android 分析

个人资料

hellowxwworld

访问： 153947次
积分： 1736
等级： BLOG 4
排名： 千里之外

原创： 20篇
转载： 50篇
译文： 0篇
评论： 10条

在 Android 通过 get_event 获得 input 设备 上报event

2013-08-31 14:31 7875人阅读 评论

分类： Android（10）

目录(?) [+]

Android 本身有一个genevnet 和 sendevent 工具用来从内核获取event事件和向内核发送event事件，具体可

Android 下收发input事件的工具 getevent 和 sendevent

o get_evnet 是一个可以获得注册成input设备上报event的调试工具。

o usage: get_event /dev/input/eventX, X is 0, 1, 2，3， 4, 5，6

o 附件为其源代码， 可以采用静态编译，然后通过adb push进 手机， 进行调试。

arm-linux-gcc -static get_event.c -o get_evnet

-> 获得 红外gp2ap 的上报 event

```
shell@android:/ # get_event /dev/input/event0
Input driver version is 1.0.1
Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0
Input device name: gp2ap
Supported events:
```

关闭

文章分类

Android (11)
C/C++ (6)
Java (0)
bootloader (0)
algorithm (7)
Alsa (13)
Embedded (0)
Arm (1)
Linux (26)
multimedia (3)
recovery (0)
tools (0)
device driver (0)

文章存档

2013年09月 (12)
2013年08月 (21)
2013年06月 (1)
2013年05月 (4)
2013年04月 (2)

展开

阅读排行

Android 下收发input事件 (17362)
ida debug android so (16360)
跟踪Android callback 调

Event type 0 (Sync)
Event type 3 (Absolute)
Event code 25 (Distance)
Value -1
Min 0
Max 1
Event code 40 (Misc)
Value 0
Min 0
Max 65535
Testing ... (interrupt to exit)

-> 获得touchscreen 的上报 event

130|shell@android:/ # get_event /dev/input/event1
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0x0 product 0x1 version 0x1005
Input device name: mx_ts
Supported events:
Event type 0 (Sync)
Event type 1 (Key)
Event code 102 (Home)
Event code 103 (Up)
Event code 105 (Left)
Event code 106 (Right)
Event code 107 (End)
Event code 108 (Down)
Event code 129 (Again)
Event code 158 (Back)

关闭

根据内核Oops 定位代码 (13158)
regmap使用介绍 (11533)
在 Android 通过 get_eve (10903)
用IntelliJ代替Eclipse，调 (7869)
基于ffmpeg截取视频帧画 (5693)
信号量 (3873)
内核container_of(ptr,type (3716)
(3085)

评论排行

根据内核Oops 定位代码 (5)
ida debug android so (4)
利用用平台GPIO LED调 (1)
ALSA SOC架构关键点 (0)
lowlevel_init.S (0)
信号量 (0)
SurfaceFlinger启动过程分 (0)
linux_sound_alsa_ALSA (0)
linux_sound_alsa_ALSA (0)
linux内核空间和用户空间 (0)

推荐文章

* CSDN新版博客feed流内测用户征集令
* Android检查更新下载安装
* 动手打造史上最简单的Recycleview 侧滑菜单
* TCP网络通讯如何解决分包粘包问题

Event code 325 (ToolFinger)
Event code 330 (Touch)
Event type 3 (Absolute)
Event code 0 (X)
Value 0
Min 0
Max 1334
Event code 1 (Y)
Value 0
Min 0
Max 2214
Event code 24 (Pressure)
Value 0
Min 0
Max 255
Event code 47 (?)
Value 0
Min 0
Max 9
Event code 48 (ABS_MT_TOUCH_MAJOR)
Value 0
Min 0
Max 15
Event code 53 (ABS_MT_POSITION_X)
Value 0
Min 0
Max 1334

关闭

Event code 54 (ABS_MT_POSITION_Y)

Value 0

Min 0

Max 2214

Event code 57 (ABS_MT_TRACKING_ID)

Value 0

Min 0

Max 65535

Event code 58 (ABS_MT_PRESSURE)

Value 0

Min 0

Max 255

Testing ... (interrupt to exit)

Event: time 1325376099.794968, type 3 (Absolute), code 57 (ABS_MT_TRACKING_ID), value 0

Event: time 1325376099.794990, type 1 (Key), code 330 (Touch), value 1

Event: time 1325376099.794999, type 1 (Key), code 325 (ToolFinger), value 1

Event: time 1325376099.795009, type 3 (Absolute), code 53 (ABS_MT_POSITION_X), value 738

