

PENMOUNT TOUCH CONTROLLER LINUX / ANDROID DEVICE DRIVER SOURCE CODE USER GUIDE

Revision A

25/May/'20

Preface

Disclaimer

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

Trademarks

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

Copyright

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

Copyright © 2020 All rights reserved

Revision Table

Date	Revision	Changes
25/May/2020	A	Revised Release

Table of Content

Preface	i
Disclaimer.....	i
Trademarks.....	i
Copyright.....	i
Revision Table.....	ii
1. Introduction	4
2. USB Interface Support.....	6
2.1. Kernel Configurations.....	6
2.2. Using Customized PenMount Device Drivers.....	6
2.3. Enable Virtual Key	7
2.3.1. Kernel Before 3.14	7
2.3.2. Kernel 3.15 to 4.9.....	8
2.3.3. Kernel 4.10 and Later	9
3. RS-232 / UART Interface Support	10
3.1. Enable Kernel Support	10
3.2. Using Customized Kernel Driver.....	10
3.3. Loading the PenMount RS-232 kernel driver.	10
4. I2C Interface Support	12
4.1. Using Customized PenMount Driver	12
4.2. Interrupt Setup.....	12
4.3. Auto launch the device driver.....	12

1. Introduction

In most cases, touch function requires Linux kernel driver support. Most modern systems support PenMount touch controllers with its inbox driver. To check whether additional device driver is required, please check the following items.

1. Linux kernel version

PenMount devices are well supported after Linux kernel 3. The cells that marked grey color in table below are systems that require additional drivers provided by PenMount.

Controller	Interface	USB Product ID	Kernel Version			
			2.6	3.X	4.X	5.X
PM9000	RS-232	---		V3.2 ~ penmount		
PM6000	USB	0x6000	generic-usb	V3.5 ~ hid-generic	V3.18 ~ hid-penmount	
		0x6005	generic-usb	V3.5 ~ hid-generic		
	RS-232	---		V3.2 ~ penmount		
P2	USB	0x3500		V3.0 ~ hid-multitouch		
		0x3502	generic-usb	V3.5 ~ hid-generic		
		0x3508		3.4 ~ hid-multitouch		
	RS-232	---		V3.2 ~ penmount		
	I2C	---				

2. Linux Distribution

- Red Hat Enterprise Linux / CENTOS 4: using kernel V2.6.9, the bug in generic-usb driver prevents from loading the PenMount 6000 USB device correctly.
- Red Hat Enterprise Linux / CENTOS 6: using kernel V2.6.32. This version has been patched to support the hid-multitouch driver, so that PenMount P2 USB can be supported.¹
- Raspbian / SuSE Enterprise Server 12 SP3 / Raspbian: Using kernel 4.X. However, these image does not include hid-penmount support, so that PenMount 6000 USB cannot function correctly on those systems.

3. Special Firmware Features

- Virtual Button: This feature of PenMount P2 USB V6.0 firmware can run without any issue in Linux kernel V4.9 or later systems. Other kernel versions will require modifications of the hid-multitouch device driver.

¹ However, since the Xorg X Server version is too old in RHEL6, touch will still require additional XINPUT drivers to function correctly.

4. APP Support

- Android APP: Most modern Android systems uses Linux kernel 3 and later versions, so with USB interface, touch can be plug-and-play. Additional drivers are needed if using resistive touch controllers such as PenMount 6000, which requires calibration with user mode APP.

2. USB Interface Support

PenMount USB interface are supported in most Linux kernel versions.

Controller	Interface	USB Product ID	Kernel Version			
			2.6	3.X	4.X	5.X
PM6000	USB	0x6000	generic-usb	V3.5 hid-generic	V3.18 hid-penmount	
		0x6005	generic-usb	V3.5 hid-generic		
P2	USB	0x3500		V3.0 hid-multitouch		
		0x3502	generic-usb	V3.5 hid-generic		
		0x3508		3.4 hid-multitouch		

2.1. Kernel Configurations

The kernel drivers must be enabled when configuring the kernel. If touch is not functioning, please check the .config file of the system and be sure that the corresponding kernel drivers are enabled.

