

PENMOUNT PCIUTILITY USERS' GUIDE FOR LINUX

Version 1.4

31/Aug/'21



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 © 2021 All rights reserved

Revision Table

Date	Revision	Changes
20/Oct/2016	1.0	(1) Initial Release
20/Apr/2017	1.1	(1) PM1415, PM1715, and PM2204 are added to the supported device list (2) Modify 4.1 and 4.2. Program can display result graphically if using X server. (3) Add 4.3 Build System Reference (SR). (4) Add 4.4 Live Test, for those without SR support.
09/Feb/2018	1.2	(1) 2.2: Update help context menu. (2) 3: Update for SR Status and Parameters CRC information. (3) 4.3: The term "ER" is renamed to "SR". (4) 4.4: New section for running diagnostic tests with command line options. (5) 7.3: New enhanced driving option. (6) 8.3: Command line options for firmware update.
03/Oct/2019	1.3	(1) 7.1: New option "HID Report". (2) 7.2: New options for advanced panel tweaking. (3) 7.7: New option "Median Filter", and "Enhanced Frequency Hopping".
08/16/2021	1.4	(1) 1.1: Add ARM64, MIPS64, and android. (2) 1.2: Ubuntu 12.04 with kernel 3.13 is not supported. (3) 1.3: Add PM2206, and PM2207. (4) 3: ND-BASE CRC information. (5) 4: Open Short test now called Panel Quality Test. (6) 5: Quick Select features for firmware 6.2 and 6.3. (7) 7.7: The Double ADC Average option.

Table of Content

PREFACE	I
Disclaimer	i
Trademarks.....	i
Copyright	i
REVISION TABLE.....	II
1. INTRODUCTION	6
1.1 Choosing the Correct Program	6
1.2 Operating System Support	6
1.3 PenMount Device Support	7
1.4 GUI Support.....	8
2. GENERAL USAGES	9
2.1 Program Version.....	9
2.2 Program Usage	9
2.3 Device Detection	10
2.3.1 Detect only USB Interface	11
2.3.2 Specify Device Path for non-USB Device	11
2.4 Reset Device	11
3. INFORMATION.....	12
4. DIAGNOSTIC	13
4.1 Basic Panel Quality (Open-Short) Test.....	13
4.2 Get Noise Level.....	13
4.3 Build System Reference (SR).....	14
4.4 Command Line Options	15
4.4.1 Live Test.....	15
4.4.2 Diagnostic Tests.....	15

5.	QUICK SELECT	16
5.1	FRAME MODE	17
6.	BASIC PARAMETERS	19
6.1	Interface	20
6.2	Panel 21	
6.3	Finger	22
6.4	Water	22
6.5	Anti-Noise	23
6.6	Coordinate	24
7.	ADVANCED PARAMETERS	26
7.1	Interface	27
7.2	Panel 28	
7.3	Finger	30
7.4	Palm 34	
7.5	Base 36	
7.6	Water	39
7.7	Anti-Noise	42
7.8	Coordinate	44
7.9	Power	46
7.10	Miscellaneous	49
7.11	Driving IC	50
8.	FIELD UPDATE	51
8.1	Load Firmware	51
8.2	Update Firmware	51
8.3	Command Line Options	52

9.	DEBUG.....	53
9.1	Log Firmware Information.....	53
9.2	Dump Device Data.....	53

1. Introduction

This document provides information on using the PenMount PCI utility for the Linux operating system. The utility is for changing PenMount touch settings and touch issue diagnosis.

1.1 Choosing the Correct Program

The utility supports several hardware architectures, and each can be found in a specific directory. After decompressing the package, there will be several directories for different platforms.

Operating System	Target Platform	Directory
Linux	ARM32	armhf
	ARM64	aarch64
	x86 (32bit)	i686
	x86 (64bit)	x86_64
	MIPS64	mipsel
Android	ARM32	android
	ARM64	android64

The utility do not require installation and can be run directly. Please note that root permission is required for running the programs.

1.2 Operating System Support

The utility supports the following Linux kernel configurations.

Kernel version	Support
2.6	Yes (requires additional kernel driver)
3.x	Yes ¹
4.x	Yes
5.x	Yes

¹ Does not support Ubuntu 12.04 with kernel 3.13.

1.3 PenMount Device Support

The utility supports the following PenMount devices.

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	
	PM1201	v	v	
	PM2201	v	v	v
PenMount P2-04 Series	PM1300A	v		
	PM1302	v	v	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

² The I²C interface can be supported if kernel supports I²C device interfaces, or it can also be supported by connecting through the PMT101 Adapter.

	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
	PM2205	v	v	v
	PM2206	v		
	PM2207	v	v	v
	PM2303	v	v	v

1.4 GUI Support

The PenMount pciutility are text based and can be run in any terminal. They also support being run inside X window systems.

For interfaces other than USB, please make sure that the serial port is not opened by other programs such as the PenMount X input driver or the inputattach utility. If so, there is a chance that these programs can prevent the PenMount pciutility from communicating with PenMount P2 devices and result in no device found. In this case, please temporarily disable these programs or device drivers before using the PenMount pciutility.

2. General Usages

The functions of the PenMount pciutility are summarized below:

- Information: Display firmware information
- Diagnostic: Check panel status
- Parameter: Change firmware settings
- Field Update: Perform firmware update
- Debug: Log firmware data in file

2.1 Program Version

The program version is shown right after the program starts.

```
=====  
|           PenMount PCI Utility           |  
=====
```

Version 1.4.0

2.2 Program Usage

The program supports several parameters and can be viewed by using the following command:

```
./pciutility -h
```

```

=====
-h      : Print Usage
-v      : Display utility version
-----
-d <path> : Assign device path for UART / RS-232 interface.
          => Will only accept device path under "/dev"
-usb    : Detects USB interface
-hid    : Detects USB-HID interface (HIDRAW required)
-i2c    : Detects I2C interface
-----
-no-crc  : Hide parameter CRC value in menu
-reload-all: The "Write factory default parameters" option will include :
          => Host interface
          => Panel parameters
-----
-fwver  : Display firmware version
-bu     : Forcing base signal to be updated immediately
-reset  : Forcing device to reset
-----
-log    : Log current device information
-dump   : Dump device data
-----
-fu <file> : Assign firmware file path and update
-----
-live <t> : Live Test for checking if base signal is updated correctly.
          => PM1110A : Triggered if no water event update after <t> seconds (Default : 120)
          => -openshort: Perform open short test every <t> seconds.
-dir     : Setup the output directory for live test
-----
-sr     : Construct new SR.
          => Can be combined with "-openshort" and "-noiselevel".
-openshort : Perform open short test.
          => Can be combined with "-sr" and "-noiselevel".
-noiselevel: Perform noise level test.
          => Can be combined with "-sr" and "-openshort".
=====

```

Users can combine some of the parameters to launch special features directly, such as reset device, log current device information, dump device data, etc. These features are introduced in later sections of this document.

2.3 Device Detection

Running the program without specifying any parameters will let program detect all PenMount devices connected to the system including USB, UART (RS-232) or I²C interfaces, so it might take some time to finish.

If no device is found, the program will terminate immediately.

```

* Touch will be disabled !
* Looking up devices, please wait ...
* No PenMount device found !

```

If device is connected, the program will show the main menu.

```
[Main Menu]
=====
                PenMount PM1310 Settings
=====
1. Information
2. Diagnostic
3. Quick Select
4. Parameter
5. Field Update
6. Debug
-----
R. Reset Device
Q. Exit
=====

Please select next action :
```

The menu content may be different according to the device firmware version used.

2.3.1 Detect only USB Interface

By default, the pciutility scans all the supported interfaces, including USB, RS-232 and may take some time to find a device. If there is only one USB device connected, users may also specify the “-usb” parameter to save time.

```
sudo ./pciutility -usb
```

2.3.2 Specify Device Path for non-USB Device

To specify a target serial port for detecting PenMount devices, please use the “-d” parameter. If this option is not set, the application will scan all the ttyS devices that might take a long time to finish.

```
sudo ./pciutility -d /dev/ttyUSB0
```

2.4 Reset Device

This option can re-initialize the connected PenMount device. In some cases where the operation environment has greatly changed, this feature may help improve touch performance.

