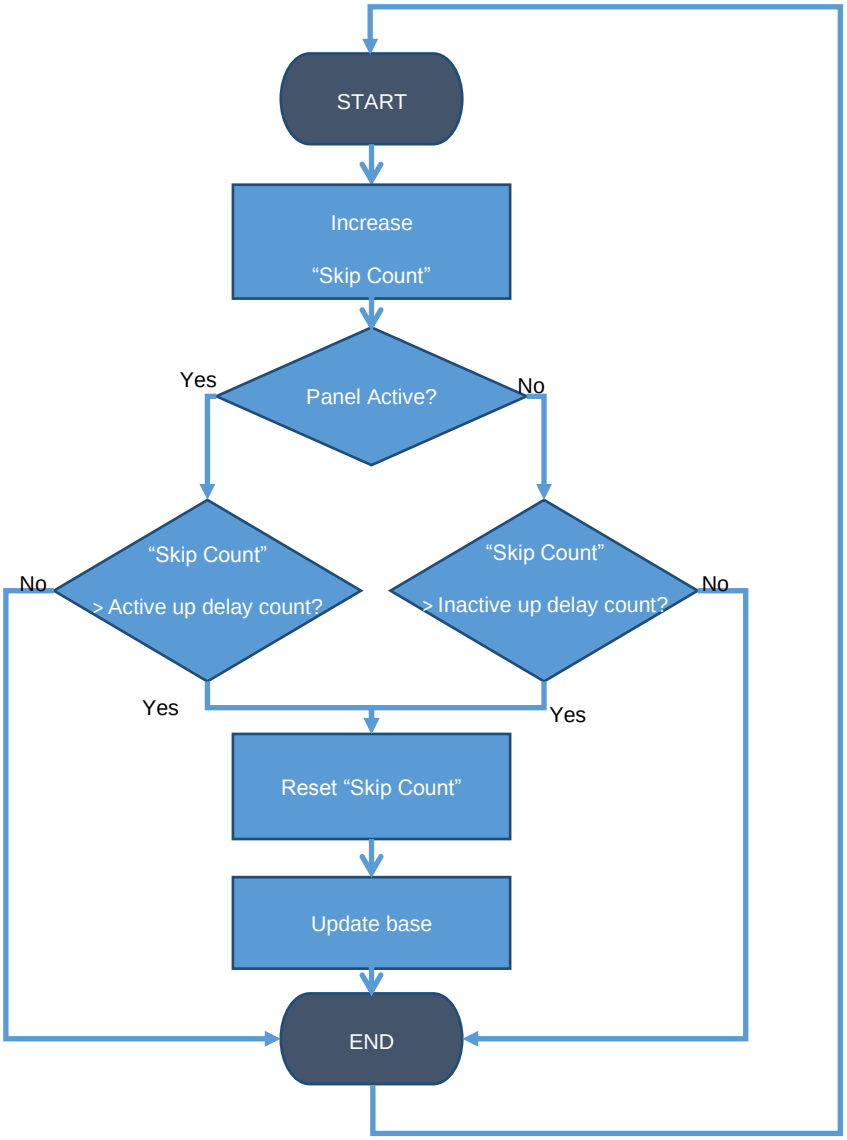


FIGURE 49

Tuning the “Active Delay Count” parameters to a proper value is extremely important. When using smaller delay count values, signals can adapt to environmental changes faster. However, since firmware keeps updating the base signal even when fingers are present, these finger signals will also be updated to become the “new base signal” eventually. In practice, user can see unexpected finger up data after press and hold in a small area for a short time.

3.7 Water Handling

The water handling feature includes choosing between different modes and tweaking parameters to compromise between performance and reliability.

3.7.1 Choosing a Water Handling Solution

The PenMount firmware supports two different water handling modes: Water Detection (WD), and Water Performance (WP).

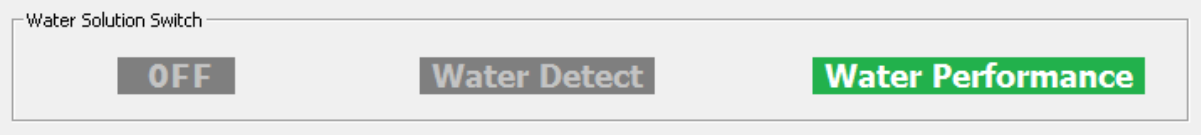


FIGURE 50

Users must choose one of the supported modes, or “OFF”, if water handling is not the main concern.

Modes	Water Handling
Water Detection Mode	Stop reporting all touch data
Water Performance Mode	Supports touch in dry area, lock to single touch if water can cause false touch points

3.7.2 Tweaking Water Performance Mode

The main feature of Water Performance (WP) mode is that it can eliminate false touch points caused by water. If false point signals keep appearing, it will cause firmware to enter a “critical state” where single touch can still be utilized in the worst case.

The parameters listed in the “Water” tab are for tuning water performance reliability.

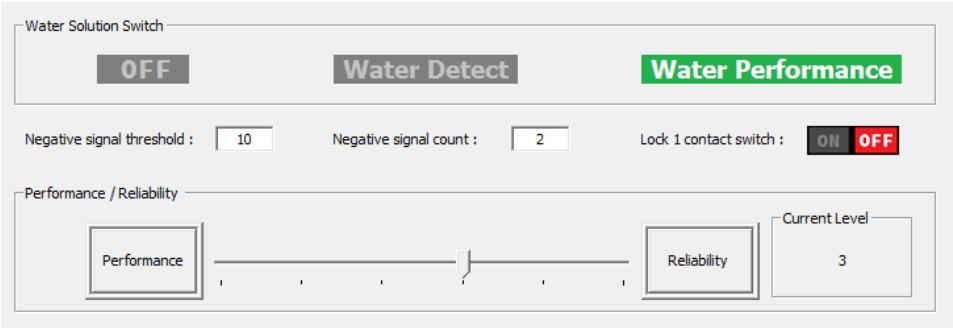


FIGURE 51

1. Lock 1 contact switch

   By default, the firmware automatically switched to single point touch mode to eliminate false touch signals. When this feature is turned OFF, the firmware does not enforce single touch mode, but it will not guarantee multi-touch results when water is present.

2. Compensate Between Performance and Reliability

   This is the main control for Water Performance mode. The settings are listed in the table below:

<table>
<tr><th>Reliability Level</th><th>Description</th><th>Touch Mode</th><th>Use Negative Signal Check</th></tr>
<tr><td>0</td><td>Some false touches to retain touch performance</td><td rowspan="3">Multiple touch in dry area;
Switch to single touch mode after false touch detected</td><td rowspan="3">YES</td></tr>
<tr><td>1</td><td>Improved line drawing in dry areas with very little reliability drop</td></tr>
<tr><td>2
(recommend)</td><td>Good performance in dry areas</td></tr>
<tr><td>3
(default)</td><td>Better reliability by locking to single touch</td><td rowspan="3">Locks to single touch mode if there is water</td></tr>
<tr><td>4</td><td>Good for harsh environments with saline, metal, or other conductors on touch panel surface</td><td>No</td></tr>
<tr><td>5</td><td>Very strict water verification that sacrifices performance</td></tr>
</table>

Unless there is extreme concern regarding performance or reliability, we do not recommend setting to level 0 or 5.

3. Tuning the Negative Signal Check Parameters

Since any touch point can be a false touch caused by water, the PenMount firmware performs several checks to filter out normal touches and those that may be caused by water.

To compensate for performance, a negative signal filter is used to quickly filter out normal touches. This is designed with a basic assumption that false touch points are always surrounded by water, which comes with a negative signal.

Please note that for above 3, the PenMount firmware will not use this filter, and will always use stricter rules to check all the touch points for false touches.

- Check Rules

The firmware checks negative signals around a touch point candidate. It will check count those that are more negative than the “Negative Signal Threshold” and, only if the amount is more than “Negative Signal Count”, the firmware will proceed with further false touch verification.

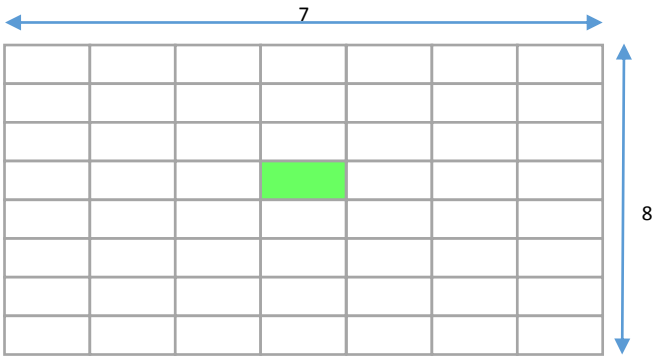


FIGURE 52

- Pass Sample

  In the following sample, there is no negative signal around the touch point (in green), so this point is identified as a normal touch.

-4	1	4	4	1	4	3
-2	0	-4	-2	-2	0	-5
-3	-1	9	25	9	1	0
3	8	42	101	36	5	6
5	3	29	67	21	6	-1
1	1	3	9	-1	-4	-5
6	5	0	3	0	0	-1
5	-1	2	-2	-1	0	-5

FIGURE 53

- Fail Sample

In the following sample, the amount of negative signals (in red) with values more negative than the "Negative Signal Threshold" is larger than the "Negative Signal Count", so it can be a false touch point and will require further verification.

9	-8	-15	-16	-15	-29	-15
24	19	-22	-29	-30	-33	21
-14	18	15	-10	-11	-7	-6
-46	-6	34	44	37	1	-23
-14	-15	-4	14	23	27	-3
-1	-12	-14	-21	-4	37	20
-5	-8	-8	-9	-14	0	31
-4	1	-1	-7	-4	0	1

FIGURE 54

Normally using the default parameter values are quite enough and it is only necessary to change them for stricter requirements.

3.8 Anti-Noise Parameters

In anti-noise parameter tab, users can configure the “Low pass filter” and switch “Enhanced Driving” to decrease the impact of noise.

