

PENMOUNT PCIUTILITY USERS' GUIDE FOR MICROSOFT WINDOWS

Version 1.2

25/Apr/17



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 2017 **PenMount Touch Solutions**

Revision Table

| Date | Revision | Changes |
|--------------------|----------|--|
| 31/May/2016 | 1.0 | First release version. |
| 03/Jun/2016 | 1.0A | Revised 1.2 Supported Devices |
| 28/Oct/2016 | 1.1 | (1) Added 1.6 Special Notice (2) Added 3.5.1 Palm Rejection Switch (3) Added 3.9.2 Edge Compensation (4) Added 6.4 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) In 3.9.1.2, the hotkey for skipping calibration checkpoint is changed to F9. |

Table of Content

| | |
|---|-----------|
| PREFACE | I |
| Disclaimer | i |
| Trademarks..... | i |
| Copyright | i |
| REVISION TABLE | II |
| 1. INTRODUCTION | 6 |
| 1.1 Supported Systems..... | 6 |
| 1.2 Supported Devices..... | 6 |
| 1.3 Utilities Summary | 7 |
| 1.4 Checking Utility Information | 7 |
| 1.4.1 Device Information | 8 |
| 1.4.2 Application Version | 8 |
| 1.4.3 Open Default Export File Folder | 9 |
| 1.5 Refresh and Reset Device | 9 |
| 1.6 Special Notice | 9 |
| 2. DIAGNOSTIC | 10 |
| 2.1 Open Short Test..... | 10 |
| 2.1.1 Test Items | 11 |
| 2.1.2 Test Result & Analysis..... | 12 |
| 2.1.3 Save Test Results | 14 |
| 2.1.4 Test Log | 14 |
| 2.2 Noise Level | 16 |
| 2.2.1 Execute test | 16 |
| 2.2.2 Test Analysis | 17 |
| 2.3 3D Module of Touch Signal..... | 18 |
| 2.4 Environmental Reference (ER) | 20 |

| | | |
|-----------|--|-----------|
| 2.4.1 | Checking Status | 20 |
| 2.4.2 | Rebuild the Environmental Reference | 21 |
| 3. | PARAMETERS..... | 24 |
| 3.1 | Main Features..... | 24 |
| 3.2 | Interface Parameters..... | 26 |
| 3.2.1 | Change Interface | 26 |
| 3.2.2 | I2C Address..... | 26 |
| 3.2.3 | Baud Rate | 26 |
| 3.3 | Panel Parameters | 26 |
| 3.3.1 | Choosing from a Preset | 27 |
| 3.4 | Finger Parameters | 27 |
| 3.4.1 | Sensitivity | 28 |
| 3.4.2 | Touch Points..... | 28 |
| 3.4.3 | Sample Rate Limit..... | 28 |
| 3.4.4 | Touch Debounce..... | 29 |
| 3.4.5 | De-jittering | 30 |
| 3.5 | Palm Parameters | 32 |
| 3.5.1 | Palm Rejection..... | 32 |
| 3.5.2 | Palm Size..... | 32 |
| 3.5.3 | Palm Detection on Edge | 33 |
| 3.5.4 | Disabled Range near Palm..... | 33 |
| 3.5.5 | Palm up De-bouncing | 34 |
| 3.6 | Base Parameters..... | 34 |
| 3.6.1 | Base Update Modes | 35 |
| 3.6.2 | Standard Base Update Configurations | 35 |
| 3.6.3 | Keep Base Update (KBU) Configurations..... | 37 |
| 3.7 | Water Handling | 38 |

| | | |
|-----------|--|-----------|
| 3.7.1 | Choosing a Water Handling Solution..... | 38 |
| 3.7.2 | Tweaking Water Performance Mode | 39 |
| 3.8 | Anti-Noise Parameters | 42 |
| 3.8.1 | Configure Low Pass Filter | 42 |
| 3.8.2 | Configure Frequency Hopping..... | 42 |
| 3.9 | Coordinate Parameters | 42 |
| 3.9.1 | Calibration | 43 |
| 3.9.2 | Edge Compensation..... | 46 |
| 3.9.3 | Orientation | 46 |
| 3.10 | Power Parameters..... | 47 |
| 3.10.1 | USB Interface Configurations | 48 |
| 3.10.2 | Idle Mode Settings..... | 49 |
| 3.11 | Miscellaneous Features | 49 |
| 3.11.1 | Watch Dog..... | 50 |
| 3.11.2 | Event Notify..... | 50 |
| 3.12 | Driving IC Parameters..... | 51 |
| 3.12.1 | High Voltage | 51 |
| 4. | FIELD UPDATE | 52 |
| 4.1 | Load Firmware File | 52 |
| 4.2 | Update Firmware..... | 52 |
| 5. | DRAW | 54 |
| 6. | DEBUG | 58 |
| 6.1 | Reset Touch Panel | 58 |
| 6.2 | Log Current Device Information | 59 |
| 6.3 | Show Notifications of Water Detection..... | 59 |
| 6.4 | Touch and Halt | 59 |

1. Introduction

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

1.1 Supported Systems

The PenMount PCIUtility can operate in the operating systems listed below:

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

1.2 Supported Devices

The devices and interfaces supported by the PenMount PCIUtility are as follows:

| Series | Control Board | USB | RS-232 / UART | I ² C 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 | | |

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

² The I²C interface is supported by connecting through the PMT101 Adapter.

| | | | | |
|------------------------------|---------|---|---|---|
| | 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 |
| | PM2300 | v | v | v |
| | PM2351 | v | | |
| PenMount P2-08 Series | PM1310 | v | v | v |
| | PM1410 | v | v | v |
| | PM1415 | v | | v |
| | PM1710 | v | v | |
| | PM1711 | v | | v |
| | PM1715 | v | | v |

1.3 Utilities Summary

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

| Utility | Usage | Applied Version | Notes |
|------------|---------------------------------|-----------------|-------|
| PCIUtility | Advanced firmware configuration | 1.2.x | |

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

1.4 Checking Utility Information

Application version, and other useful information are listed under the “Information” page.

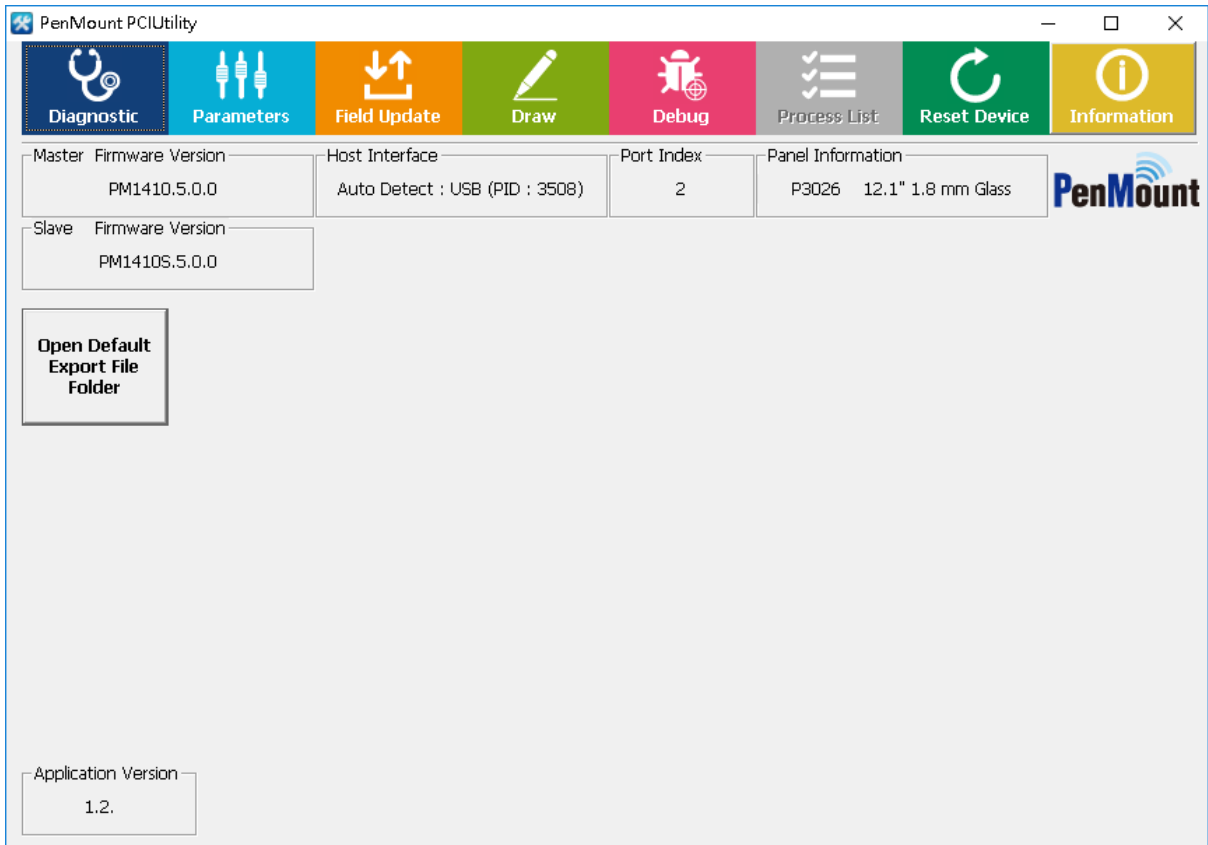


FIGURE 1

1.4.1 Device Information

The connected device information consists of several parts:

- Master / Slave Firmware Version
It is the firmware version of the connected device
- Host Interface
It is the connected interface. Most P2-06 and P2-08 devices supports interface auto detection.
- Port Index
It is the connected port index. For example, when using UART / RS-232 interface, the Port Index 1 means that device is connected to COM1.
- Panel Information
It is the configuration of panel size and cover lens thickness.

1.4.2 Application Version

It is the version of the utility.

1.4.3 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

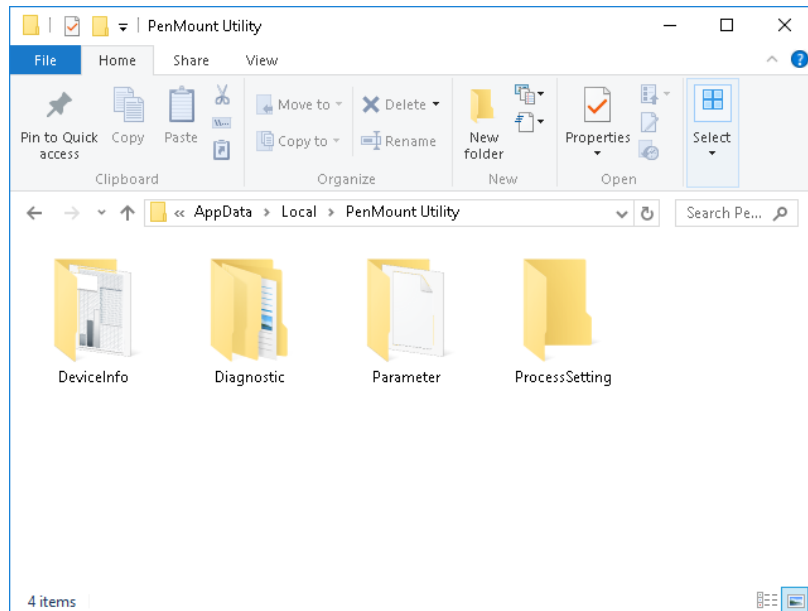


FIGURE 2

1.5 Refresh and Reset Device

If device is not connected when utility starts up, it can still automatically detects 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.

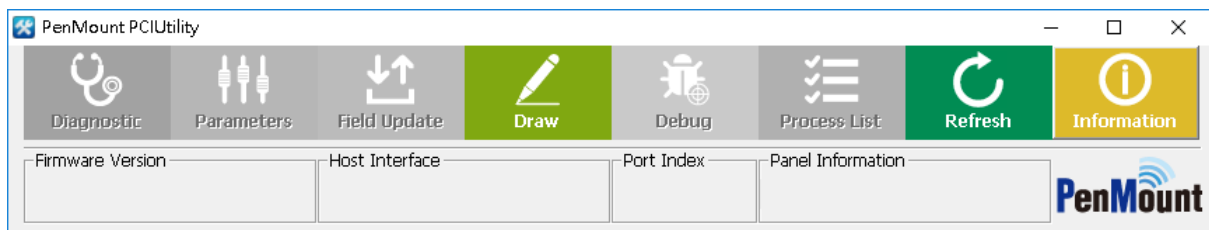


Figure 3

1.6 Special Notice

1. The 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.

2. Diagnostic

The 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
- Environmental Reference: allow user to rebuild the reference data used for adjusting environmental changes. This feature is only supported in some device firmware versions.

