







<u>\*\* xda-developers</u> → <u>Android Development and Hacking</u> → <u>Android Software and Hacking General [Developers Only]</u> → TUTORIAL: Remote Control Touch Screen by TwinOne



FORUMS

Android Software and Hacking General [Developers Only]

**Android General** 

Android Q&A, Help & Troubleshooting

**Android Software Development** 

Miscellaneous Android Development

[More]

Remove All Ads from XDA

Do you want to activate the screen wirelessly, or play some music at your phone over SSH because you don't know where you left it?

THIS IS YOUR PLACE!!!

This tutorial will show you how to create a file, that simulates a keypress, swipe, button press.

You can even draw something, save a file of data, and then replay it so android draws the same something.

You can simulate somebody writing text with the default android keyboard, over SSH, etc.



LOGIN ▶ REGISTER ▶







Let's start by the beginning: I will add a zip package with my working directory at my phone, all tools are included there.

The dropbear ssh server is included because it's practical because it can be turned on with your android terminal. You can also access the other tools using another ssh server.

The media player control is a very simple script that takes an argument and then translates it to the android "input keyevent" command.

And the low level screen events is what i'm going to talk about most.

From here on, this will be a tutorial that will focus on screen events I will cover the following topics:

- 1) Understanding and getting screen events.
- 2) Simulating keypressess, creating and sending screen events.
- 3) Automating the tasks.

## UNDERSTANDING AND GETTING SCREEN EVENTS

Android provides two commandline tools for managing screen events: getevent and sendevent.

These commands are kinda cryptic:

We'll take a look at the "getevent" command. This command outputs all events of a given device (or all if no device is given)

#### TOP FORUM DISCUSSIONS

How to Root the Samsung Galaxy Note 8 (Exynos)

② September 25, 2017

How to Enable Fingerprint Selfie on the Xiaomi Redmi 3s

② September 25, 2017

"Can't Talk" Lets You Auto-Reply to WhatsApp, Slack and More

② September 20, 2017

Google Nexus 4 Receives a Port of Android 8.0 Oreo

② September 19, 2017

Transcriber Converts Voice Notes in WhatsApp to Text

② September 10, 2017

Install the H2OS Boot Animation from the OnePlus 5 using a Magisk Module

② September 6, 2017

OpenSans and Roboto Fonts for Unrooted Huawei and Honor Devices on EMUI 4.0+

② September 6, 2017

More Forum Links ▶



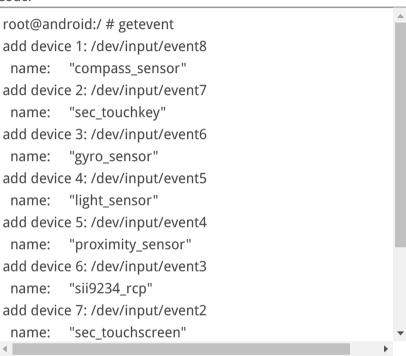






I just use getevent (hint, press Control+C to terminate the command and stop getting events):

#### Code:



You may notice the "add device X" number is not matching the /dev/input/eventX number. I will use the latter to refer to an event, as we will use this command a lot, believe me.

I will be only talking about devices 1, 2 and 7, i don't care about the rest.

/dev/input/event1 (gpio-keys): This are the hardware keys.

On my Samsung Galaxy SII, these keys are POWER, HOME, VOLUP, and VOLDOWN keys.

/dev/input/event2 (sec\_touchscreen): is the touch-screen (they're not very creative at android developing









/dev/input/event7 (sec\_touckey): The touch keys.

On my Galaxy, i have the MENU and BACK buttons as touchkeys. This touchkeys are not part of the touchscreen, thats why its another device.

Note though that the touch\_screen and the touck\_keys devices ARE connected, i'll later have you remember this line.

The numbers of the devices may change from device to device, as not all phones have touchkeys, or a touch screen.

Well, now you know how to list the devices.

The next step is to see how each device works:

#### **Device 1: HWKEYS**

I recommend following these steps on your device, and comment any changes at this thread, thank you.

Let's try out short-pressing just the POWER button of our phone (or unlock button)

When i refer to short-pressing, i mean pressing and immediately releasing the button. You'll soon understand why. Use getevent, press the key, and finish with Control + C.

#### Code:



WTF is this?! Yes, i said that too when i first saw this, (and you havent seen a swipe with 4 fingers yet, haha)









What can we observe? There are 3 fields, and all numbers.