Event: time 1325376099.795018, type 3 (Absolute), code 54 (ABS_MT_POSITI

Event: time 1325376099.795027, type 3 (Absolute), code 58 (ABS_MT_PRESSURE), value 84

Event: time 1325376099.795036, type 3 (Absolute), code 48 (ABS_MT_TOUCH_MAJOR), value 4

Event: time 1325376099.795090, ---------- Report Sync ----------

Event: time 1325376099.903355, type 3 (Absolute), code 57 (ABS_MT_TRACKING_ID), value -1

Event: time 1325376099.903432, ---------- Report Sync ----------

-> 获得物理按键key 的上报 event

130|shell@android:/ # get_event /dev/input/event2

Input driver version is 1.0.1

Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100

Input device name: gpio-keys

Supported events:

Event type 0 (Sync)

Event type 1 (Key)

Event code 102 (Home)

Event code 114 (VolumeDown)

Event code 115 (VolumeUp)

Event code 116 (Power)

Testing ... (interrupt to exit)

Event: time 1325376161.600192, type 1 (Key), code 115 (VolumeUp), value 1

Event: time 1325376161.600214, ---------- Report Sync ----------

Event: time 1325376161.794925, type 1 (Key), code 115 (VolumeUp), value 0

Event: time 1325376161.794933, ---------- Report Sync ----------

Event: time 1325376162.645193, type 1 (Key), code 102 (Home), value 1

Event: time 1325376162.645215, ---------- Report Sync ----------

Event: time 1325376162.794977, type 1 (Key), code 102 (Home), value 0

Event: time 1325376162.794995, ---------- Report Sync ----------

Event: time 1325376164.970050, type 1 (Key), code 114 (VolumeDown), value 1

Event: time 1325376164.970072, ---------- Report Sync ----------

Event: time 1325376164.975038, type 1 (Key), code 114 (VolumeDown), value 0

Event: time 1325376164.975058, ---------- Report Sync ----------

Event: time 1325376165.000110, type 1 (Key), code 114 (VolumeDown), value 1

Event: time 1325376165.000135, ---------- Report Sync ----------

Event: time 1325376165.005173, type 1 (Key), code 114 (VolumeDown), value 0

Event: time 1325376165.005190, ---------- Report Sync ----------

Event: time 1325376166.700714, type 1 (Key), code 116 (Power), value 1

Event: time 1325376166.700737, ---------- Report Sync ----------

Event: time 1325376167.005092, type 1 (Key), code 116 (Power), value 0

Event: time 1325376167.005110, ---------- Report Sync ----------

-> 获得headset 的上报 event

130|shell@android:/ # get_event /dev/input/event3

Input driver version is 1.0.1

Input device ID: bus 0x0 vendor 0x0 product 0x0 version 0x0

Input device name: Headset

Supported events:

  Event type 0 (Sync)

  Event type 1 (Key)

    Event code 226 (Media)

    Event code 259 (Btn3)

    Event code 260 (Btn4)

    Event code 261 (Btn5)

Testing ... (interrupt to exit)