3. Information

Choosing the “Information” option in the main menu will display information about the PenMount device.

```
[PenMount Device Summary]
=====
Device       : PM1310 Control Board
Location     : USB 2-2.1
Firmware Version : PM1310.6.3.0
Core Version  : 3.8.1.8
Panel Size   : P3007 8.4"
Cover Lens   : 2.8 mm (Glass)
SR Status    : ACTIVE
Parameters CRC : xA132
ND-BASE CRC  : x8AE4
=====
* Please press any key to continue ...
```

- Device: The PenMount P2 controller model number.
- Location: The interface and the port to which the PenMount P2 device is connected.
- Firmware Version, Master Version: Firmware version for main IC. Versions that start with Dxx indicate ODM firmware.
- Slave Version: Firmware version for slave IC. Versions that start with Dxx indicate ODM firmware.
- Panel Size: The target touchscreen size.
- Cover Lens: The target cover glass thickness.
- SR Status: The System Reference status, can be either “NOT INITIALIZED” or “ACTIVE”.
- Parameters CRC: The calculated CRC-16 value of the firmware parameters.
- ND-BASE CRC: The CRC-16 value of signal compensation parameters.

The program stays on this screen until user presses any key to return to the main menu.

4. Diagnostic

Choosing the “Diagnostic” option in the menu will perform some basic diagnosis on touch panels. The PenMount PCIScan for Windows provides full features for manufacturers.

```
[Diagnostic]
=====
1. Panel Quality (Open-Short Test)
2. Evaluate Noise Level
3. Build SR
4. Reset SR
-----
Q. Exit to upper menu
=====

Please select action : 1
```

4.1 Basic Panel Quality (Open-Short) Test

Test if every drive line and sense line is working normally. To test if the touch panel is functioning normally, the following screen will appear and start testing.

```
* Log Firmware Information 100%
* Panel Quality (Open Short) Test : 100%
* Test Aborted !
* Reason: Check Base Flat Failed.
* Test result saved to : "./Diagnostic/OpenShort_PM1310_6_3_0_20210816_115818.csv"
v"
↳ Open/Short diagnostic data : "./Diagnostic/OpenShort_PM1310_6_3_0_20210816_115818.csv"

↳ Diagnose result : [FAIL]
```

4.2 Get Noise Level

Test the level of noise interference to the touch panel. It will collect 50 sets of raw data and proceed to noise level calculation.

```
* Log Firmware Information 100%
* Noise Level Test : 100%
* Test Finishes Successfully !
* Test result saved to : "./Diagnostic/NoiseLevel_PM1310_6_3_0_20210816_115947.csv"
sv"
-----
* Noise Level Rating : {GREAT} ⊙ No False touch ⊙ Line Drawing ⊙ Noise Susceptibility
↳ Noise level diagnostic data : "./Diagnostic/NoiseLevel_PM1310_6_3_0_20210816_115947.csv" !
```

Test results will be saved automatically in “.CSV” file, and the utility will also rate the surrounding criteria with 5 different levels: “GREAT”, “GOOD”, “OK”, “POOR”, and “BAD”.

The definitions of the surroundings criteria are shown in the diagram below. If finger or noise is present during test, it could be seriously affected and result in BAD.

	No False Touch	Line Drawing	Noise Susceptibility
GREAT	⊙	⊙	⊙
GOOD	⊙	⊙	○
OK	⊙	○	△
POOR	○	△	×
BAD	×	×	×

4.3 Build System Reference (SR)

This option is only available if device firmware supports the “System Reference (SR)” feature.

System Reference (SR) is a feature that supports environmental changes which is usually built during system assembly.

With a new touch screen controller without System Reference (SR), the program will display a warning messages when it starts.

```
* [WARNING] SR is not initialized !
```

After choosing this option, program will first display the current SR status, which can be:

Status	Description
NOT INITIALIZED	This device does not contain a valid set of System Reference (SR).
ACTIVE	The device is already working with System Reference (SR).

User should be take note of the onscreen messages before proceed to building SR. It is extremely important that the touch screen is clean and noise-free before proceeding.

```
-----  
[Building SR]  
-----  
* SR Status : [ACTIVE]  
  
-----  
[CAUTION]  
(1) The touch panel should be connected properly.  
(2) The touch panel should be clean (without water).  
(3) Please run in a noise-free environment.  
-----  
* Continue building new SR ? (y|n) : y
```

After user confirms proceeding SR building by entering the “y” key, System Reference (SR) is evaluated under a full blank screen.

```
* Building process will start in 2 seconds ...  
* The screen will turn to white during calibration.  
* Log Firmware Information 100%  
* Build System Reference Result Saved : "./Diagnostic/SR_PM1310_6_3_0_20210816_1  
20116.csv"  
* Built Result ... [OK]  
=====
```

After the System Reference (SR) is built, the program will start a post-check to ensure that the SR just built is good to use.

If check fails, the user can still choose whether to save the result. If user chooses not to save, the program will rollback changes.

Please note that if the device firmware supports several touch panel options, changing the panel size will invalidate the System Reference (SR) and program will prompt to building the SR again!

4.4 Command Line Options

Diagnostic tests are also available by running utility with command line options.

4.4.1 Live Test

Several command line options can be combined to perform batch live test.

Option	Description
-live <t>	Perform Live Test every <t> seconds
-openshort	Perform Open Short Test every <t> seconds
-sniff	Protocol Analyzing

4.4.2 Diagnostic Tests

Several command line options can be combined to perform batch diagnostic tests.

Option	Description	Return Value
-openshort	Open Short Test	SUCCESS : 0 FAIL : Non-zero values
-noiselevel	Noise Level Test	SUCCESS : 0 FAIL : Non-zero values
-sr	Build SR	SUCCESS : 0 FAIL : Non-zero values

When combined together, the operation sequence will be: SR construction -> open short test -> noise level test.

5. Quick Select

For firmware V6.2 and later, the utility supports switching between several predefined operation modes conveniently.

```
[Mode Selection]
=====
1. Fast Drawing Mode
2. Anti-Noise Mode
3. Water Proof Mode
4. Thick Glove Mode
-----
5. Frame Mode : OFF
-----
Q. Exit to upper menu
=====
Please select action : 1
```

Figure 1

The supports modes are listed below:

Mode	Feature	Side Effect
Fast Drawing Mode	Allow smooth drawing	Environmental noise will have greater impact.
Anti-Noise Mode	Maintain normal touch operation in environments with high noise.	Lesser sampling rate.
Water Proof Mode	Prevents abnormal touches caused by water.	Lesser sampling rate.
Thick Glove Mode	Allow touch operation with thick gloves	Environmental noise will have greater impact.

The following chart compares the major parameter differences between each mode when using device firmware V6.3 or later versions.

Modes	Standard	Fast Drawing	Anti-Noise	Water Proof	Thick Glove
Sensitivity	0	0	0	0	+3
Water Handling	OFF	OFF	OFF	Water Performance	OFF
Palm Rejection	ON	OFF	ON	ON	OFF

Anti-Noise	Spike Filter	Normal	Lite	Enhanced	Normal	Normal
	Enhanced Driving	OFF	OFF	ON	OFF	ON
	Double ADC Average	OFF	OFF	ON	OFF	OFF
	Frequency Hopping	Normal	Lite	Enhanced	Normal	Normal
Max Contact Number		Default	Default	Default	1	Default

5.1 Frame Mode

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.

For firmware V6.3 and later, switch to frame mode enables the “Thick Frame Bezel” parameter.

For firmware V6.2, switch to frame mode adjust the sensor / driver lines used by device firmware to work with thicker frames.

