

android input调试工具

Last edited by **caoquanli** 1 month ago

Android input调试工具

一、介绍

- 在开发Android input时，我们可以使用Android提供的调试工具来进行调试。主要有三种，分别是 getevent,sendevent和input。
 - getevent 用于获取当前系统input设备的一些参数和实时事件的数据，比如获取按键上报信息、获取触摸屏上报信息等。
 - sendevent用于模拟input事件，写入设备节点。
 - input也是用来模拟input事件的，但是使用上与sendevent不同，用法更加简单。

二、getevent使用

- 首先通过getevent -h帮助信息，我们可以去了解一下getevent有哪些可用的操作，如下显示：

```
mt2712:/ # getevent -h
Usage: getevent [-t] [-n] [-s switchmask] [-S] [-v [mask]] [-d] [-p] [-i] [-l] [-q] [-c cou
-t: show time stamps
-n: don't print newlines
-s: print switch states for given bits
-S: print all switch states
-v: verbosity mask (errs=1, dev=2, name=4, info=8, vers=16, pos. events=32, props=64)
-d: show HID descriptor, if available
-p: show possible events (errs, dev, name, pos. events)
-i: show all device info and possible events
-l: label event types and names in plain text
-q: quiet (clear verbosity mask)
-c: print given number of events then exit
-r: print rate events are received
```

- getevent: 显示当前有那些输入设备，数量与 /dev/input 目录下相同

```
130|mt2712:/ # getevent
add device 1: /dev/input/event0
name:      "kpd"
could not get driver version for /dev/input/mice, Not a typewriter
add device 2: /dev/input/event3
name:      "rtq5115_pwrkey"
add device 3: /dev/input/event2
name:      "Atmel_mXT540E_Key"
add device 4: /dev/input/event1
name:      "Atmel_mXT540E_TouchPanel"
```

- t 显示时间戳

```
130|mt2712:/ # getevent -t
add device 1: /dev/input/event0
name:      "kpd"
could not get driver version for /dev/input/mice, Not a typewriter
add device 2: /dev/input/event3
name:      "rtq5115_pwrkey"
add device 3: /dev/input/event2
name:      "Atmel_mXT540E_Key"
add device 4: /dev/input/event1
name:      "Atmel_mXT540E_TouchPanel"
[ 3615.252209] /dev/input/event2: 0001 0073 000000001
[ 3615.252209] /dev/input/event2: 0000 0000 000000000
[ 3615.329776] /dev/input/event2: 0001 0073 000000000
[ 3615.329776] /dev/input/event2: 0000 0000 000000000
```

- n: 不换行打印

```
130|mt2712:/ # getevent -n
add device 1: /dev/input/event0
  name:      "kpd"
could not get driver version for /dev/input/mice, Not a typewriter
add device 2: /dev/input/event3
  name:      "rtq5115_pwrkey"
add device 3: /dev/input/event2
  name:      "Atmel_mXT540E_Key"
add device 4: /dev/input/event1
  name:      "Atmel_mXT540E_TouchPanel"
/dev/input/event2: 0001 0071 00000001/dev/input/event2: 0000 0000 00000000/dev/input/event2
```

- s: 显示指定位的开关状态（暂时不懂，仅限翻译）

```
1|mt2712:/ # getevent -s1
could not get driver version for /dev/input/mice, Not a typewriter
add device 1: /dev/input/event3
  name:      "rtq5115_pwrkey"
add device 2: /dev/input/event2
  name:      "Atmel_mXT540E_Key"
add device 3: /dev/input/event1
  name:      "Atmel_mXT540E_TouchPanel"
add device 4: /dev/input/event0
  name:      "kpd"
0000
0000
0000
0000
```

- S: 显示所有位的开关状态（暂时不懂，仅限翻译）

```
could not get driver version for /dev/input/mice, Not a typewriter
add device 1: /dev/input/event3
  name:      "rtq5115_pwrkey"
add device 2: /dev/input/event2
  name:      "Atmel_mXT540E_Key"
add device 3: /dev/input/event1
  name:      "Atmel_mXT540E_TouchPanel"
add device 4: /dev/input/event0
  name:      "kpd"
6400
6400
6400
6400
```