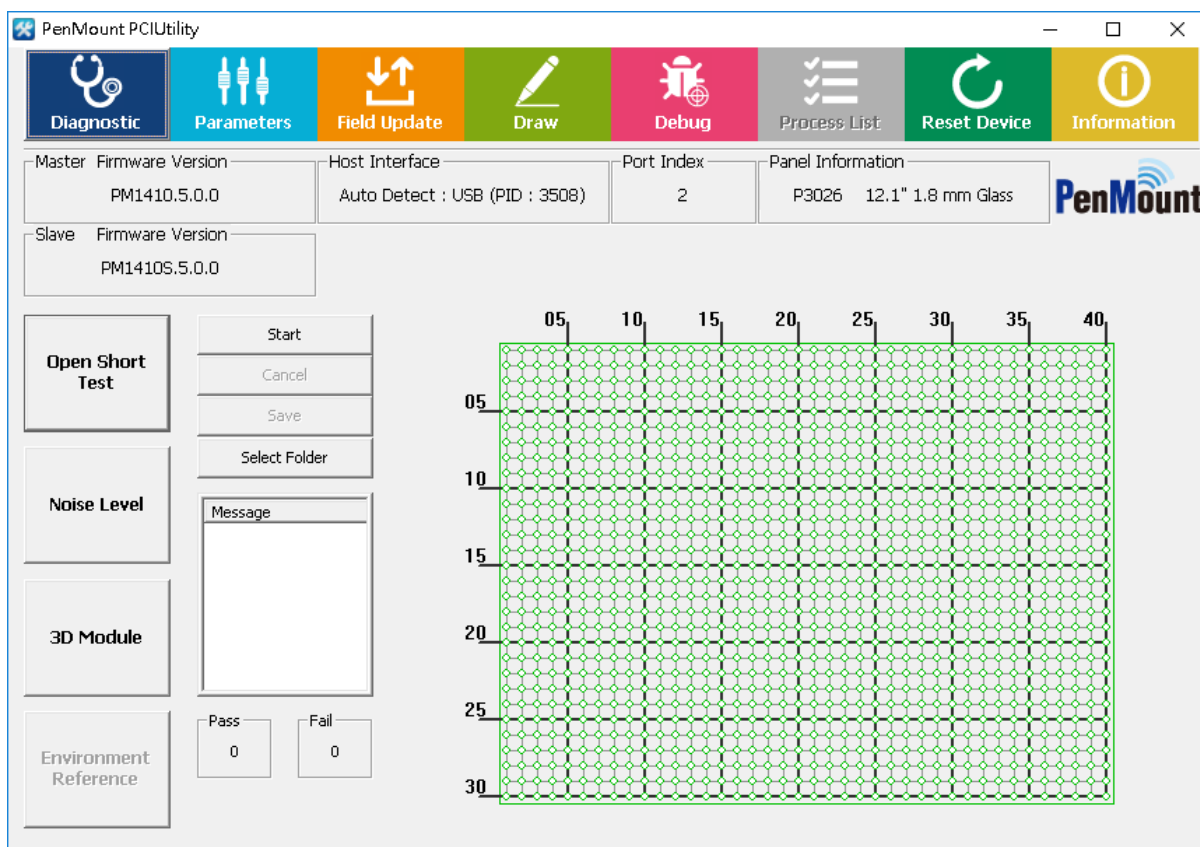


FIGURE 4

Please notice that you are using a customized PenMount firmware version, please use the corresponding PCIScan utility for advanced inspection.

2.1 Open Short Test

The “Open Short” tests sensor and driver pins for possible open / short issues. To make sure the accuracy of the test, please avoid any touch action after the test is running.

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



FIGURE 5

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

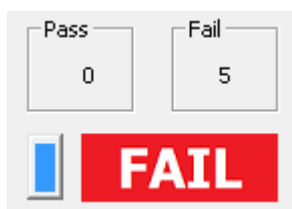


FIGURE 6

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.

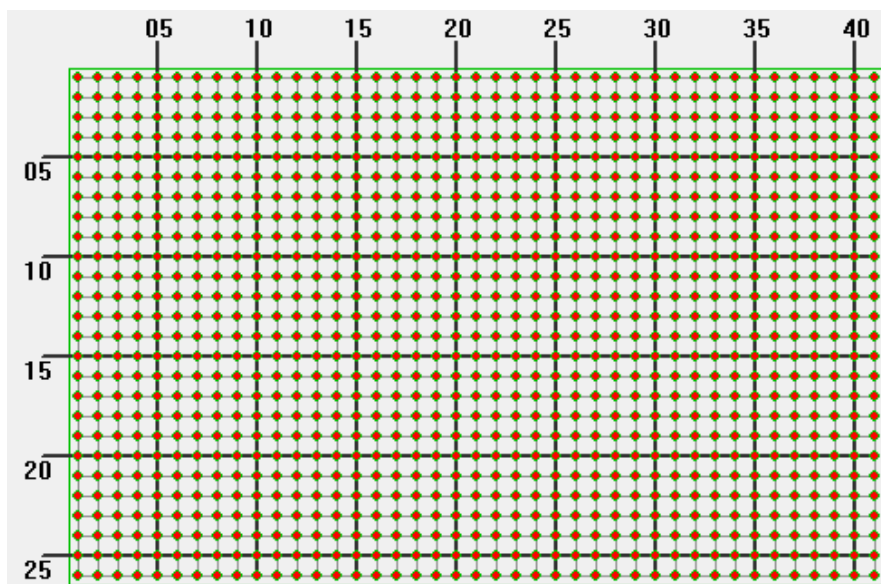


FIGURE 7

2.1.1 Test Items

When diagnose starts, the program measures raw data of every nodes on touch panel. Three sets of raw data are collected for signal testing. The primary test items are indicated 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 LevelC, 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 LevelE, 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 be too much. If the raw data of over one third of the nodes of the two neighboring sensor pins is larger than LevelE, 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 if raw data is negative. |
| Absolute Data Check | Checks if raw data is within valid range. |

2.1.2 Test Result & Analysis

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

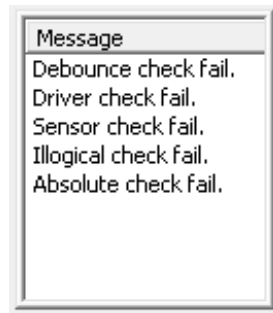


FIGURE 8

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 |
| Self check fail | Self-Signal 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 |

After the test is completed, the nodes with abnormal signals will be immediately indicated. They are usually classified as 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.

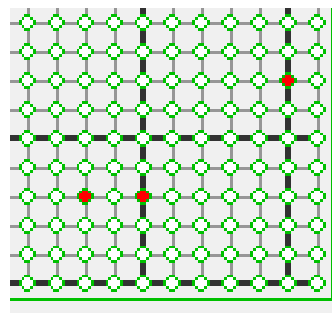


FIGURE 9

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.

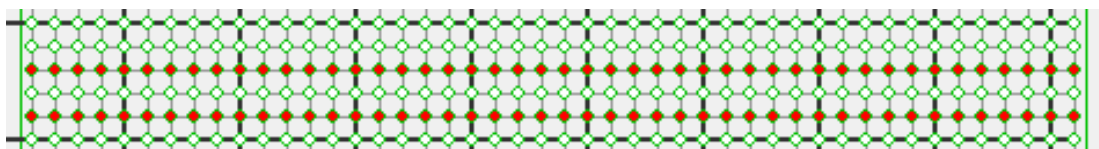


FIGURE 10

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 the target folder for saving test results.

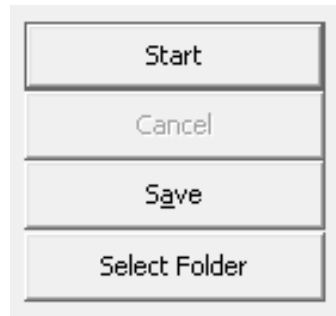


FIGURE 11

Then, the dialogue box below will appear. The user has to 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 the name \$ProductNumber_ \$BatchNumber_ \$Month\$Day\$Hour\$Minute\$Second”.csv.

2.1.4 Test Log

The program will save the detected signal in the format of CSV file. The 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.

| | A | B | C | D | E |
|----|--|---|---|---|---|
| 1 | | | | | |
| 2 | Write Log Time : 2016/12/24 19:07:09 | | | | |
| 3 | | | | | |
| 4 | Application Version : PenMount PCIMSet 2.2.0 | | | | |
| 5 | | | | | |
| 6 | Function : Open Short Test | | | | |
| 7 | | | | | |
| 8 | Master Version : PM1410.5.0.0 Boot : 6.2 | | | | |
| 9 | Slave Vers PM1410S.5.0.0 Boot : 6.2 | | | | |
| 10 | | | | | |
| 11 | Panel Information : P3026 12.1" | | | | |
| 12 | Cover Lens : 1.8 mm Glass | | | | |

FIGURE 12

- 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 13

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.

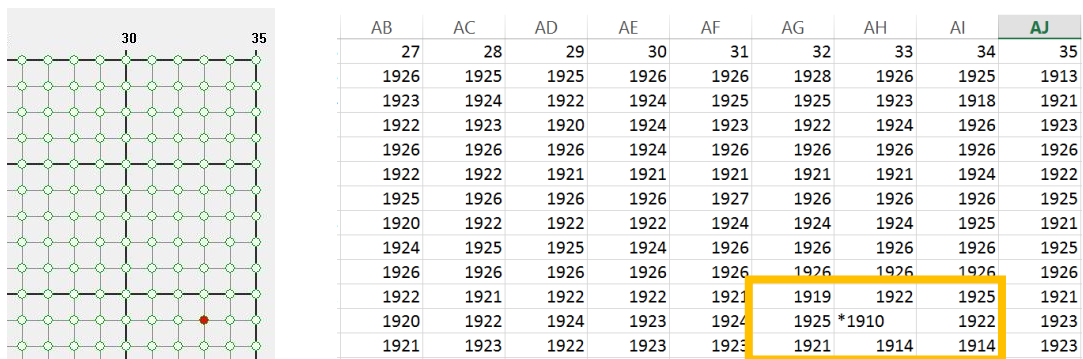


FIGURE 14

- 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.

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 a large fluctuations will be produced. The program will display the “noise level” interface:

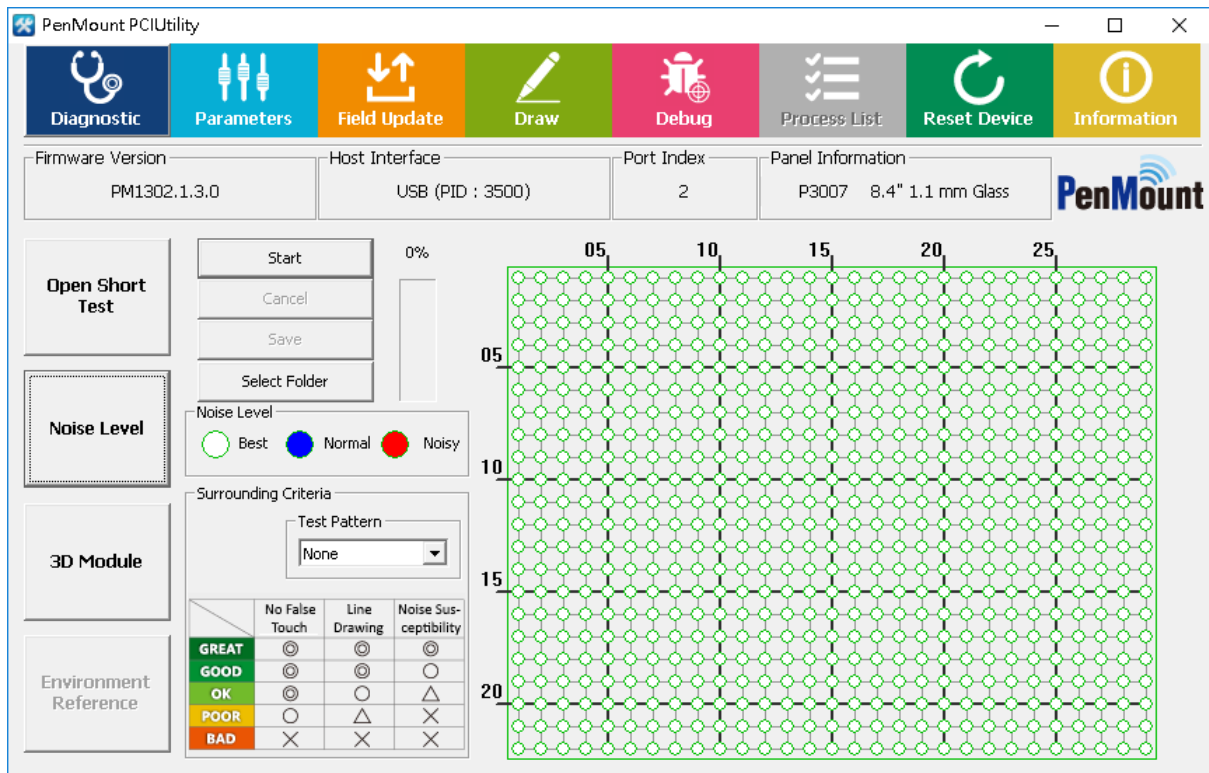


FIGURE 15

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 temporarily be disabled during the test. Usually the test takes longer when using larger panel sizes. Please click “Cancel” if want to abort for using other features.

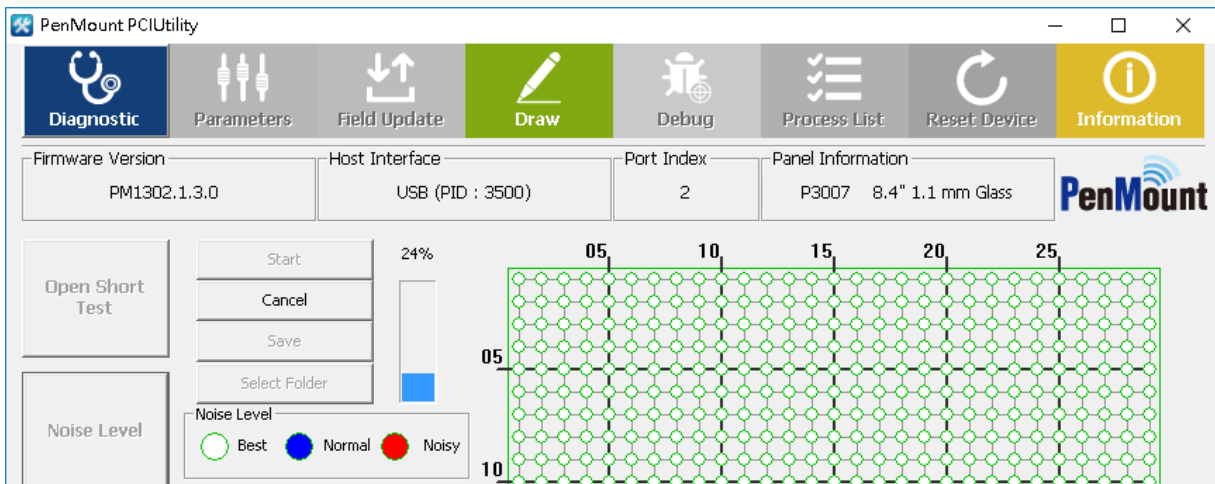


FIGURE 16

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

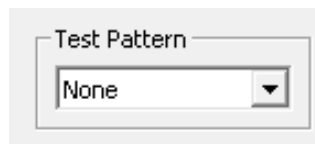


FIGURE 17

2.2.2 Test Analysis

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

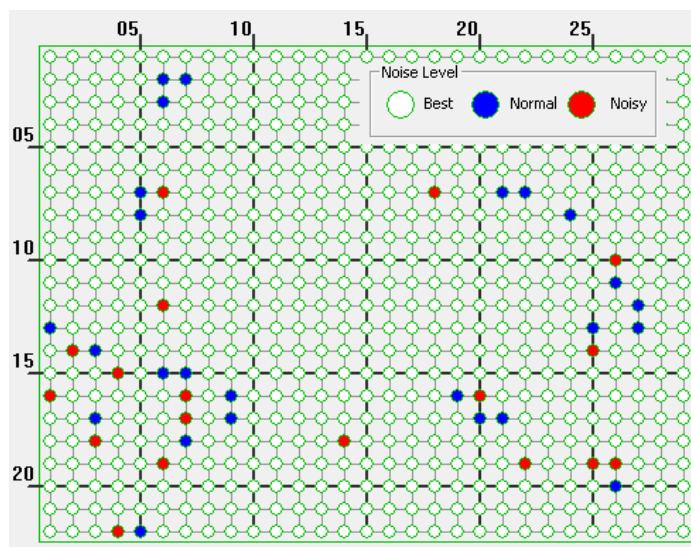


FIGURE 18

The right side of the program indicates the degree of signal changes in the nodes on the touch panel. The node marked with a red circle indicates a greater change in the signal value measured on this node and it may be unstable.

