

A better audio jack console cable for Google Nexus devices

Copyright © 2015 pabr@pabr.org
All rights reserved.

Many Google Nexus smartphones and tablets have a serial console which can be accessed through the headset socket. This is an invaluable tool for software development. Unfortunately there are no suitable off-the-shelf adapters on the market; therefore some tinkering is required.



<http://www.pabr.org/consolejack>

READ THE HYPERTEXT VERSION HERE:

<http://www.pabr.org/consolejack/consolejack.en.html>

Revision History		
1.0	2015-09-04	Initial release.

Table of Contents

1. Motivations	3
2. State of the art	3
3. Reference design (2015)	4
4. USB to audio jack console adapters with 1.8 V levels	4
4.1. Modifying a FTDI 1.8 V TTL-232RG cable	4
4.2. Adding a 1.8 V supply to a FT232R-based adapter	6
Bibliography	7

1. Motivations

In the process of porting Linux kernels to poorly documented embedded devices, one of the first milestones is seeing log messages on a serial console. Sometimes the serial port is exposed as such. Sometimes it is hidden in a USB OTG port and enabled by a magic resistor value on the ID pin. Sometimes the serial port is merely a bunch of unlabeled pads on the circuit board.

It was revealed in 2013 that many Google Nexus Android devices have a serial port multiplexed with audio signals in their 3.5 mm headphone socket. This is an interesting design decision because:

- Many mobile devices have an audio socket, even very small ones.
- The USB port remains available for other debugging tasks, allowing a productive workflow (load kernel with **fastboot**, watch console, repeat)
- USB ports may become extinct if inductive charging becomes popular.
- There are hints that other Android devices have similar serial consoles. I will maintain a list below:
 - some Lenovo Yoga tablets according to `osc_4_blade2_L_result/linux/kernel/drivers/misc/uart_audiojack_sw.c` in `yoga_tab_2_osc_android_to_lollipop_201505.rar` [<http://support.lenovo.com/fr/en/products/Tablets/Yoga-Series/Yoga-Tablet-2-830/downloads/DS102874>]
 - ASUS ME103K (MemoPad) [<http://www.thanassis.space/android.html#serialport>]

Note that alternate functionality on 3.5 mm audio sockets is not a new idea. Mobile phones used to output analog video that way before the advent of MHL and SlimPort.

2. State of the art

Early attempts to access the serial console on Nexus devices are documented in these blog posts, with lots of speculation about permissible voltage levels:

- *Building a Nexus 4 UART Debug Cable* [<https://www.optiv.com/blog/building-a-nexus-4-uart-debug-cable>]
- *Serial Console on Google Nexus 5* [<http://www.abclinuxu.cz/blog/Lorris/2013/12/serial-console-on-google-nexus-5>] (also Nexus 7 2012 and 2013)
- *Enabling the Nexus 7 UART debug port* [<http://delta.zauner.one/154-nexus-7-uart-unlocked/>]

WARNING. I do not recommend following the instructions in these posts. The proposed designs use 3.3 V USB-UART adapters and resistive voltage dividers, resulting in poor signal integrity, unnecessary power dissipation and mismatched logic levels. The point of the present article is to do better. Failing to obtain console output from a freshly compiled kernel on undocumented hardware is frustrating enough; I would rather not have to worry about faulty cables or fried UARTs when this happens.

We learn from these documents that:

- Injecting a certain voltage into the MIC pin causes the socket to switch from audio mode to serial mode.
- RX/TX logic levels are probably 1.8 V instead of the usual 3.3 V.
- On some devices the serial console must be enabled by appending parameters on the kernel command-line.
- On the Nexus 7 2013 it looks like the serial console is already enabled by **console=ttyHSL0,115200,n8**, but for some reason the bootloader skips that part of the command-line.

- The relevant source code is referenced in these commits:
 - *mako: add the earjack debugger trigger* [<https://android.googlesource.com/kernel/msm/+43a79436bed5499b78f140a73003975576987505>] (Nexus 4 "mako")
 - *misc: earjack_debugger: Add earjack debugger* [<https://android.googlesource.com/kernel/msm/+bd1b310d27d1b043e361904ca5d0aa5ddb8bc1a>] (Nexus 5 "hammerhead");
 - *Audio: headset: add debug board detection mechanism.* [<https://android.googlesource.com/kernel/msm.git/+52e2c2eecacbf9c6259c5c9d5b0acbf3585b3bf7>] (Nexus 7 2013 "flo").