Event: time 1325376242.312919, type 1 (Key), code 261 (Btn5), value 1

Event: time 1325376242.312944, ---------- Report Sync ----------

Event: time 1325376242.463815, type 1 (Key), code 261 (Btn5), value 0

Event: time 1325376242.463837, ---------- Report Sync ----------

Event: time 1325376243.583124, type 1 (Key), code 261 (Btn5), value 1

Event: time 1325376243.583150, ---------- Report Sync ----------

Event: time 1325376243.690958, type 1 (Key), code 261 (Btn5), value 0

Event: time 1325376243.690979, ---------- Report Sync ----------

Event: time 1325376245.200309, type 1 (Key), code 226 (Media), value 1

关闭

Event: time 1325376245.200334, ---------- Report Sync ----------

Event: time 1325376245.302111, type 1 (Key), code 226 (Media), value 0

Event: time 1325376245.302145, ---------- Report Sync ----------

get_event.c 源码， 具体可调整里面对应的key-value 值进行调试

```
#include <stdint.h>
#include <linux/input.h>
#include <string.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
#ifndef EV_SYN
#define EV_SYN 0
#endif
char *events[EV_MAX + 1] = {
    [0 ... EV_MAX] = NULL,
    [EV_SYN] = "Sync",            [EV_KEY] = "Key",
    [EV_REL] = "Relative",            [EV_ABS] = "Absolute",
    [EV_MSC] = "Misc",            [EV_LED] = "LED",
    [EV_SND] = "Sound",             [EV_REP] = "Repeat",
    [EV_FF] = "ForceFeedback",        [EV_PWR] = "Power",
    [EV_FF_STATUS] = "ForceFeedbackStatus",
};

char *keys[KEY_MAX + 1] = {
    [0 ... KEY_MAX] = NULL,
    [KEY_RESERVED] = "Reserved",        [KEY_ESC] = "Esc",
    [KEY_1] = "1",                [KEY_2] = "2",
    [KEY_3] = "3",                [KEY_4] = "4",
    [KEY_5] = "5",                [KEY_6] = "6",
    [KEY_7] = "7",                [KEY_8] = "8",
    [KEY_9] = "9",                [KEY_0] = "0",
    [KEY_MINUS] = "Minus",            [KEY_EQUAL] = "Equal",
    [KEY_BACKSPACE] = "Backspace",        [KEY_TAB] = "Tab",
    [KEY_Q] = "Q",                [KEY_W] = "W",
    [KEY_E] = "E",                [KEY_R] = "R",
    [KEY_T] = "T",                [KEY_Y] = "Y",
    [KEY_U] = "U",                [KEY_I] = "I",
    [KEY_O] = "O",                [KEY_P] = "P",
    [KEY_LEFTBRACE] = "LeftBrace",        [KEY_RIGHTBRACE] = "RightBrace",
    [KEY_ENTER] = "Enter",            [KEY_LEFTCTRL] = "LeftControl",
    [KEY_A] = "A",                [KEY_S] = "S",
    [KEY_D] = "D",                [KEY_F] = "F",
    [KEY_G] = "G",                [KEY_H] = "H",
    [KEY_J] = "J",                [KEY_K] = "K",
    [KEY_L] = "L",                [KEY_SEMICOLON] = "Semicolon",
    [KEY_APOSTROPHE] = "Apostrophe",    [KEY_GRAVE] = "Grave",
    [KEY_LEFTSHIFT] = "LeftShift",        [KEY_BACKSLASH] = "BackSlash",
    [KEY_Z] = "Z",                [KEY_X] = "X",
    [KEY_C] = "C",                [KEY_V] = "V",
```

关闭

```
[KEY_B] = "B",                [KEY_N] = "N",
[KEY_M] = "M",                [KEY_COMMA] = "Comma",
[KEY_DOT] = "Dot",            [KEY_SLASH] = "Slash",
[KEY_RIGHTSHIFT] = "RightShift",    [KEY_KPASTERISK] = "KPAsterisk",
[KEY_LEFTALT] = "LeftAlt",        [KEY_SPACE] = "Space",
[KEY_CAPSLOCK] = "CapsLock",        [KEY_F1] = "F1",
[KEY_F2] = "F2",            [KEY_F3] = "F3",
[KEY_F4] = "F4",            [KEY_F5] = "F5",
[KEY_F6] = "F6",            [KEY_F7] = "F7",
[KEY_F8] = "F8",            [KEY_F9] = "F9",
[KEY_F10] = "F10",            [KEY_NUMLOCK] = "NumLock",
[KEY_SCROLLLOCK] = "ScrollLock",    [KEY_KP7] = "KP7",
[KEY_KP8] = "KP8",            [KEY_KP9] = "KP9",
[KEY_KPMINUS] = "KPMinus",        [KEY_KP4] = "KP4",
[KEY_KP5] = "KP5",            [KEY_KP6] = "KP6",
[KEY_KPPLUS] = "KPPlus",        [KEY_KP1] = "KP1",
[KEY_KP2] = "KP2",            [KEY_KP3] = "KP3",
[KEY_KP0] = "KP0",            [KEY_KPDOT] = "KPDot",
[KEY_ZENKAKUHANKAKU] = "Zenkaku/Hankaku", [KEY_102ND] = "102nd",
[KEY_F11] = "F11",            [KEY_F12] = "F12",
[KEY_RO] = "RO",            [KEY_KATAKANA] = "Katakana",
[KEY_HIRAGANA] = "HIRAGANA",        [KEY_HENKAN] = "Henkan",
[KEY_KATAKANAHIRAGANA] = "Katakana/Hiragana", [KEY_MUHENKAN] = "Muhenkan",
[KEY_KPJPCOMMA] = "KPJpComma",        [KEY_KPENTER] = "KPEnter",
[KEY_RIGHTCTRL] = "RightCtrl",        [KEY_KPSLASH] = "KPSlash",
[KEY_SYSRQ] = "SysRq",            [KEY_RIGHTALT] = "RightAlt",
[KEY_LINEFEED] = "LineFeed",        [KEY_HOME] = "Home",
[KEY_UP] = "Up",            [KEY_PAGEUP] = "PageUp",
[KEY_LEFT] = "Left",            [KEY_RIGHT] = "Right",
[KEY_END] = "End",            [KEY_DOWN] = "Down",
[KEY_PAGEDOWN] = "PageDown",        [KEY_INSERT] = "Insert",
[KEY_DELETE] = "Delete",        [KEY_MACRO] = "Macro",
[KEY_MUTE] = "Mute",            [KEY_VOLUMEDOWN] = "VolumeDown",
[KEY_VOLUMEUP] = "VolumeUp",        [KEY_POWER] = "Power",
[KEY_KPEQUAL] = "KPEqual",        [KEY_KPPLUSMINUS] = "KPPlusMi
[KEY_PAUSE] = "Pause",            [KEY_KPCOMMA] = "KPComma",
[KEY_HANGUEL] = "Hanguel",        [KEY_HANJA] = "Hanja",
[KEY_YEN] = "Yen",            [KEY_LEFTMETA] = "LeftMeta",
[KEY_RIGHTMETA] = "RightMeta",        [KEY_COMPOSE] = "Compose",
[KEY_STOP] = "Stop",            [KEY_AGAIN] = "Again",
[KEY_PROPS] = "Props",            [KEY_UNDO] = "Undo",
[KEY_FRONT] = "Front",            [KEY_COPY] = "Copy",
[KEY_OPEN] = "Open",            [KEY_PASTE] = "Paste",
[KEY_FIND] = "Find",            [KEY_CUT] = "Cut",
[KEY_HELP] = "Help",            [KEY_MENU] = "Menu",
[KEY_CALC] = "Calc",            [KEY_SETUP] = "Setup",
[KEY_SLEEP] = "Sleep",            [KEY_WAKEUP] = "WakeUp",
[KEY_FILE] = "File",            [KEY_SENDFILE] = "SendFile",
[KEY_DELETEFILE] = "DeleteFile",    [KEY_XFER] = "X-fer",
[KEY_PROG1] = "Prog1",            [KEY_PROG2] = "Prog2",
[KEY_WWW] = "WWW",            [KEY_MSDOS] = "MSDOS",
[KEY_COFFEE] = "Coffee",        [KEY_DIRECTION] = "Direction",
[KEY_CYCLEWINDOWS] = "CycleWindows",    [KEY_MAIL] = "Mail",
[KEY_BOOKMARKS] = "Bookmarks",        [KEY_COMPUTER] = "Computer",
[KEY_BACK] = "Back",            [KEY_FORWARD] = "Forward",
```