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

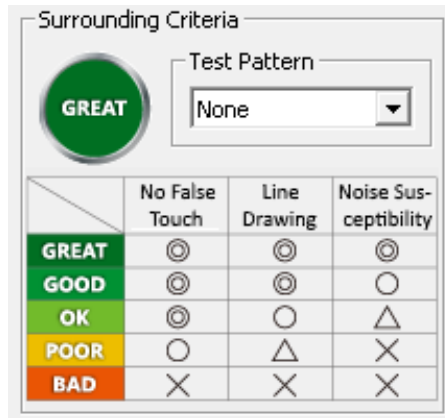


FIGURE 19

Generally speaking, please try to lower the noise influence by increasing the air gap between touch panel and LCD, and let the surrounding criteria to be 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 signal. Please prepare a mouse or keyboard before launching this feature, because touch will be disabled during the whole process.

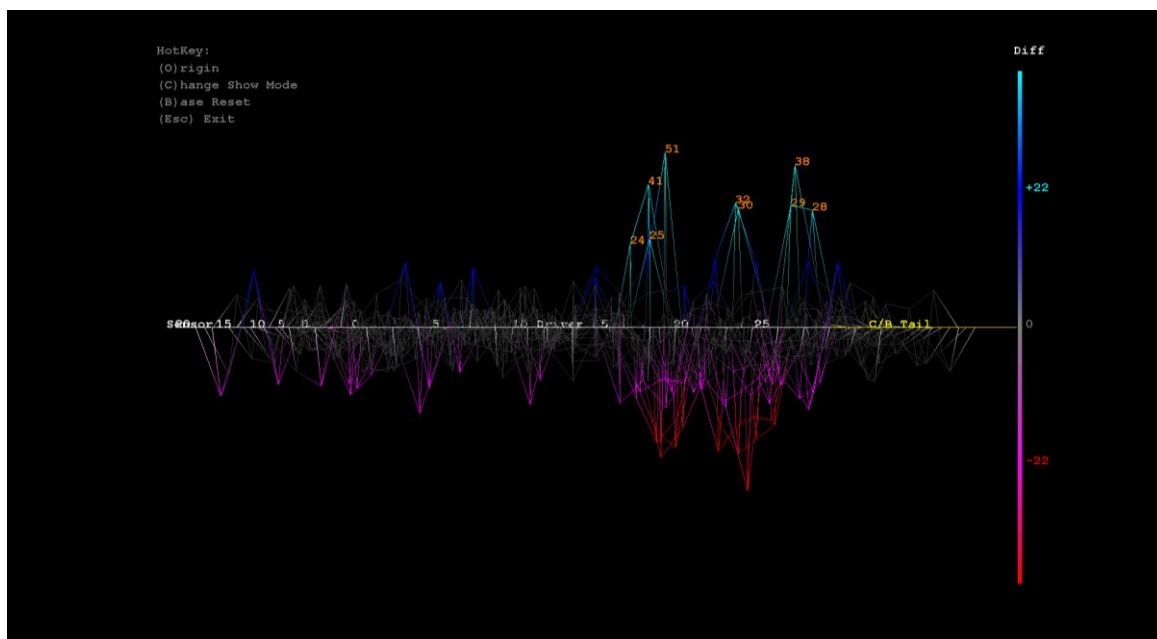


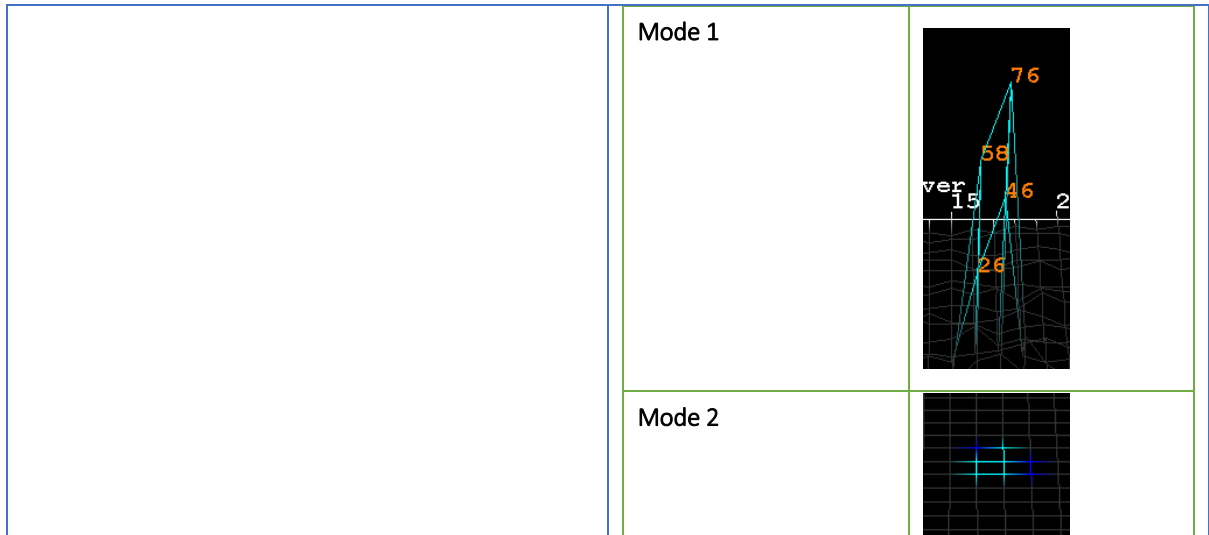
FIGURE 20

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 hotkey listed in the table below.

| Colors | Actions |
|--------|---|
| top | 5 degree X axis counterclockwise rotation |
| down | 5 degree X axis clockwise rotation |
| left | 5 degree Y axis counterclockwise rotation |
| right | 5 degree Y axis clockwise rotation |
| + | Zoom in |
| - | Zoom out |
| O | Restore to default angle: 225 degree X axis counterclockwise rotation, 270 degree Z axis clockwise rotation |
| B | Reinitialize the base signal |
| C | Change displayed mode. |
| | Mode |



2.4 Environmental Reference (ER)

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

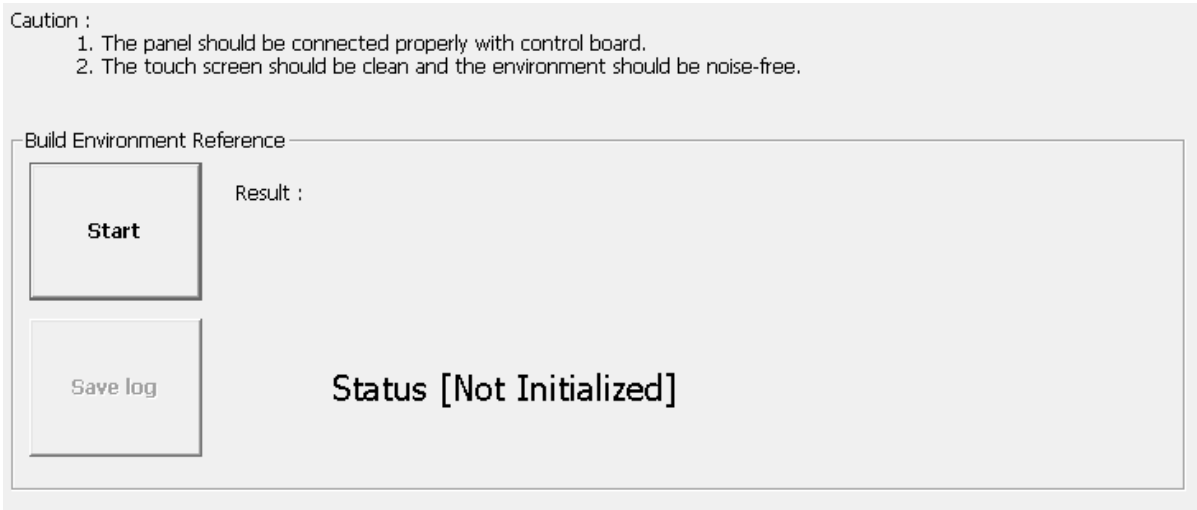


FIGURE 21

2.4.1 Checking Status

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

| Status | Description |
|------------------------|---|
| Active | The environmental reference is built and in use |
| Not Initialized | The device does not have a valid environmental reference, please press start to rebuild it. |

In the following cases, the utility will check the reference status and prompt for rebuilding 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.

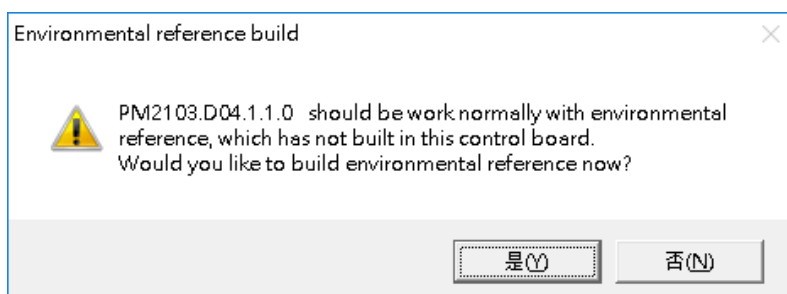


Figure 22

2.4.2 Rebuild the Environmental Reference

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

1. The touch panel is connected to control board properly.
2. The touch screen surface is clean, without any water, hand, or other obstacles on it.
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%.

Work in progress. Please wait....1%

Figure 23

In case that the touch panel is not connected properly, or build environmental is too noisy, the utility will display a warning message after build completes. User need to confirm whether to accept the build result by pressing Yes or No button.

Environment too noisy. Build anyway ?



Figure 24

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

3. Parameters

Parameters provides an interface to flexibility fine tuning the PenMount firmware settings.

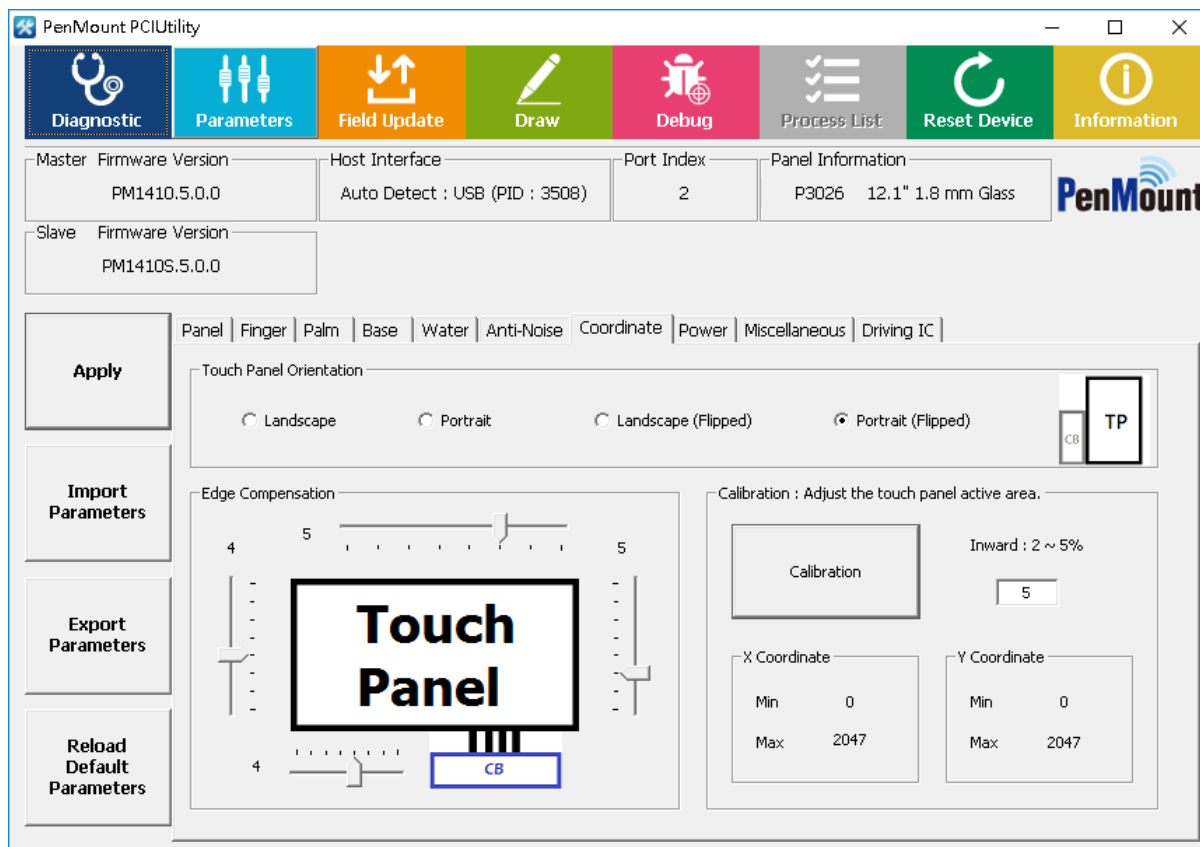


FIGURE 25

3.1 Main Features

The Parameters configuration interfaces includes following features.