3. Reference design (2015)

The official Google requirements for audio sockets on all Android devices ([WAHS]) confirm that the pinout order is (usually) CTIA/AHJ: **(tip) LEFT - RIGHT - GROUND - MIC (sleeve)**.

Notably, the MIC bias voltage can be anywhere between 1.8 V and 2.9 V. Therefore it is reasonable to expect that any threshold intended to trigger the serial functionality should be higher than 2.9 V, although Google may have enforced a MIC bias voltage in the bottom of that range on their Nexus devices.

In May 2015 Google published the schematic for (presumably) the adapter they use internally ([DEBUG-CABLE], announced here [<https://plus.google.com/111524780435806926688/posts/S4ajVewveSR>]), albeit with some disclaimers ("No Warranty expressed or implied"). Actually, they released the whole manufacturing information, which you could send to a fab shop instead of reading further.

- The adapter contains a 1.8 V voltage regulator which powers the VCCIO input of a FT232R USB-UART chip.
- The adapter contains another voltage regulator apparently intended to supply 3.36 V. Based on component values, tolerances and datasheets ([TPS717]) the actual output range is 3.30 - 3.49 V.

This voltage is injected into the MIC input through a 33 Ohm PTC thermistor (Ihold=36mA and Itrip=71mA according to datasheets; Ihold=29mA and Itrip=61mA according to the BOM).

Why did they not use the 3.3 V regulator that is built into the FT232R ? Maybe it cannot supply enough current to overcome the MIC bias on some devices, or maybe they decided that it is not accurate enough (the guaranteed output range is 3.0 - 3.6 V according to [FT232R]).

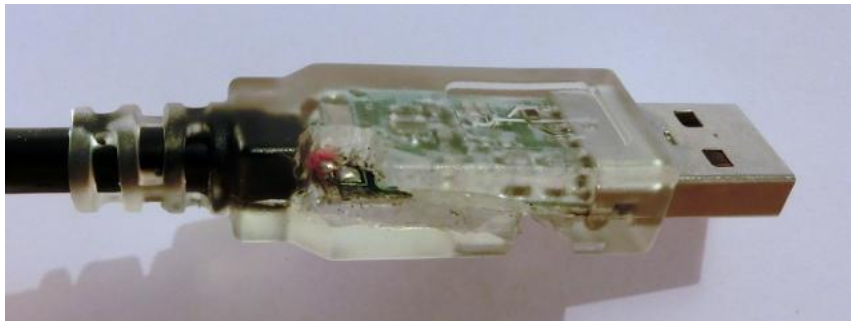
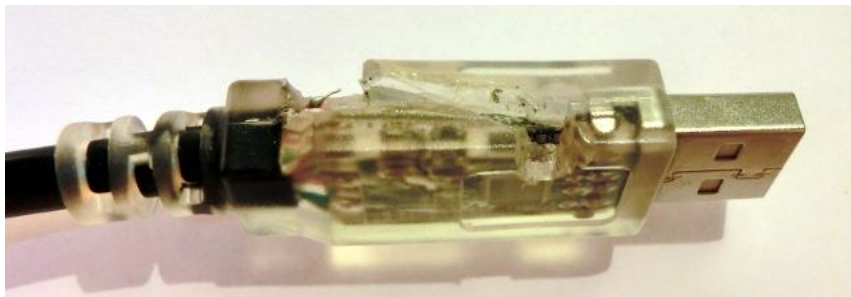
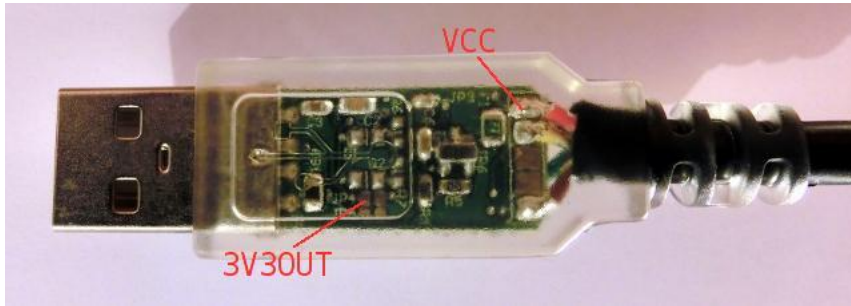
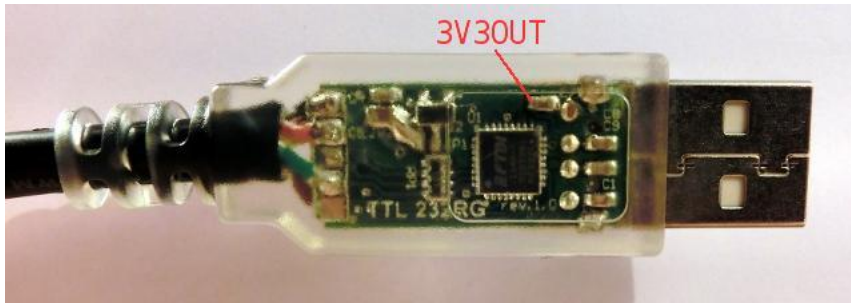
4. USB to audio jack console adapters with 1.8 V levels