- v: 根据 mask 值显示相关信息，执行后会一直显示上报数据(**errs=1, dev=2, name=4, info=8, vers=16, pos. events=32, props=64**),默认显示 dev| name| info| vers = 30;

```
130|mt2712:/ # getevent -v
add device 1: /dev/input/event3
  bus:      0000
  vendor    0000
  product   0000
  version   0000
  name:      "rtq5115_pwrkey"
  location: ""
  id:        ""
  version:   1.0.1
add device 2: /dev/input/event2
  bus:      0018
  vendor    0000
  product   0000
  version   0000
  name:      "Atmel_mXT540E_Key"
  location: ""
  id:        ""
  version:   1.0.1
add device 3: /dev/input/event1
  bus:      0018
  vendor    0000
  product   0000
```

```
version    0000
name:      "Atmel_mXT540E_TouchPanel"
location:  ""
id:        ""
version:   1.0.1
add device 4: /dev/input/event0
bus:       0019
vendor     0000
product    0000
version    0000
name:      "kpd"
location:  ""
id:        ""
version:   1.0.1
```

- d: 如果设备可用，显示设备隐藏的描述信息

```
mt2712:/ # getevent -d
could not get driver version for /dev/input/mice, Not a typewriter
add device 1: /dev/input/event3
  name:      "rtq5115_pwrkey"
add device 2: /dev/input/event2
  name:      "Atmel_mXT540E_Key"
add device 3: /dev/input/event1
  name:      "Atmel_mXT540E_TouchPanel"
add device 4: /dev/input/event0
  name:      "kpd"
```

- p: 显示设备支持的事件类型和编码方式

```
130|mt2712:/ # getevent -p
could not get driver version for /dev/input/mice, Not a typewriter
add device 1: /dev/input/event3
  name:      "rtq5115_pwrkey"
  events:
    KEY (0001): 0074
  input props:
    <none>
add device 2: /dev/input/event2
  name:      "Atmel_mXT540E_Key"
  events:
    KEY (0001): 0066  0071  0072  0073  0074
  input props:
    <none>
add device 3: /dev/input/event1
  name:      "Atmel_mXT540E_TouchPanel"
  events:
    ABS (0003): 0030   : value 0, min 0, max 255, fuzz 0, flat 0, resolution 0
                  0035   : value 0, min 0, max 1279, fuzz 0, flat 0, resolution 0
                  0036   : value 0, min 0, max 719, fuzz 0, flat 0, resolution 0
                  003a   : value 0, min 0, max 255, fuzz 0, flat 0, resolution 0
  input props:
    <none>
add device 4: /dev/input/event0
  name:      "kpd"
  events:
    KEY (0001): 0001  0002  0003  0004  0005  0006  0007  0008
                  0009  000a  000b  000c  000d  000e  000f  0010
                  0011  0012  0013  0014  0015  0016  0017  0018
                  0019  001a  001b  001c  001d  001e  001f  0020
                  0021  0022  0023  0024  0025  0026  0027  0028
                  0029  002a  002b  002c  002d  002e  002f  0030
                  0031  0032  0033  0034  0035  0036  0037  0038
                  0039  003a  003b  003c  003d  003e  003f  0040
                  0041  0042  0043  0044  0045  0046  0047  0048
                  0049  004a  004b  004c  004d  004e  004f  0050
                  0051  0052  0053  0054  0055  0056  0057  0058
                  0059  005a  005b  005c  005d  005e  005f  0060
    MSC (0004): 0004
  input props:
    <none>
```