The first field will always be 1 except in separator lines (see the second line)

The second field is the identifier for the POWER key.

The third field is a boolean value, it will be 1 when the key is pressed and 0 when it is released.

#### 0000 0000 00000000

This is a simple separator, it it means android just processed everything ultil here.

#### 0001 0074 00000000

This is a KEYUP line, as you can see, the identifier field is 1, the keycode field is 74, and the third boolean field is set to 0, what indicates the key is released.

#### 0000 0000 00000000

And another separator so android process it.

^C

This is the Control+C character, it is used to terminate the running program in linux terminal.

Actually quite simple, isn't it? Lets take a look at the touchkeys device (it's simpler as the screen)

## **Device 7: MENU and BACK**

Yes, android has set up a whole device driver only for 2 stupid keys.

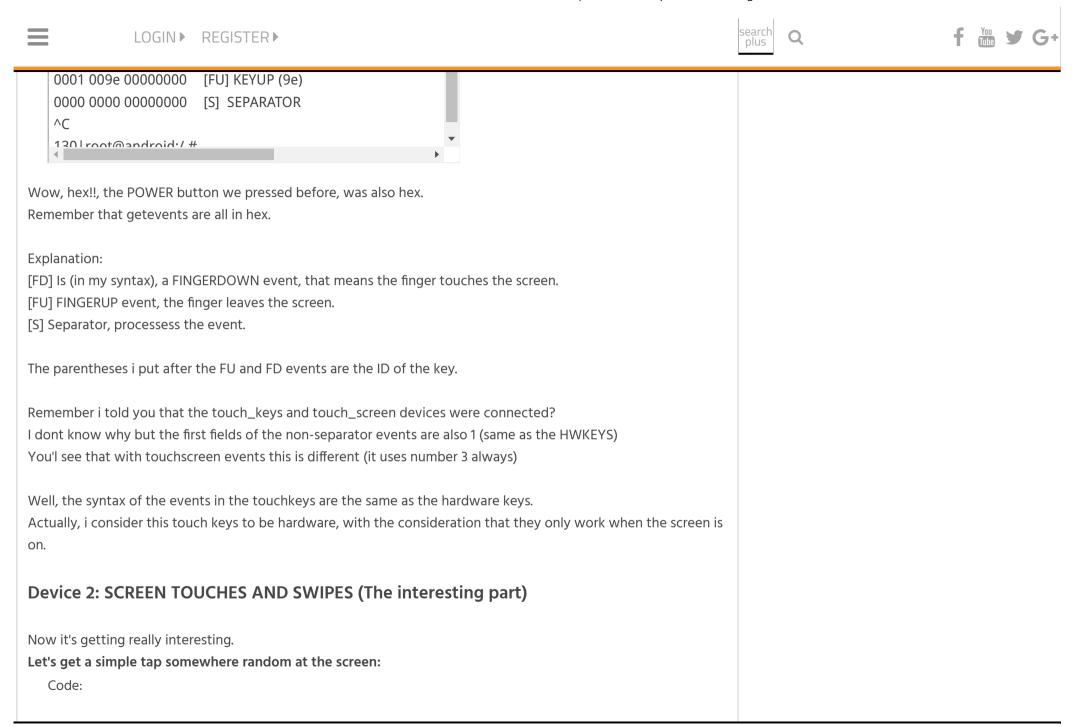
Let's see them in action. I will first press the MENU key, and then the BACK key.

Note that now i use "getevent /dev/input/event7", and not event1.

Oh, i will make some comments on the output so you can see it more clear.

Code:

0000 0000 00000000 [S] SEPARATOR 0001 008b 00000000 [FU] KEYUP (8b)











0003 0036 000001dd [Y] Y coordinate

// NOTE LINE

0000 0000 00000000 [S] Separator

0003 0039 ffffffff [FU] FingerUp (24f)

NOTE: At the NOTE LINE you MAY get 2 other events, 0030 and 003a. I have no idea what these two are for.

Besides, to simulate a click we don't need them, so if anyone know what these two events are, you're welcome to comment.

If you enable the Show touches & Pointer location, you see that it has some fields: PRS and SIZE, Those are for pressure and size, of course, but i don't know if they have something to do with this 0030 and 003a.

Tip for those of you that are debugging: if you increase pressure, your finger's contact area with the device grows, as your sking gets down.

I realised this thinking of somebody pushing his face to a glass surface and somebody watching at the other side.