关闭

```
[KEY_CLOSECD] = "CloseCD",          [KEY_EJECTCD] = "EjectCD",
[KEY_EJECTCLOSECD] = "EjectCloseCD",     [KEY_NEXTSONG] = "NextSong",
[KEY_PLAYPAUSE] = "PlayPause",         [KEY_PREVIOUSSONG] = "PreviousSong",
[KEY_STOPCD] = "StopCD",         [KEY_RECORD] = "Record",
[KEY_REWIND] = "Rewind",         [KEY_PHONE] = "Phone",
[KEY_ISO] = "ISOKey",             [KEY_CONFIG] = "Config",
[KEY_HOMEPAGE] = "HomePage",          [KEY_REFRESH] = "Refresh",
[KEY_EXIT] = "Exit",             [KEY_MOVE] = "Move",
[KEY_EDIT] = "Edit",             [KEY_SCROLLUP] = "ScrollUp",
[KEY_SCROLLDOWN] = "ScrollDown",     [KEY_KPLEFTPAREN] = "KPLeftParenthesis",
[KEY_KPRIGHTPAREN] = "KPRightParenthesis", [KEY_F13] = "F13",
[KEY_F14] = "F14",             [KEY_F15] = "F15",
[KEY_F16] = "F16",             [KEY_F17] = "F17",
[KEY_F18] = "F18",             [KEY_F19] = "F19",
[KEY_F20] = "F20",             [KEY_F21] = "F21",
[KEY_F22] = "F22",             [KEY_F23] = "F23",
[KEY_F24] = "F24",             [KEY_PLAYCD] = "PlayCD",
[KEY_PAUSECD] = "PauseCD",          [KEY_PROG3] = "Prog3",
[KEY_PROG4] = "Prog4",             [KEY_SUSPEND] = "Suspend",
[KEY_CLOSE] = "Close",             [KEY_PLAY] = "Play",
[KEY_FASTFORWARD] = "Fast Forward",     [KEY_BASSBOOST] = "Bass Boost",
[KEY_PRINT] = "Print",             [KEY_HP] = "HP",
[KEY_CAMERA] = "Camera",          [KEY_SOUND] = "Sound",
[KEY_QUESTION] = "Question",          [KEY_EMAIL] = "Email",
[KEY_CHAT] = "Chat",             [KEY_SEARCH] = "Search",
[KEY_CONNECT] = "Connect",          [KEY_FINANCE] = "Finance",
[KEY_SPORT] = "Sport",             [KEY_SHOP] = "Shop",
[KEY_ALTERASE] = "Alternate Erase",     [KEY_CANCEL] = "Cancel",
[KEY_BRIGHTNESSDOWN] = "Brightness down", [KEY_BRIGHTNESSUP] = "Brightness up",
[KEY_MEDIA] = "Media",             [KEY_UNKNOWN] = "Unknown",
[BTN_0] = "Btn0",             [BTN_1] = "Btn1",
[BTN_2] = "Btn2",             [BTN_3] = "Btn3",
[BTN_4] = "Btn4",             [BTN_5] = "Btn5",
[BTN_6] = "Btn6",             [BTN_7] = "Btn7",
[BTN_8] = "Btn8",             [BTN_9] = "Btn9",
[BTN_LEFT] = "LeftBtn",              [BTN_RIGHT] = "RightBtn",
[BTN_MIDDLE] = "MiddleBtn",          [BTN_SIDE] = "SideBtn",
[BTN_EXTRA] = "ExtraBtn",         [BTN_FORWARD] = "ForwardBtn",
[BTN_BACK] = "BackBtn",            [BTN_TASK] = "TaskBtn",
[BTN_TRIGGER] = "Trigger",         [BTN_THUMB] = "ThumbBtn",
[BTN_THUMB2] = "ThumbBtn2",          [BTN_TOP] = "TopBtn",
[BTN_TOP2] = "TopBtn2",            [BTN_PINKIE] = "PinkieBtn",
[BTN_BASE] = "BaseBtn",            [BTN_BASE2] = "BaseBtn2",
[BTN_BASE3] = "BaseBtn3",         [BTN_BASE4] = "BaseBtn4",
[BTN_BASE5] = "BaseBtn5",         [BTN_BASE6] = "BaseBtn6",
[BTN_DEAD] = "BtnDead",            [BTN_A] = "BtnA",
[BTN_B] = "BtnB",             [BTN_C] = "BtnC",
[BTN_X] = "BtnX",             [BTN_Y] = "BtnY",
[BTN_Z] = "BtnZ",             [BTN_TL] = "BtnTL",
[BTN_TR] = "BtnTR",              [BTN_TL2] = "BtnTL2",
[BTN_TR2] = "BtnTR2",              [BTN_SELECT] = "BtnSelect",
[BTN_START] = "BtnStart",          [BTN_MODE] = "BtnMode",
[BTN_THUMBL] = "BtnThumbL",          [BTN_THUMBR] = "BtnThumbR",
[BTN_TOOL_PEN] = "ToolPen",          [BTN_TOOL_RUBBER] = "ToolRubber",
[BTN_TOOL_BRUSH] = "ToolBrush",         [BTN_TOOL_PENCIL] = "ToolPencil",
```