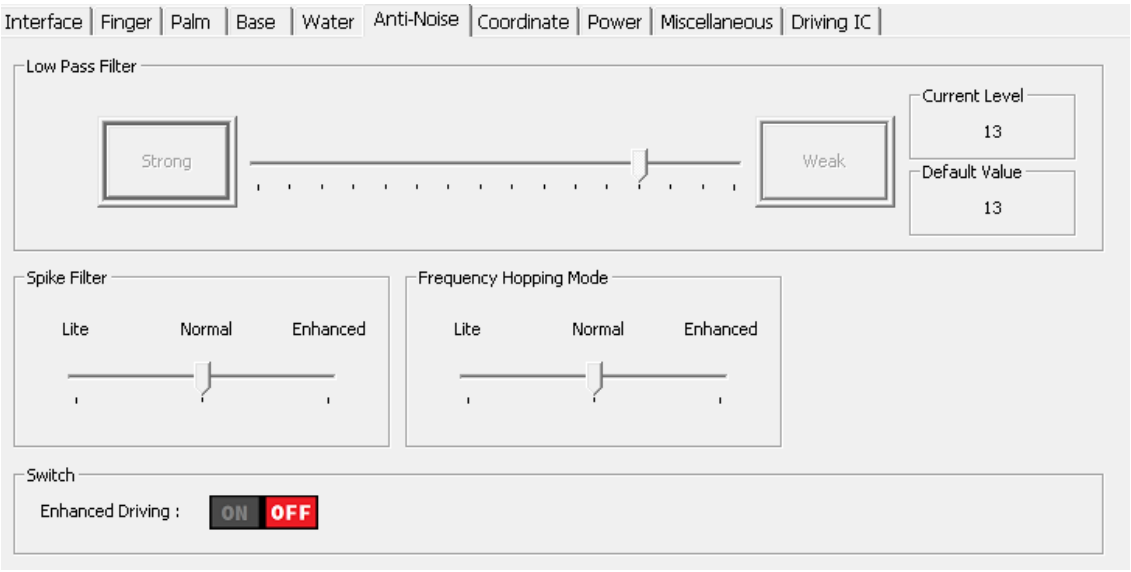


FIGURE 55

3.8.1 Configure Spike Noise Filtering

The PenMount firmware may use one of the following noise filtering methods:

Firmware Support	Spike Filter Settings		
	Lite	Normal	Enhanced
V6.2 and later	V	V	V
V6.1 and later	V		V
V1.0 and later	V		

Both methods are designed for eliminating the influence of noise jitter, and “Spike Filter” can give better result when dealing with ESD.

The “Low Pass Filter” setting will only be effective if “Spike Filter” is switched to “Lite” mode. Choosing a stronger value of “Low Pass Filter” will cause slower signal changes. The lower filter value can affect the base update time in KBU (keep base update) mode, so it needs to be fine-tuned for the best result.

Low Pass Filter	Keep Base Update Speed
Weak	Fast
Strong	Slow

3.8.2 Enhanced Driving

Enable enhanced driving for noisy environments, thicker cover glass, or glove touch to achieve better SNR and finger resolution at the expense of accuracy on the edge.

3.8.3 Configuring Frequency Hopping

For firmware versions before V6.1, this option is fixed to “Normal” and not adjustable. For firmware V6.1, the “Enhanced” mode can be used, and for firmware V6.2, an extra “Lite” mode can be used for performance tuning.

Firmware Support	Lite(0)	Normal (1)	Enhanced (2)
V6.2 and later	V	V	V
V6.1 and later		V	V
V1.0 and later		V	

3.9 Coordinate Parameters

After the PenMount firmware applies low pass, smoothing, and other filters to the sensed touch data, it will proceed to the final step according to the parameters in the “coordinate” tab to map data to the user specified display size, before sending them to the system.

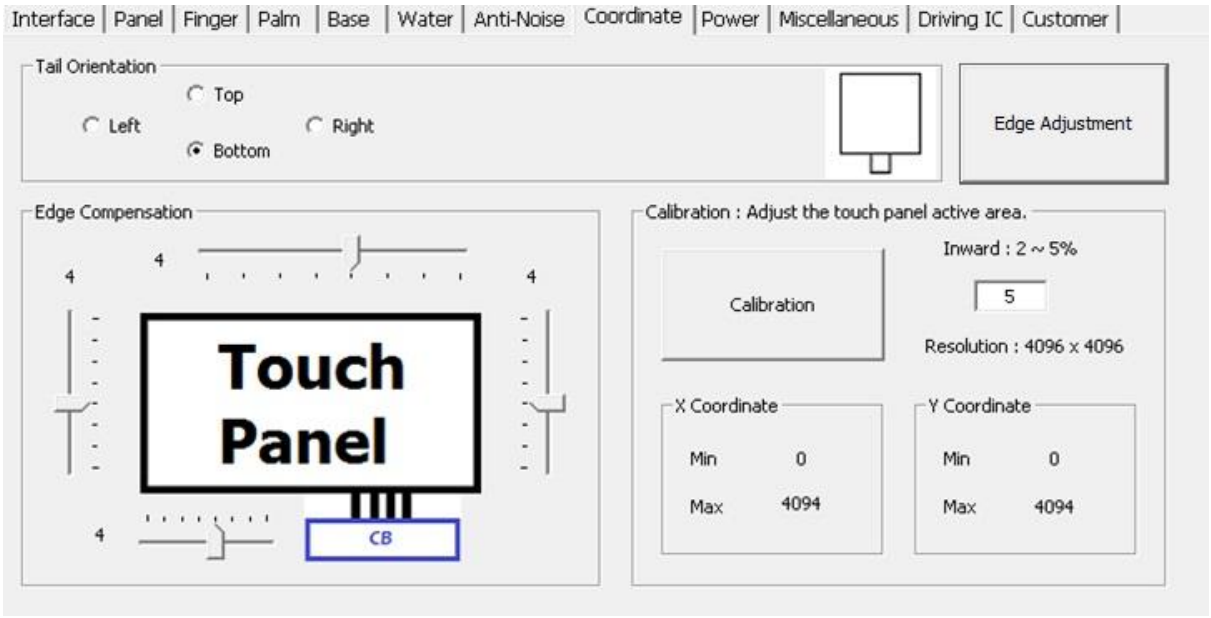


FIGURE 56

3.9.1 Calibration

Although the term “calibration” for some PCAP devices means calibrate the “base signal“ for the current environment, for PenMount it means calibrate the coordinates of the touch area to match the user defined region.

By default, the PenMount firmware maps the whole touch panel active area to the display, meaning when touching the top left corner, it sends (0,0) as its position, and sends (RES_X, RES_Y) for the lower right corner.

FIGURE 57

There are two common cases that need calibration:

(1) When Touch panel is larger than the LCD

On systems that use touch panels larger than the actual display size, the display will be mapped to the part of the touch panel from (min_x, min_y) to (max_x, max_y).

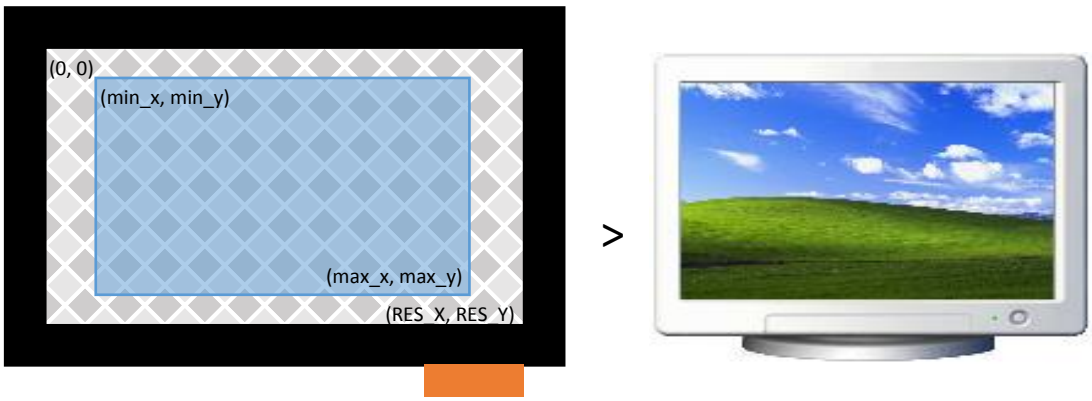


FIGURE 58

(2) When Using Front Bezel

On systems with displays designed with a front bezel, it can be harder for fingers to reach the edges. The touch position reported by firmware is near the center of a finger, and in the following diagram, it's represented by min "min_x".

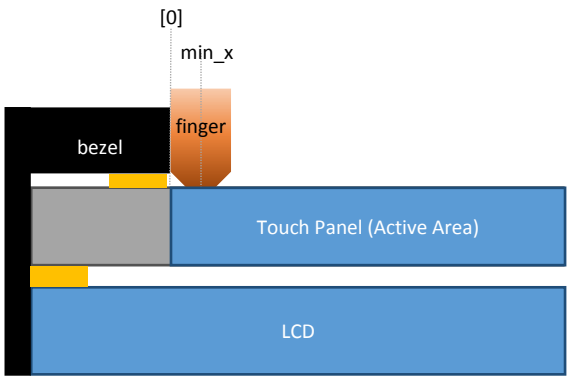


FIGURE 59

Calibration can be completed by:

(1) Manually setting up the minimum and maximum values for the X and Y axes.

The values should be calculated and converted to a 0 to DIP_RESO_X and DIP_RESO_Y[5] range.

(2) Pressing the “Calibration” button to launch the helper UI.

3.9.1.1 Data Collection Stage

Please follow the instructions on screen by pressing the two red squares in top-left and bottom-right corner. The program will calculate the minimum and maximum values for the X and Y axes when finished.