- i: 显示设备的所有信息和支持的事件，比 -p 显示更多信息

```
130|mt2712:/ # getevent -i
could not get driver version for /dev/input/mice, Not a typewriter
add device 1: /dev/input/event3
  bus:      0000
  vendor    0000
  product   0000
  version   0000
  name:      "rtq5115_pwrkey"
  location: ""
  id:        ""
  version:   1.0.1
  events:
    KEY (0001): 0074
  input props:
    <none>
add device 2: /dev/input/event2
  bus:      0018
  vendor    0000
  product   0000
  version   0000
  name:      "Atmel_mXT540E_Key"
  location: ""
  id:        ""
  version:   1.0.1
  events:
    KEY (0001): 0066  0071  0072  0073  0074
  input props:
    <none>
add device 3: /dev/input/event1
  bus:      0018
  vendor    0000
  product   0000
  version   0000
  name:      "Atmel_mXT540E_TouchPanel"
  location: ""
  id:        ""
  version:   1.0.1
  events:
    ABS (0003): 0030   : value 0, min 0, max 255, fuzz 0, flat 0, resolution 0
                  0035   : value 0, min 0, max 1279, fuzz 0, flat 0, resolution 0
                  0036   : value 0, min 0, max 719, fuzz 0, flat 0, resolution 0
                  003a   : value 0, min 0, max 255, fuzz 0, flat 0, resolution 0
  input props:
    <none>
add device 4: /dev/input/event0
  bus:      0019
  vendor    0000
  product   0000
  version   0000
  name:      "kpd"
  location: ""
  id:        ""
  version:   1.0.1
  events:
    KEY (0001): 0001  0002  0003  0004  0005  0006  0007  0008
                  0009  000a  000b  000c  000d  000e  000f  0010
                  0011  0012  0013  0014  0015  0016  0017  0018
                  0019  001a  001b  001c  001d  001e  001f  0020
                  0021  0022  0023  0024  0025  0026  0027  0028
                  0029  002a  002b  002c  002d  002e  002f  0030
                  0031  0032  0033  0034  0035  0036  0037  0038
                  0039  003a  003b  003c  003d  003e  003f  0040
                  0041  0042  0043  0044  0045  0046  0047  0048
                  0049  004a  004b  004c  004d  004e  004f  0050
                  0051  0052  0053  0054  0055  0056  0057  0058
                  0059  005a  005b  005c  005d  005e  005f  0060
    MSC (0004): 0004
  input props:
    <none>
```

- l: 以文本形式输出事件类型和名称，比直接getevent更清楚直观， 建议调试的时候使用

```
130|mt2712:/ # getevent -l
could not get driver version for /dev/input/mice, Not a typewriter
add device 1: /dev/input/event3
  name:      "rtq5115_pwrkey"
add device 2: /dev/input/event2
  name:      "Atmel_mXT540E_Key"
add device 3: /dev/input/event1
  name:      "Atmel_mXT540E_TouchPanel"
add device 4: /dev/input/event0
  name:      "kpd"
/dev/input/event2: EV_KEY      KEY_VOLUMEUP      DOWN
/dev/input/event2: EV_SYN      SYN_REPORT         00000000
/dev/input/event2: EV_KEY      KEY_VOLUMEUP      UP
/dev/input/event2: EV_SYN      SYN_REPORT         00000000
```

- -q 清空mask信息（暂时不懂，仅限翻译）

- -c: 打印固定数量的事件并退出

```
130|mt2712:/ # getevent -c 10
could not get driver version for /dev/input/mice, Not a typewriter
add device 1: /dev/input/event3
  name:      "rtq5115_pwrkey"
add device 2: /dev/input/event2
  name:      "Atmel_mXT540E_Key"
add device 3: /dev/input/event1
  name:      "Atmel_mXT540E_TouchPanel"
add device 4: /dev/input/event0
  name:      "kpd"
/dev/input/event2: 0001 0073 00000001
/dev/input/event2: 0000 0000 00000000
/dev/input/event2: 0001 0073 00000000
/dev/input/event2: 0000 0000 00000000
/dev/input/event2: 0001 0073 00000001
/dev/input/event2: 0000 0000 00000000
/dev/input/event2: 0001 0073 00000000
/dev/input/event2: 0000 0000 00000000
/dev/input/event2: 0001 0073 00000001
/dev/input/event2: 0000 0000 00000000
```