Kernel Driver	Configuration	Description
generic-usb	CONFIG_USB_HID=y	USB Human Interface Device (full HID) support
hid-generic	CONFIG_HID_GENERIC=m	Generic HID driver
hid-penmount	CONFIG_HID_PENMOUNT=m	Penmount touch device
hid-multitouch	CONFIG_HID_MULTITOUCH=m	HID Multitouch panels

2.2. Using Customized PenMount Device Drivers

PenMount provide sources codes for systems that do not have building support of PenMount devices.

Item	Description	
Kernel Driver	hid-penmount	
Configuration	CONFIG_HID_PENMOUNT=m	
Source Code	Source Code Path	<penmount_src_dir>/linux/driver/hid/hid-penmount.c
	Copy To	<kernel_src_dir>/drivers/hid/hid-penmount.c
Kconfig File	Path	<kernel_src_dir>/drivers/hid/Kconfig
	Add New Entry	config HID_PENMOUNT tristate "Penmount touch device" depends on USB_HID

		<div>---help---</div> <div>This selects a driver for the PenMount USB touch controller.</div>
Makefile File	Path	<kernel_src_dir>/drivers/hid/Makefile
	Add New Entry	obj-\$(CONFIG_HID_PENMOUNT) += hid-penmount.o

For kernel 2.6, additional setup is required for PenMount P2 USB devices.

Path	Kernel	File	Modification or New Entries
<kernel_src_dir>/drivers/hid		hid-ids.h	#define USB_VENDOR_ID_PENMOUNT 0x14E1 #define USB_DEVICE_ID_PENMOUNT_P2 0x3500 #define USB_DEVICE_ID_PENMOUNT_P2_WIN8 0x3508
	~ 2.6.32	hid.h	#define IS_INPUT_APPLICATION(a) (((a >= 0x00010000) && (a <= 0x00010008)) (a == 0x00010080) (a == 0x000c0001) ((a >= 0x000d0002) && (a <= 0x000d0006)))
	2.6.33 ~ 2.6.37	hid-core.c	static const struct hid_device_id hid_blacklist[] = { // ... { HID_USB_DEVICE(USB_VENDOR_ID_PENMOUNT, USB_DEVICE_ID_PENMOUNT_P2) }, { HID_USB_DEVICE(USB_VENDOR_ID_PENMOUNT, USB_DEVICE_ID_PENMOUNT_P2_WIN8) }, {} }
	2.6.38 ~ 2.6.39	hid-core.c	static const struct hid_device_id hid_have_special_driver[] = { // ... { HID_USB_DEVICE(USB_VENDOR_ID_PENMOUNT, USB_DEVICE_ID_PENMOUNT_P2) }, { HID_USB_DEVICE(USB_VENDOR_ID_PENMOUNT, USB_DEVICE_ID_PENMOUNT_P2_WIN8) }, {} }

2.3. Enable Virtual Key

When using the virtual kernel feature of PenMount P2 USB V5.2 ~ V6.X firmware versions with Linux kernel before 4.9, by default design, these hotkeys will be filtered out by hid-multitouch device drivers. In this case, please follow the steps below to modify hid-multitouch device drivers.

2.3.1. Kernel Before 3.14

Please modify hid-multitouch.c source file and find the following function:

```
static int mt_input_mapping(struct hid_device *hdev, struct hid_input *hi,
                          struct hid_field *field, struct hid_usage *usage,
                          unsigned long **bit, int *max)
{
    /* Only map fields from TouchScreen or TouchPad collections.
     * We need to ignore fields that belong to other collections
```



```

        * such as Mouse that might have the same GenericDesktop usages. */
        if (field->application != HID_DG_TOUCHSCREEN &&
            field->application != HID_DG_PEN &&
            field->application != HID_DG_TOUCHPAD) {
            return -1;
        }
    }
}