At the time of writing there are apparently no off-the-shelf USB-UART adapters with 1.8 V logic levels and an additional 3.3 V output. Until someone decides to bring the Google design to market, I am testing the following homebrew solutions.

4.1. Modifying a FTDI 1.8 V TTL-232RG cable

The FTDI TTL-232RG-VREG1V8-WE is a USB-UART adapter cable with 1.8 V logic levels. Unfortunately the VCC output (red wire) is also at 1.8 V, and the 3.3 V line is buried in plastic moulding ([TTL-232RG-VREG1V8-WE]).

The catalogue showed an all-black product but the unit I received came with a transparent plug, so I was able to cut and melt my way to the relevant pads on the PCB.



I disconnected the red wire and rerouted it through a 33 ohm resistor to 3V3OUT at C7 (also reachable at JP4).



I checked that 3V3OUT actually outputs 3.30 V. As already mentioned, the datasheet only guarantees a range of 3.0 V (possibly too low to trigger serial mode) to 3.6 V (possibly dangerous).

Then I re-sealed the moulding with epoxy.

It is worth remembering that the good old 3.5 mm audio jack is not a very safe connector. If the sleeve gets shorted to ground, the FT232R will not be able to hold 3V3OUT at 3.3 V (maximum current is 50 mA according to [FT232R]). Since the FT232R uses 3V3OUT internally to power its USB interface, this might cause problems. For the same reason, any high voltage accidentally applied to the sleeve might feed into the host computer through the USB port. I would have used a PTC fuse, as Google did, if I had had one in my parts bin.

Table 1. Wiring for adapter based on TTL-232RG cable

TTL-232RG signal	TTL-232RG wire	Audio wire	Audio signal (CTIA/AHJ)	Jack contact	Device console signal
RX	Yellow	White	Left	Tip	TX
TX	Orange	Red	Right	Ring 1	RX
Ground	Black	Shield/Yellow	Ground	Ring 2	Ground
3V3OUT	Red (re-wired)	Yellow/Shield	Mic	Sleeve	Detect

Figure 2. Android accessing its own kernel console

What could possibly go wrong ? #turinglocks #singularity #skynet #RotM



4.2. Adding a 1.8 V supply to a FT232R-based adapter

This variant should be easier to build, but less compact.

Parts list:

- FT232R-based USB-UART adapter board whose VCCIO pin can be jumpered to VBUS (5 V) or 3V3OUT (3.3 V), or connected to an external supply between 1.8 V and 5 V.
- Generic 1.8 V regulator
- TRRS jack and cable, e.g. from an old headset.
- 33 ohm resistor.

NOT TESTED YET.

Bibliography

[WAHS] *Wired Audio Headset Specification (v1.1)*. <http://source.android.com/accessories/headset/specification.html>.

[DEBUGCABLE] *USB debug cable design documents*. <https://android.googlesource.com/device/google/debugcable>.

[FT232R] *FT232R USB UART IC Datasheet*. http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf.

[TPS717] *TPS717xx Low-Noise, High-Bandwidth PSRR, Low-Dropout, 150-mA Linear Regulator*. <http://www.ti.com/lit/gpn/TPS717>.

[TTL-232RG-VREG1V8-WE] *TTL-232RG TTL to USB Serial Converter Generic Cables*. http://www.ftdichip.com/Documents/DataSheets/Cables/DS_TTL-232RG_CABLES.pdf.