NOTE 2: You are recommended to enable Settings -> Developer Options -> Show touches & Pointer location

Understanding touchscreen events is more complicated. All buttons from HWKEYS and TOUCHKEYS, were quite simple: they only have a key identifier, and a boolean state: Pressed (1) or not pressed (0).

However, when we use the touchscreen, every single tap has the following elements (events):









etc. I don't know yet when it is resetted, but for sure at reboot as this counter will probably reside in ram.

- [X] Number of pixels counted from left side.
- [Y] Number of pixels counted from top.
- [S] Separator / process trigger.
- [FU] Finger Up (The finger exits the screen completely)
- [S] Another separator.

Notice that the 0030 and 003a events are not in my list. I've made a tool to simulate keypressess, and those two events are NEVER included, but my script works always. So something tells me these two events are worthless.

### Introduction to swipe

With the pointer location enabled in settins, try swiping somewhere random.

As you may notice, you aren't perfect, so you can't draw a straight line.

Everytime your finger changes direction, a new point is added. this means that if you want to make a square, you would need 4 points, imagining all lines are straight.

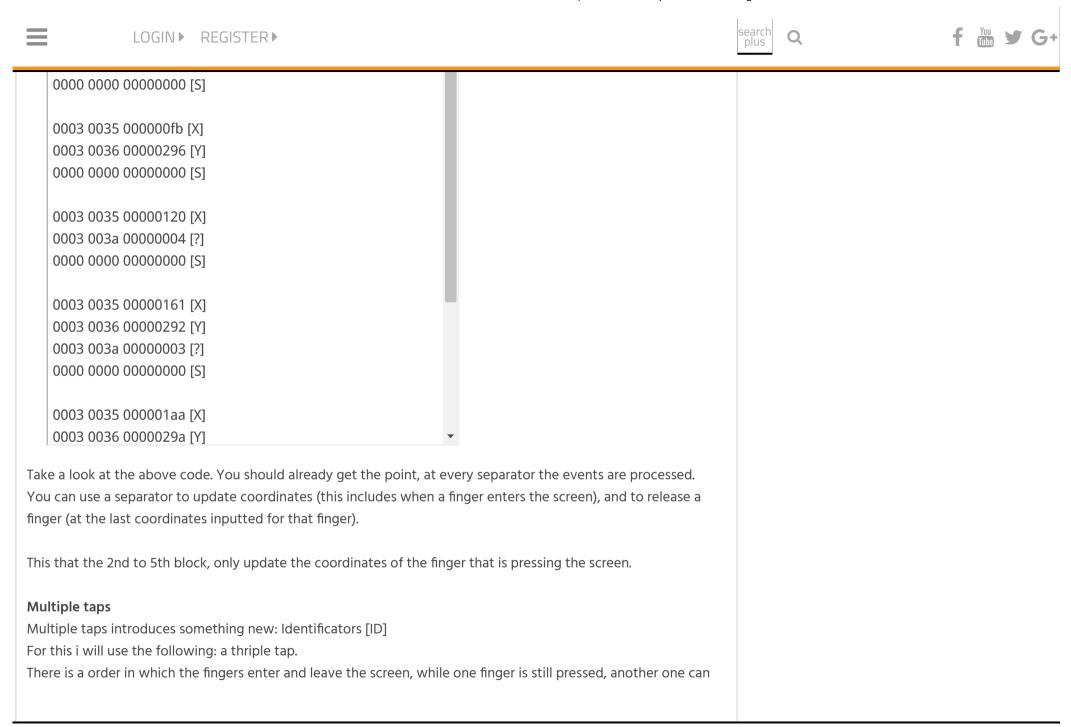
If you take a close look at the screen, the POINTER LOCATION shows a colored dot, everytime you set some coordinates (this means, by a single click, or by a direction change with a swipe, or with a double tap it would make two dots).