1. The Parameters Panel

The 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 feature and side effect will show up for quick reference.

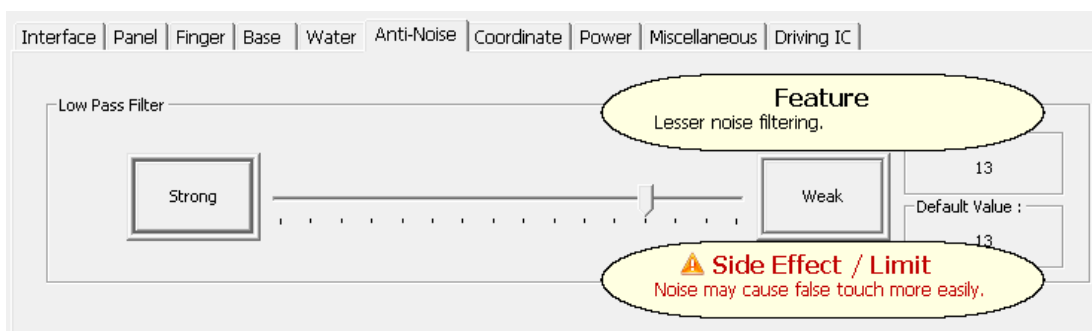


FIGURE 26

- Grayed out Options

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



FIGURE 27

- Hidden Options

Any parameter that is not displayed also indicates that it is not adjustable. In the following example, when using older firmware versions such as PM1410 V2.1, the utility only displayed “Panel”, “Finger”, “Water”, “Anti-Noise” and “Coordinate” tabs with a few adjustable options.

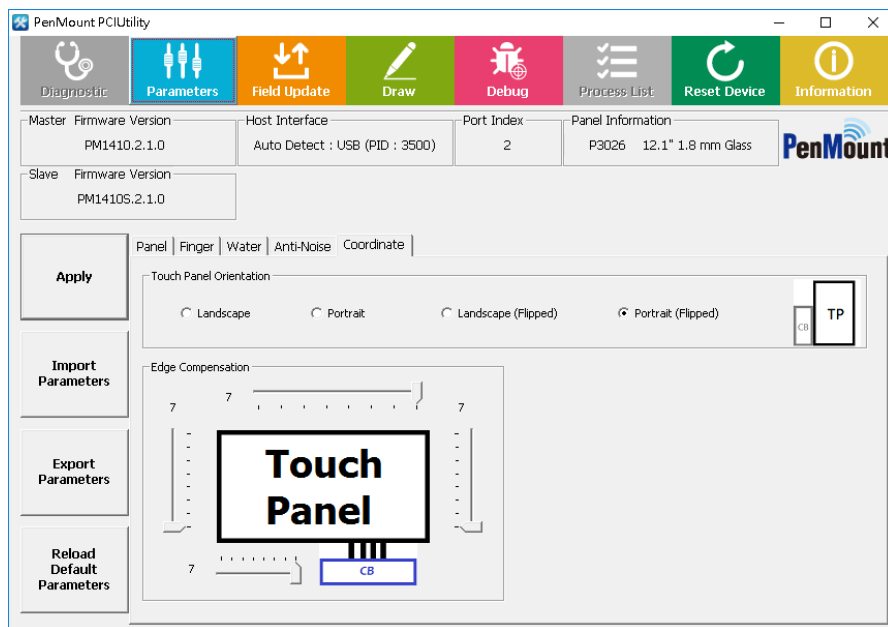


Figure 28

2. The Apply Button

Any parameter change needs to be saved by pressing the “Apply” button. This might also reset the device if necessary.

3. Import Parameters

Sets up device by selecting a “PMP” format file in the “Open” dialog window.

4. Export Parameters

Saves active device settings to a “PMP” format file. The “PMP” file stores binary data, and cannot be edited with text editors directly.

5. Reload Default Parameters

Reset all parameter changes and use the firmware default value. Some configurations, however, such as touch panel size and cover lens thickness will still be kept.

3.2 Interface Parameters

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



Figure 29

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 I²C connection. Please change this value if the default slave address 0x38 conflicts with other devices.

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

3.2.3 Baud Rate

The transfer baud rate when using UART/RS-232 connection. The supported values are listed in the drop down menu which are: 115200, 57600, 38400, 19200 and 9600.

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

3.3 Panel Parameters

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

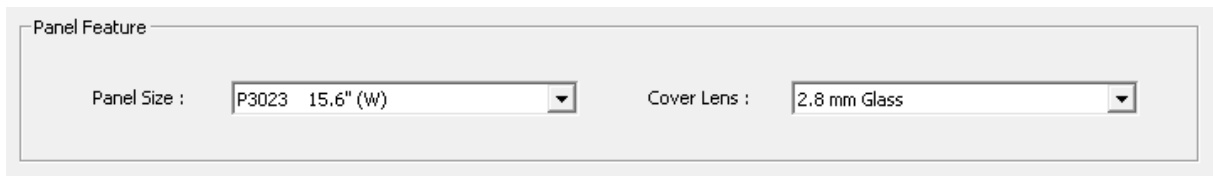


FIGURE 30

3.3.1 Choosing from a Preset

Standard PenMount firmware versions supports a number of panel sizes and cover lens thickness. By selecting a preset, firmware will apply different configurations on parameters.

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.

If your touch panel is not listed, please consider manually setting up the "Skip Driver" / "Used Driver" / "Skip Sensor" / "Used Sensor" combination.

2. Cover Lens

Each PenMount firmware supports several cover lens thickness parameters, and the standard ones are: 1.1mm, 1.8mm, and 2.8mm. Please choose one with the nearest thickness from the list.

For thicker cover lens support, PenMount provides special firmware that supports thicker cover lenses up to 6.0mm. Users can also manually change "Scan" parameters to increase sensitivity and stability.

3.4 Finger Parameters

The parameters in this tab are for tuning finger detection.

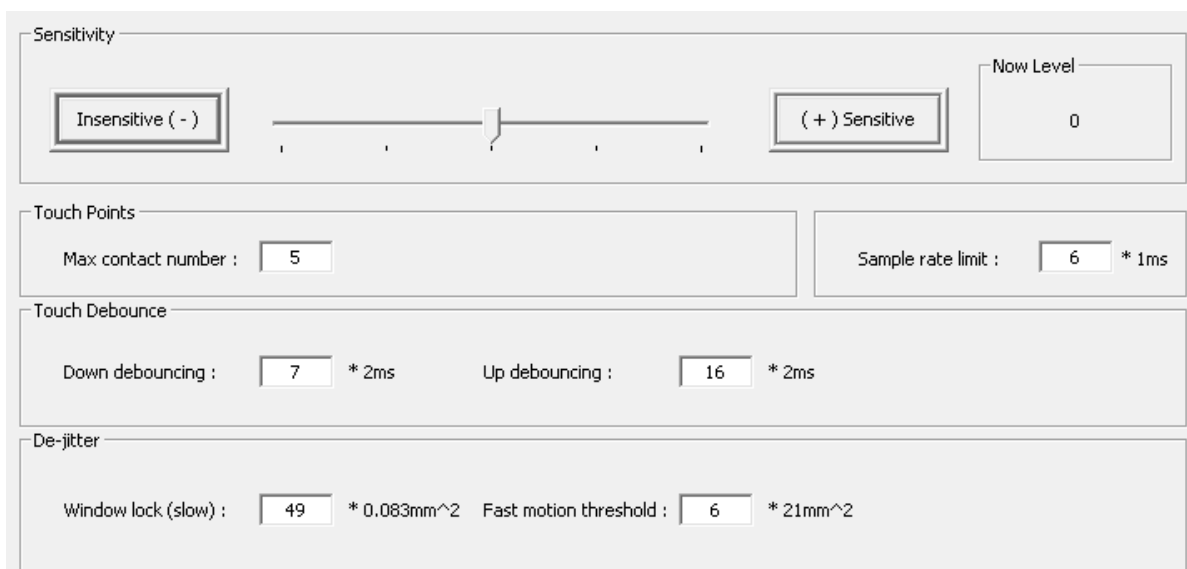


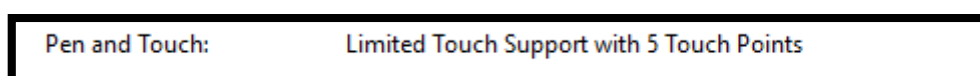
FIGURE 31

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, the sensitivity level is 0, setting to positive values increases the sensitivity and setting to negative values decreases sensitivity.

3.4.2 Touch Points

The maximum contact number limits how many touch contacts will be reported to the system. For example, if the value is 1, it means that 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 a 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, it might 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:

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

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 make sure that the signal is stable. The same situation applies when the value differences drop under the threshold.

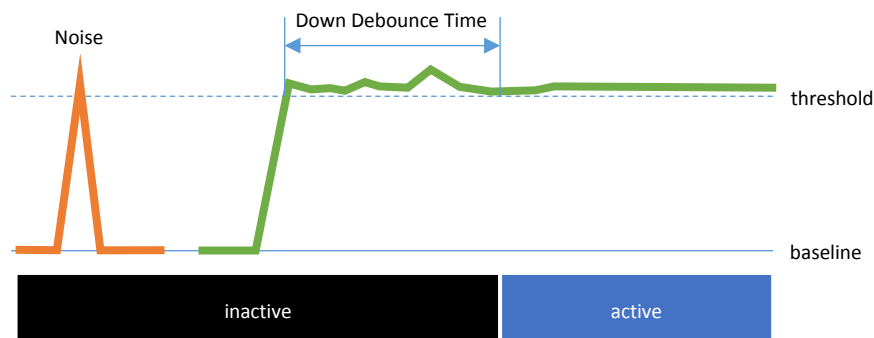


FIGURE 32

In some situations, the “Up Debouncing” 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 Debouce Time” ends. In this case, users can lower the “Up debounce” parameter to optimize for double click operations.

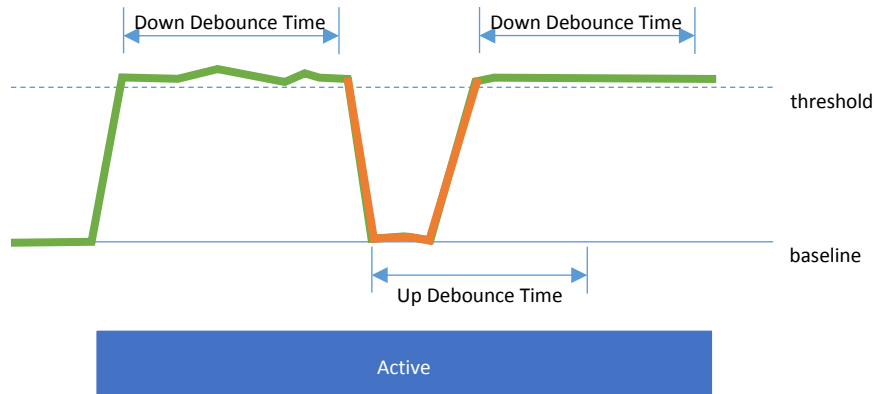


FIGURE 33

3.4.5 De-jittering

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

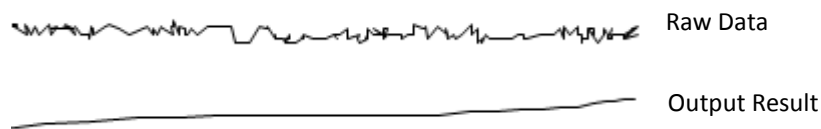


Figure 34

The “Window Lock” parameters are adjustable to control the 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 below:

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

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

$$\text{dist}^2 = dx^2 + dy^2$$

In the formula above, the dx and dy each represents the offset on x and y axis, and the resolution is the “pitch resolution” * X lines and “pitch resolution” * Y lines respectively³.

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

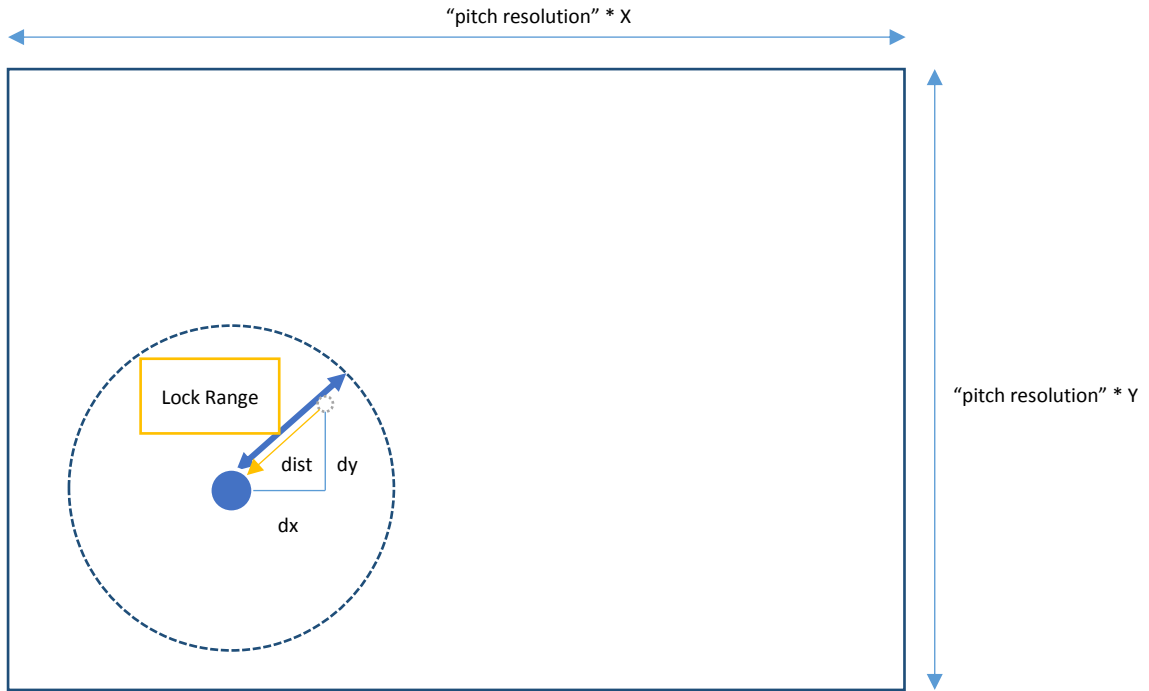


FIGURE 35

The parameter value can be set from 0 to 255. Larger values help avoid jitters 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 | \leq Window Lock (Slow) | Window Locked |
| Normal | $>$ Window Lock (Slow) \leq 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 might 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:

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

As mentioned earlier, the logical maximum is also 2047. If the parameter value is 9, it means that when moving 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.