关闭

```
    [BTN_TOOL_AIRBRUSH] = "ToolAirbrush",    [BTN_TOOL_FINGER] = "ToolFinger",
    [BTN_TOOL_MOUSE] = "ToolMouse",        [BTN_TOOL_LENS] = "ToolLens",
    [BTN_TOUCH] = "Touch",            [BTN_STYLUS] = "Stylus",
    [BTN_STYLUS2] = "Stylus2",        [BTN_TOOL_DOUBLETAP] = "Tool Doubletap",
    [BTN_TOOL_TRIPLETAP] = "Tool Tripletap", [BTN_GEAR_DOWN] = "WheelBtn",
    [BTN_GEAR_UP] = "Gear up",        [KEY_OK] = "Ok",
    [KEY_SELECT] = "Select",        [KEY_GOTO] = "Goto",
    [KEY_CLEAR] = "Clear",            [KEY_POWER2] = "Power2",
    [KEY_OPTION] = "Option",        [KEY_INFO] = "Info",
    [KEY_TIME] = "Time",            [KEY_VENDOR] = "Vendor",
    [KEY_ARCHIVE] = "Archive",        [KEY_PROGRAM] = "Program",
    [KEY_CHANNEL] = "Channel",        [KEY_FAVORITES] = "Favorites",
    [KEY_EPG] = "EPG",            [KEY_PVR] = "PVR",
    [KEY_MHP] = "MHP",            [KEY_LANGUAGE] = "Language",
    [KEY_TITLE] = "Title",            [KEY_SUBTITLE] = "Subtitle",
    [KEY_ANGLE] = "Angle",            [KEY_ZOOM] = "Zoom",
    [KEY_MODE] = "Mode",            [KEY_KEYBOARD] = "Keyboard",
    [KEY_SCREEN] = "Screen",        [KEY_PC] = "PC",
    [KEY_TV] = "TV",            [KEY_TV2] = "TV2",
    [KEY_VCR] = "VCR",            [KEY_VCR2] = "VCR2",
    [KEY_SAT] = "Sat",            [KEY_SAT2] = "Sat2",
    [KEY_CD] = "CD",            [KEY_TAPE] = "Tape",
    [KEY_RADIO] = "Radio",            [KEY_TUNER] = "Tuner",
    [KEY_PLAYER] = "Player",        [KEY_TEXT] = "Text",
    [KEY_DVD] = "DVD",            [KEY_AUX] = "Aux",
    [KEY_MP3] = "MP3",            [KEY_AUDIO] = "Audio",
    [KEY_VIDEO] = "Video",            [KEY_DIRECTORY] = "Directory",
    [KEY_LIST] = "List",            [KEY_MEMO] = "Memo",
    [KEY_CALENDAR] = "Calendar",        [KEY_RED] = "Red",
    [KEY_GREEN] = "Green",            [KEY_YELLOW] = "Yellow",
    [KEY_BLUE] = "Blue",            [KEY_CHANNELUP] = "ChannelUp",
    [KEY_CHANNELDOWN] = "ChannelDown",    [KEY_FIRST] = "First",
    [KEY_LAST] = "Last",            [KEY_AB] = "AB",
    [KEY_NEXT] = "Next",            [KEY_RESTART] = "Restart",
    [KEY_SLOW] = "Slow",            [KEY_SHUFFLE] = "Shuffle",
    [KEY_BREAK] = "Break",            [KEY_PREVIOUS] = "Previous",
    [KEY_DIGITS] = "Digits",        [KEY_TEEN] = "TEEN",
    [KEY_TWEN] = "TWEN",            [KEY_DEL_EOL] = "Delete EOL",
    [KEY_DEL_EOS] = "Delete EOS",        [KEY_INS_LINE] = "Insert line",
    [KEY_DEL_LINE] = "Delete line",
};
char *absval[5] = { "Value", "Min  ", "Max  ", "Fuzz ", "Flat " };
char *relatives[REL_MAX + 1] = {
    [0 ... REL_MAX] = NULL,
    [REL_X] = "X",            [REL_Y] = "Y",
    [REL_Z] = "Z",            [REL_HWHEEL] = "HWheel",
    [REL_DIAL] = "Dial",        [REL_WHEEL] = "Wheel",
    [REL_MISC] = "Misc",
};
char *absolutes[ABS_MAX + 1] = {
    [0 ... ABS_MAX] = NULL,
    [ABS_X] = "X",            [ABS_Y] = "Y",
    [ABS_Z] = "Z",            [ABS_RX] = "Rx",
    [ABS_RY] = "Ry",        [ABS_RZ] = "Rz",
    [ABS_THROTTLE] = "Throttle",    [ABS_RUDDER] = "Rudder",
```