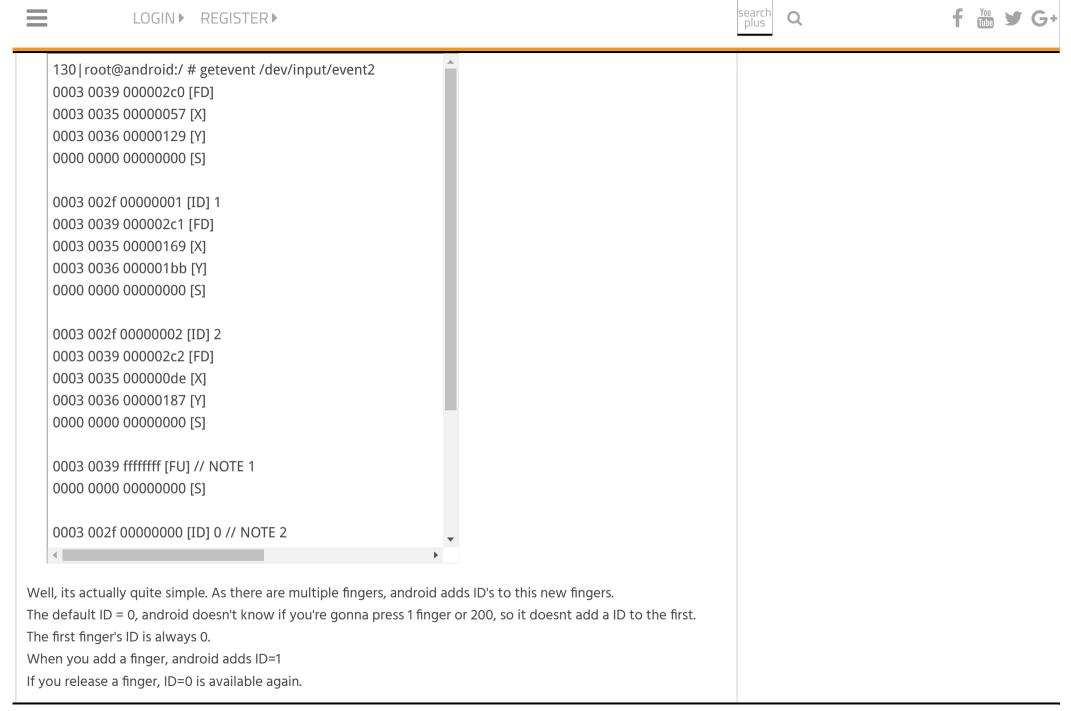
I will unlock my screen with my finger, and then analyze the output of getevent: Lets analize this simple swipe, and quicly move on to the last teorichal part: Multiple taps.

Code:

root@android:/ # getevent /dev/input/event2
0003 0039 000002ba [FD]

0003 0035 000000e6 [X]













just learned to combine swipe with multiple touches)

Thats why at the line i marked with NOTE 1, there is no ID, because the first finger i released, was finger ID=2 (the last that touched the screen)

**Important:** When you release a finger, you MUST set an ID, because android only keeps track of the CID (Current ID), but not of the previus.

So if you release a finger, and then set some coordinates, they will affect an ID that is no longer pressed, and it could either do NOTHING, or output some error message :P

## SIMULATING KEYPRESSESS, CREATING AND SENDING EVENTS

NOTE: When we use "getevent", we get some hex codes. When we use "sendevent" we have to use decimal values.

Why? I don't know, maybe bothering is the point. Anyway, it's like that, and we'll have to manage, right?

Sendevent takes the EXACT same input that getevent outputs, except that GETEVENT outputs [HEX]adecimal, and Sendevents wants [DEC]imal characters

To simulate the POWER button key, for one single click, it would take this steps:

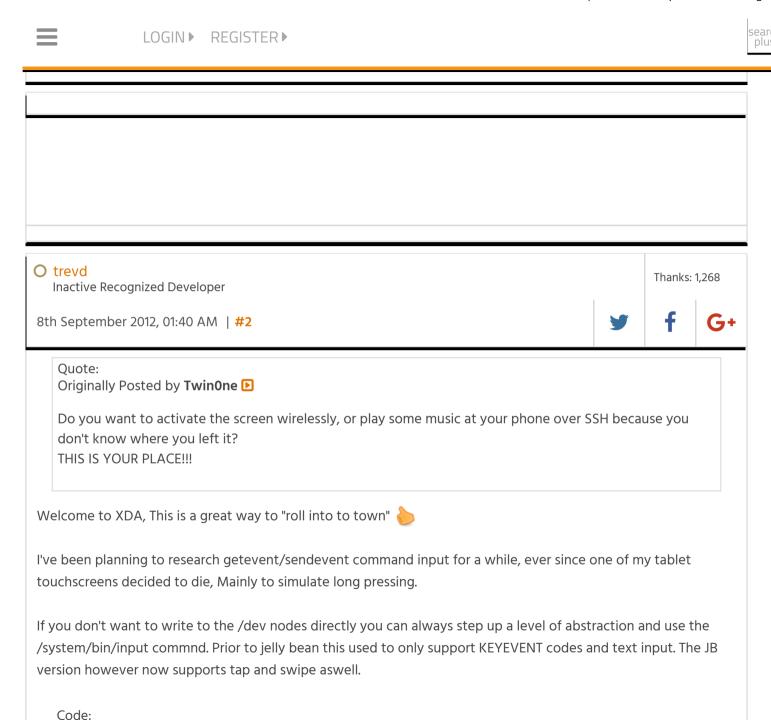
STEP 1

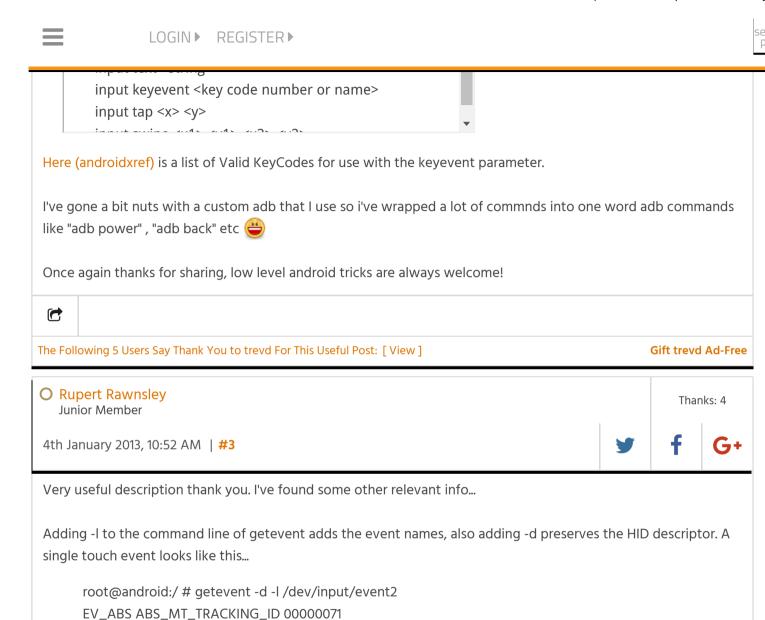
STEP 2

STEP 3

Edit: I'm currently making the SENDEVENT TUTORIAL part, in about 2 hours it will be done.

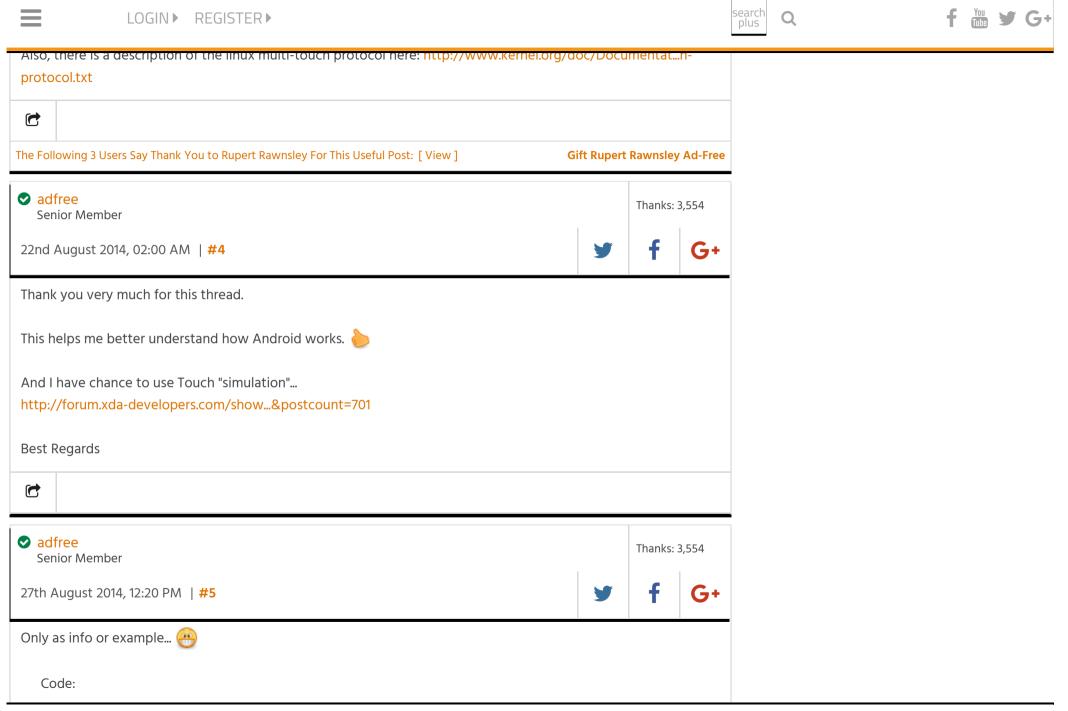
Vote for newsworthy if you think it was!!!