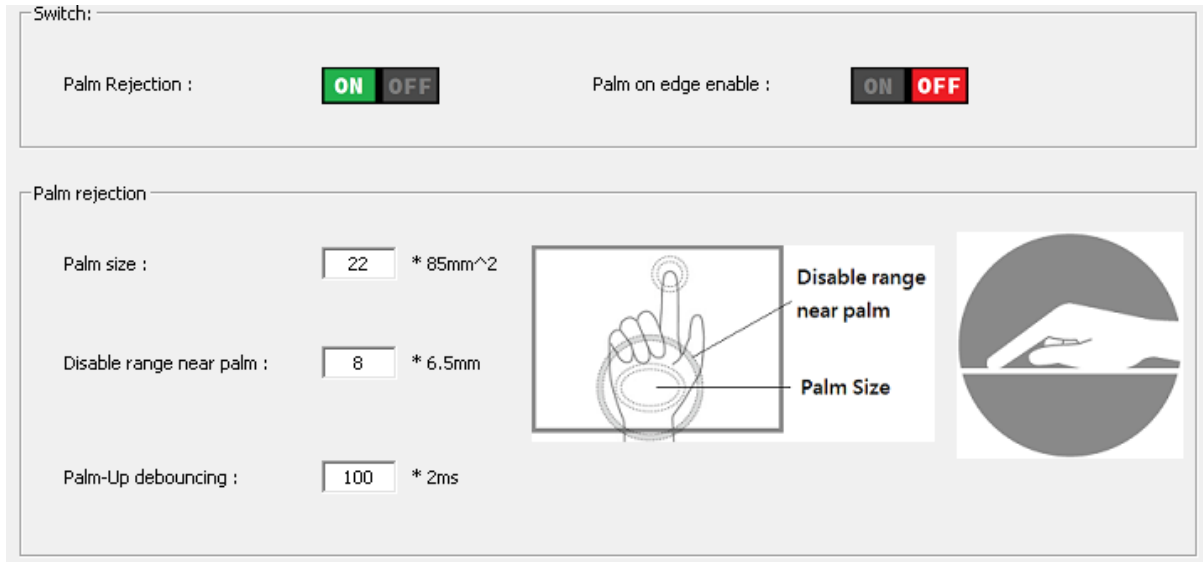


FIGURE 36

3.5.1 Palm Rejection

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

3.5.2 Palm Size

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

| | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| -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 37

To tune the palm size, please approximately count the nodes that will be covered by palm. Using a value smaller than the palm size can make firmware detects palm objects faster. However, the smaller this value, the easier an object is treated as a “palm.” It will also increase the chance of misjudgment when touching with multiple “fat fingers”.

3.5.3 Palm Detection on Edge

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



After turning on “Palm on edge enable”, the firmware will start checking any newly detected objects on the panel edge and will ignore 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 38

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.

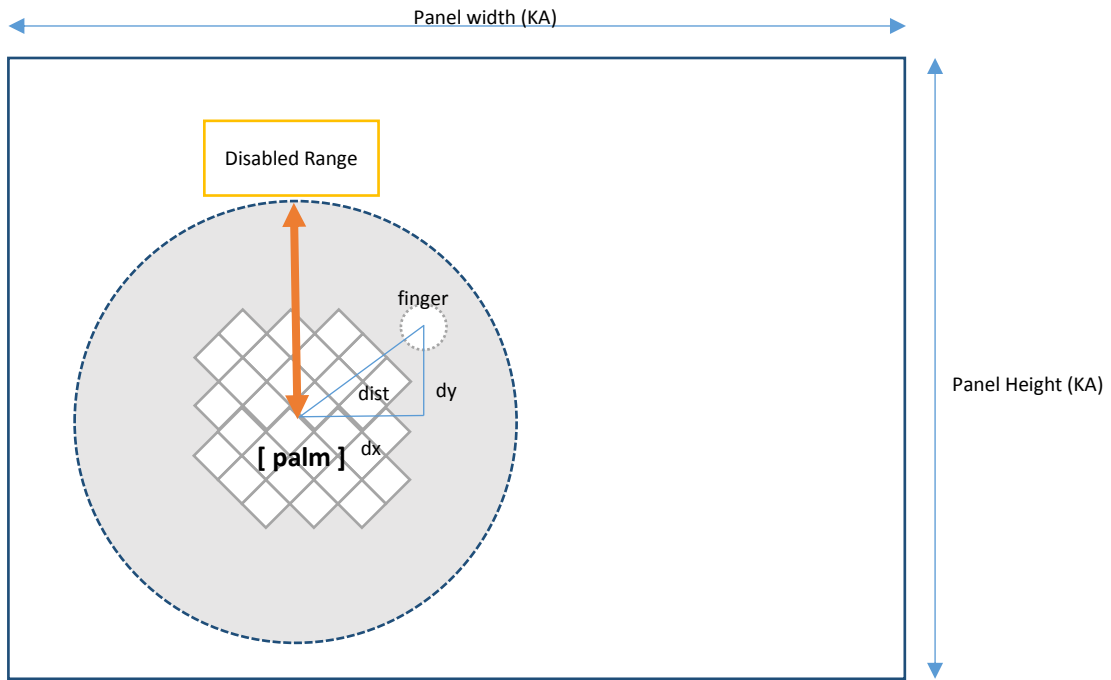


FIGURE 39

The “disabled range” can be calculated as below:

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

Any finger at a distance from the center of 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-denouncing” 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 not to set this value too long 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.

Base Update Pattern

Update driver number :

Start update delay time : * 100ms

Keep Base Update (KBU)

Inactive update delay count :

Active update delay count :

FIGURE 40

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 or not.

| 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.

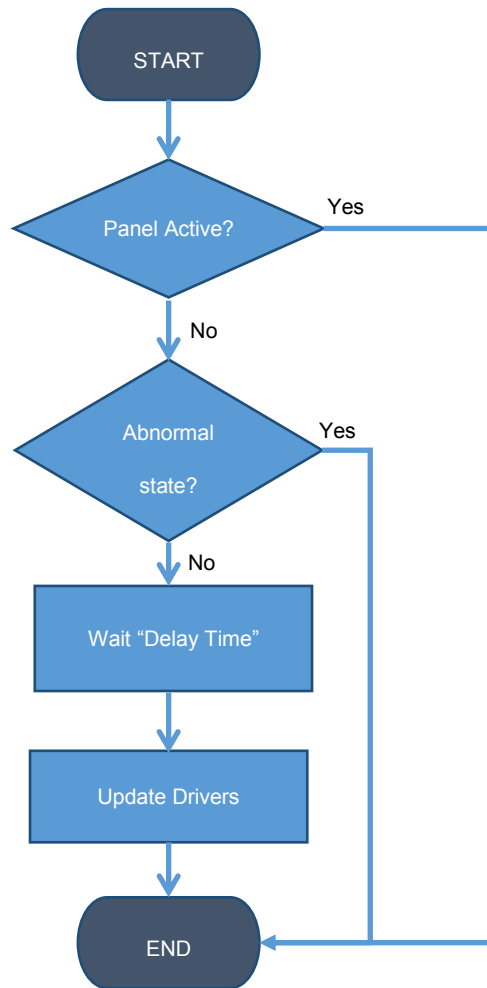


FIGURE 41

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.

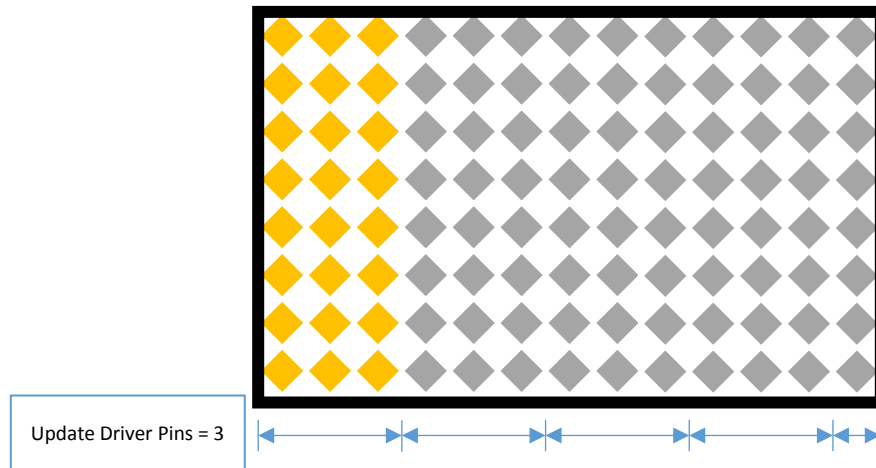


FIGURE 42

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.

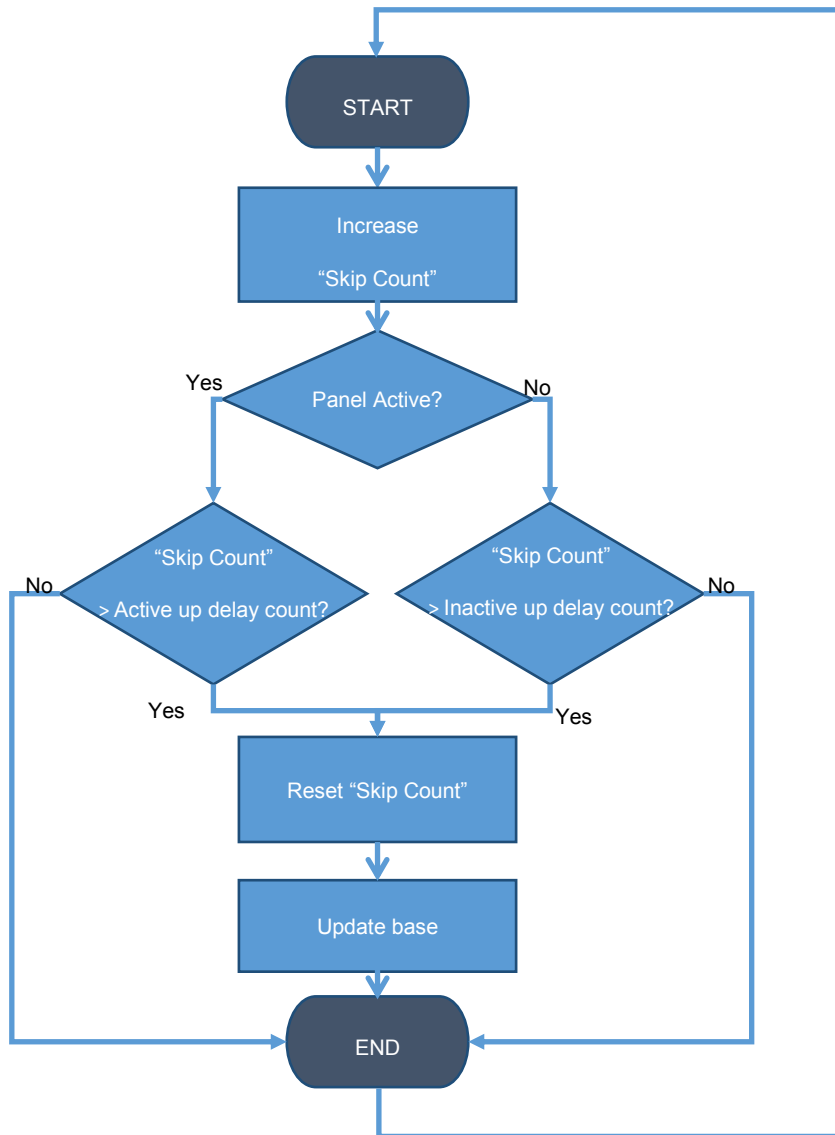


FIGURE 43

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 features including 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).

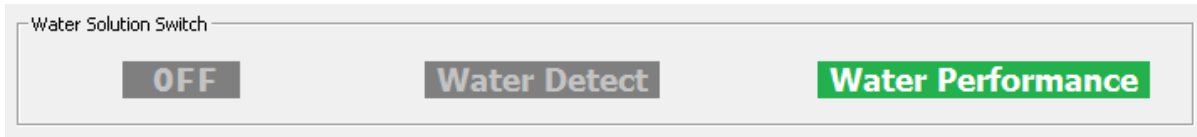


FIGURE 44

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

| Modes | Water Handling |
|-------------------------------|--|
| Water Detection Mode | Stop report any 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 keeps appearing, it will cause firmware to enter a “critical state” and single touch can still be utilized in the worst case.

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

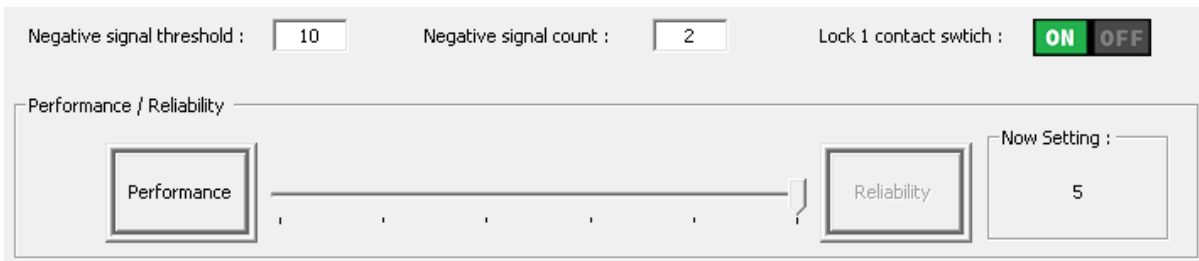


FIGURE 45

1. Lock 1 contact switch

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

2. Compensate Between Performance and Reliability

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

| Reliability Level | Description | Touch Mode | Use Negative Signal Check |
|-------------------|---|-----------------------------|---------------------------|
| 0 | Allows some false touches to retain touch performance | Multiple touch in dry area; | YES |

| | | | |
|------------------|--|--|--|
| 1 | Improve line drawing in dry areas with very little reliability drop | Switch to single touch mode after false touch detected | |
| 2 (recommend) | Good performance in dry areas | | |
| 3 (default) | Better reliability by locking to single touch | Locks to single touch mode if there is water | |
| 4 | Good for harsh environments with saline, metal, or other conductors on touch panel surface | | |
| 5 | Very strict water checking that sacrifices performance | | |

Unless there is extreme concern about 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 might 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 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 checking.