- -r: 显示事件上报速率（暂时不懂，仅限翻译）

```
130|mt2712:/ # getevent -r
could not get driver version for /dev/input/mice, Not a typewriter
add device 1: /dev/input/event3
  name:      "rtq5115_pwrkey"
add device 2: /dev/input/event2
  name:      "Atmel_mXT540E_Key"
add device 3: /dev/input/event1
  name:      "Atmel_mXT540E_TouchPanel"
add device 4: /dev/input/event0
  name:      "kpd"
/dev/input/event2: 0001 0073 00000001
/dev/input/event2: 0000 0000 00000000
/dev/input/event2: 0001 0073 00000000
/dev/input/event2: 0000 0000 00000000 rate 9

/dev/input/event2: 0001 0073 00000001
/dev/input/event2: 0000 0000 00000000 rate 0
/dev/input/event2: 0001 0073 00000000
/dev/input/event2: 0000 0000 00000000 rate 12
/dev/input/event2: 0001 0073 00000001
/dev/input/event2: 0000 0000 00000000 rate 10
/dev/input/event2: 0001 0073 00000000
/dev/input/event2: 0000 0000 00000000 rate 17
/dev/input/event2: 0001 0073 00000001
/dev/input/event2: 0000 0000 00000000 rate 8
/dev/input/event2: 0001 0073 00000000
/dev/input/event2: 0000 0000 00000000 rate 34
```

三、sendevent使用

- sendevent是通过往设备节点写入数据，去模拟事件，但是个人认为比较麻烦，不太适合调试使用，input命令可能会更适合。通过sendevent --help，会发现，介绍很简单，就两句话：

```
OnePlus5T:/ $ sendevent --help
usage: sendevent DEVICE TYPE CODE VALUE

Sends a Linux input event.
```

可以看到sendevent需要4个参数即：device，type，code，value。这些值可以由input子系统定义，也可以从getevent里面获取。DEVICE代表设备，type代表事件的类型，相当于getevent中的events，code代表事件中具体的类型，value代表事件的值。我们可能用到的也就两种，key事件和touch事件。设备支持的事件类型，我们可以通过getevent -p去显示出来：

```
mt2712:/ # getevent -p
could not get driver version for /dev/input/mice, Not a typewriter
add device 1: /dev/input/event3
  name:      "rtq5115_pwrkey"
  events:
    KEY (0001): 0074
  input props:
    <none>
add device 2: /dev/input/event2
  name:      "Atmel_mXT540E_Key"
  events:
    KEY (0001): 0066  0071  0072  0073  0074
  input props:
    <none>
add device 3: /dev/input/event1
  name:      "Atmel_mXT540E_TouchPanel"
  events:
    ABS (0003): 0030   : value 0, min 0, max 255, fuzz 0, flat 0, resolution 0
                  0035   : value 0, min 0, max 1279, fuzz 0, flat 0, resolution 0
                  0036   : value 0, min 0, max 719, fuzz 0, flat 0, resolution 0
                  003a   : value 0, min 0, max 255, fuzz 0, flat 0, resolution 0
  input props:
    <none>
add device 4: /dev/input/event0
  name:      "kpd"
  events:
    KEY (0001): 0001  0002  0003  0004  0005  0006  0007  0008
                  0009  000a  000b  000c  000d  000e  000f  0010
                  0011  0012  0013  0014  0015  0016  0017  0018
                  0019  001a  001b  001c  001d  001e  001f  0020
                  0021  0022  0023  0024  0025  0026  0027  0028
                  0029  002a  002b  002c  002d  002e  002f  0030
                  0031  0032  0033  0034  0035  0036  0037  0038
                  0039  003a  003b  003c  003d  003e  003f  0040
                  0041  0042  0043  0044  0045  0046  0047  0048
                  0049  004a  004b  004c  004d  004e  004f  0050
                  0051  0052  0053  0054  0055  0056  0057  0058
                  0059  005a  005b  005c  005d  005e  005f  0060
    MSC (0004): 0004
  input props:
    <none>
mt2712:/ #
```