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

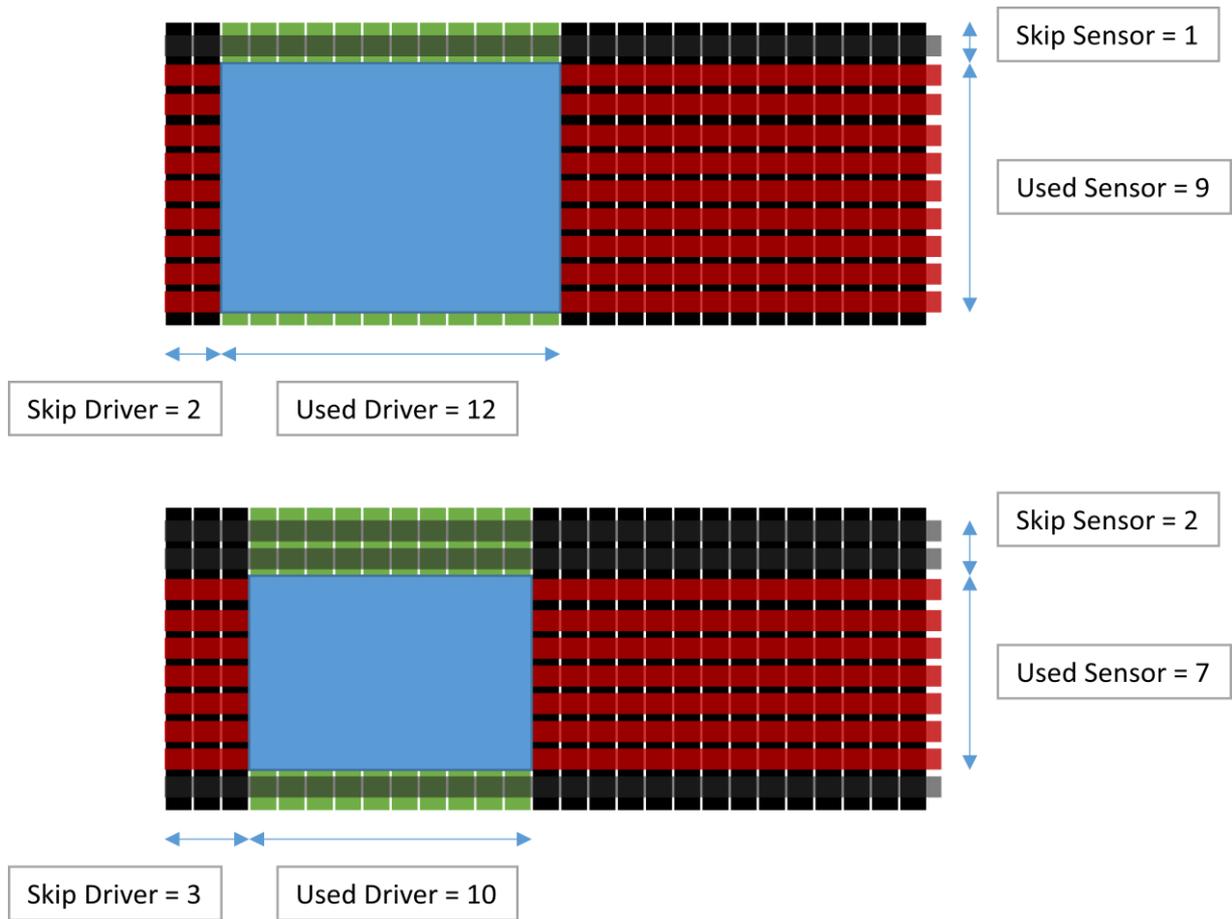


Figure 2

Please notice that running this click button several times might significantly decrease accuracy. To revert settings, please use the "Write Factory Default Parameter" button or do firmware update again.

6. Basic Parameters

For PenMount P2-02 / P2-03 / P2-04 and some older versions of P2-06 / P2-08 firmware, only basic parameters can be adjusted. These parameters are classified in several categories.

```
[Parameter] (0x239D)
=====
1. Interface
2. Panel
3. Finger
4. Water
5. Anti-Noise
6. Coordinate
-----
(I) Import Parameters
(E) Export Parameters
(R) Write Factory Default Parameters
-----
S. Apply (Exit to upper menu)
Q. Cancel (Exit to upper menu)
=====
Please select action :
```

- Import Parameters

This feature allows user to select a “INI” format file and import the parameter settings inside. The utility will search the “Parameters” sub-directory for a compatible parameter file.

```
Please select action : i
[Import Parameters]

* Default file : "Parameters/PM1410_2_1_0_x239D.ini" ?
* Loading firmware parameters ? (Y|N|Q) : y
* Loading firmware parameters : "Parameters/PM1410_2_1_0_x239D.ini"
* Open parameter file "Parameters/PM1410_2_1_0_x239D.ini" ...
=> Update AP parameter #0x7F = 0xD6
=> Update BT parameter #0x3F = 0xE6
* {INFO} Waiting for device ready .....

* {INFO} Waiting for device ready .....
```

This parameter file is compatible when using in Windows PCIUtility V1.4 and later.

- Export Parameters

This feature allows user to export the device parameters to a “INI” format file. The file will be automatically saved in the “Parameters” sub-directory.

```
Please select action : e
[Export Parameters]

* Parameters exported to Parameters/PM1410_2_1_0_x239D.ini !
```

- Write Factory Default Parameters

This feature reverts all AP parameter changes to factory default values.

For BT parameters, host interface and panel size parameters will be kept. To reload all BT parameters, please add the “-reload-all” option when running pciutility.

- Apply

Choosing this will apply parameter changes, reset device, and then go back to the upper menu.

- Cancel

Choosing this will drop all parameter changes and go back to the upper menu.

6.1 Interface

In the Interface Parameters category, users may change the parameters related to connection with host PC.

```
[Interface Parameters]
=====
1. Host Connection      : AUTO
-----
Q. Exit to upper menu
=====
```

- Host Connection

View and change Interface used for connection. Please prepare the connection cable in advance.

```
[Supported connection]
=====
[U] USB
[R] RS-232
[I] I2C
-----
[Q] Exit to upper menu
=====

Please specify a new connection : █
```

For some PenMount devices, it is not possible to change the interface, please check the table below.

PenMount Device	Host interface Configurable
PM1302 / PM1401	YES
PM2101 / PM2201	YES

PM1210	NO (Always AUTO)
PM2103 / PM2203 / PM2300	YES
PM1310 / PM1410 / PM1710 / PM1711	NO (Always AUTO)

6.2 Panel

In standard PenMount P2 firmware, choosing the “Panel” Parameters option in the main menu allows user to change the target panel size or cover lens thickness settings. In ODM firmwares, however, this settings may not be adjustable and will not be listed in the menu.

```
[Panel Parameters]
=====
1. Change Panel Size : 12.1"
2. Change Cover Lens : 1.8 mm (Glass)
-----
Q. Exit to upper menu
=====
Please select action :
```

- Change Panel Size

Displays the active panel size setting and lists the supported sizes for modification.

```
[Supported Panel Sizes]
=====
[1] P3026 12.1"
[2] P3023 15.6" (W)
[3] P3029 15.0"
[4] P3030 12.1" (W)
-----
[Q] Exit to upper menu
=====
Please specify new panel size ID : |
```

- Change Cover Lens

Displays the active cover lens setting and lists the supported thickness for modification.

```
[Supported Cover Lens]
=====
[1] 1.1 mm (Glass)
[2] 1.8 mm (Glass)
[3] 2.8 mm (Glass)
-----
[Q] Exit to upper menu
=====
Please specify new cover lens ID :
```

6.3 Finger

In finger parameters, users can change the sensitivity level and the maximum touch counts.

```
[Finger Parameters]
=====
1. Sensitivity      : Default (0)
2. Touch Mode      : Multiple Touch
-----
0. Exit to upper menu
=====
Please select action :
```

- Sensitivity

For sensitivity, users can choose between “less sensitive,” “more sensitive,” or “default”.

Sensitivity	Value
Default	0
Less Sensitive	Negative value
More Sensitive	Positive value

```
[Changing sensitivity]
* Higher value increase touch sensitivity !
Please specify a new value (-2 ~ 2) : 0
```

- Touch Mode

Only standard firmware versions allow user to change between “single touch,” “dual touch,” or “multiple touch”. For ODM versions, the firmware is locked into using one specific touch mode.

```
[Changing touch mode]
Please specify a new value (1|2|m) : 1
* Change to single touch mode
```

6.4 Water

The water handling feature is supported in newer firmware versions. By default, the firmware uses the best modes for performance and reliability. Users can still disable this feature if necessary.

```
[Water Parameters]
=====
1. Water Handling      : WD (Water Detection)
-----
Q. Exit to upper menu
=====

Please select action :
```

- Water Handling

The PenMount firmware supports several water handling modes such as WP and WD which can be changed dynamically.

```
[Changing Water Handling Mode]
* Please specify a new mode (WP|WD|n) : WP
* Enable Water Handling : WP (Water Performance)
* No false touch when wet, dry areas retain touch function
```

The table below summarizes the differences between different water handling modes.