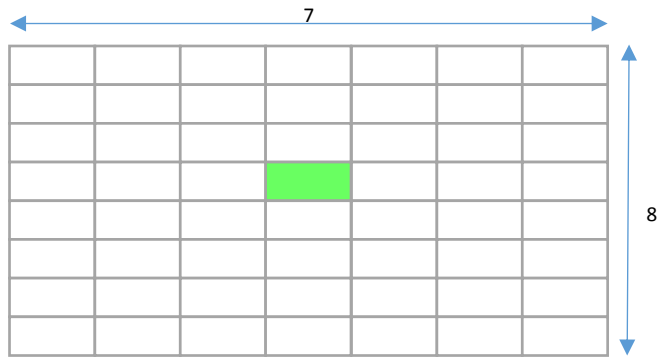


FIGURE 46

- Pass Sample

In the following sample, there is no negative signal around the touch point (in green color), 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 47

- 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 need further checking.

| | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|
| 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 48

Normally using the default parameter values is 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” to decrease the impact of noise.



FIGURE 49

3.8.1 Configure Low Pass Filter

PenMount firmware uses a low pass filter to eliminate the influence of noise jitter. Choosing a stronger value will cause slower signal changes.

Please note that 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 Configure Frequency Hopping

Currently PCI Parameters do not support manually configuring frequency hopping because the firmware will choose the best settings automatically.

3.9 Coordinate Parameters

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

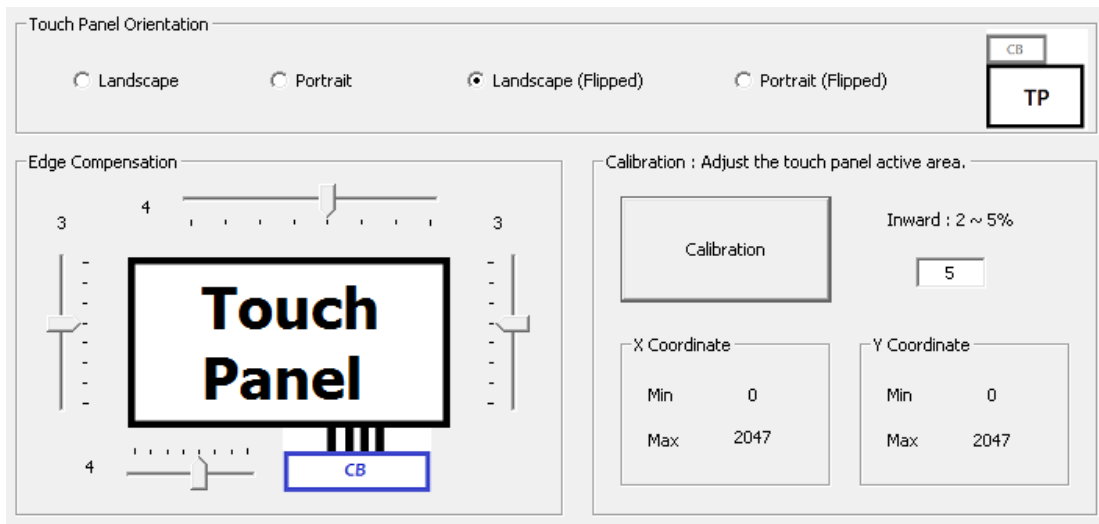


FIGURE 50

3.9.1 Calibration

Although the term “calibration” for some P-CAP 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, which means when touching the top left corner, it sends (0,0) as its position, and sends (2047, 2047) for the lower right corner.

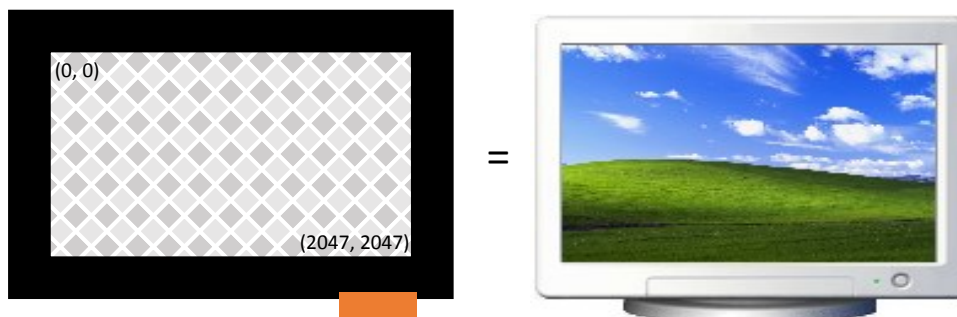


FIGURE 51

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).

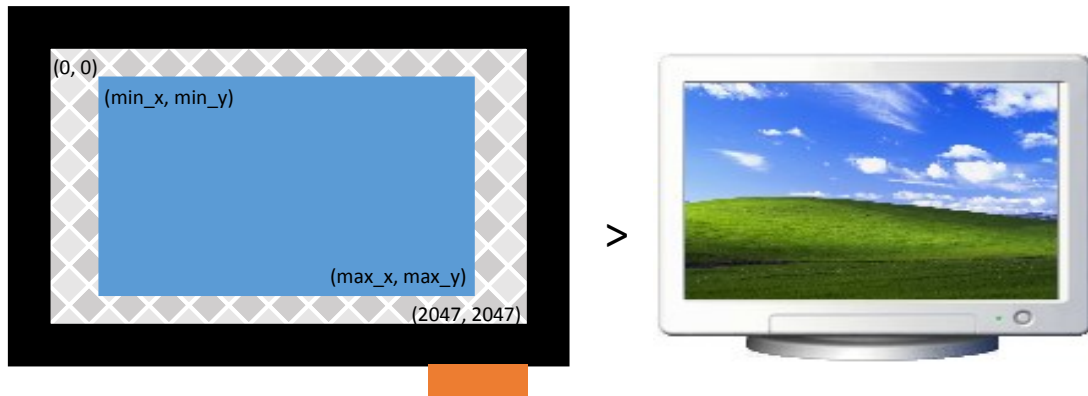


FIGURE 52

(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 as "min_x".

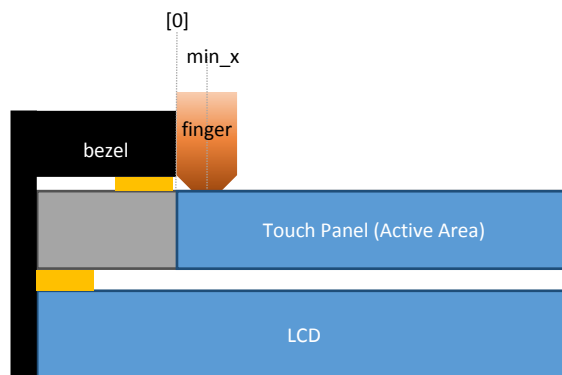


FIGURE 53

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 2047 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.

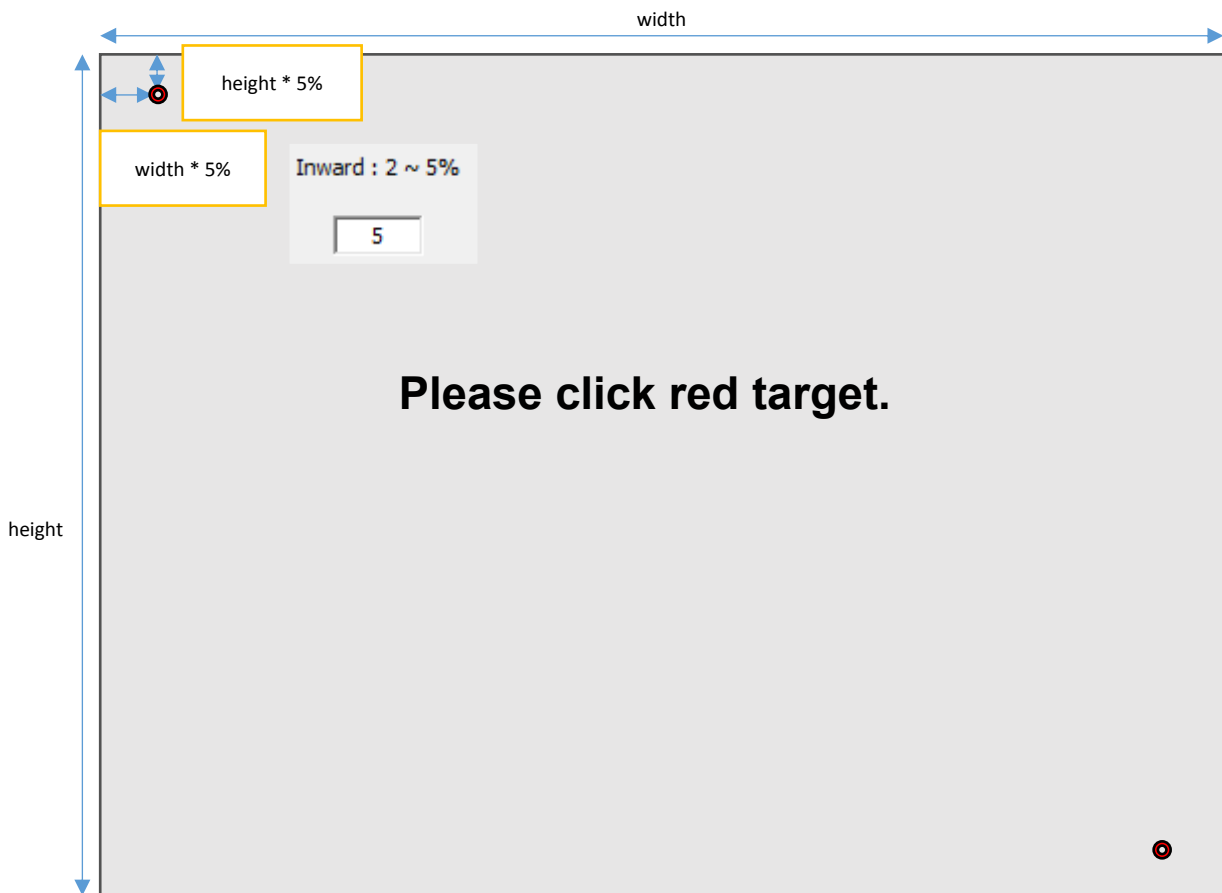


FIGURE 54

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

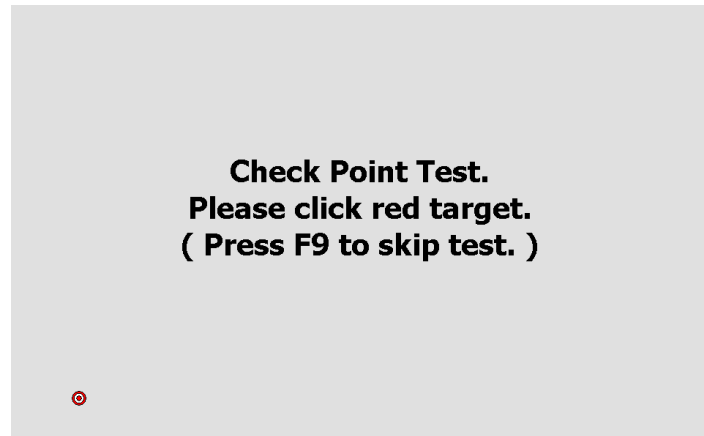


FIGURE 55

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.

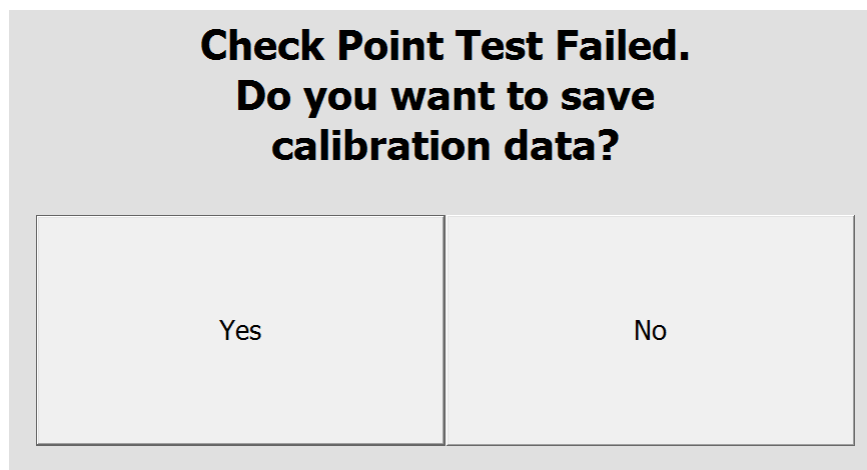


FIGURE 56

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 Orientation

The firmware supports four rotational degrees listed in the table below.

| Orientation | Rotational Degree (Counter Clockwise) |
|-------------|---------------------------------------|
| Landscape | 0 |

| | |
|---------------------|-----|
| Portrait | 90 |
| Landscape (Flipped) | 180 |
| Portrait (Flipped) | 270 |

Please notice that the 0 degree of a touch panel is defined as the diagram below, which “tail” is on the bottom side.