EV\_SYN SYN\_REPORT 00000000

EV\_ABS ABS\_MT\_POSITION\_X 0000020d EV\_ABS ABS\_MT\_POSITION\_Y 00000160













adb shell sendevent /dev/input/event3 3 48 28 adb shell sendevent /dev/input/event3 0 2 0 adb shell sendevent /dev/input/event3 0 0 0 adb shell sendevent /dev/input/event3 3 57 0 adb shell sendevent /dev/input/event3 3 53 4 adb shell sendevent /dev/input/event3 3 54 327 adb shell sendevent /dev/input/event3 3 48 32 adb shell sendevent /dev/input/event3 0 2 0 adb shell sendevent /dev/input/event3 0 0 0 adb shell sendevent /dev/input/event3 3 57 0 adb shell sendevent /dev/input/event3 3 53 5 adb shell sendevent /dev/input/event3 3 54 327 adb shell sendevent /dev/input/event3 3 48 48 adb shell sendevent /dev/input/event3 0 2 0 adb shell sendevent /dev/input/event3 0 0 0 adb shell sendevent /dev/input/event3 3 57 0 adb shell sendevent /dev/input/event3 3 53 6 adb shell sendevent /dev/input/event3 3 54 327 adb shell sendevent /dev/input/event3 3 48 49 adb shell sendevent /dev/input/event3 0 2 0

I have created simple Batch file... \*.bat

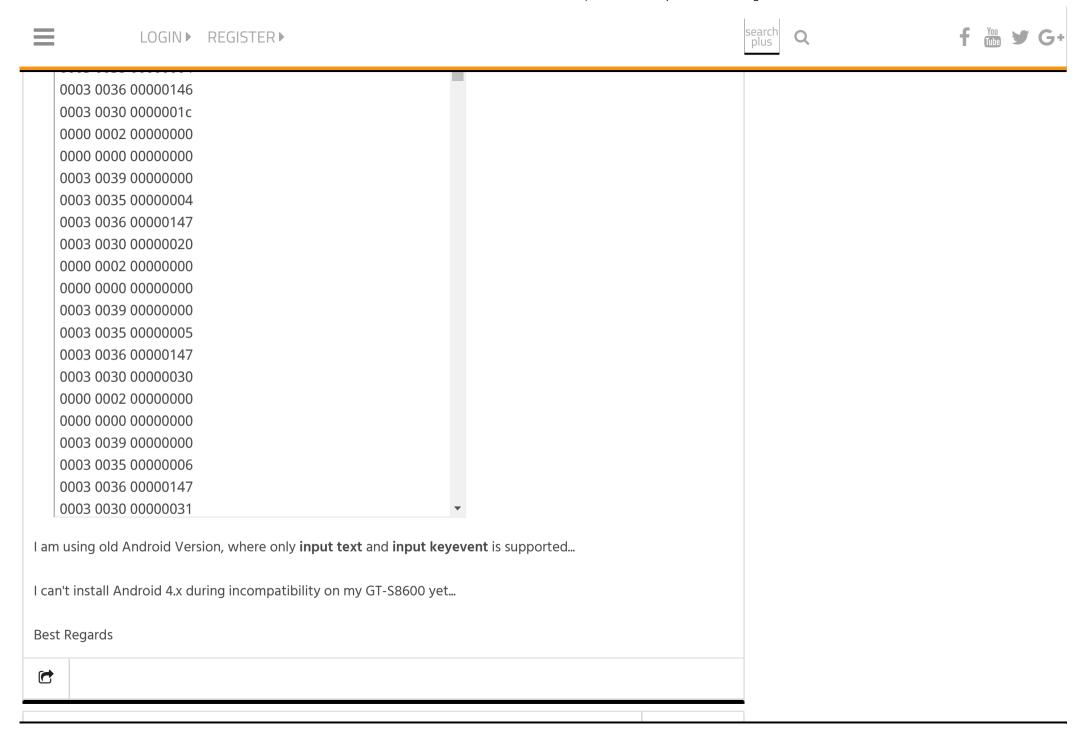
I can unlock Screenlock/Screensaver with this...