Option	Water Handling	Description
n	Disabled	Water handling is disabled. False touch might occur when there is water on touch panel surface but touch function returns to normal after water is removed.
WD	Water Detection	PCI touch function disabled when water detected on touchscreen surface.
WP	Water Performance	No false touch when wet, dry areas retain touch function.

6.5 Anti-Noise

By default PenMount firmware uses optimized anti-noise parameters for best performance and noise filtering. Users can increase the filtering level for operation in noisier environments.

```
[Anti-Noise Parameters]
=====
1. Low Pass Filter    : Stronger (12)
-----
Q. Exit to upper menu
=====

Please select action :
```

- Low Pass Filter

The low pass filter level is used to filter out randomly occurred noise signals and is quantized in 16 levels.

```
[Changing Low Pass Filter level]
* {stronger} 1 <- 13 -> 16 {weaker}
* Lower value increase noise filtering, but also decrease touch sensitivity !
Please specify a new value : 13
```

The table below summarizes the effects for using different levels of low pass filter.

Low Pass Filter Level	Performance	Noise Filtering
< Default	Slower	Stronger
Default (13)	Optimized	Optimized
> Default	Faster	Weaker

6.6 Coordinate

The coordinate parameters control how PenMount firmware reports touch point positions.

```
[Touch Parameters]
=====
1. Orientation (Rotation) : Landscape (0)
-----
2. Edge Adjust (Left)      : 7
3. Edge Adjust (Right)    : 7
4. Edge Adjust (Up)       : 7
5. Edge Adjust (Down)     : 7
-----
* AMT touch sensor / PenMount PCI do not need calibration,
If the LCD display [AA] Active Area matches the [KA] Key Area.
-----
Q. Exit to upper menu
=====
Please select action :
```

- Orientation

The rotation degree of the touch panel, measured counter clockwise. The supported values are:

Rotation Degree (CCW)	Orientation
0	Landscape
90	Portrait
180	Landscape (Flipped)
270	Portrait (Flipped)

- Edge Adjust

Controls the tendency of touch point being adjusted toward screen edges. Setting value to 0 will turn off edge compensation.

- Active Area Calibration

For older firmware versions, calibration is not needed when LCD display [AA] active area matches the [KA] key area.

7. Advanced Parameters

For most P2-06 and P2-08 touch controllers, more advanced parameters can be adjusted. These parameters are classified in several categories.

```
[Parameter] (0x8F14)
=====
(1) Interface
(2) Panel
(3) Finger
(4) Palm
(5) Base
(6) Water
(7) Anti-Noise
(8) Coordinate
(9) Power
(10) Miscellaneous
(11) Driving IC
-----
(I) Import Parameters
(E) Export Parameters
(R) Write Factory Default Parameters
-----
S. Apply (Exit to upper menu)
Q. Cancel (Exit to upper menu)
=====
Please select next action : █
```

- Import Parameters

This feature allows user to select a “INI” format file and import the parameter settings inside. The utility will search the “Parameters” sub-directory for a compatible parameter file.

```
Please select action : i
[Import Parameters]

* Default file : "Parameters/PM1410_2_1_0_x239D.ini" ?
* Loading firmware parameters ? (Y|N|Q) : y
* Loading firmware parameters : "Parameters/PM1410_2_1_0_x239D.ini"
* Open parameter file "Parameters/PM1410_2_1_0_x239D.ini" ...
=> Update AP parameter #0x7F = 0xD6
=> Update BT parameter #0x3F = 0xE6
* {INFO} Waiting for device ready .....

* {INFO} Waiting for device ready .....
```

This parameter file is compatible when using in Windows PCIUtility V1.4 and later.

- Export Parameters

This feature allows user to export the device parameters to a “INI” format file. The file will be automatically saved in the “Parameters” sub-directory.

```
Please select action : e
[Export Parameters]

* Parameters exported to Parameters/PM1410_2_1_0_x239D.ini !
```

- Write Factory Default Parameters

This feature reverts all AP parameter changes to factory default values.

For BT parameters, host interface and panel size parameters will be kept. To reload all BT parameters, please add the “-reload-all” option when running pciutility.

- Apply

Choosing this will apply parameter changes, reset device, and then go back to the upper menu.

- Cancel

Choosing this will drop all parameter changes and go back to the upper menu.

7.1 Interface

In the Interface Parameters category, users may change the parameters related to connection with host PC.

```
[Interface]
=====
(*) Interface           : USB (Not Adjustable)
(1) I2C Address        : 0x38
(*) RS-232/UART Baudrate : 38400 bps (Not Adjustable)
(2) HID Report         : Windows 8 Protocol
-----
Q. Exit to upper menu
=====
```

- Interface

View and change Interface used for connection. Please prepare the connection cable in advance. For some PenMount devices, it is not possible to change the interface, please check the table below.

PenMount Device	Host interface Configurable
PM1302 / PM1401	YES
PM2101 / PM2201	YES
PM1210	NO (Always AUTO)

PM210X / PM220X / PM230X	YES
PM1310 / PM141X / PM171X	NO (Always AUTO)

- I2C Address

Adjust the slave address used when connecting to host through I2C interface.

- RS-232/UART Baudrate

The pciutility does not support dynamically changing baudrate settings yet. By default, the baudrate used is 38400 bps.

- HID Report

Some firmware versions supports switching between different data reporting formats.

Available options are:

```
* Supported HID Report :
[1] : Windows 8 Protocol
[2] : Mouse Protocol
* Please specify new HID Report : █
```

- **Windows 8 Protocol**

HID Data Report type: Touch Screen, supported by Linux kernel 3.4 and later versions.

- **Mouse Protocol**

HID Data Report type: Mouse, supported by all Linux versions.

7.2 Panel

In standard PenMount P2 firmware, choosing the “Panel Parameters” option in the main menu

```
[Panel]
=====
(1) Panel Size      : 18.5" (W)
(2) Cover Lens     : 1.8 mm (Glass)
-----
Advanced Panel Tweaking Options
-----
↳ 44 Available Sensor Pins = 4 + 36 + 4
-----
(3) Skip Sensor Pins : 4
(4) Sensor Pin Number : 36
-----
↳ 76 Available Driver Pins = 1 + 64 + 11
-----
(5) Skip Driver Pins : 1
(6) Driver Pin Number : 64
-----
Q. Exit to upper menu
=====
```

allows user to change the target panel size or cover lens thickness settings. In ODM firmwares, however, this settings is not adjustable and will not be listed in the menu.

- Panel Size

Displays the active panel size setting and lists the supported sizes for modification.

```
[Supported Panel Sizes]
=====
[1] P3026 12.1"
[2] P3023 15.6" (W)
[3] P3029 15.0"
[4] P3030 12.1" (W)
-----
[Q] Exit to upper menu
=====

Please specify new panel size ID : 1
```

- Cover Lens

Displays the active cover lens setting and lists the supported thickness for modification.

```
[Supported Cover Lens]
=====
[1] 1.1 mm (Glass)
[2] 1.8 mm (Glass)
[3] 2.8 mm (Glass)
-----
[Q] Exit to upper menu
=====

Please specify new cover lens ID : █
```

- Advanced Panel Tweaking Options

Allows user to manually specify the driver and sensing pin numbers used by a touch panel.

- Skip Sensor Pins: The number of skipped sensors from pin 0.
- Sensor Pin Number: The number of pins used for sensing.
- Skip Driver Pins: The number of skipped driving pins from pin 0.
- Driver Pin Number: The number of pins used for driving high voltage.

7.3 Finger

The parameters in this tab are for tuning finger detection.

```
[Finger]
=====
(1) Sensitivity                : 0 (default)
(2) Maximum contact number    : 5
(3) Sample rate limit         : 6 ms
(4) Down debouncing           : 14 ms (7)
(5) Up Debouncing             : 32 ms (16)
(6) Window lock (slow)        : 4.067 mm (49)
(7) Fast motion threshold     : 126 mm^2 (6)
-----
Q. Exit to upper menu
=====
```

- Sensitivity

For sensitivity, users can choose between “less sensitive,” “more sensitive,” or “default”.