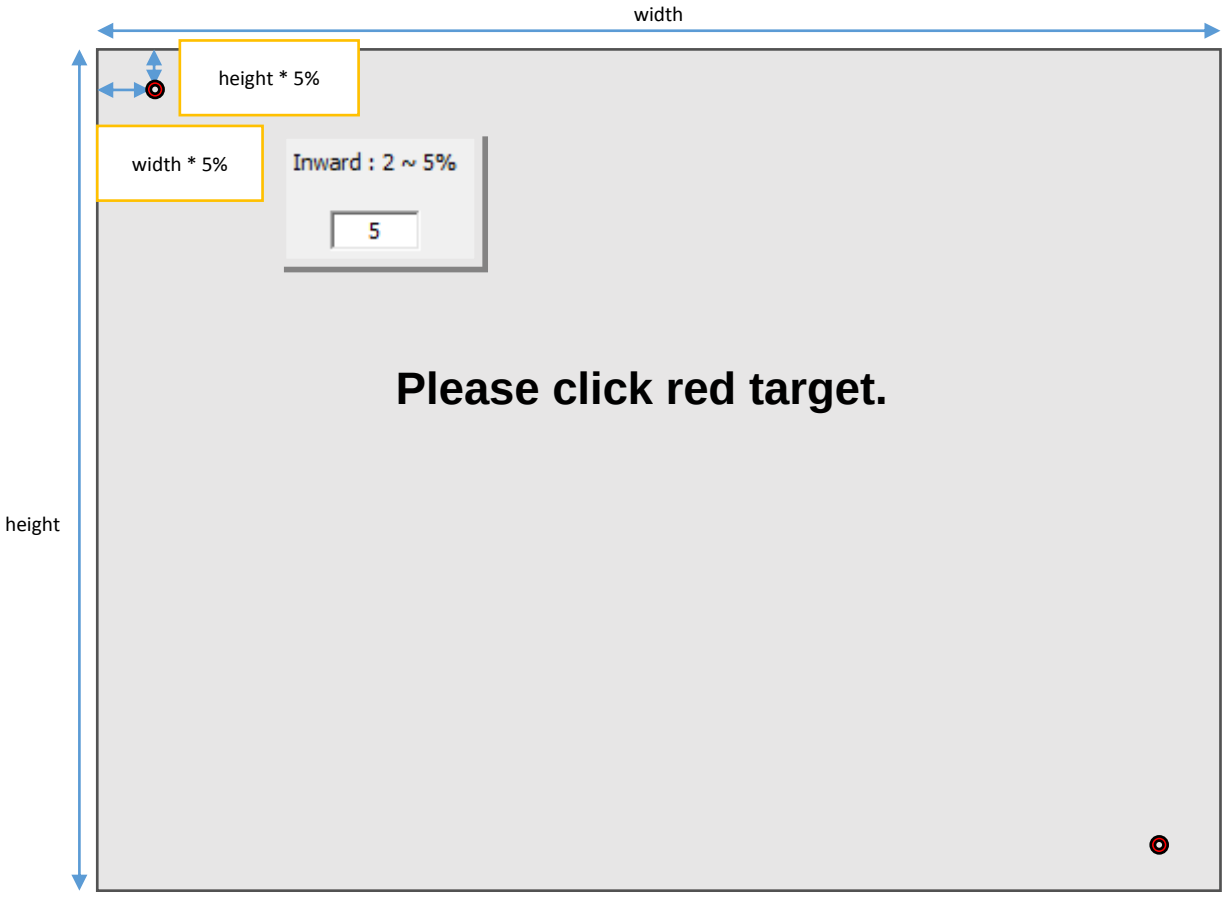


FIGURE 60

The “inward” setting helps users touch the red square easier, especially when performing calibration on a LCD with front bezels.

3.9.1.2 Test Stage

The calibration UI proceeds to the test stage after it collects the minimum and maximum axes values. During this stage, there is a third red square and users must press it. This stage can be skipped by pressing

[5] The “DIP_RESO_X” and “DIP_RESO_Y” value can be 4095 when using firmware V6.1 USB interface, or 2047 for any other case.

the F9 key. The firmware will automatically apply new settings if users skip the test, or when the test is successful.

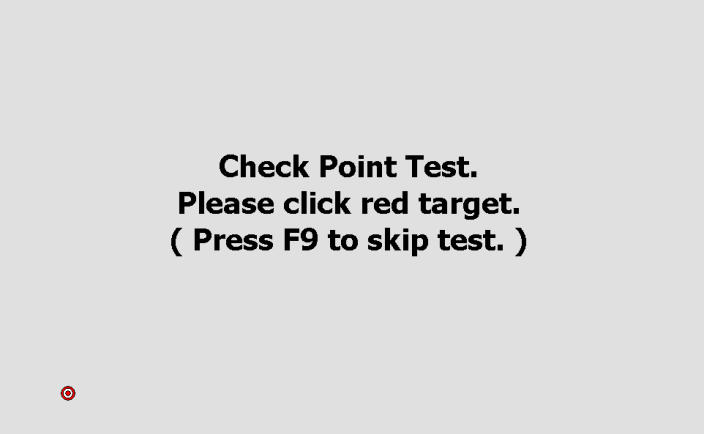


FIGURE 61

If the test fails, the program will prompt for another confirmation and users can still choose to apply the calibration result by pressing the “yes” button on screen.

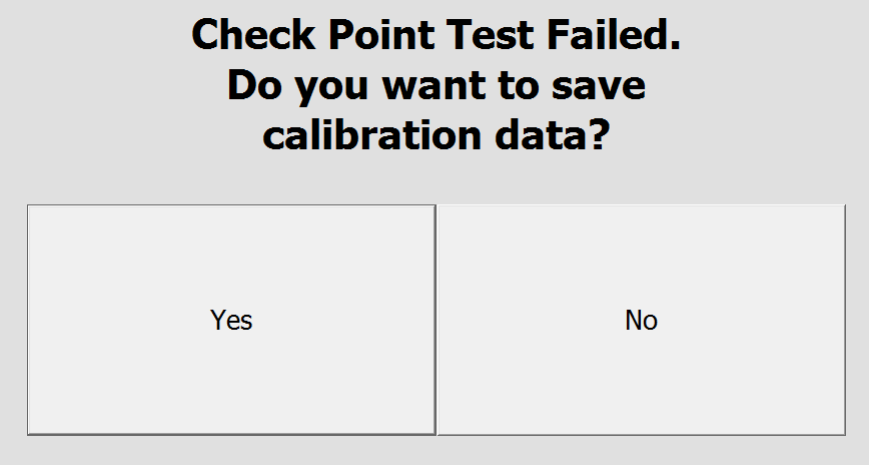


FIGURE 62

3.9.2 Edge Compensation

When the PenMount controller is released from the factory, the firmware’s default active area setting for the LCD monitor corresponds to that of the touch panel and the active area sizes of hardware and software are the same. However, due to resolution, the active areas for hardware and software can be slightly different. To adjust this, please click “Edge Compensation”.

When the edges of the touch panels do not function smoothly, please set the edge compensation value higher to improve functionality.

3.9.3 Tail Orientation

The firmware supports four rotational degrees compared to tail orientation.

FIGURE 63

3.9.4 Edge Adjustment

When touch panels are mounted on LCD with thicker frame design, it would not be easy for a finger to touch the UI elements near the edge. In this case, users can use this feature to automatically adjust the sensor / driver lines used by device firmware to work with thicker frames.

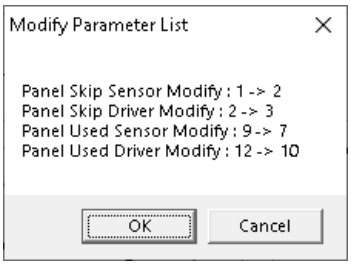


FIGURE 64

The definitions of Skip Sensor, Skip Driver, Used Sensor, and Used Driver are defined as below.

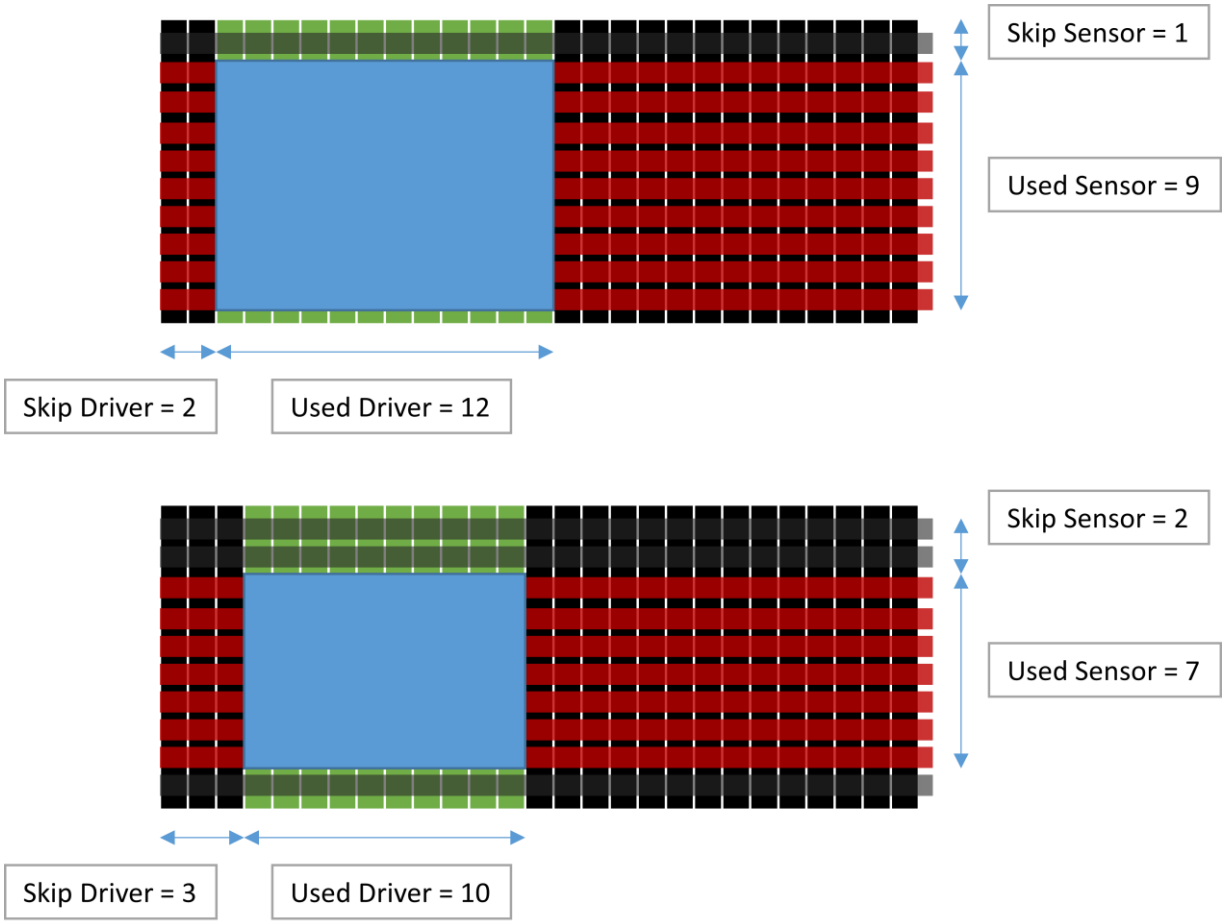


Figure 65

Please notice that click on the “Edge Adjustment” button several times might significantly decrease accuracy. To revert settings, please use the “Write Factory Default Parameter” button or do firmware update again.

