

PENMOUNT PCIMSET USERS' GUIDE FOR LINUX

Version 1.13

03/Oct/'19



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 © 2019 All rights reserved

Revision Table

Date	Revision	Changes
03/10/2019	1.13	Documentation for Linux pcimset V1.13.x

Table of Content

PREFACE	I
Disclaimer	i
Trademarks.....	i
Copyright	i
REVISION TABLE.....	II
1. INTRODUCTION	5
1.1 Choosing the Correct Program	5
1.2 Operating System Support	5
1.3 PenMount Device Support	5
1.4 GUI Support.....	7
2. GENERAL USAGES	8
2.1 Program Version.....	8
2.2 Program Usage	8
2.3 Device Detection	9
2.3.1 Detect only USB Interface	10
2.3.2 Specify Device Path for non-USB Device	10
2.4 Reset Device	10
3. INFORMATION.....	11
4. DIAGNOSTIC	12
4.1 Basic Panel Open-Short Test.....	12
4.2 Get Noise Level.....	13
4.3 Build System Reference (SR).....	13
4.4 Live Test.....	15
4.5 Command Line Options	15
4.5.1 Live Test.....	15

4.5.2	Diagnostic Tests	16
5.	PARAMETERS	17
5.1	Interface	18
5.2	Panel 19	
5.3	Finger	20
5.4	Palm 21	
5.5	Water	21
5.6	Anti-Noise	22
5.7	Coordinate	24
6.	FIELD UPDATE	25
6.1	Load Firmware	25
6.2	Update Firmware	25
6.3	Command Line Options	26
7.	DEBUG	27
7.1	Log Firmware Information	27
7.2	Dump Device Data	27

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

Target Platform	Directory
ARM (soft float)	arm
ARMV7 (hard float)	armhf
x86 (32bit)	i686
x86 (64bit)	x86_64

The pcimset 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 PenMount pcimset supports the following Linux kernel configurations.

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

Please note that the Android system is not supported yet.

1.3 PenMount Device Support

The PenMount pcimset support the following PenMount devices.

Series	Control Board	USB	RS-232 /	I ² C ¹
--------	---------------	-----	----------	-------------------------------

¹ The I²C interface can be supported if kernel supports I²C device interfaces, or it can also be supported by connecting through the PMT101 Adapter.

UART				
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
	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

1.4 GUI Support

The PenMount pcimset 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 pcimset 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 pcimset.

2. General Usages

The functions of the PenMount pcimset 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 PCIMSet   |
|                               |
|=====
Version 1.13.0
```

2.2 Program Usage

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

```
./pcimset -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. Parameter
4. Field Update
5. 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 pcimset 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 ./pcimset -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 ./pcimset -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 : PM1310.5.2.0
Panel Size   : 10.4"
Cover Lens   : 1.8 mm (Glass)
SR Status    : NOT INITIALIZED
Parameters CRC : xC09F
=====
* 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.

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. Open-Short Test
2. Get Noise Level
3. Build SR
-----
Q. Exit to upper menu
=====

Please select action : █
```

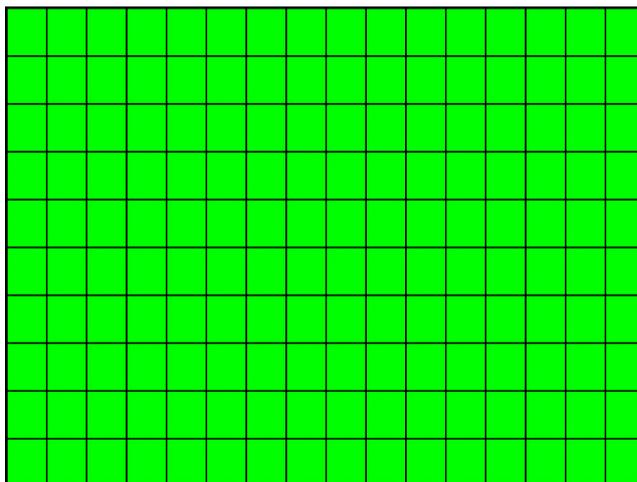
4.1 Basic Panel Open-Short Test

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.

```
[1] SR built environment is good !
[2] Open / short test for SR
* No open / short issue found !
[3] Verify base signal flatness ...
* Touch panel with good base signal regularity !