Tested on my S8600 with I9001 Firmware... because unsupported Hardware and Touch not working yet....

Also on 18150 tested successfully... because taken from 18150...

Here Log, which I have converted into Dec Values by my little brain... line by line...

Code:





LOGIN ▶ REGISTER ▶









19th April 2015, 12:18 AM | #6



Hi, what i understand so far.

for each device you have different "targets" to simulate e.g. the press/hold of a key. Actually i found it for my device and managed to simulate my hardware/soft keys with tasker-run shell and sendevent...

Question1: how do key-remapper handle that? Do they question the system e.g. what device the menu-key is? But more important to me is **Question2:** How can i simulate CTRL-C assuming that it is copy (and CTRL-V,CTRL-a) for a floating addition for my soft keyboard. I already managed to have a taker-scene, which floats as an overlay an by pressing my button called "test" the scene(overlay) hides and i can simulate a (sequence of keys or i can sendevent-stuff)...and there i would call Select-All,copy,paste.... Any idea (and no, the standard bar popping up when editing text with these functions does not pop up with swiftkey or in other situations i might want to use that). The keyboard "programmer keyboard" has this as buttons..so the function must be there somewhere...

In short: how to emulate CTRL-C?





12th January 2016, 07:01 PM | #7







Thanks: 1,088

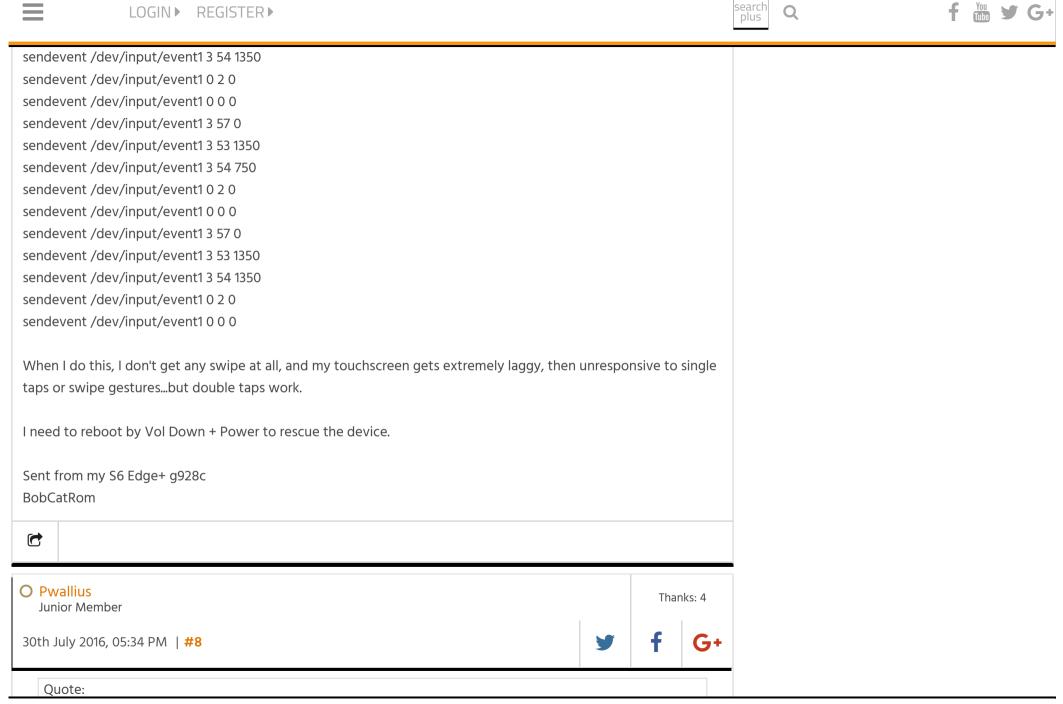
I need to emulate a swipe up and down the side of my screen, in a single touch. I'm trying to go from x=1350 y=1350 to x=1350 y=750 and back down to x=1350 y=1350