3.10 Power Parameters

PenMount supports three power modes: Active Mode, Idle Mode, and Sleep Mode. The transitions are illustrated in the diagram below:

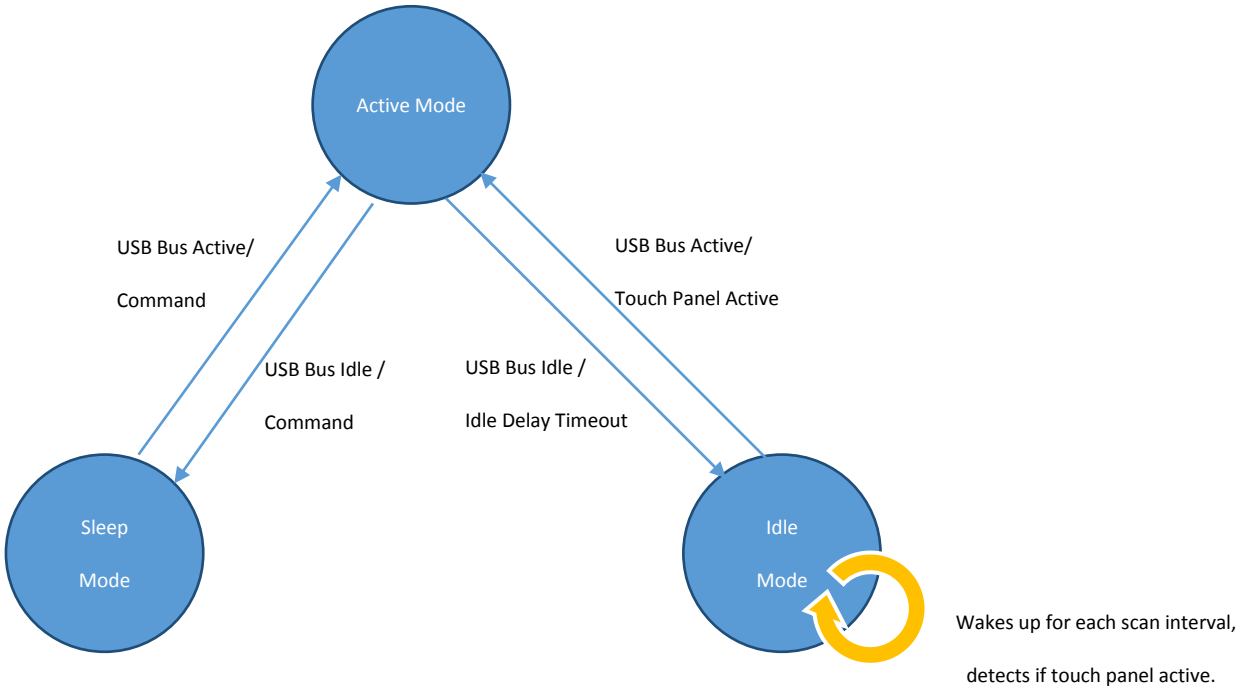


FIGURE 66

Both sleep mode and idle mode can be enabled or disabled with a PCI Parameter.

FIGURE 67

1. Idle Mode

If enabled, the device will enter idle mode if it meets one of the following conditions:

- When remote wake up is enabled and USB bus becomes idle (USB interface only)
- When touch panel is inactive for a certain time period.

During idle mode, device wakes up from time to time to scan for touch signals. If the scan result is active, it will return device to active mode.

2. Sleep mode

If enabled, the device will enter sleep mode if it meets one of the following conditions:

- When remote wake up is disabled and USB bus becomes idle (USB interface only)
- When the device receives sleep command (RS-232 and I^2C interface only).

Once device enters sleep mode, touch will be disabled, and the device will only wake up when system resumes or when it receives any other command.

3.10.1 USB Interface Configurations

PenMount supports the USB “Remote Wakeup” feature and, when this feature is enabled, user can wake up the system by touch. For Microsoft Windows systems, this can be configured in the device’s “Properties” page in “Device Manager”.

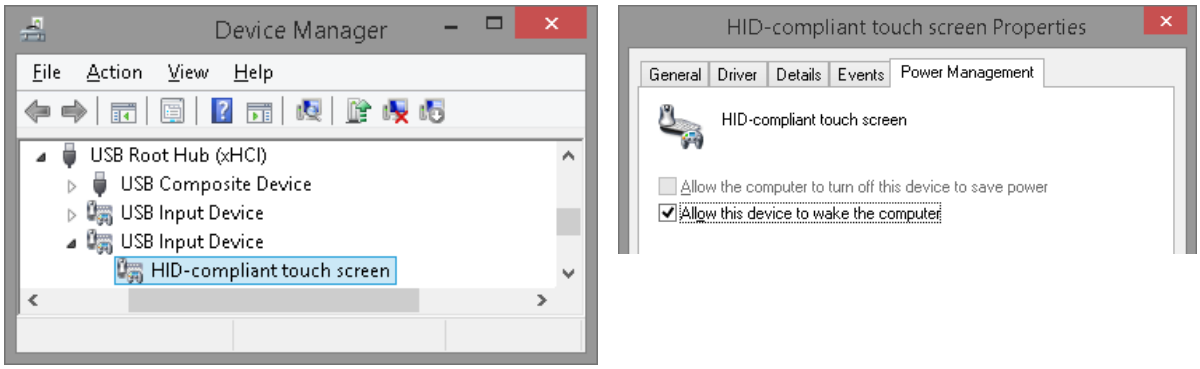


FIGURE 68

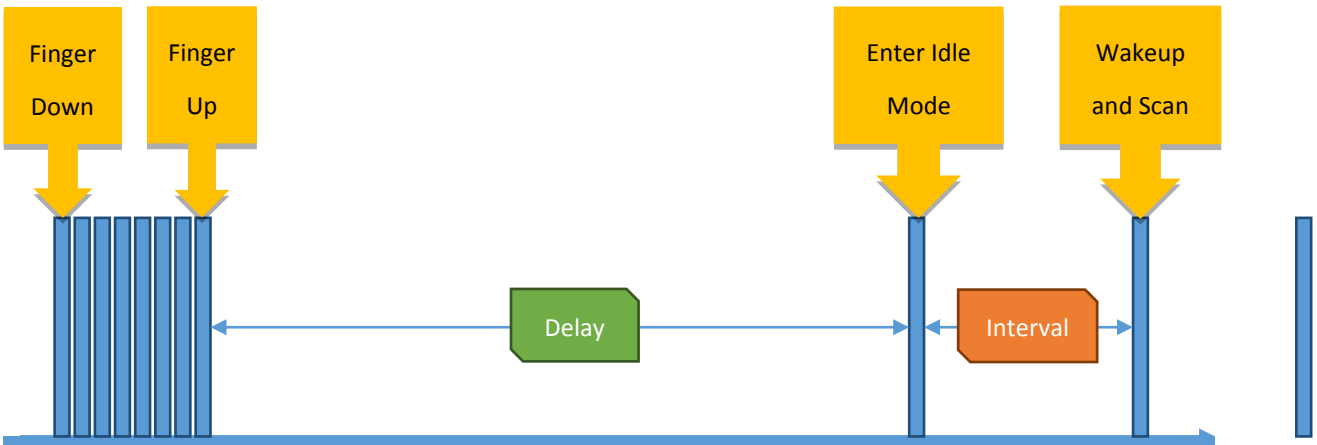
When the USB Remote Wakeup feature is enabled, it will also let PenMount device go into idle mode after system is suspended, otherwise it will go to sleep mode and will not wake up until system is resumed.

USB States / Remote Wake-up	Suspended States	Enabled States
Enable	Idle mode	Active Mode (With Touch)
Disable	Sleep mode	Idle Mode (Without Touch)

3.10.2 Idle Mode Settings

For idle mode, user can configure how fast the device becomes idle and how frequently it wakes up to scan for touch signals.

FIGURE 69



Power Modes	Scan Interval
Idle Mode	Configurable 10ms, 20ms, 30ms, 50ms
Sleep Mode	Scan Stops

FIGURE 70

3.11 Miscellaneous Features

In the “misc” tab, users can enable or disable firmware supported features.

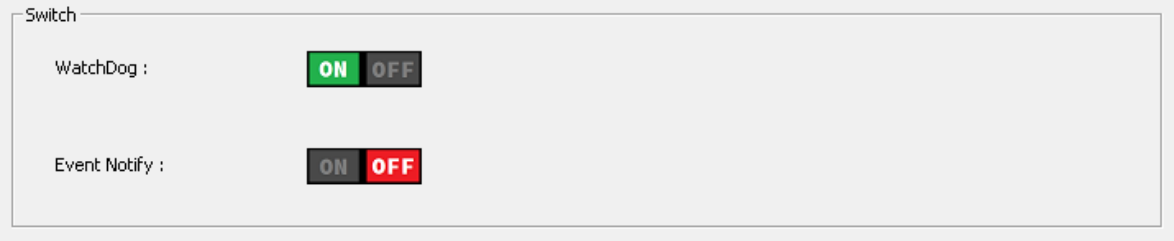


FIGURE 71

3.11.1 Watch Dog

When the watch dog feature is enabled, it will automatically reset device when it becomes unresponsive.

3.11.2 Event Notify

This switch controls two kinds of event notifications:

(1) The helper feature for "Water Detection" (WD) mode. When firmware enters an abnormal state, it will send a special notification data packet as the listed in the table below.

Interface: USB — Abnormal Notification

State	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Enter	0x01	0x8A	---	---	---	---	---	---	---	---	---	---	---	---
Leave	0x01	0x0A	---	---	---	---	---	---	---	---	---	---	---	---

Values of data byte 2 to 13 are not used and can be ignored.