↳ SR diagnostic data : "./Diagnostic/SR_PM1310_5_2_0_20180205_181713.csv" !
* Using threshold values : 100, 100, 100, 100
-----
* Preparing Data Collection ...
-----
* Collecting Data {#0}
-----
* Collecting Data {#1}
-----
* Collecting Data {#2}
* Collecting data ... 100%
-----
* Checking Non-Driving Sensing ...
-----
* Collecting Data {#0}
-----
* Collecting Data {#1}
-----
* {INFO} Open/Short Test : {OK} !
↳ Open/Short diagnostic data : "./Diagnostic/OpenShort_PM1310_5_2_0_20180205_181727.csv"
↳ Diagnose result : [OK]
```

After test finishes, user can choose whether to show the diagnose result graphically.



The program will stay in the result screen for 3 seconds and return to menu. Please notice that this option is only available if Xorg X Server or framebuffer is available.

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.

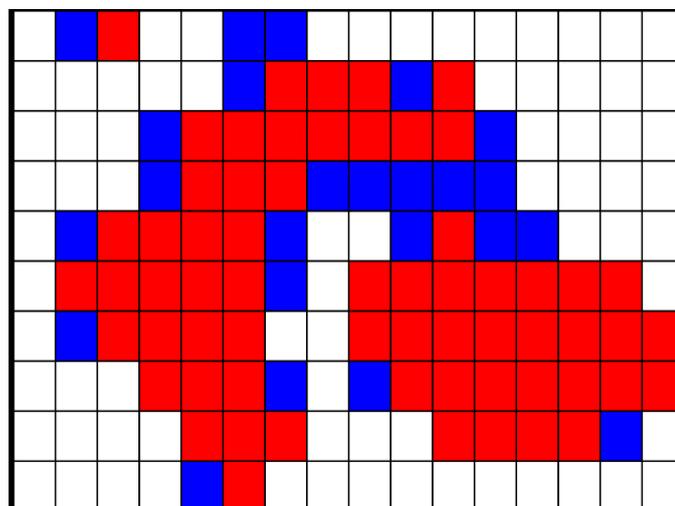
Test results will be saved automatically in “.CSV” file, and the utility will also rate the surrounding criteria with 5 different levels: “GREAT”, “GOOD”, “OK”, “POOR”, and “BAD”.

```
* Collecting data ... 100% >> Data Set {#50/50}
* Calculating noise level ...
* Saving noise level data ...
↳ Noise level diagnostic data : "./Diagnostic/NoiseLevel_PM1310_5_2_0_20180205_182325.csv" !
-----
Result : {GREAT} ⊙ No False touch ⊙ Line Drawing ⊙ Noise Susceptibility
-----
Display diagnose result (y/n) ? █
```

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	×	×	×

User can also display the diagnose result when test finishes.



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 : [NOT INITIALIZED]
-----
                [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.

```
* Continue building new SR ? (y|n) : y
* Building process will start in 2 seconds ...
* The screen will turn to white during calibration.
```

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.

```
=====
Checking Built Result ...
[1] SR built environment is good !
[2] Open / short test for SR
* No open / short issue found !
[3] Verify base signal flatness ...
* Touch panel with good base signal regularity !

↳ SR diagnostic data : "./Diagnostic/SR_PM1310_5_2_0_20180205_182722.csv" !
Checking Built Result ... [OK]
=====

↳ Building SR : [PASS] ! (WARNING: ENV NG)
```

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

```
* The built result may not be good enough to handle environmental changes !
* Do you still wish to save it ? (y|n) : n
* {WARNING} Revert 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 Live Test

The live test feature is only supported in device firmware without the “System Reference (SR)” feature.

This test will monitor the “base update count” value in device firmware. In most cases, the base update count should increase by time, and if it’s not updated correctly, user can provide the output file to PenMount team for further checking.

```
=====
* Live Test [START]
=====
[1] Checking Base Signal Update Status ...
[2017/04/17 21:06:23] Base update count : 0032
==> Firmware updates base signal normally !
-----
[2] Logging Base Signal ...
* Open device for logging ...
* Collecting device information ...
* Collecting Base Signal ...
* Collecting Raw Count {#1/2} ...
* Collecting Raw Count {#2/2} ...
-----
[3] Checking Base Signal Update Status ...
[2017/04/17 21:06:36] Base update count : 0059
==> Firmware updates base signal normally !
-----
==> Output file : "./Diagnostic/LiveTest_PM1410_5_0_5_20170417_210612.csv"
=====
* Live Test [END]
=====
```