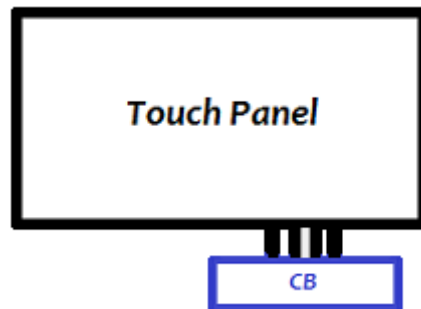


FIGURE 57

3.10 Power Parameters

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

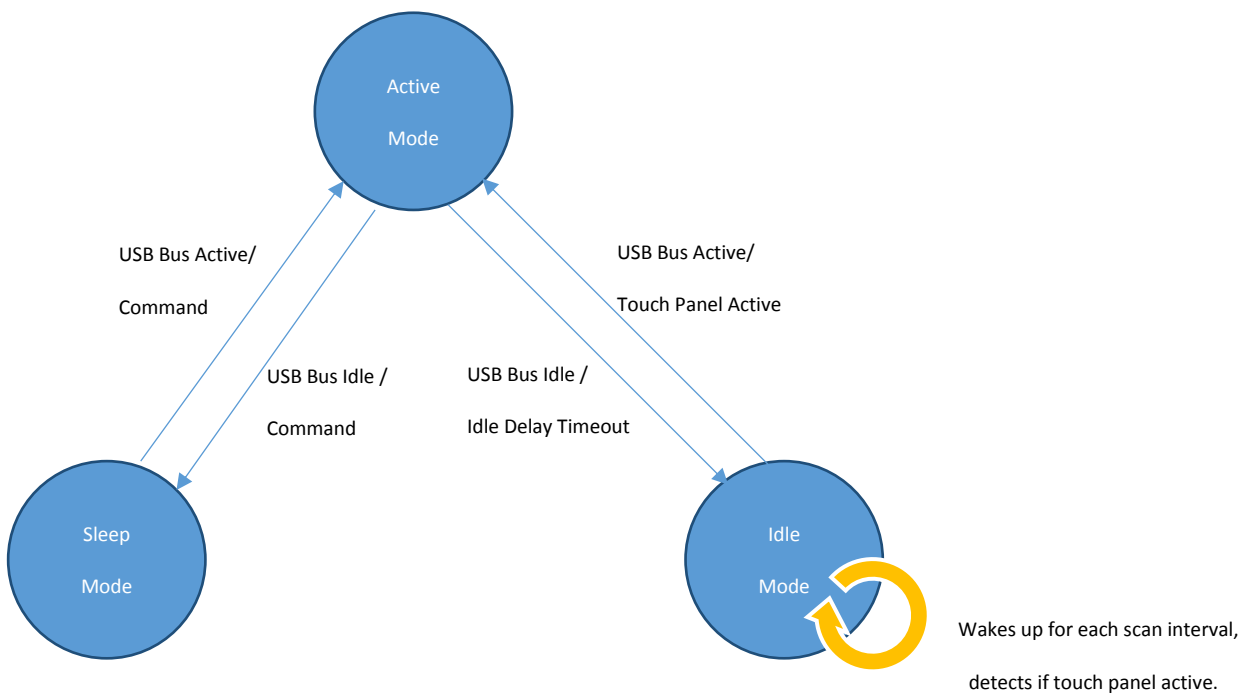


FIGURE 58

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



FIGURE 59

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 I²C 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 the “Device Manager”.

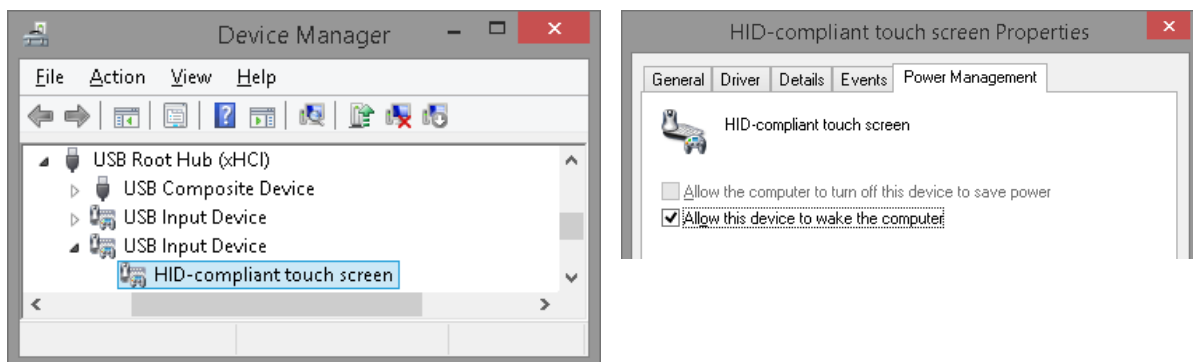


FIGURE 60

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 wakeup until system is resumed.

| Remote Wake-up \ USB States | 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.

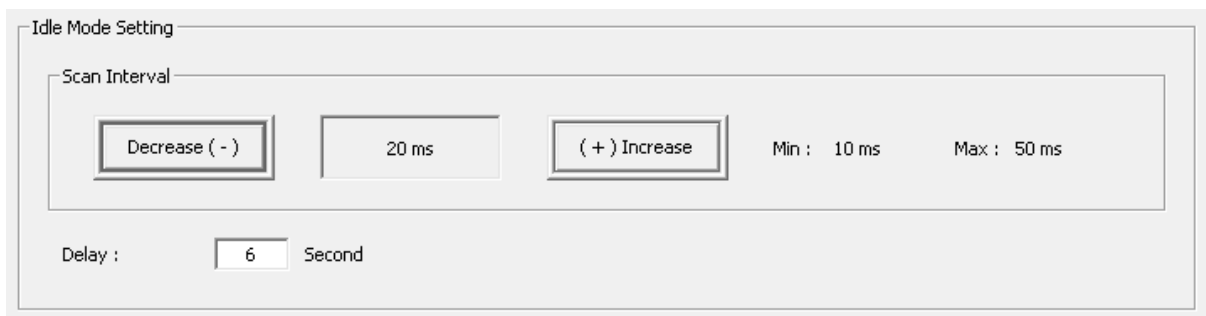
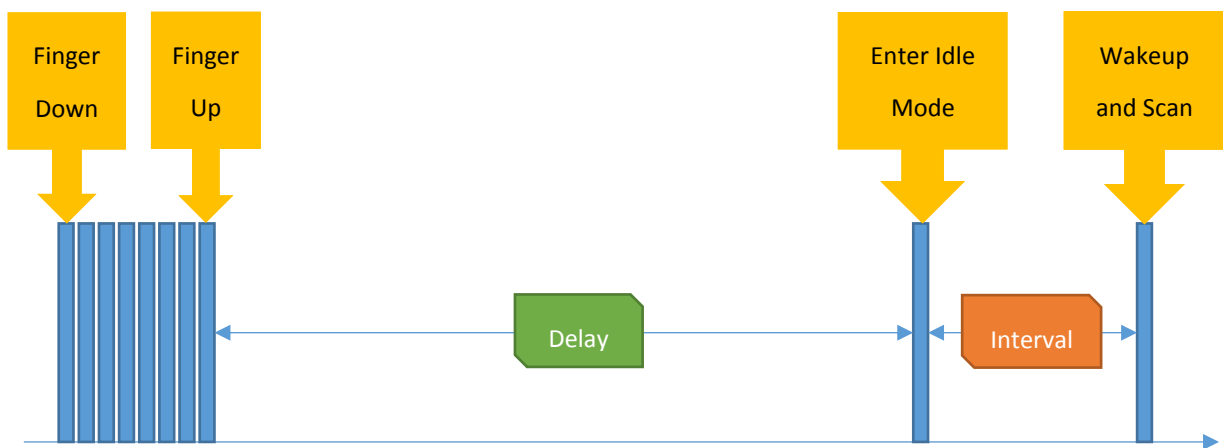


FIGURE 61



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

FIGURE 62

3.11 Miscellaneous Features

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

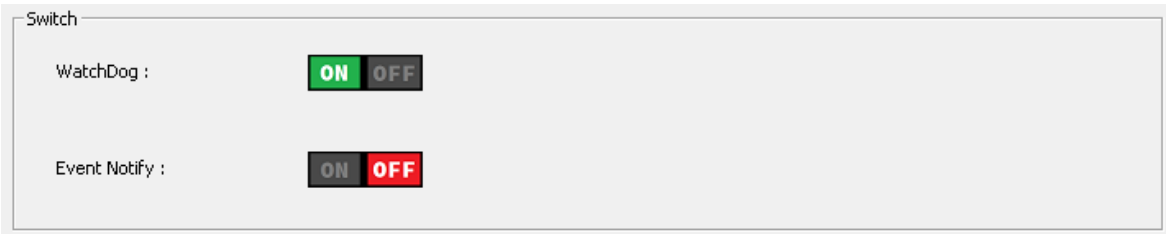


FIGURE 63

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 | Abnormal Notification | | | | | | | | | | | | | | | |
|--|-----------------------|------|------|------|------|------|------------|-----|-----|-----|-----|-----|-----|-----|-----|--|
| USB | 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. | | | | | | | | | | | | | | | | |
| RS-232 | State | 0 | 1 | 2 | 3 | 4 | 5 | | | | | | | | | |
| | Enter | 0x8A | 0x00 | 0x00 | 0x00 | 0x00 | <CheckSUM> | | | | | | | | | |
| | Leave | 0x0A | 0x00 | 0x00 | 0x00 | 0x00 | <CheckSUM> | | | | | | | | | |
| I2C | 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 in Debug Page, device will also send 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 the listed in the table below.

| Interface | Abnormal Notification | | | | | | | | | | | | | | |
|--|-----------------------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|--|
| USB | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | |
| | 0x01 | 0x8F | 0x00 | 0x00 | 0x00 | 0x01 | --- | --- | --- | --- | --- | --- | --- | --- | |
| Values of data byte 6 to 13 are not used and can be ignored. | | | | | | | | | | | | | | | |

| | | | | | | | |
|--------|------|------|------|------|------|------------|--|
| RS-232 | 0 | 1 | 2 | 3 | 4 | 5 | |
| | 0x8F | 0x00 | 0x00 | 0x00 | 0x01 | <Checksum> | |
| I2C | 0 | 1 | 2 | 3 | 4 | 5 | |
| | 0x8F | 0x00 | 0x00 | 0x00 | 0x01 | <Checksum> | |

3.12 Driving IC Parameters

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

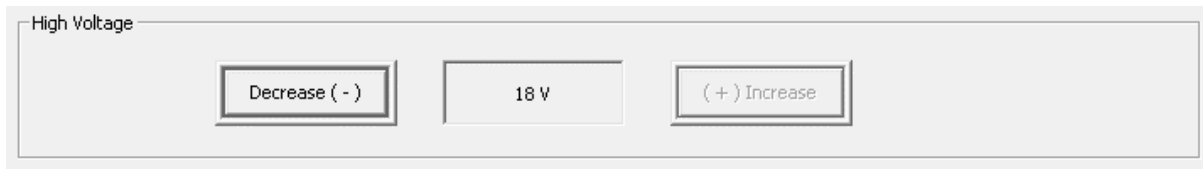


FIGURE 64

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.

4. Field Update

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

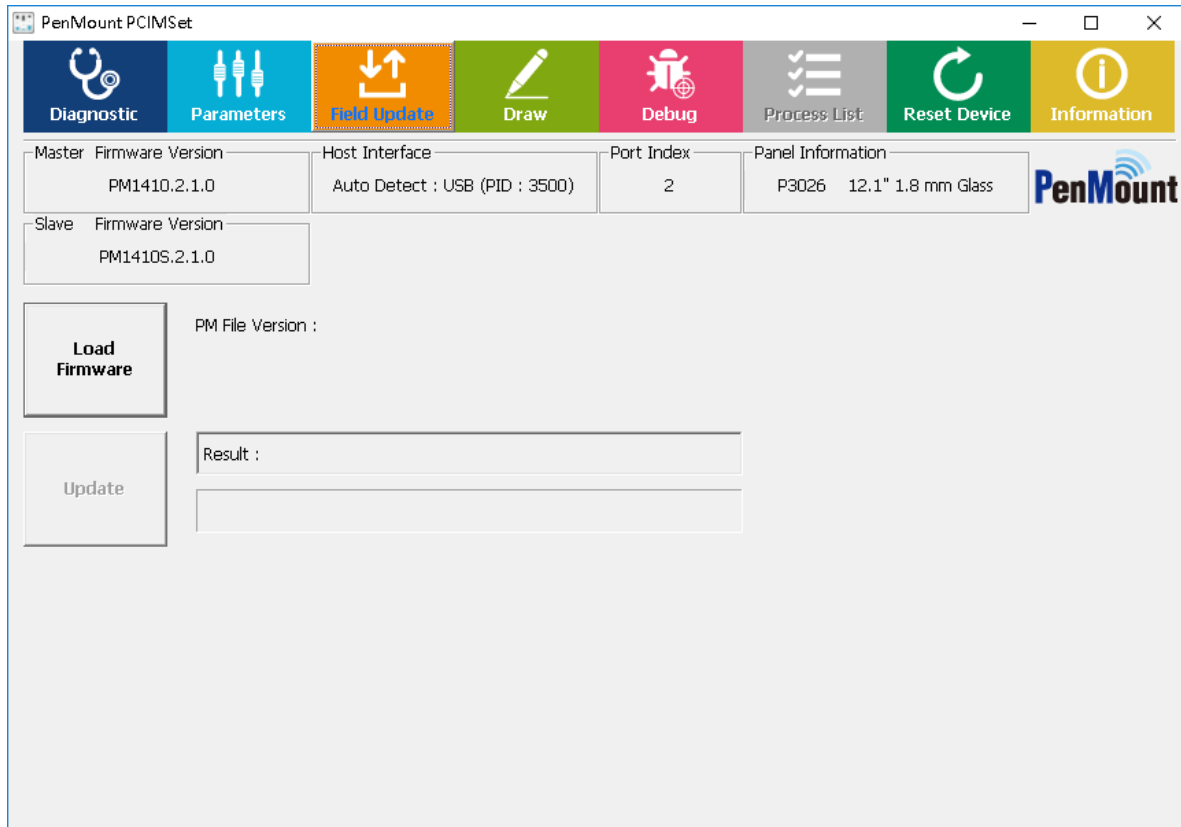


FIGURE 65

4.1 Load Firmware File

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

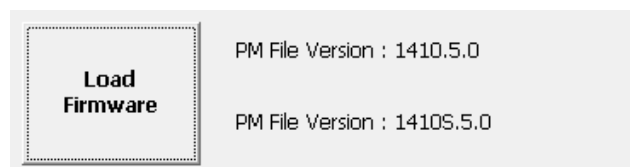


FIGURE 66

The format of the firmware supported by PenMount is suitable for “pmf format” for one single IC and “pmc format” for multiple ICs.