- 如上所示，我们能用到的是 **KEY (0001)** 和 **ABS (0003)** ,分别代表的是key事件和touch事件。它们后面显示的是支持哪些具体的事件，相当于sendevent的code.code具体值代表的含义可以在bionic/libc/kernel/uapi/linux/input-event-codes.h文件中查看，在此就不多说了。
- 首先我们来模拟一个key事件，第一件事情是去确认往哪个设备节点，不知道的话，我们可以按按下真正的实体key的时候，通过getevent去查看是哪个设备需要写入。第二个是需要确认写入的key的keycode,同样通过getevent去确认，但是需要注意一点 **getevent中code显示的是十六进制,而sendevent时需要用十进制** 。通过getevent，我在Leepi机器上知道power键的code对应的是0074，转换成十进制是116.写入设备是/dev/input/event2。接下来就可以模拟power key事件了。为了更直观的看出效果，我同时使用了getevent和getevent -l，可以看出来getevent只是单纯的一串数值显示，具体的含义还需要自己去查询，而getevent -l直接显示了power key，而且连down还是up都显示了出来。最后的1和0分别是代表的是按下和抬起。需要注意的是输入完事件后，要sync一下，事件才能生效，命令行为 **sendevent /dev/input/event2 0 0 0** ,如果想模拟的是短压的话，down和up都要输入，如果是长压的话，只需要down就行了，长压结束后再去输入up。从方便的角度

了来看，sendevent使用起来是比较繁琐的，远远比不上input,但是若是模拟key长压的话，input是无法与sendevent相比的，稍后我会在input使用方法中加以说明。

```
mt2712:/ # sendevent /dev/input/event2 1 116 1
mt2712:/ # sendevent /dev/input/event2 0 0 0
/dev/input/event2: EV_KEY      KEY_POWER      DOWN
/dev/input/event2: EV_SYN      SYN_REPORT     00000000
/dev/input/event2: 0001 0074 00000001
/dev/input/event2: 0000 0000 00000000

mt2712:/ # sendevent /dev/input/event2 1 116 0
mt2712:/ # sendevent /dev/input/event2 0 0 0
/dev/input/event2: 0001 0074 00000000
/dev/input/event2: 0000 0000 00000000
/dev/input/event2: EV_KEY      KEY_POWER      UP
/dev/input/event2: EV_SYN      SYN_REPORT     00000000
```

- 模拟完key事件之后，我们来模拟一下touch,touch模拟起来比较复杂，我也只是略懂，稍微提一下如何使用单指，对多指感兴趣的同学，请自行研究一下吧。touch首先你得确定你的x,y坐标，也就是 **点**，确认完之后，再去选择是down，dwon。因为这个我也不是很清楚，也是通过getevent去查看touch的事件然后去模拟的。如下所示，是一个完整的touch事件的模拟，包括了down和up。先来个down的，首先输入的是ABS_MT_TOUCH_MAJOR，然后再输入ABS_MT_PRESSURE，这两个命令行是固定的，我也不懂，接下来就是输入touch事件的坐标了，先是x,再是y,53代表的是这是x坐标，173表示x坐标值,54代表的y坐标，549表示y坐标值。然后接下来的两条命令是sync。至此touch down事件就发送出去了。发送touch up比较简单，只需要将之前的down的六句输入语句中关于x,y两句删除，就行了。

```
mt2712:/ # sendevent /dev/input/event1 3 48 15
mt2712:/ # sendevent /dev/input/event1 3 58 15
mt2712:/ # sendevent /dev/input/event1 3 53 173
mt2712:/ # sendevent /dev/input/event1 3 54 549
mt2712:/ # sendevent /dev/input/event1 0 2 0
mt2712:/ # sendevent /dev/input/event1 0 0 0
/dev/input/event1: EV_ABS      ABS_MT_TOUCH_MAJOR 0000000f
/dev/input/event1: EV_ABS      ABS_MT_PRESSURE    0000000f
/dev/input/event1: EV_ABS      ABS_MT_POSITION_X  000000ad
/dev/input/event1: EV_ABS      ABS_MT_POSITION_Y  00000225
/dev/input/event1: EV_SYN      SYN_MT_REPORT     00000000
/dev/input/event1: EV_SYN      SYN_REPORT        00000000
mt2712:/ # sendevent /dev/input/event1 3 48 0
mt2712:/ # sendevent /dev/input/event1 3 58 0
mt2712:/ # sendevent /dev/input/event1 0 2 0
mt2712:/ # sendevent /dev/input/event1 0 0 0
/dev/input/event1: EV_ABS      ABS_MT_TOUCH_MAJOR 00000000
/dev/input/event1: EV_ABS      ABS_MT_PRESSURE    00000000
/dev/input/event1: EV_SYN      SYN_MT_REPORT     00000000
/dev/input/event1: EV_SYN      SYN_REPORT        00000000
```