关闭

```
    [ABS_WHEEL] = "Wheel",          [ABS_GAS] = "Gas",
    [ABS_BRAKE] = "Brake",          [ABS_HAT0X] = "Hat0X",
    [ABS_HAT0Y] = "Hat0Y",          [ABS_HAT1X] = "Hat1X",
    [ABS_HAT1Y] = "Hat1Y",          [ABS_HAT2X] = "Hat2X",
    [ABS_HAT2Y] = "Hat2Y",          [ABS_HAT3X] = "Hat3X",
    [ABS_HAT3Y] = "Hat 3Y",          [ABS_PRESSURE] = "Pressure",
    [ABS_DISTANCE] = "Distance",     [ABS_TILT_X] = "XTilt",
    [ABS_TILT_Y] = "YTilt",          [ABS_TOOL_WIDTH] = "Tool Width",
    [ABS_VOLUME] = "Volume",    [ABS_MISC] = "Misc",

#define ABS_MT_TOUCH_MAJOR  0x30    /* Major axis of touching ellipse */
#define ABS_MT_TOUCH_MINOR  0x31    /* Minor axis (omit if circular) */
#define ABS_MT_WIDTH_MAJOR  0x32    /* Major axis of approaching ellipse */
#define ABS_MT_WIDTH_MINOR  0x33    /* Minor axis (omit if circular) */
#define ABS_MT_ORIENTATION  0x34    /* Ellipse orientation */
#define ABS_MT_POSITION_X   0x35    /* Center X ellipse position */
#define ABS_MT_POSITION_Y   0x36    /* Center Y ellipse position */
#define ABS_MT_TOOL_TYPE    0x37    /* Type of touching device */
#define ABS_MT_BLOB_ID      0x38    /* Group a set of packets as a blob */
#define ABS_MT_TRACKING_ID  0x39    /* Unique ID of initiated contact */
#define ABS_MT_PRESSURE     0x3a    /* Pressure on contact area */
#define ABS_MT_DISTANCE     0x3b    /* Contact hover distance */

[ABS_MT_TOUCH_MAJOR] = "ABS_MT_TOUCH_MAJOR",            // 0x30 /* Major axis of touching
[ABS_MT_TOUCH_MINOR] = "ABS_MT_TOUCH_MINOR",        // 0x31 /* Minor axis (omit if circula
[ABS_MT_WIDTH_MAJOR] = "ABS_MT_WIDTH_MAJOR",        // 0x32 /* Major axis of approaching e
[ABS_MT_WIDTH_MINOR] = "ABS_MT_WIDTH_MINOR",        // 0x33 /* Minor axis (omit if circular
[ABS_MT_ORIENTATION] = "ABS_MT_ORIENTATION",        // 0x34 /* Ellipse orientation */
[ABS_MT_POSITION_X]  = "ABS_MT_POSITION_X" ,        // 0x35 /* Center X ellipse position */
[ABS_MT_POSITION_Y]  = "ABS_MT_POSITION_Y" ,        // 0x36 /* Center Y ellipse position */
[ABS_MT_TOOL_TYPE]   = "ABS_MT_TOOL_TYPE"  ,        // 0x37 /* Type of touching device */
[ABS_MT_BLOB_ID]     = "ABS_MT_BLOB_ID"    ,    // 0x38 /* Group a set of packets as a blob
[ABS_MT_TRACKING_ID] = "ABS_MT_TRACKING_ID",        // 0x39 /* Unique ID of initiated conta
[ABS_MT_PRESSURE]    = "ABS_MT_PRESSURE"   ,        // 0x3a /* Pressure on contact area */
[ABS_MT_DISTANCE]    = "ABS_MT_DISTANCE"   ,        // 0x3b /* Cont

};
char *misc[MSC_MAX + 1] = {
    [ 0 ... MSC_MAX] = NULL,
    [MSC_SERIAL] = "Serial",    [MSC_PULSELED] = "Pulseled",
    [MSC_GESTURE] = "Gesture",    [MSC_RAW] = "RawData",
    [MSC_SCAN] = "ScanCode",
};
char *leds[LED_MAX + 1] = {
    [0 ... LED_MAX] = NULL,
    [LED_NUML] = "NumLock",         [LED_CAPSL] = "CapsLock",
    [LED_SCROLLL] = "ScrollLock",    [LED_COMPOSE] = "Compose",
    [LED_KANA] = "Kana",        [LED_SLEEP] = "Sleep",
    [LED_SUSPEND] = "Suspend",    [LED_MUTE] = "Mute",
    [LED_MISC] = "Misc",
};
char *repeats[REP_MAX + 1] = {
    [0 ... REP_MAX] = NULL,
    [REP_DELAY] = "Delay",        [REP_PERIOD] = "Period"
};
```