Sensitivity	Value
Default	0
Less Sensitive	Negative value
More Sensitive	Positive value

- Maximum Contact Number

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.

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.

- 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 at a fixed rate and the reporting rate for single touch can be approximately calculated with the equation below:

$$\text{Estimated Reporting Rate (single touch)} = \frac{1000}{\text{Sample Rate Limit Value}}$$

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.

- Down Debouncing

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.

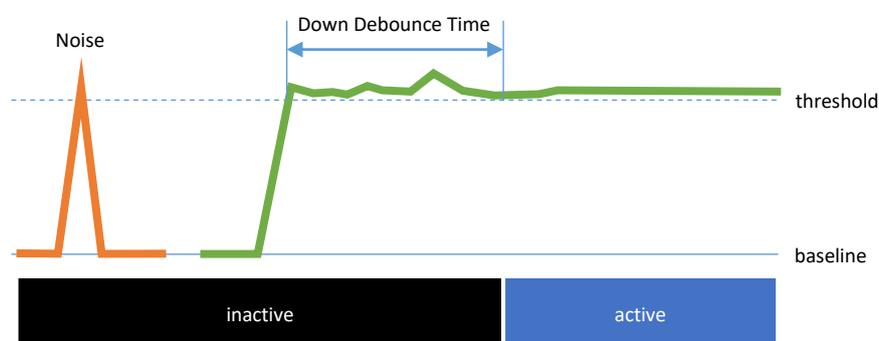


Figure 3

- Up Debouncing

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

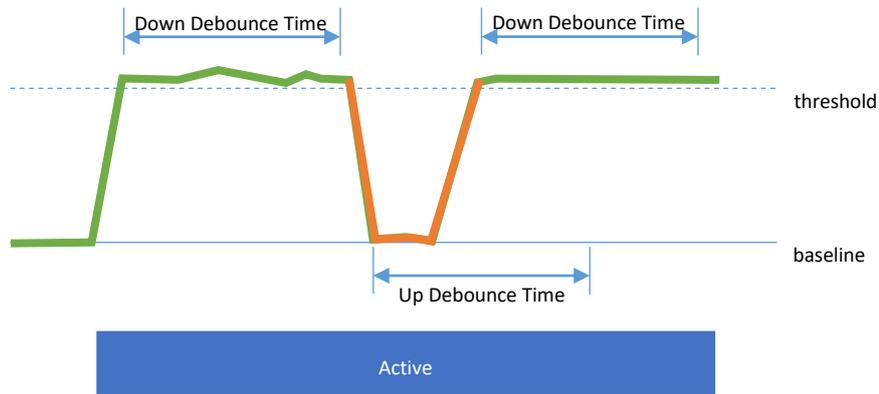


Figure 4

- 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, with logical maximum 2047.

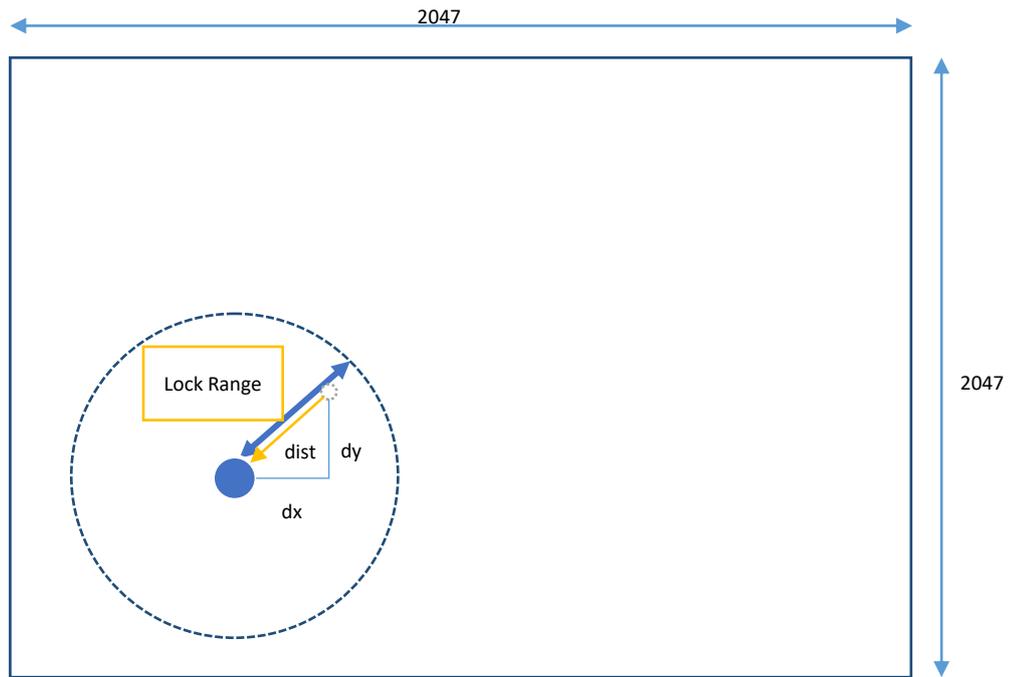


Figure 5

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.

- 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.

7.4 Palm

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.

```
[Palm]
=====
(1) Palm Rejection      : Enabled
(2) Palm size          : 22 * 85mm ^ 2
(3) Disable range near palm : 8
(4) Palm-Up debouncing  : 200 ms (100)
-----
(5) Palm on edge enable : Disabled
-----
Q. Exit to upper menu
=====

Please select action :
```

- Palm Rejection

This is the switch that controls whether palm rejection is enabled.

- 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.

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”.

- Disabled Range Near Palm

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

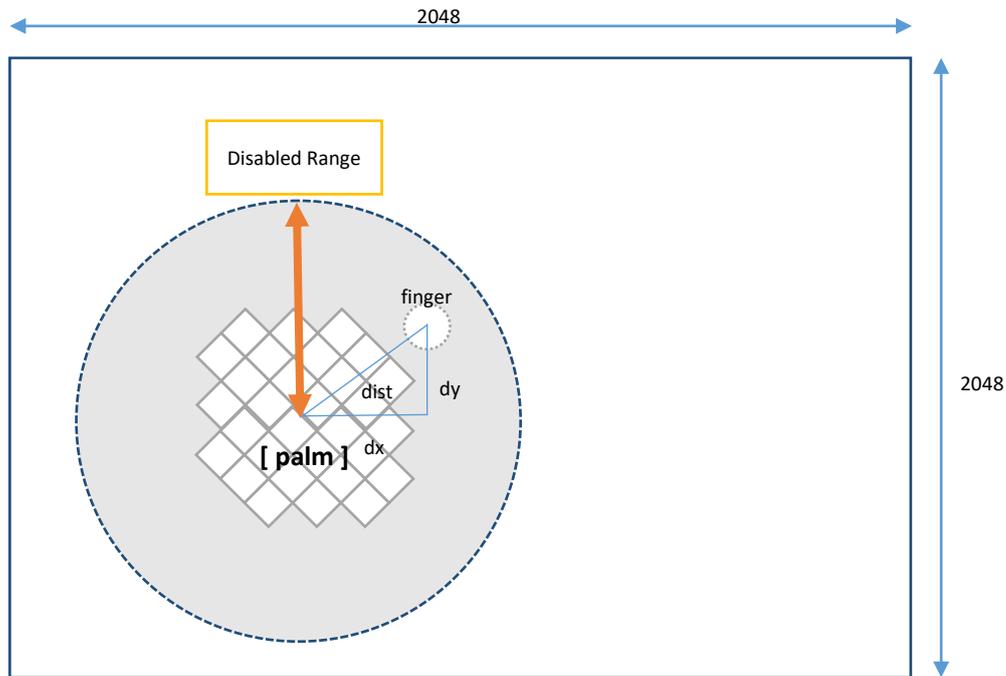


Figure 6

The “disabled range” can be calculated as below:

$$Disabled\ range^2 = (2 * Parameter\ Value * Sensor\ Pitch)^2$$

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

- Palm-Up Debouncing

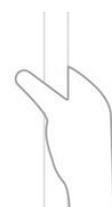
When a “palm” object is about to be removed from the touch panel surface, its decreasing size could also make it small enough to look like a normal finger touch. Without setting up a de-bouncing time for this, the firmware could send some unwanted touch data.