Interface: RS-232 — Abnormal Notification

State	0	1	2	3	4	5
Enter	0x8A	0x00	0x00	0x00	0x00	<CheckSUM>
Leave	0x0A	0x00	0x00	0x00	0x00	<CheckSUM>

Interface: I2C — Abnormal Notification

State	0	1	2	3	4	5
Enter	0x8A	0x00	0x00	0x00	0x00	<CheckSUM>
Leave	0x0A	0x00	0x00	0x00	0x00	<CheckSUM>

The PCIUtility will be able to display warning messages after receiving this message.

If the device drivers in the operating system do not support this feature, receiving the message could cause some issues. In this case, please consider turning this feature off.

(2) When the “Touch and Halt” feature is enabled on the Debug Page, device will also send a notification when touch is detected. If it is turned off, the “Touch and Halt” button will also be greyed out and cannot be changed.

The data packet of the event notification is as listed in the table below.

Interface	Abnormal Notification
USB	<table><tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th></tr><tr><td>0x01</td><td>0x8F</td><td>0x00</td><td>0x00</td><td>0x00</td><td>0x01</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td></tr></table> Values of data byte 6 to 13 are not used and can be ignored.
RS-232	<table><tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr><tr><td>0x8F</td><td>0x00</td><td>0x00</td><td>0x00</td><td>0x01</td><td><CheckSUM></td></tr></table>
I2C	<table><tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr><tr><td>0x8F</td><td>0x00</td><td>0x00</td><td>0x00</td><td>0x01</td><td><CheckSUM></td></tr></table>

3.12 Driving IC Parameters

PenMount controllers use the driving IC to help improve SNR. Driving voltage and current limits can also be configured.

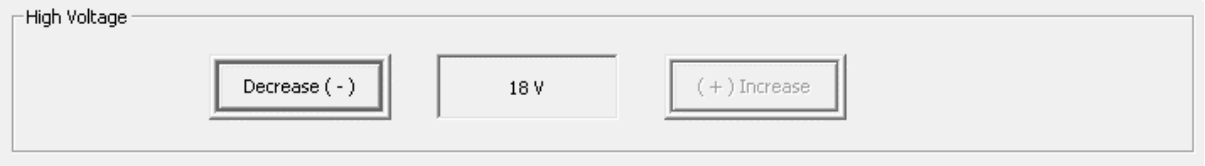


FIGURE 72

3.12.1 High Voltage

The default voltage used is 18V for better SNR. However, if using a higher voltage will affect system EMI testing results, please consider using smaller values. The supported voltage range is:

Driving Voltage	Description
18V	Default value
16V	
14V	
12V	

10V	
8V	Not recommended
6V	Not recommended

Please note that some voltages are not recommended, because the measured signal will be too small and can affect normal operation.

3.13 Customer Parameters

Some ODM firmware versions support special parameter settings requested by customer.

For example, the “Point Report Ratio”, which controls the overall data report rate to enhance touch experience on slower systems, can be adjusted in the “Customer” tab.

FIGURE 73

4. Field Update

The field update page allows user to change controller firmware version.

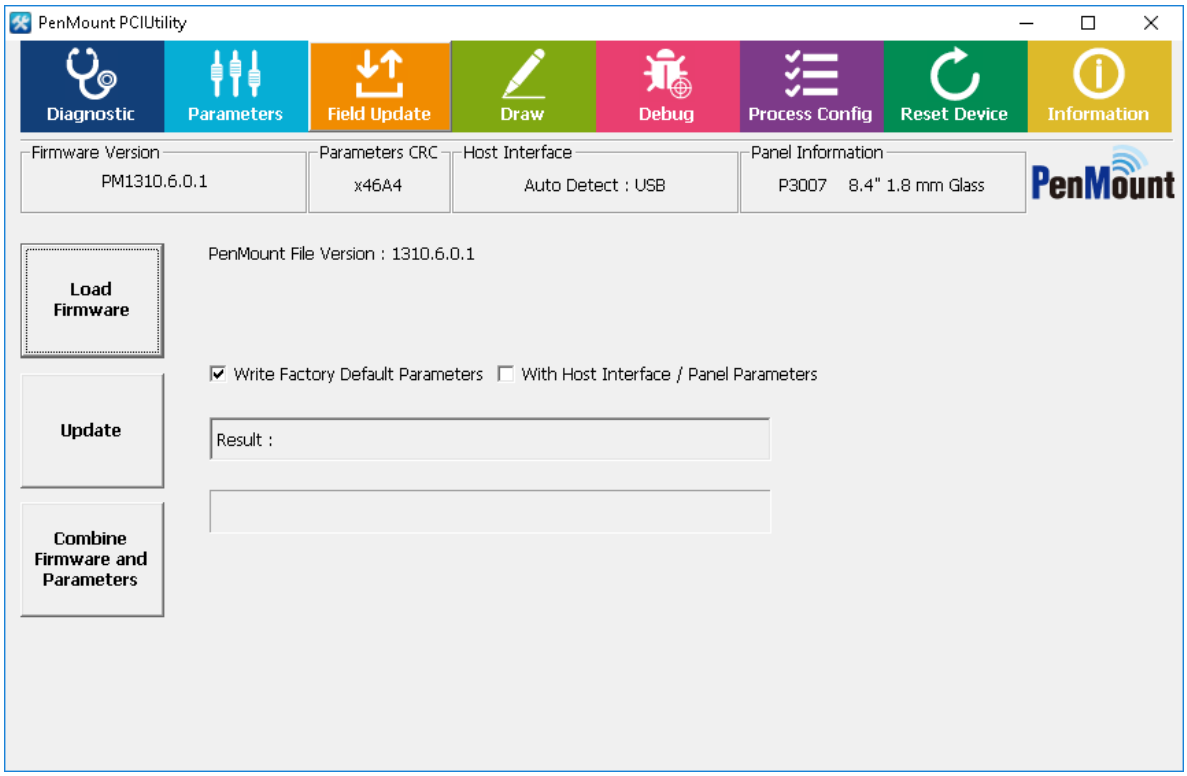


FIGURE 74

4.1 Load Firmware File

Please press “Load Firmware” under “Field Update” and click the file you want to update in the dialogue box.

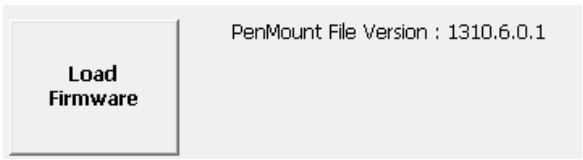


FIGURE 75

The format of the firmware supported by PenMount is suitable for

(1) “pmf”, “pnf”, or “pxf” format for one single IC

(2) “pmc”, “pnc”, or “pxc” format for multiple ICs.

(3) “pfp” format for combination of firmware and parameters.

The pfp file format that includes virtual button settings are also supported in this utility version, which may also display the CRC values of “Parameter”, “ND Base”, or “Virtual Key” when corresponding data is included in target file.

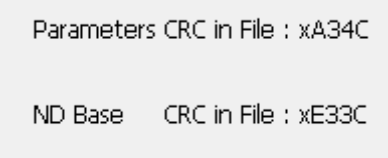


FIGURE 76

Please note, if the firmware file is not consistent with the controller connected during the update, the update will not proceed.

4.2 Update Firmware

After you load firmware files, please click “Update” button to proceed. There are two options available:

- Write Factory Default Parameters

This option is selected by default when updating with PMF or PMC file format. This option is not selectable when using PFP file format.

With this option set, firmware parameters will be reset to default values after update finishes. When only check this option, host interface / panel and other firmware parameters will remain unchanged.

- Write Host Interface / Panel Parameters

This option is only adjustable when “Write Factory Default Parameters” is turned ON. With this option set, host interface and panel parameters will also be reset to default values after update finishes.

During the update, all the button settings will be locked temporarily to prevent update failure due to inadvertent selection.

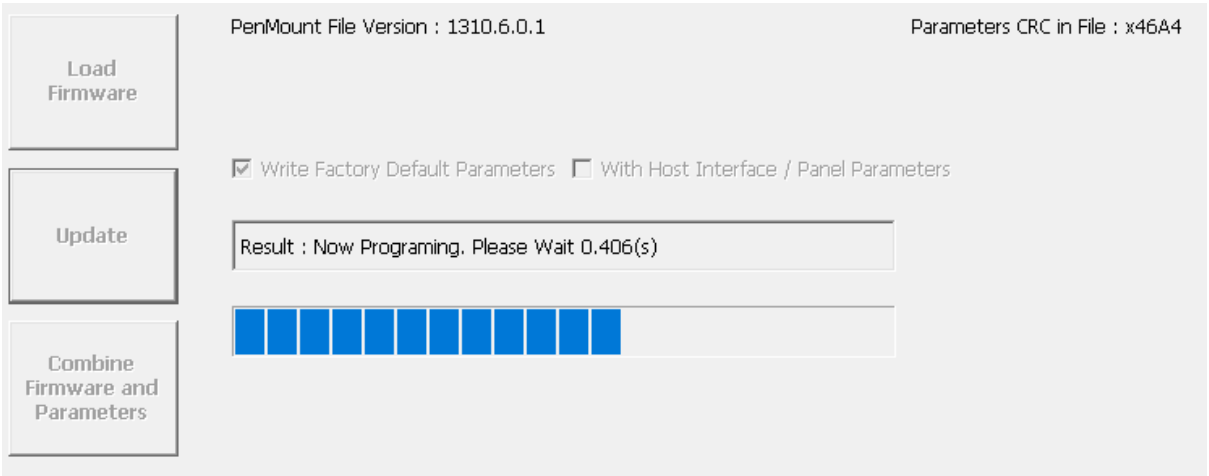


FIGURE 77

Please wait until update reaches 100%. During the update, please do not interrupt the power supply. After the firmware update is completed, you will see the “PASS” mark.