三、input使用

- input与sendevent也是输入数据的，但是与sendevent不同的是，input不是往设备节点里写数据，而是往InputDispatcher线程中写入数据。对于InputDispatcher而言是属于不被信任的事件，但是也能使用。根据adb来看，input支持的模拟设备还挺多的，但是我们一般用的也只有key和touch,对应来说就是keyevent,tap(touch 单击),swipe (touch滑动) draganddrop（拖放），所以我们就讲一下这两种事件。

```
Usage: input [<source>] <command> [<arg>...]

The sources are:
    dpad
    keyboard
    mouse
    touchpad
    gamepad
    touchnavigation
    joystick
    touchscreen
    stylus
    trackball

The commands and default sources are:
    text <string> (Default: touchscreen)
```

```
keyevent [--longpress] <key code number or name> ... (Default: keyboard)
tap <x> <y> (Default: touchscreen)
swipe <x1> <y1> <x2> <y2> [duration(ms)] (Default: touchscreen)
draganddrop <x1> <y1> <x2> <y2> [duration(ms)] (Default: touchscreen)
press (Default: trackball)
roll <dx> <dy> (Default: trackball)
```

- input 模拟key事件，需要说一下的是，input模拟的key事件不分down,up，它是一个完整的key事件派发，包括了key和up。模拟一个短压key事件是 input keyevent ,可以是keycode,也可以是key的名字，具体的key可以去 frameworks/base/core/java/android/view/KeyEvent.java查看，我们以back键为例，在代码中的定义是 `public static final int KEYCODE_BACK= 4;` ,因此我们既可以使用key code 4,也可以使用key name BACK(去除KEYCODE_前缀)。长压也可以模拟，加上--longpress即可，但是这种模拟的长压，只是将keyevent的flag将变为了FLAG_LONG_PRESS，从效果来看的话，还是sendevent模拟长压更合适。

```
mt2712:/ # input keyevent 4
mt2712:/ # input keyevent BACK
mt2712:/ # input keyevent --longpress 4
```

- input 模拟touch事件，input模拟touch事件也是不分down,up，也是一个完整的touch事件派发，也就是说input模拟事件都是不分down和up的，完整的派发一个事件。如下所示，模拟一个touch事件，使用 `input tap` 加上点的坐标即可，而模拟一个滑动事件则是稍微多了一点设置，可以设置滑动后松开的时间 `input swipe x1 y1 x2 y2 duration(ms)]` ,除了滑动之外，touch事件中还有一个可能平时用的比较多，那就是拖放，比如长按图标然后移动，这个与swipe的模拟差不多， `input draganddrop x1 y1 x2 y2 duration(ms)]` ,看来下源码，这两种的区别就在于模拟拖放的时候，加了一个sleep去模拟down的时候是长压，有兴趣的同学，可以看一下源码， `frameworks/base/cmds/input/src/com/android/commands/input/Input.java`

```
mt2712:/ # input tap 173 549
mt2712:/ # input swipe 173 549 73 549
mt2712:/ # input swipe 173 549 73 549 100
mt2712:/ # input draganddrop 173 549 73 549
mt2712:/ # input draganddrop 173 549 73 549 100
```