```

Since the HID Top Level Collection Usage Page of Virtual Keyis HID_UP_KEYBOARD, it will be ignored. Please change the code and return 0, instead of -1, if the target device is PenMount.

```

static int mt_input_mapping(struct hid_device *hdev, struct hid_input *hi,
                          struct hid_field *field, struct hid_usage *usage,
                          unsigned long **bit, int *max)
{
    /* Only map fields from TouchScreen or TouchPad collections.
     * We need to ignore fields that belong to other collections
     * such as Mouse that might have the same GenericDesktop usages. */
    if (field->application != HID_DG_TOUCHSCREEN &&
        field->application != HID_DG_PEN &&
        field->application != HID_DG_TOUCHPAD) {
        if (hi->input->id.vendor == USB_VENDOR_ID_PENMOUNT) {
            return 0;
        } else {
            return -1;
        }
    }
}

```

2.3.2. Kernel 3.15 to 4.9

Please open hid-multitouch source file, and find the following array.

```

static const struct hid_device_id mt_devices[] = {

    /* Generic MT device */
    { HID_DEVICE(HID_BUS_ANY, HID_GROUP_MULTITOUCH, HID_ANY_ID, HID_ANY_ID) },

    /* Generic Win 8 certified MT device */
    { .driver_data = MT_CLS_WIN_8,
      HID_DEVICE(HID_BUS_ANY, HID_GROUP_MULTITOUCH_WIN_8,
                  HID_ANY_ID, HID_ANY_ID) },
    {}
};

```

Please add a new entry before “Generic MT device”:

```

static const struct hid_device_id mt_devices[] = {

    /* PenMount devices */
    { .driver_data = MT_CLS_EXPORT_ALL_INPUTS,
      MT_USB_DEVICE(USB_VENDOR_ID_PENMOUNT,

```

```
0x3508) },

/* Generic MT device */
{ HID_DEVICE(HID_BUS_ANY, HID_GROUP_MULTITOUCH, HID_ANY_ID, HID_ANY_ID) },

/* Generic Win 8 certified MT device */
{ .driver_data = MT_CLS_WIN_8,
  HID_DEVICE(HID_BUS_ANY, HID_GROUP_MULTITOUCH_WIN_8,
    HID_ANY_ID, HID_ANY_ID) },
{}

};
```

2.3.3. Kernel 4.10 and Later

Kernel 4.10 and later versions can support the PenMount virtual key functions correctly without any modification required.

3. RS-232 / UART Interface Support

The PenMount RS-232 interface are supported after kernel V3.2.

Controller	Interface	ID	Kernel Version			
			2.6	3.X	4.X	5.X
PM9000	RS-232	0		V3.2 penmount		
PM6000	RS-232	1		V3.2 penmount		
P2	RS-232	2		V3.2 penmount		

3.1. Enable Kernel Support

The kernel drivers must be enabled when configuring the kernel. If touch is not functioning, please check the .config file of the system and be sure that the corresponding kernel drivers are enabled.

驱动程序	内核设置	说明
serio-penmount	CONFIG_TOUCHSCREEN_PENMOUNT=m	Penmount serial touchscreen

Please notice that the building driver supports dual touch only. For multitouch support, please use the customized device driver provided by PenMount.

3.2. Using Customized Kernel Driver

PenMount provides customized kernel driver that supports Linux kernel before 3.2.

Item	内容	
Kernel Driver	serio-penmount	
Configuration	CONFIG_TOUCHSCREEN_PENMOUNT =m	
Source Code	Source Code Path	<penmount_src_dir>/linux/driver/serial/penmount.c
	Copy To	<kernel_src_dir>/drivers/input/touchscreen/penmount.c
Kconfig File	Path	<kernel_src_dir>/drivers/input/touchscreen/Kconfig
	Add New Entry	Not required
Makefile File	Path	<kernel_src_dir>/drivers/input/touchscreen/Makefile
	Add New Entry	Not required

3.3. Loading the PenMount RS-232 kernel driver.

The PenMount RS-232 kernel driver is a SERIO type device driver, which requires additional utility to launch manually.