Please notice 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.

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

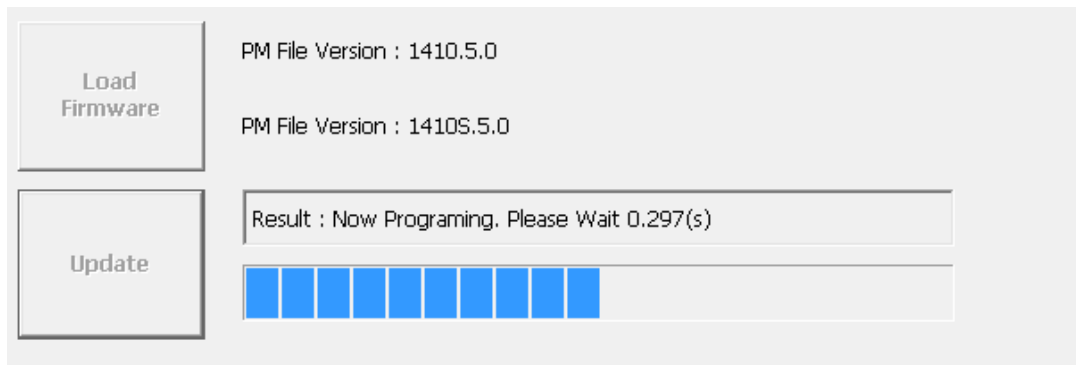


FIGURE 67

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.

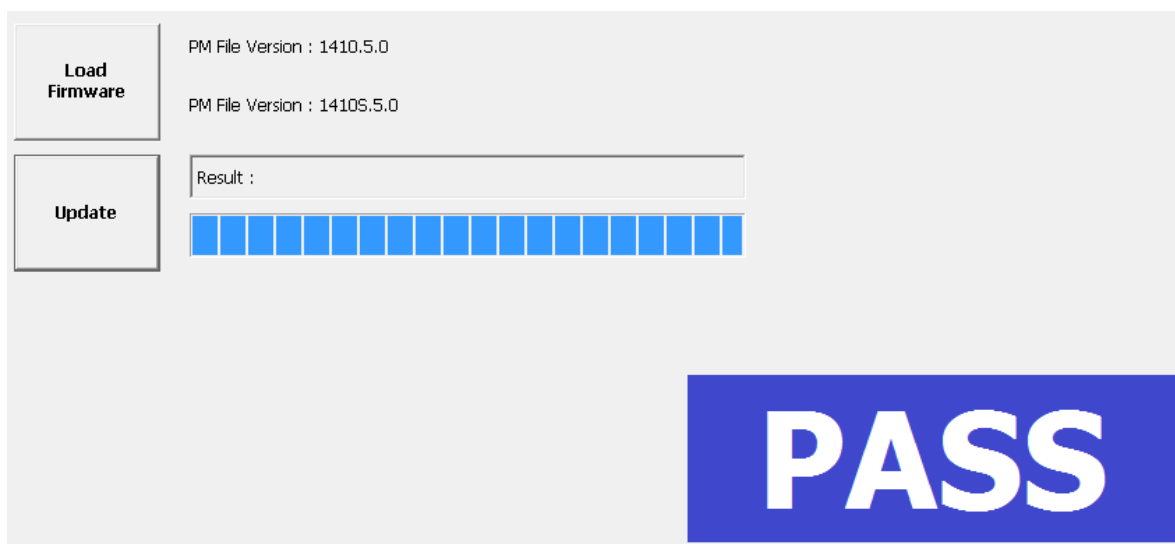


FIGURE 68

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



FIGURE 69

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.

5. Draw

The Drawing Test mainly examines whether breaking or point hopping will occur when drawing lines and provides a mechanism for measuring sample rate. Please click “Draw Test” in “Advanced test” and PCIScan will try to run “pmDraw3.exe.”

| 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 the advanced functions, please click “Press and Hold” with your finger or click the right mouse button for the functions as indicated below:

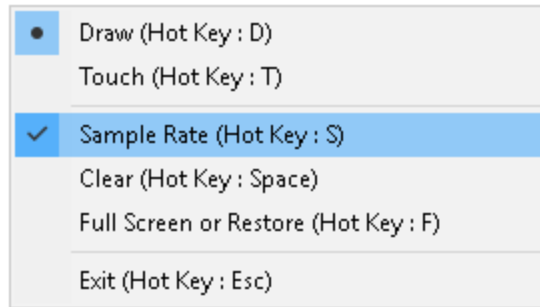


FIGURE 70

- Draw Mode

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

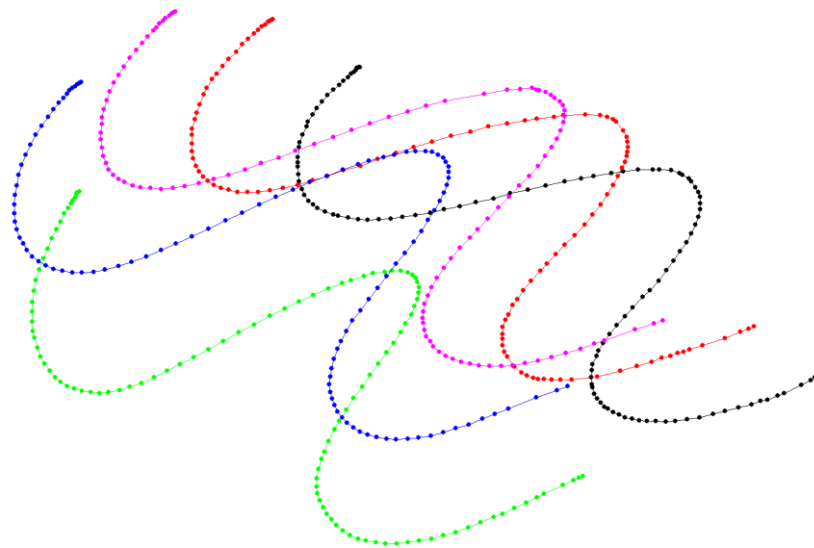


FIGURE 71

- Touch Mode

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

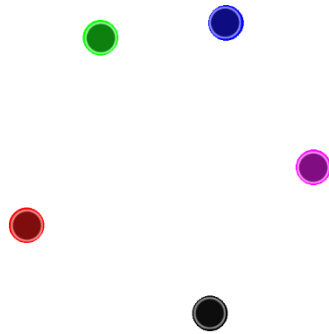


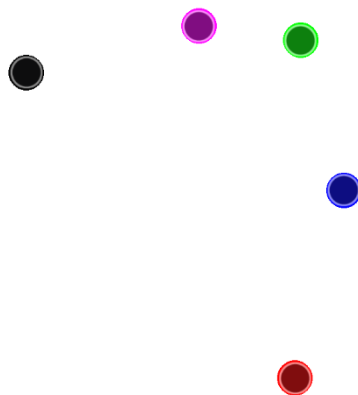
FIGURE 72

- 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 73

Point 0 : 109
 Point 1 : 109
 Point 2 : 109
 Point 3 : 109
 Point 4 : 109



- Clear

Clean the canvas.

- Full Screen or Restore

Switch between full screen display and windowed display.

- Exit

Exit the program.

Note: After entering pmDraw3, please do not connect or disconnect devices. Otherwise, touch will not function.

6. Debug

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

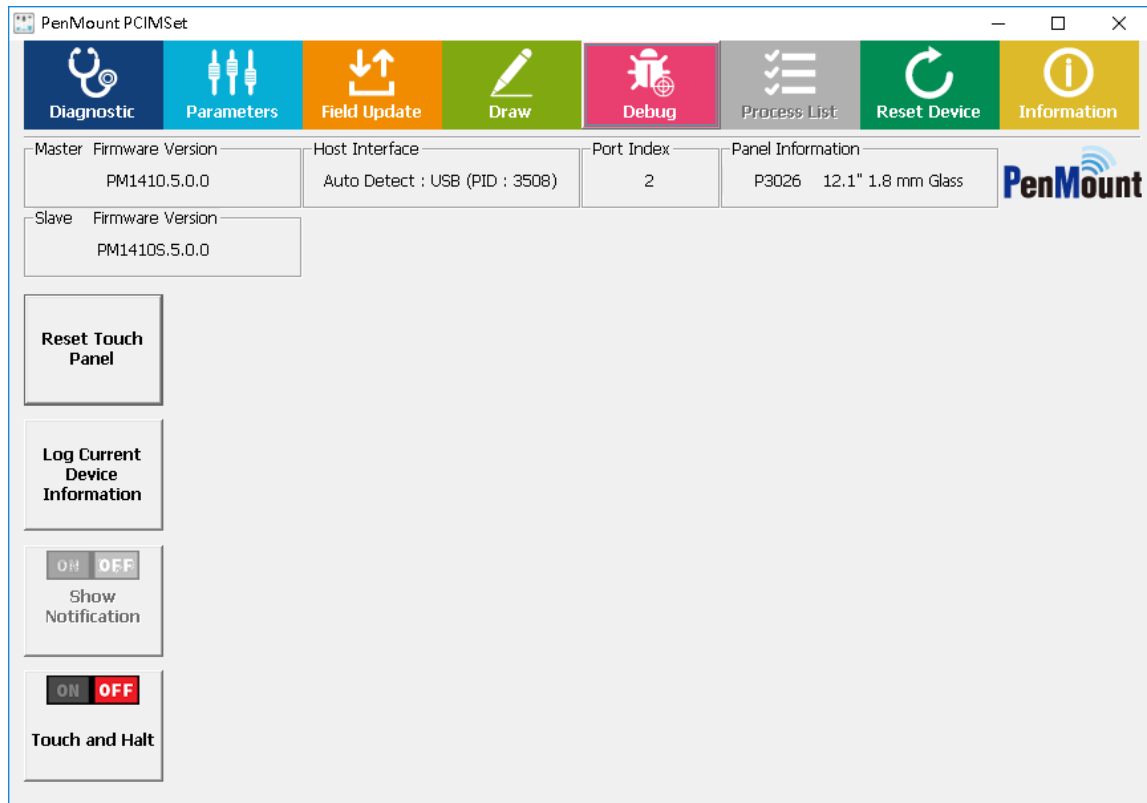


FIGURE 74

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.

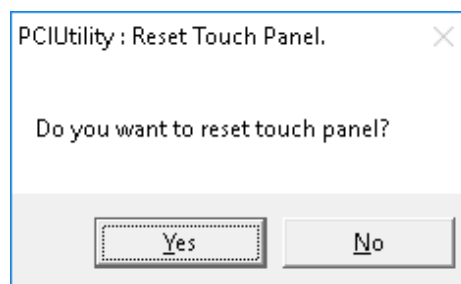


FIGURE 75

Reset touch panel may help resolve some touch malfunction issues when being 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 process could take several seconds. Please wait for the “Write log finish” message to appear.

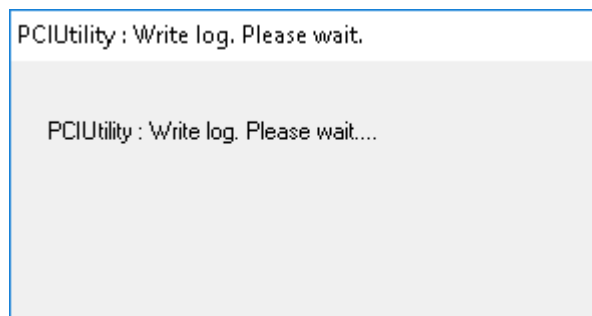


FIGURE 76

The firmware information is saved as “<Model> Firmware Information.csv” file in the program directory. For example, when using PM1410, the file name will be “PM1410 Firmware Information.log”. The file consists of the useful information for debugging touch issues.

6.3 Show Notifications of 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.

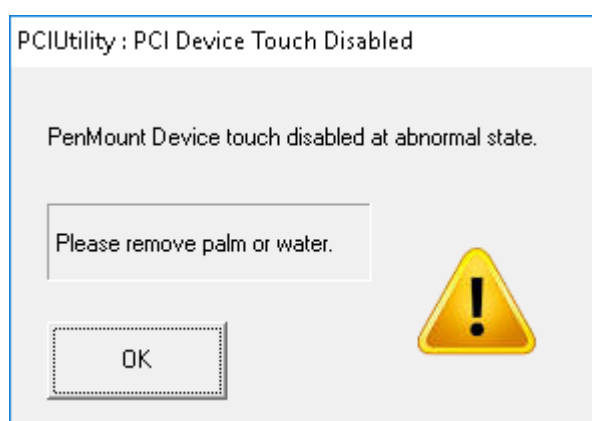


FIGURE 77

6.4 Touch and Halt

The touch and halt feature is supported by 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 enabled, utility will ask user to choose a folder to store firmware information.

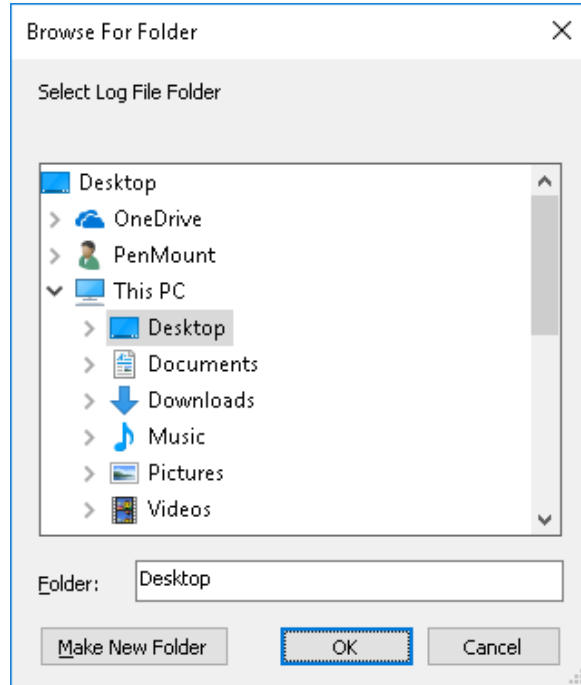


FIGURE 78

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