The “up-denouncing” time used for palms is different from the one introduced in previous section, 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.

- Palm on Edge Enable

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 because the object can be either a palm or finger. Currently the PCI Parameter does not support changing the threshold value.

7.5 Base

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.

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 Mode

7.5.1 Standard Base Update Configurations

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

```
[Base]
=====
(1) Update driver number      : 2
(2) Start update delay time   : 800 ms (8)
-----
Q. Exit to upper menu
=====
Please select action :
```

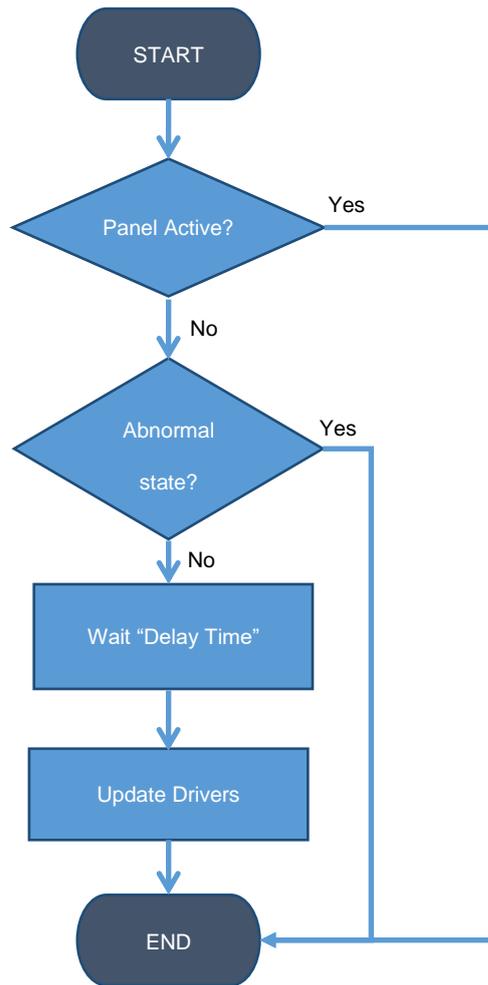


Figure 7

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

(1) 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.

(2) 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.

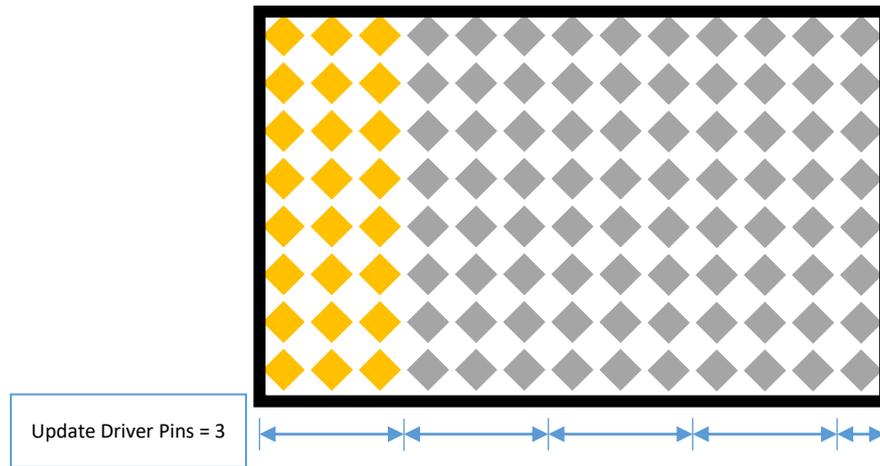


Figure 8

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.

7.5.2 Keep Base Update 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” to prevent base signals from changing rapidly, so it will take several rounds for the base signal to be completely updated.

```

(3) Inactive update delay count      : 0
(4) Active update delay count       : 0
  
```

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.

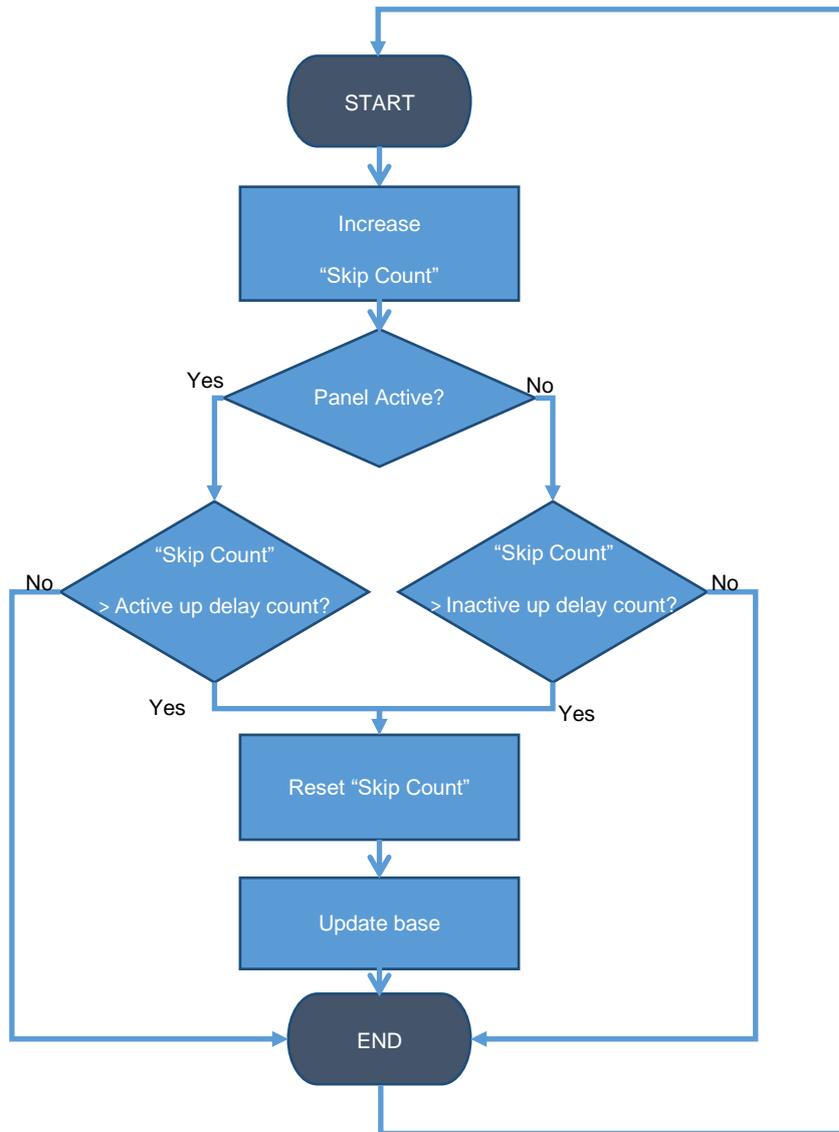


Figure 9

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.

7.6 Water

The water handling feature is supported in newer firmware versions. By default, the firmware uses the best modes for performance and reliability. Users can still disable this feature if necessary.

```
[Water]
=====
(1) Water Solution Switch (0) : Disabled
-----
Q. Exit to upper menu
=====
Please select action :
```

- Water Solution Switch

The PenMount firmware supports several water handling modes such as WP and WD which can be changed dynamically.

The table below summarizes the differences between different water handling modes.

Option	Water Handling	Description
n	Disabled	Water handling is disabled. False touch might occur when there is water on touch panel surface but touch function returns to normal after water is removed.
d	Water Detection	PCI touch function disabled when water detected on touchscreen surface.
p	Water Performance	No false touch when wet, dry areas retain touch function.

7.6.1 Water Performance Setting

```
[Water]
=====
(1) Water Solution Switch (17) : Water Performance
-----
Water Perfomance Settings
-----
(2) Performance / Reliability : 3
(3) Negative signal threshold : 8
(4) Negative signal count : 2
(5) Lock 1 contact switch : OFF
-----
Q. Exit to upper menu
=====
Please select action :
```

- Performance / 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; Switch to single touch mode after false touch detected	YES
1	Improve line drawing in dry areas with very little reliability drop		
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	No
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.

- Negative Signal Check

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 reliability level 3, the PenMount will not use this filter, and will always use stricter rules to check all the touch points for false touches.

- 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.