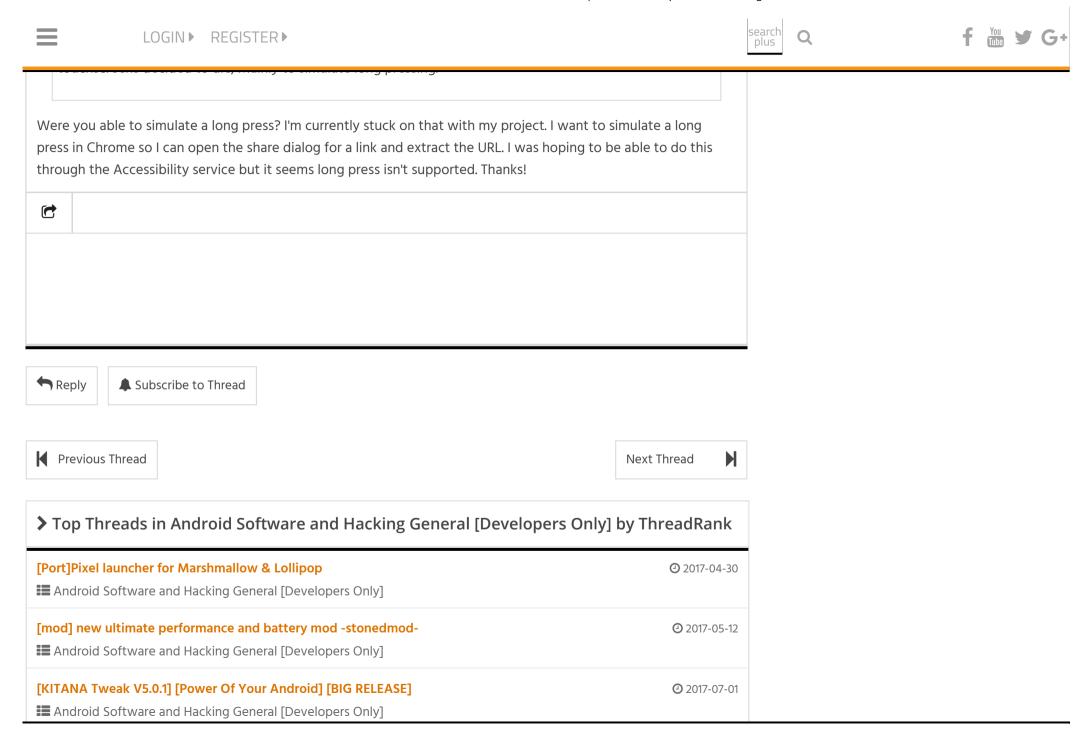
I can do this in two distinct touches with:

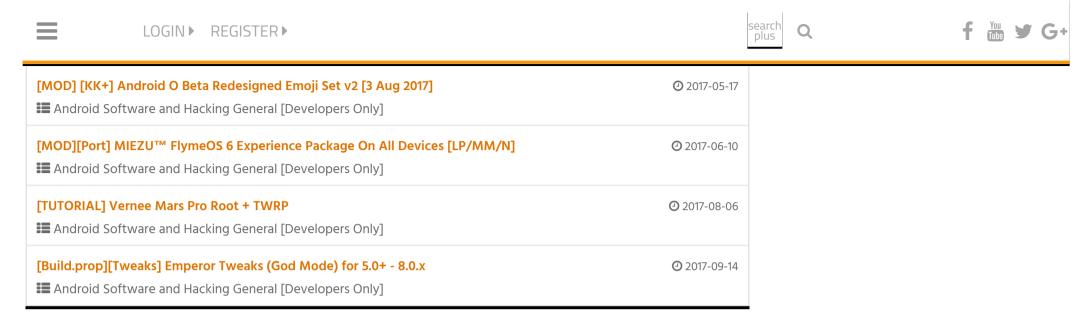
input swipe 1350 1350 1350 750

input swipe 1350 750 1350 1350

But I need it completed in one single touch, so I'm trying to use sendevent for the first time. I thought this would







★ xda-developers → Android Development and Hacking → Android Software and Hacking General [Developers Only] → TUTORIAL: Remote Control Touch Screen by TwinOne

# xda

XDA Developers was founded by developers, for developers. It is now a valuable resource for people who want to make the most of their mobile devices, from customizing the look and feel to adding new functionality.

More info









Hosted by **elaseweb** 

Are you a developer? | Terms of Service

Contact Rules Suggest Content Security Privacy Policy XDA App Root Any Device Remove ads on XDA

Copyright © xda-developers. Hosted by Leaseweb

VD 4 2045