关闭

```
char *sounds[SND_MAX + 1] = {
    [0 ... SND_MAX] = NULL,
    [SND_CLICK] = "Click",        [SND_BELL] = "Bell",
    [SND_TONE] = "Tone"
};
char **names[EV_MAX + 1] = {
    [0 ... EV_MAX] = NULL,
    [EV_SYN] = events,            [EV_KEY] = keys,
    [EV_REL] = relatives,            [EV_ABS] = absolutes,
    [EV_MSC] = misc,           [EV_LED] = leds,
    [EV_SND] = sounds,            [EV_REP] = repeats,
};
#define BITS_PER_LONG (sizeof(long) * 8)
#define NBITS(x) ((((x)-1)/BITS_PER_LONG)+1)
#define OFF(x)  ((x)%BITS_PER_LONG)
#define BIT(x)  (1UL<<OFF(x))
#define LONG(x) ((x)/BITS_PER_LONG)
#define test_bit(bit, array)    ((array[LONG(bit)] >> OFF(bit)) & 1)
int main (int argc, char **argv)
{
    int fd, rd, i, j, k;
    struct input_event ev[64];
    int version;
    unsigned short id[4];
    unsigned long bit[EV_MAX][NBITS(KEY_MAX)];
    char name[256] = "Unknown";
    int abs[5];
    if (argc < 2) {
        printf("Usage: evtest /dev/input/eventX\n");
        printf("Where X = input device number\n");
        return 1;
    }
    if ((fd = open(argv[argc - 1], O_RDONLY)) < 0) {
        perror("evtest");
        return 1;
    }
    if (ioctl(fd, EVIOCGVERSION, &version)) {
        perror("evtest: can't get version\n");
        return 1;
    }
    printf("Input driver version is %d.%d.%d\n",
        version >> 16, (version >> 8) & 0xff, version & 0xff);
    ioctl(fd, EVIOCGID, id);
    printf("Input device ID: bus 0x%x vendor 0x%x product 0x%x version 0x%x\n",
        id[ID_BUS], id[ID_VENDOR], id[ID_PRODUCT], id[ID_VERSION]);
    ioctl(fd, EVIOCGNAME(sizeof(name)), name);
    printf("Input device name: %s\n", name);
    memset(bit, 0, sizeof(bit));
    ioctl(fd, EVIOCGBIT(0, EV_MAX), bit[0]);
    printf("Supported events:\n");
    for (i = 0; i < EV_MAX; i++)
        if (test_bit(i, bit[0])) {
            printf("  Event type %d (%s)\n", i, events[i] ? events[i] : "?");
            if (!i) continue;
            ioctl(fd, EVIOCGBIT(i, KEY_MAX), bit[i]);
```