7.7 Anti-Noise

By default PenMount firmware uses optimized anti-noise parameters for best performance and noise filtering. Users can increase the filtering level for operation in noisier environments.

```
[Anti-Noise]
=====
(1) Low Pass Filter           : 13
(2) Enhanced Driving         : OFF
(3) Spike Filter             : Normal (1)
(4) Frequency Hopping Mode   : Normal (1)
(5) Double ADC Average       : OFF
-----
Q. Exit to upper menu
=====
Please select action :
```

- Low Pass Filter

This option is only available when “Median Filter” is switched OFF. It is used to filter out randomly occurred noise signals and is quantized in 16 levels.

```
-----
[Low Pass Filter]
-----
{Default}
* Value      : 13
-----
{Strong}
* Feature    : Stronger noise filtering.
* Range      : 0 ~ 12
* Side Effect: Too strong a filter can cause longer touch reponse time. May become unresponsive if value is under 3.
-----
{Weak}
* Feature    : Less noise filtering.
* Range      : 14 ~ 16
* Side Effect: Noise may cause false touch more easily.
-----
```

The table below summarizes the effects for using different levels of low pass filter.

Low Pass Filter Level	Performance	Noise Filtering
< Default	Slower	Stronger
Default (13)	Optimized	Optimized
> Default	Faster	Weaker

- 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.

- Median Filter

This option allows firmware to filter against pulsed noise interference such as ESD.

```
-----  
[Median Filter Switch]  
-----  
{ON}  
* Feature      : Better noise filtering against pulsed noise.  
* Value        : 1  
* Side Effect: Slightly reduced sample rate, touch response.  
-----  
{OFF}  
* Feature      : Legacy low pass filter.  
* Value        : 0  
* Side Effect: Normal operation.  
-----
```

- Frequency Hopping Mode

This option enables the firmware to perform even more complex algorithms to filter out noises.

```
-----  
[Frequency Hopping Mode]  
-----  
{Enhanced}  
* Value        : 2  
* Side Effect: Reduce sample rate.  
-----  
{Normal}  
* Value        : 1  
* Side Effect: Default setting.  
-----  
{Lite}  
* Value        : 0  
* Side Effect: Reduce the CS immunity.  
-----  
* Please specify new setting :
```

- Double ADC Average

This is another Feature that supports pulse type noise suppression.

```

-----
[Double ADC Average]
-----
{ON}
* Feature      : Strengthen anti-noise ability for spike
* Value        : 1
* Side Effect  : Reduced Performance
-----
{OFF}
* Feature      : Normal operation
* Value        : 0
* Side Effect  : Normal operation
-----
* Please specify new setting :

```

7.8 Coordinate

The coordinate parameters control how PenMount firmware reports touch point positions.

```

[Coordinate]
=====
Adjust Touch Panel Active Area
-----
(1) X Coordinate Min      : 0
(2) X Coordinate Max     : 2047
(3) Y Coordinate Min     : 0
(4) Y Coordinate Max     : 2047
-----
Adjust Edge Compensation (Tail on bottom edge)
-----
(5) Left Edge            : 4
(6) Right Edge           : 4
(7) Top Edge             : 4
(8) Bottom Edge          : 4
-----
(9) Touch Panel Orientation : Landscape (0)
-----
Q. Exit to upper menu
=====
Please select action :

```

- Touch Panel Active Area

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 10

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 $(\text{min}_x, \text{min}_y)$ to $(\text{max}_x, \text{max}_y)$.

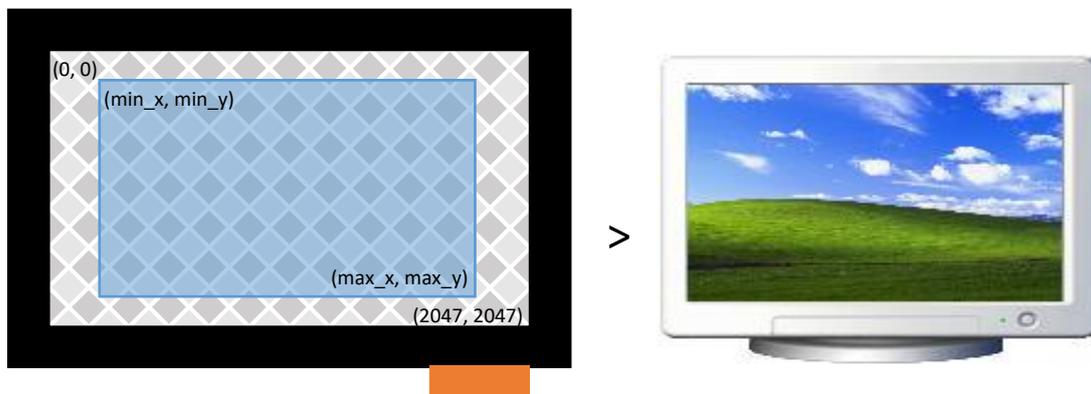


Figure 11

(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 min "min_x".

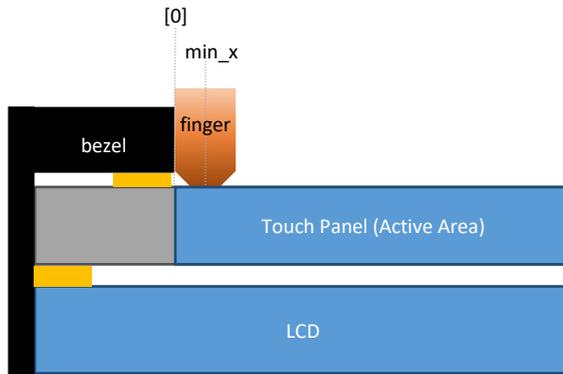


Figure 12

Calibration can be completed by 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.

- **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 areas 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. If a response is produced before the edge of the touchscreen is reached, please set the edge compensation value lower. If you want to restore the default setting, please click “Default”.

- **Orientation**

The rotation degree of the touch panel, measured counter clockwise. The supported values are:

Rotation Degree (CCW)	Orientation
0	Landscape
90	Portrait
180	Landscape (Flipped)
270	Portrait (Flipped)

7.9 Power

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

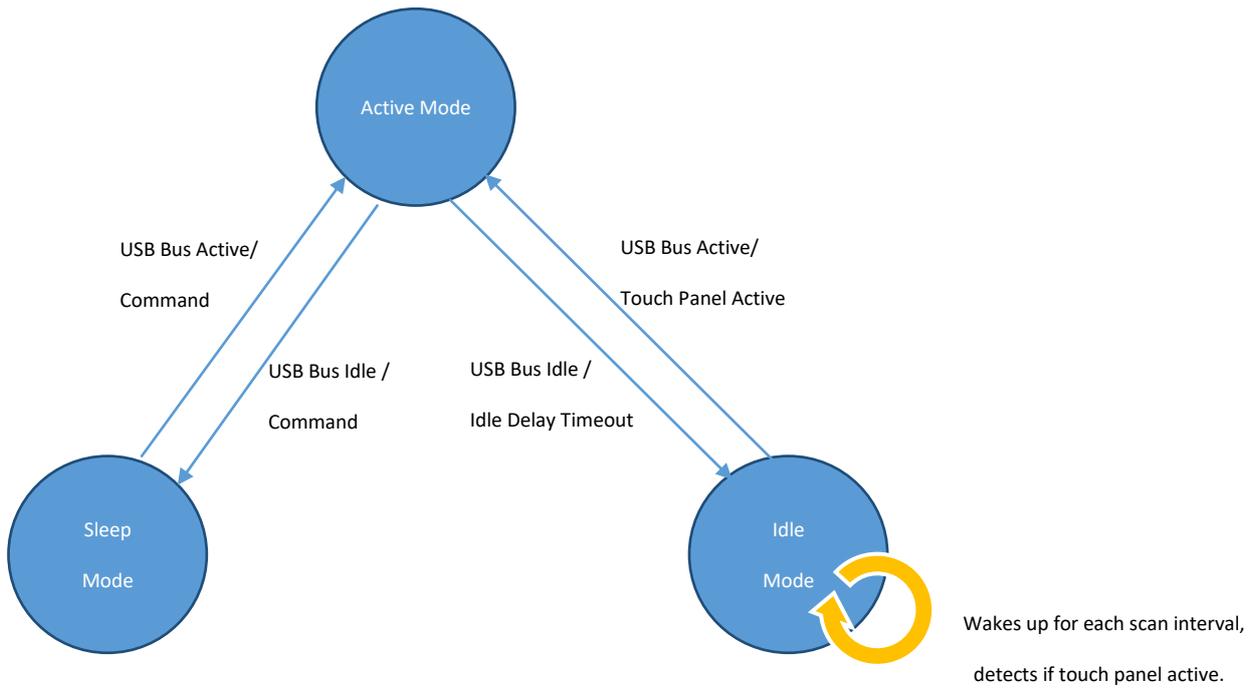


Figure 13

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

```

[Power-Saving]
=====
(1) Sleep Mode Switch      : Enabled
(2) Idle Mode Switch       : Enabled
-----
Idle Mode Setting
-----
(3) Scan Interval         : 20 ms
(4) Delay                 : 6 s
-----
Q. Exit to upper menu
=====
Please select action : █
  
```

I. 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.

II. 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.

7.9.1 USB Interface Configurations

PenMount supports the USB “Remote Wakeup” feature and, when this feature is enabled, user can wake up the system by touch.