There is a public utility called inputattach which supports various RS-232 devices. PenMount also provides it's own utility. Please select one for compilation.

Utility	Source Code Path	Example (Suppose device on /dev/ttyS1)
pmsAttach	<penmount_src_dir>/linux/driver/serial/pmsAttach	pmsAttach 9000 /dev/ttyS1 19200
		pmsAttach 6000 /dev/ttyS1 19200
		pmsAttach PCI /dev/ttyS1 38400
inputattach	https://github.com/flosse/linuxconsole/	inputattach -baud 19200 -pm9k /dev/ttyS1
		inputattach -baud 19200 -pm6k /dev/ttyS1
		inputattach -baud 38400 -pm3k /dev/ttyS1

4. I2C Interface Support

For I2C interface, please use the customized PenMount kernel driver.

4.1. Using Customized PenMount Driver

Item	Description	
Kernel Driver	penmount_i2c	
Configuration	CONFIG_TOUCHSCREEN_PENMOUNT_I2C=m	
Source Code	Source Code Path	<penmount_src_dir>/linux/driver/i2c/penmount-i2c.c
	Copy To	<kernel_src_dir>/drivers/input/touchscreen/penmount-i2c.c
Kconfig File	Path	<kernel_src_dir>/drivers/input/touchscreen/Kconfig
	Add New Entry	<pre>config PENMOUNT_TOUCHSCREEN_I2C tristate "Penmount I2C touchscreen" depends on I2C ---help--- This selects a driver for the PenMount I2C touch controller.</pre>
Makefile File	Path	<kernel_src_dir>/drivers/input/touchscreen/Makefile
	Add New Entry	obj-\$(CONFIG_TOUCHSCREEN_PENMOUNT_I2C) += penmount-i2c.o

4.2. Interrupt Setup

The PenMount I2C device driver can operate with Interrupt driven mode or polling mode. In most cases, it is recommended using interrupt driven mode by connecting the INT PIN to a selected GPIO pin on board.

There are several ways to configure the interrupt pin.

Option	Actions	Description	Example
Modify PenMount Driver Source Code	Define the PENMOUNT_I2C_GPIO_IRQ	Define the GPIO pin	#define GPIO_TO_PIN(bank, gpio) (32 * (bank) + (gpio)) GPIO_TO_PIN(3,19)
	Implement the penmount_i2c_init_gpio ()	Configure GPIO to be input, pull high, and low level triggered.	
Use Device Tree Overlays	Configure the "interrupt-parent" "Interrupts" items	Define the GPIO pin and configure to be input, pull high, and low level triggered.	interrupt-parent = <&gpio3>; interrupts = <19 8>; /* IRQ_TYPE_LEVEL_LOW */

4.3. Auto launch the device driver

The PenMount I2C driver is a I2C client driver that requires additional configurations to be launched correctly. There are two possible options.

Option	Modifications	Path (Example)	Configurations (Example)
--------	---------------	----------------	--------------------------

Modify platform settings directly.	i2c_board_info[]	(S3C2440) <kernel_src_dir>/kernel/arch/arm/mach-s3c2440/mach-s3c2440.c	Find friendly_arm_i2c_devices[] and add the following entry. #include <linux/i2c.h> static struct i2c_board_info friendly_arm_i2c_devices[] __initdata = { { I2C_BOARD_INFO("penmount_i2c", 0x38) }, }; static void __init mini2440_machine_init(void) { i2c_register_board_info(0, friendly_arm_i2c_devices, ARRAY_SIZE(friendly_arm_i2c_devices)); s3c_i2c0_set_platdata(NULL); }
Use DeviceTree Overlays	compatible reg		compatible = "penmount,penmount_i2c"; reg = <0x38>;