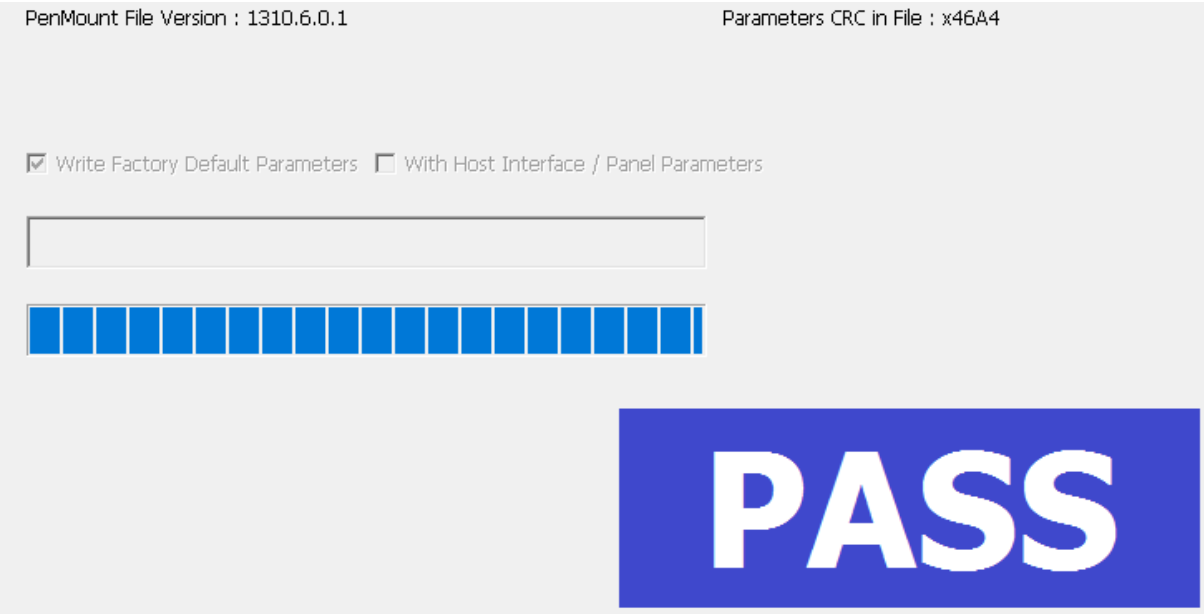


FIGURE 78

If the update fails due to an unavoidable factor, you will see the “FAIL” mark and the error messages.

FIGURE 79

If this happens, you will find firmware info remains in “Boot Loader” mode. You can click the “Update” button again to restart the update process.

4.3 Generate New PFP File

With PCIUtility V1.6 and later, users can generate new PFP files with existing firmware files. The “Combine Firmware and Parameters” button will be enabled after user specified a compatible firmware file. After pressing the button, a “Browser for Folder” dialog will appear to allow saving the PFP file to the target folder.

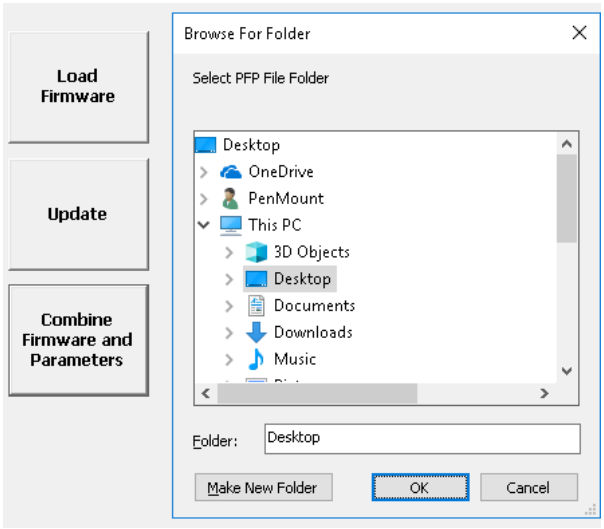


FIGURE 80

The newly generated PFP file will contain all the parameter settings of the connected device.

5. Draw

In Drawing mode, user can examine whether breaking or point hopping will occur when drawing lines and provides a mechanism for measuring sample rate.

Item	Description	
Operating systems	Windows XP and later versions	
Support Index1	1P	Black
	2P	Red
	3P	Green
	4P	Blue
	5P	Magenta
	6P	Cyan
	7P	Brown
	8P	Yellow
	9P	Purple
	10P	Orange
Switch to Draw Mode	“D” key	
Switch to Touch Mode	“T” key	
Sample Rate Test Hotkey	“S” key	
Clear Screen	“Space” key	
Toggle Full Screen	“F” key	

The program will show a white canvas in full screen mode, allowing the user to test the touch drawing feature. If you need advanced functions, please “Press and Hold” with your finger or click the right mouse button for the functions indicated below:

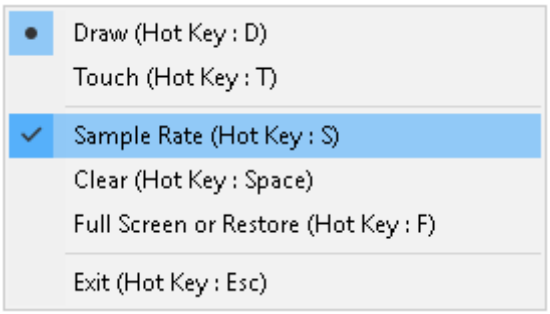


FIGURE 81

- Draw Mode

  Switch to draw mode. In this mode, the program will label touch points in different colors.

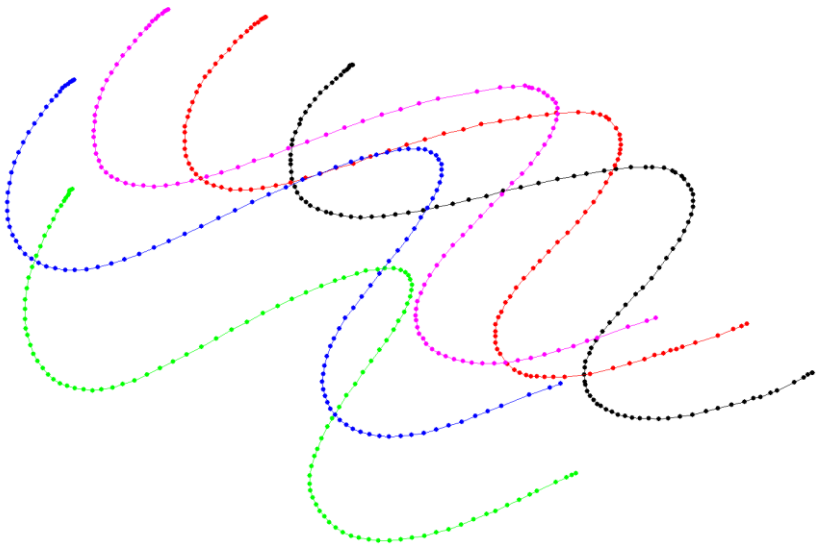


FIGURE 82

- Touch Mode

  Switch to touch mode. In this mode, only the touch point location of your finger is shown.

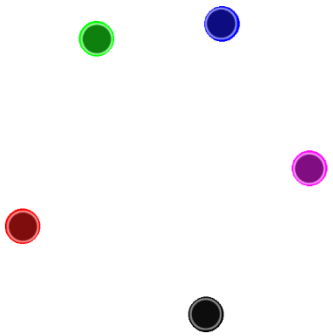


FIGURE 83

- Sample Rate

  After touch input, it will take approximately one second for the program to calculate and show the Sample Rate of each touch point. For example, Point 0: 116 means 116 points per second. When the Sample Rate is higher, drawing will be more precise.

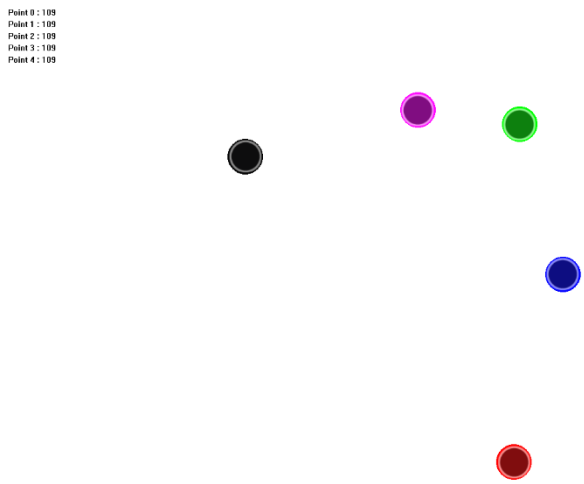


FIGURE 84

- Clear

  Clean the canvas.

- Full Screen or Restore

  Switch between full screen display and windowed display.

- Exit

  Exit the program.

Note: After entering Draw mode, please do not connect or disconnect devices. Otherwise, touch will not function.

6. Debug

The debug panel provides several features for analysis of touch status and debugging touch issues.

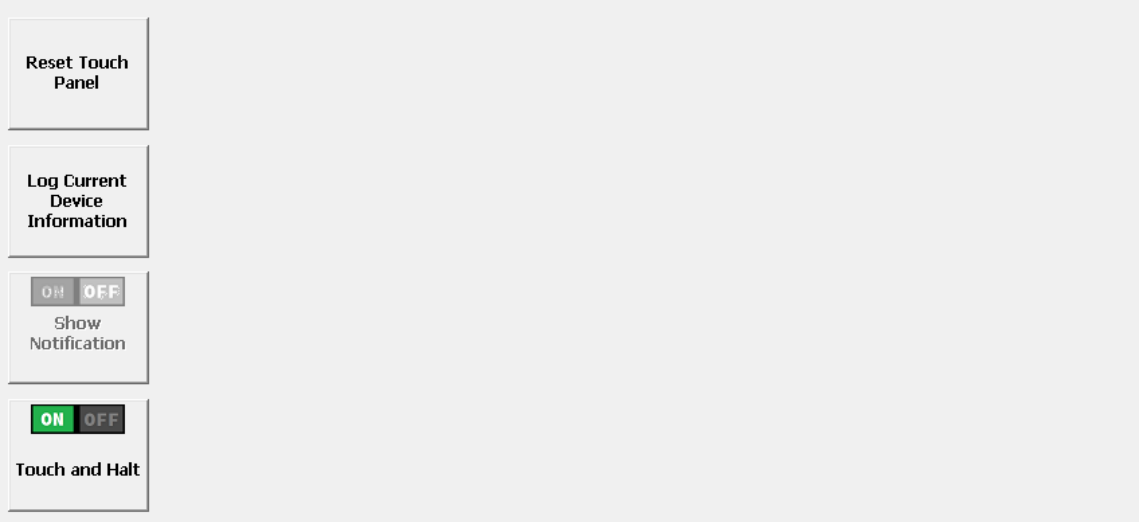


FIGURE 85

6.1 Reset Touch Panel

Choosing this option will force the PenMount touch device to be reinitialized. There will be a message box asking for user confirmation, please choose “Yes” to perform device reset and wait until the “Reset touch panel finish” message appears.