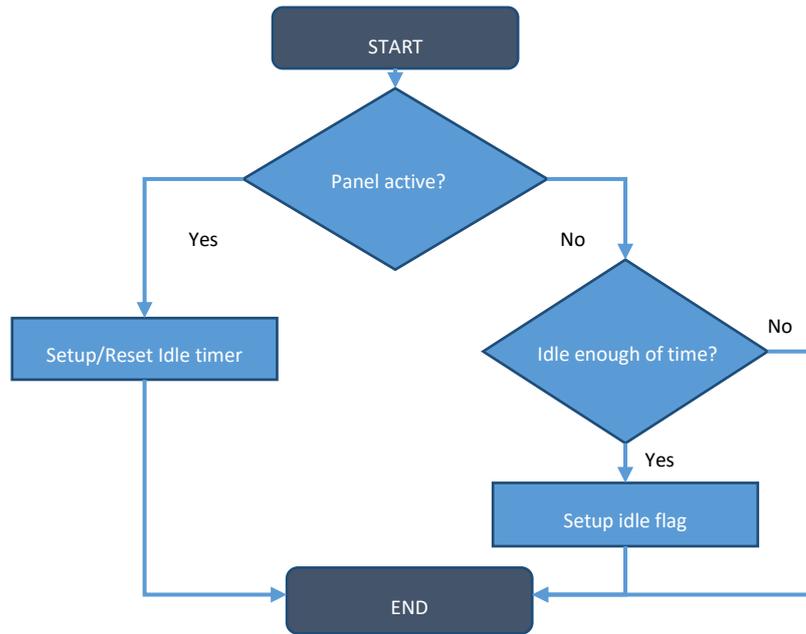
When the USB Remote Wakeup feature is enabled, it will also let PenMount device go into idle mode after system is suspended, otherwise it will go to sleep mode and will not wakeup until system is resumed.

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

7.9.2 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.

Power Modes	Scan Interval
Idle Mode	Configurable 10ms, 20ms, 30ms, 40ms, 50ms
Sleep Mode	Scan Stops



7.10 Miscellaneous

In the “miscellaneous” menu, users can enable or disable firmware supported features.

```

[Miscellaneous]
=====
(1) WatchDog      : Enabled
(2) Abnormal Message : Enabled
-----
Q. Exit to upper menu
=====
Please select action :
  
```

7.10.1 Watch Dog

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

7.10.2 Abnormal Message

This is a 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>

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.

7.11 Driving IC

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

```
[DriveIC]
=====
(1) High Voltage      : 18 V
-----
Q. Exit to upper menu
=====

Please select action :
```

7.11.1 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	* Touch may not work properly
6V	* Touch may not work properly

Please note that some low voltages may cause touch not working properly, because the measured signal will be too small and can affect normal operation.

8. Field Update

User can update firmware in the “Field Update” option. For advanced update features, please consider using the pmfu utility.

```
[Field Update]
=====
1. Load Firmware
2. Update firmware
-----
Q. Exit to upper menu
=====
Please select action :
```

8.1 Load Firmware

Please enter a relative or full path to the PenMount firmware file formats, such as pxc, pmc, or pfp.

Please make sure that the file extension is included.

The utility will not update the path until the specified file path is correct.

```
Firmware file path : PM1410_v21.pmc
Please enter the firmware file path : dummy.pmc
{WARNING} Specified file path is invalid !
```

8.2 Update Firmware

If no file path is specified, the utility will search for PenMount firmware files in the current directory, and user will need to choose one of them to proceed.

```
Please choose a firmware file listed below :
[0] PM1410_V12.pmc
[1] PM1410_v21.pmc
[2] PM1410_V20.pmc
[3] PM1410_V11.pmc
[4] PM1410_V10.pmc
File Index >
```

Once a valid firmware file is chosen, the utility will start update the firmware.

```
* Using firmware file "PM1410_v21.pmc"
=====
File      : PM1410_v21.pmc
Target    : PM1410
Version   : 2.1
Slave     : 2.1
=====

* {INFO} Waiting for device ready .

*****
|          Updating Firmware for Slave IC 0          |
|                      WARNING                      |
| DO NOT disconnect device until update finish    |
| Unexpected disconnection might damage device    |
*****
* {INFO} Updating firmware ... 100%
* {INFO} Waiting for device ready .

*****
|          Updating Firmware for Master IC          |
|                      WARNING                      |
| DO NOT disconnect device until update finish    |
| Unexpected disconnection might damage device    |
*****
* {INFO} Updating firmware ... 100%
* {INFO} Waiting for device ready .

*****
|          PenMount Firmware Update Finished !          |
*****
```

8.3 Command Line Options

Option	Description	Return Value
-fu	Firmware update	SUCCESS : 0 FAIL : Non-zero values
-reload-all	Reload all parameters after firmware update, including the host interface and panel parameters	N/A

9. Debug

In the Debug option, user can export firmware information to file for touch issue inspection.

```
[Debug]
=====
1. Log Current Device Information
2. Dump Device Data
-----
Q. Exit to upper menu
=====
Please select action :
```

9.1 Log Firmware Information

Saves firmware information, including memory dump, raw data and base line signal for debugging.

```
* Open device for logging ...
* Collecting device information ...
* Open device for dumping data ...
=====
[IC CONFIG]
* Data size : 0xA
* Dump data : 100%
=====
[SFR]
* Data size : 0x1000
* Dump data : 100%
=====
[DATA RAM]
* Data size : 0x8000
* Dump data : 100%
=====
* Collecting Base Signal ...
* Collecting Raw Count {#1/5} ...
* Collecting Raw Count {#2/5} ...
* Collecting Raw Count {#3/5} ...
* Collecting Raw Count {#4/5} ...
* Collecting Raw Count {#5/5} ...
* Collecting {Mode 0:1} Raw Count {#1/2} ...
* Collecting {Mode 0:1} Raw Count {#2/2} ...
* Collecting {Mode 0:2} Raw Count {#1/2} ...
* Collecting {Mode 0:2} Raw Count {#2/2} ...
* Collecting {Mode 0:3} Raw Count {#1/2} ...
* Collecting {Mode 0:3} Raw Count {#2/2} ...
* Collecting {Mode 1:1} Raw Count {#1/2} ...
* Collecting {Mode 1:1} Raw Count {#2/2} ...
* Collecting {Mode 1:2} Raw Count {#1/2} ...
* Collecting {Mode 1:2} Raw Count {#2/2} ...
* Collecting {Mode 1:3} Raw Count {#1/2} ...
* Collecting {Mode 1:3} Raw Count {#2/2} ...
↳ Firmware information : "Diagnostic/firmware_info_PM1310_6_1_0_20191003_174356.csv" !
↳ Log firmware info result : [OK]
```

9.2 Dump Device Data

Similar to Log Firmware Information, but this option will only dump current device memory to file.

```
* Open device for dumping data ...
* Total data size : 0x8000
* Dump data : 100%
* Device data is dumped to "Diagnostic/PM1310_T_1_0_0_dump.log" !

Dump device data result : [OK]
```

Please provide this file to PenMount team for analysis.

If this option is not shown in menu, it means that the feature is not supported.