关闭

```
            for (j = 0; j < KEY_MAX; j++)
                if (test_bit(j, bit[i])) {
                    printf("    Event code %d (%s)\n", j, names[i] ? (names[i][j] ? names[i]
[j] : "?") : "?");
                    if (i == EV_ABS) {
                        ioctl(fd, EVIOCGABS(j), abs);
                        for (k = 0; k < 5; k++)
                            if ((k < 3) || abs[k])
                                printf("      %s %6d\n", absval[k], abs[k]);
                    }
                }
        }

    printf("Testing ... (interrupt to exit)\n");
    while (1) {
        rd = read(fd, ev, sizeof(struct input_event) * 64);
        if (rd < (int) sizeof(struct input_event)) {
            printf("yyy\n");
            perror("\nevtest: error reading\n");
            return 1;
        }
        for (i = 0; i < rd / sizeof(struct input_event); i++)
            if (ev[i].type == EV_SYN) {
                printf("Event: time %ld.%06ld, -------------- %s ------------\n",
                    ev[i].time.tv_sec, ev[i].time.tv_usec, ev[i].code ? "Config Sync" : "R
            } else if (ev[i].type == EV_MSC && (ev[i].code == MSC_RAW || ev[i].code == MSC_
                printf("Event: time %ld.%06ld, type %d (%s), code %d (%s), value %02x\n",
                    ev[i].time.tv_sec, ev[i].time.tv_usec, ev[i].type,
                    events[ev[i].type] ? events[ev[i].type] : "?",
                    ev[i].code,
                    names[ev[i].type] ? (names[ev[i].type][ev[i].code] ? names[ev[i].type]
[ev[i].code] : "?") : "?",
                    ev[i].value);
            } else {
                printf("Event: time %ld.%06ld, type %d (%s), code %
                    ev[i].time.tv_sec, ev[i].time.tv_usec, ev[i].type,
                    events[ev[i].type] ? events[ev[i].type] : "?",
                    ev[i].code,
                    names[ev[i].type] ? (names[ev[i].type][ev[i].code] ? names[ev[i].type]
[ev[i].code] : "?") : "?",
                    ev[i].value);
            }
    }
}
```

关闭

上一篇 根据内核Oops 定位代码
下一篇 Android 下收发input事件的工具 getevent 和 sendevent

相关文章推荐

- getevent & setevent
- Presto的服务治理与架构在京东的实践与应用--王…
- 常用adb shell命令：getevent和sendevent
- 深入掌握Kubernetes应用实践--王渊命
- Android getevent用法
- Python基础知识汇总
- Binder---- Android 的IPC 通信机制
- Android核心技术详解
- Android sendevent/getevent 用法
- Retrofit 从入门封装到源码解析
- Android 下收发input事件的工具 getevent 和 sen
- 自然语言处理工具Word2Vec
- android测试之getevent/sendevent
- 通过getevent获
- Multi-touch (MT) Protocol 小结
- Android 中input event的分析

关闭

关闭