4.5 Command Line Options

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

4.5.1 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

4.5.2 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. Parameters

Adjusting basic parameters for PenMount P2 touch controllers. These parameters are classified in several categories.

```
[Parameter] (0x18CB)
=====
(1) Interface
(2) Panel
(3) Finger
(4) Palm
(5) Water
(6) Anti-Noise
(7) Coordinate
-----
(I) Import Parameters
(E) Export Parameters
(R) Write Factory Default Parameters
-----
S. Apply (Exit to upper menu)
Q. Cancel (Exit to upper menu)
=====
```

- 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 PCIMset V2.4 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 pcimset.

- 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.

5.1 Interface

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

```
[Interface]
=====
(*) Interface           : USB (Not Adjustable)
-----
Q. Exit to upper menu
=====
```

- Interface

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)
PM210X / PM220X / PM230X	YES
PM1310 / PM141X / PM1710X	NO (Always AUTO)

5.2 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]
=====
(1) Panel Size      : 10.4"
(2) Cover Lens     : 1.8 mm (Glass)
-----
Q. Exit to upper 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 : |
```

- 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 :
```

5.3 Finger

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

```
[Finger]
=====
(1) Sensitivity : 0 (default)
(2) Maximum contact number : 5
-----
Q. Exit to upper menu
=====
```

- Sensitivity

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

Sensitivity	Value
Default	0
Insensitive	Negative value
Sensitive	Positive value

```
-----
[Sensitivity]
-----
{Default}
* Value : 0
-----
{Insensitive}
* Feature : Touch becomes less sensitive.
* Range : -1 ~ -4
-----
{Sensitive}
* Feature : Touch becomes more sensitive for operating with gloves.
* Range : 1 ~ 4
-----
```

- Maximum contact number

This option allows switching between single touch (1), dual touch (2), or multiple touch (more than 2).

```
[Maximum contact number]
-----
* Feature      : Maximum supported touch points
* Range       : 1 ~ 10
-----
```

5.4 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      : Disabled
-----
Q. Exit to upper menu
=====
```

- Palm Rejection

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

5.5 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
=====
```

- Water Solution Switch

The PenMount firmware supports several water handling modes such as WP and WD which can be changed dynamically. The value in “()” is the value that represents the corresponding mode.

```

-----
[Water Solution Switch]
-----
{Disable}
* Feature   : Disable water detect mode.
* Value    : 0
* Side Effect: Water may generate false touch, but will recover after water is removed.
-----
{Water Detect}
* Feature   : Touch function will be disabled when water is detected on touchscreen surface.
* Value    : d
* Side Effect: Touch will be disabled when water is detected.
-----
{Water Performance}
* Feature   : No false touch in wet areas, dry areas retain touch function.
* Value    : p
* Side Effect: Firmware will automatically switch to water detect mode when using bar type panel.
-----

```

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

Option	Water Handling	Description
0	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.

5.6 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) Median Filter            : OFF
(4) Enhanced Frequency Hopping : OFF
-----
Q. Exit to upper menu
=====

```

- 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.
-----

```

- Enhanced Frequency Hopping

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

```

-----
[Enhanced Frequency Hopping Switch]
-----
{ON}
* Feature      : Double or higher SNR at specified frequency.
* Value       : 1
* Side Effect  : Reduce sample rate and extend touch response time
-----
{OFF}
* Feature      : Normal noise filtering.
* Value       : 0
* Side Effect  : Normal operation.
-----

```

5.7 Coordinate

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

```

[Coordinate]
=====
Adjust Edge Compensation (Tail on bottom edge)
-----
(1) Left Edge           : 2
(2) Right Edge          : 2
(3) Top Edge            : 2
(4) Bottom Edge         : 2
-----
(5) Touch Panel Orientation : Landscape (0)
-----
Q. Exit to upper menu
=====

```

- Touch Panel 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)

- Adjust Edge Compensation

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

6. 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 :
```

6.1 Load Firmware

Please enter a relative or full path to the PenMount “pmc” or “pmf” file. 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 !
```

6.2 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 !          |
*****
```

6.3 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

7. 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 :
```

7.1 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]
```

7.2 Dump Device Data

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

```
* 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]
```