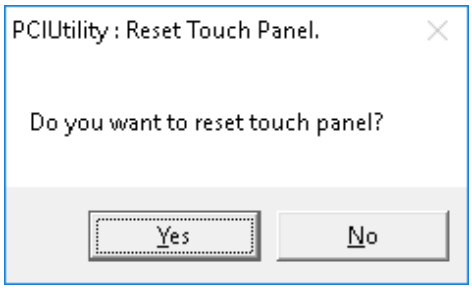


FIGURE 86

Reset touch panel may help resolve some touch malfunction issues when panel is affected by environmental changes such as temperature, water, or noise.

6.2 Log Current Device Information

Choosing this option will save firmware information to file which can be provided to the PenMount team for issue debugging. Due to the amount of information, the whole process could take several seconds.

As a practical condition, the human body and the system cannot share the same ground. The touch panel performs normally when it is isolated. However, the touch panel may be interfered with when a human touched the panel. This is because noise may come from a different ground.

This function can guide the user to obtain information step by step and save this information in the directory that the user specified.

When user starts the test, tap the black mark to start noise test.

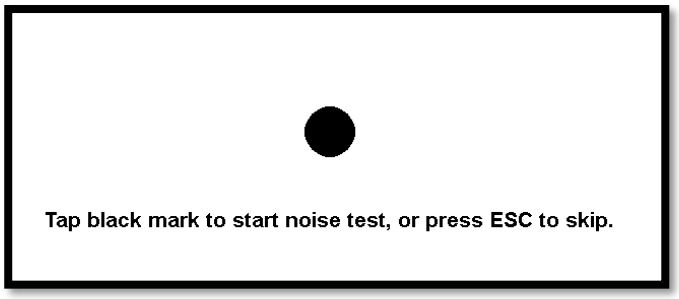


FIGURE 87

Then, do not touch panel while the firmware is sampling.

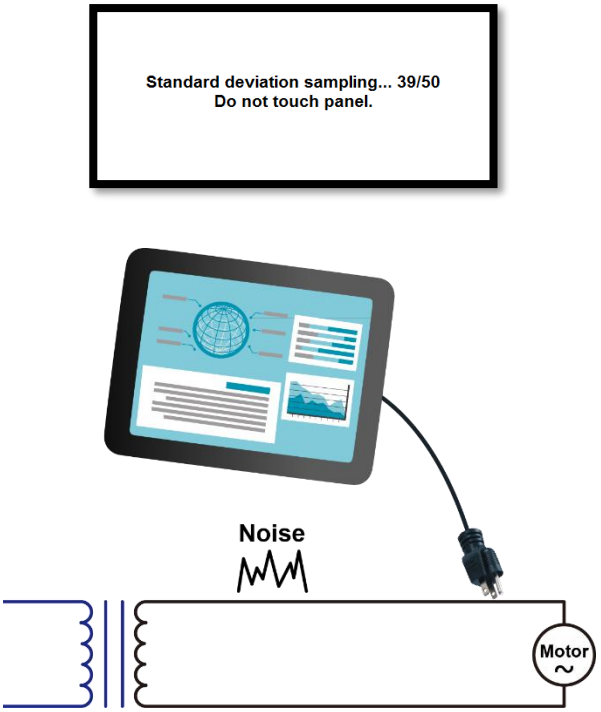


FIGURE 88

When the screen shows the black point, please keep touch it until the process is finished.

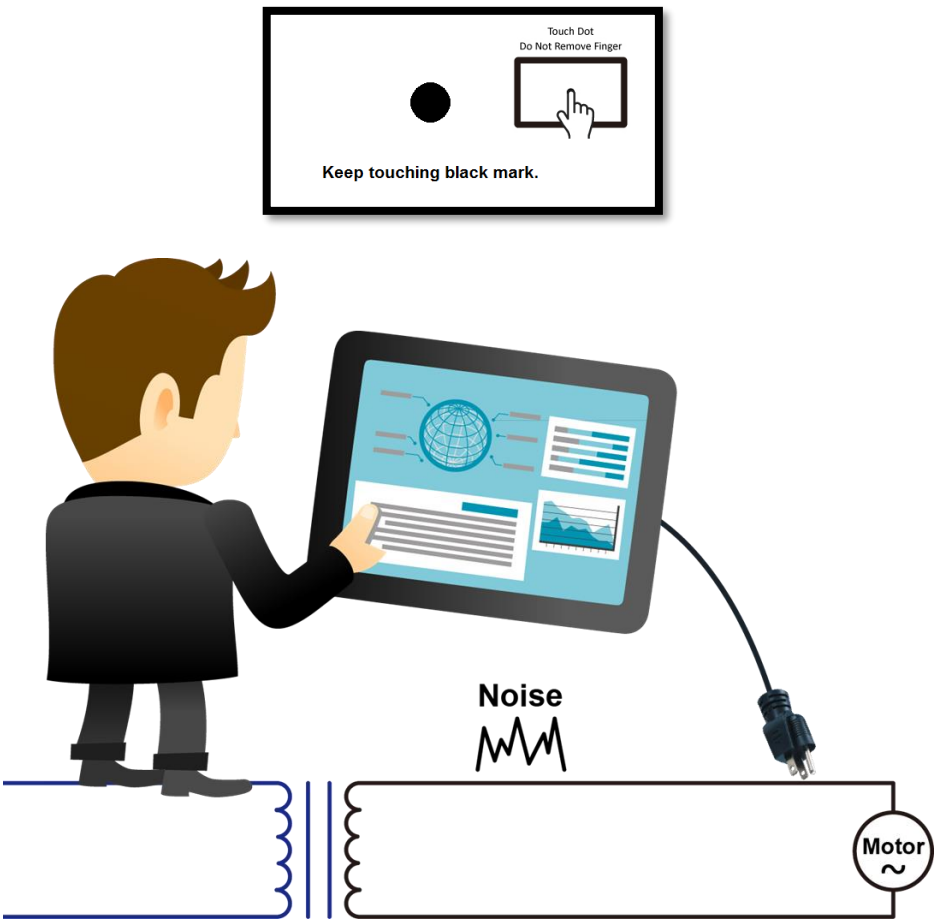


FIGURE 89

Finally, please wait for the “Write log finish” message to appear.

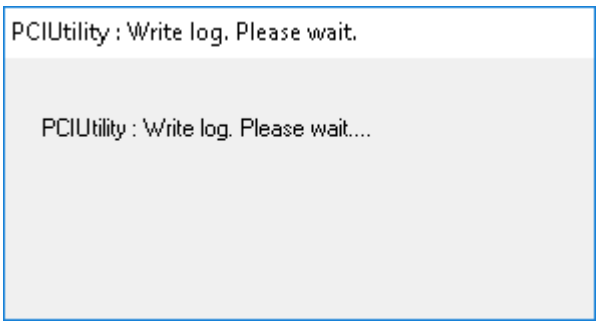


FIGURE 90

The file consists of the useful information for debugging touch issues.

6.3 Show Notifications for Water Detection

When firmware is using “Water Detection” mode, touch will be disabled when firmware detects water on touch panel surface.

Choosing the “Show Notification” option enables more conspicuous notifications when touch is disabled. The warning dialog box will disappear after several seconds.

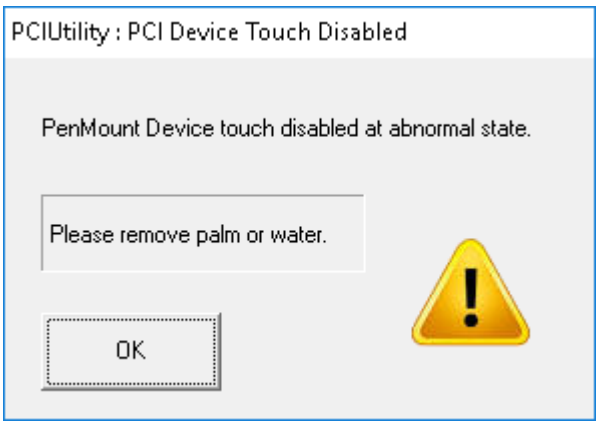


FIGURE 91

6.4 Touch and Halt

The touch and halt feature is supported by the latest PenMount firmware, which is used for debugging touch issues. If this field is grayed-out, it means that the firmware does not support this feature.

When enabling, utility will ask user to choose a folder to store firmware information.

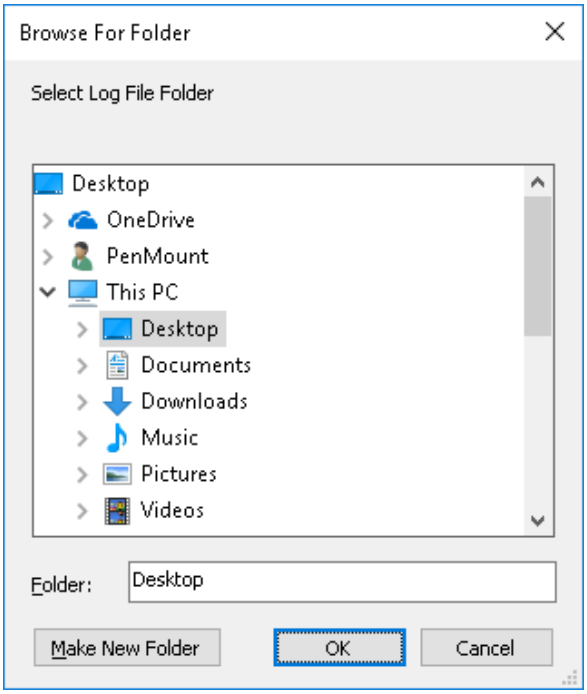


FIGURE 92

After enabling, each touch operation will trigger the firmware to cease scanning and allow the utility to retrieve the most accurate information from device. This feature will be switched back to disabled state after utility finishes saving data.

7. Process Config

“Process Config” is the feature that allows generating process configuration files to be used in PenMount production line utility for processing the configured tasks.

This feature is accessible without PenMount device connected.

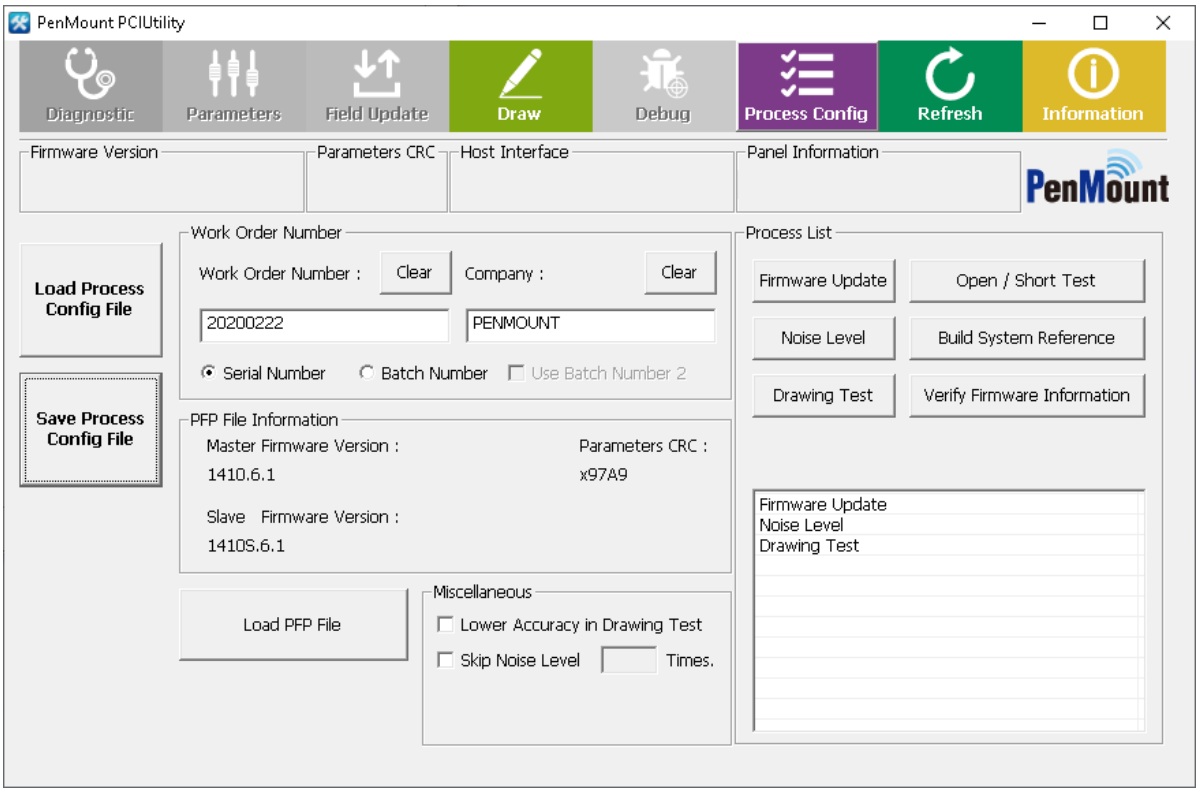


FIGURE 93

- Load Process Config File

  Log the existed process configuration file to check detail or modify the setting.
- Save Process Config File

  Save the current setting the in the process configuration file.
- Load PFP File

  Load the PFP file to set up the process configuration file

7.1 Work Order Number

A “Process Config” file is a plan text file with INI extension. The following information is necessary to create a new file.

The work order number is defined by project or by company.

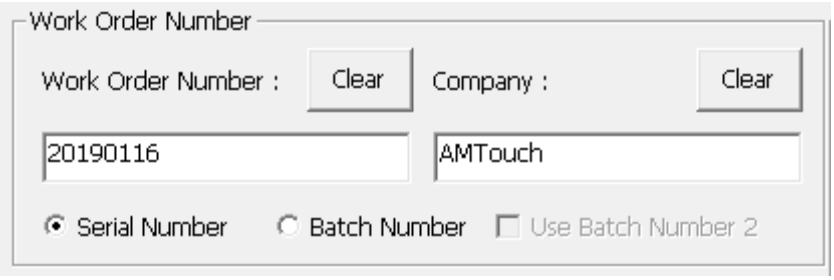


FIGURE 94

Choose the “Batch Number” or “Use Batch Number 2” option when output file is for batch test.

Choose the “Serial Number” option when output file is for individual test.

7.2 Load PFP File

Please provide a valid PFP file by using the “Load PFP File” button. After that, the information of the file will be displayed in utility.

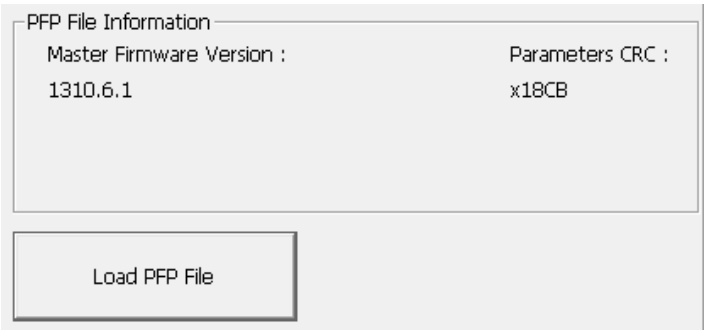


FIGURE 95

7.3 Miscellaneous

Several options are available for tweaking the test flow.

- Lower Accuracy in Drawing Test

User can select drawing test to check if the PCAP can be draw smoothly. However, sometimes the user test without LCM and the accuracy may be affected. The user can switch on “Lower Accuracy Test in Drawing Test” to improve the production line efficiency.

The next two figures show the normal test area for drawing test, and the test area when the user can switch on “Lower Accuracy Test in Drawing Test”.

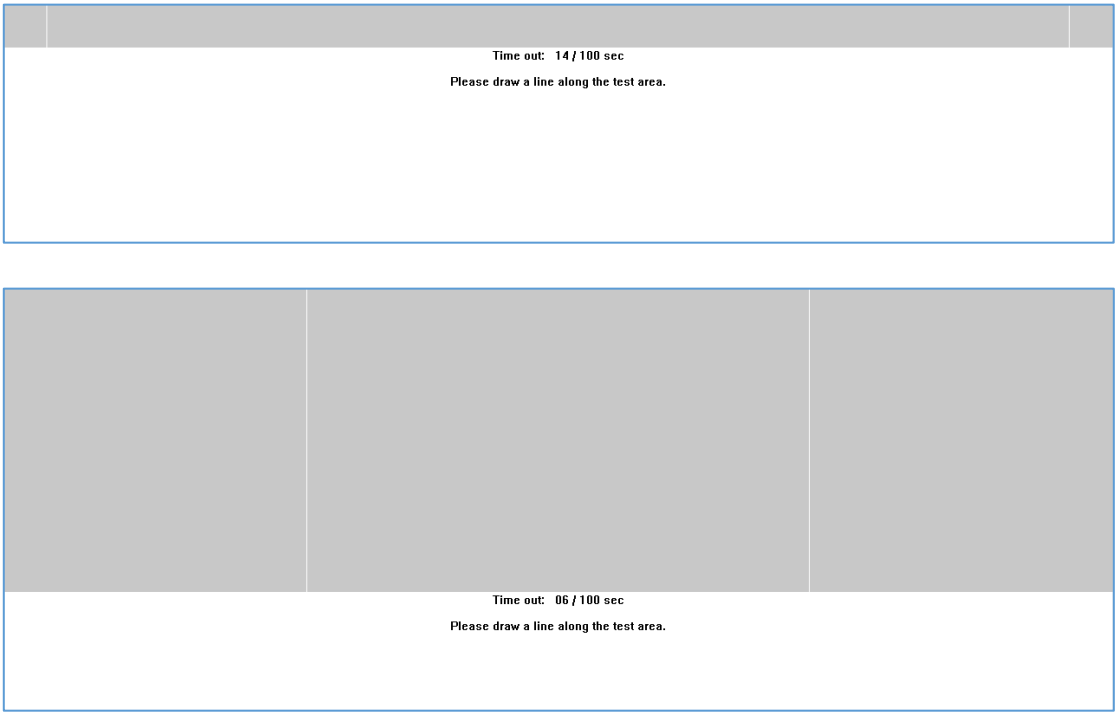


FIGURE 96

- Skip Noise Level N Times

Noise Level Test costs a lot of time. When performing batch tests for multiple control boards, it is not always necessary to do noise level test each run. In this case, user can specify the skip times in this field.

For example, if set up “Skip Noise Level 2 Times”, the test flow may become as below.

Test #	1	2	3	4	5	6	7
Do Noise Test	YES	SKIP	SKIP	YES	SKIP	SKIP	YES

7.4 Select Tests for Process List

Each button under the Process List group is represents a specific task. Please click on the desired button to add the task to Process List. To remove it, please click the button again.

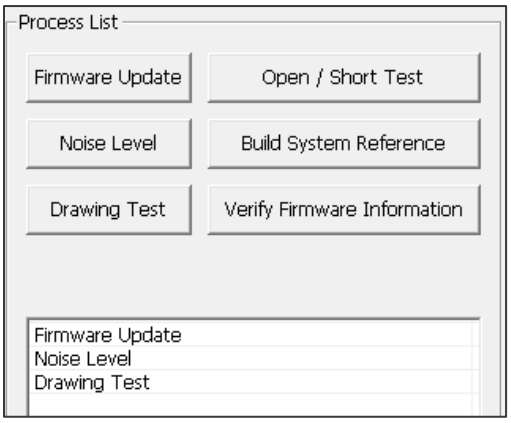


FIGURE 97

The process order is determined by the task selection sequence. In the example above, the selection order is “Firmware Update”, “Build System Reference”, and “Drawing Test”, and the PenMount production line utility will also follow this order to perform tasks.

Task	Description	Option
Firmware Update	Update firmware with the selected PFP file.	
Open / Short Test	Do open short test	
Noise Level	Do noise level test with pattern “WHITE”, “BLACK”, and “CHECKERBOARD”	
Build System Reference	Build system reference	
Drawing Test	Do instructed draw test to verify touch functionality and within acceptable accuracy.	Low Accuracy in Draw Test When this option is selected, “Draw Test” will allow much relaxed passing conditions.
Verify Firmware Information	To verify if the device firmware matches the selected PFP file.	