Mainline kernel on ARM Tegra20 devices that are left behind on 2.6 kernels

Dobrica Pavlinušić
https://blog.rot13.org/
@dpavlin

Is it possible to take obsolete Android device and port it to mainline kernel with Debian? (make RaspberryPi-like device)

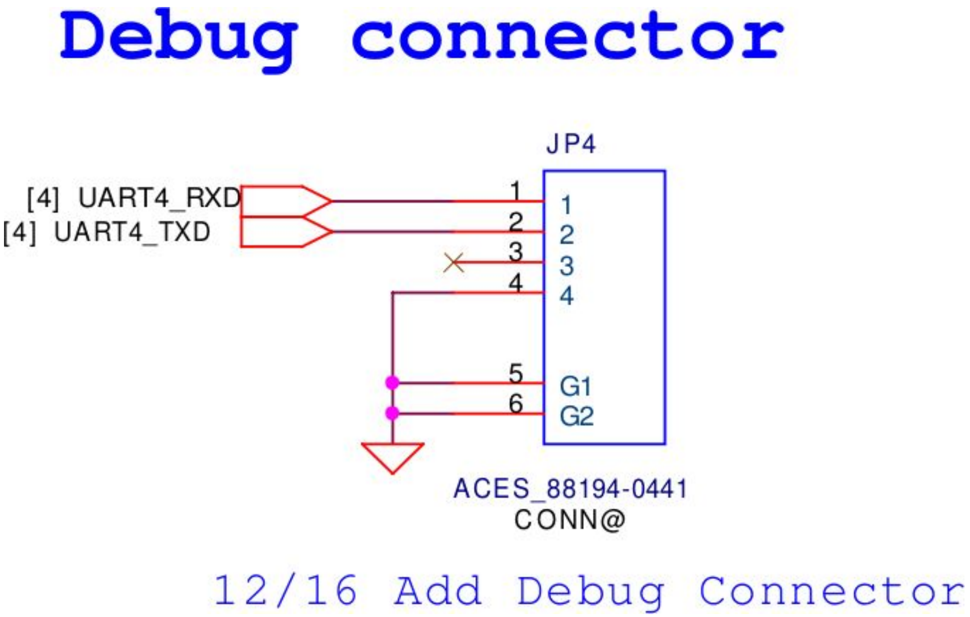
Thinkpad Tablet - Tegra20 PHJ00LA-7461P

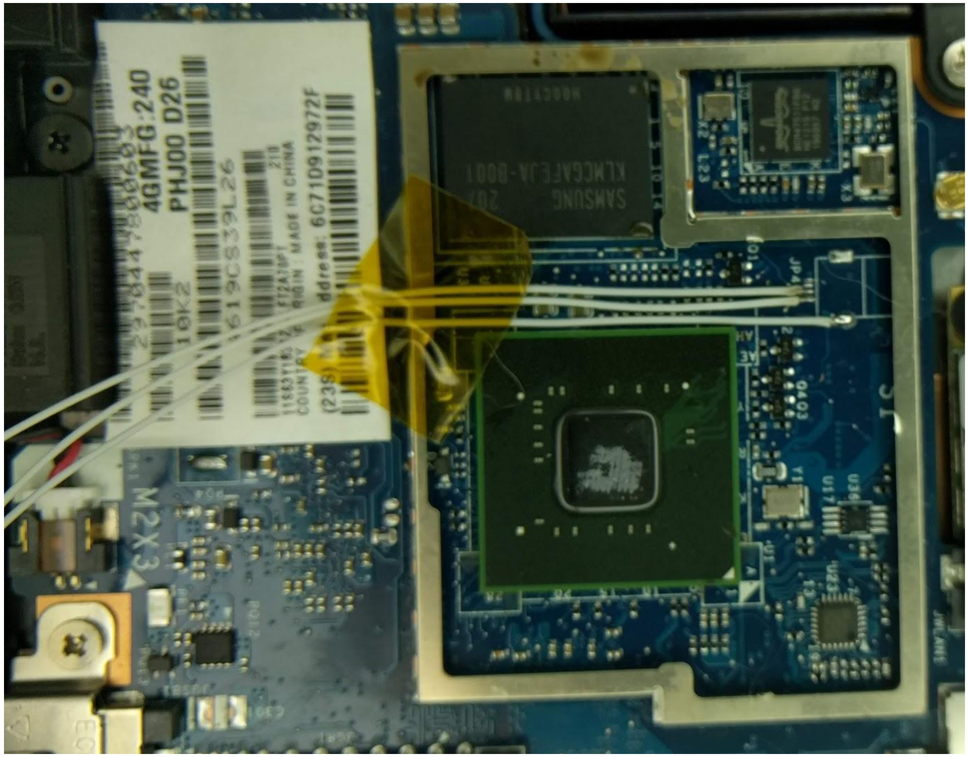
2011 model, Tegra T20 **2 cores @1Ghz, 1 GB RAM, 64 GB mmc flash, 1280x800 screen, wifi**, 3g/gps, camera, light sensor, accelerometer

- kernel 2.6.36 available from Lenovo, based on Tegra ventana board
  - 213 files changed, 39153 insertions(+), 1947 deletions(-)
- paz00 schematic exists from OEM (nice, but not as useful as you think)
  - compal_la-7461p_r0.3_schematics.pdf
- mainline has support for Tegra
  - Opensource drivers for NVIDIA Tegra20+ https://github.com/grate-driver
- cheaply available locally
  - somewhat broken with soldering marks and hotglue

serial port

- suggested first step to begin anything -- it's much easier to bootstrap u-boot or kernel if you have working serial port
- schematics shows UART on 4 pin connector
    - connector hidden under metal shield of cpu
- 1.8V direct connection to Tegra CPU
    - cheap "iphone" 1.8V usb serials available in China
    - wire-wrapping wire is small enough for job





Tegra APX mode

- early boot loader in CPU ROM (can be locked, it isn't in this case)
- enter APX by holding rotate key while pressing power key
- nvflash can be used to modify flash on device (binary blob, so we won't use it here)
- tegrarcm can create image that makes device bootable over usb

```
[Tue Oct  9 14:30:41 2018] usb 2-4: new high-speed USB device number 16 using xhci_hcd
[Tue Oct  9 14:30:42 2018] usb 2-4: New USB device found, idVendor=0955, idProduct=7820, bcdDevice= 1.04
[Tue Oct  9 14:30:42 2018] usb 2-4: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[Tue Oct  9 14:30:42 2018] usb 2-4: Product: APX
[Tue Oct  9 14:30:42 2018] usb 2-4: Manufacturer: NVIDIA Corp.
```

- this makes device unbrickable, so safe and easy to experiment with

So far, we have:

★ diff of 2.6 kernel on device
★ serial port
★ ability to run our code

u-boot

```
$ export CROSS_COMPILE="arm-none-eabi-" ARCH=arm
```

```
$ make ventana_defconfig
```

port display from 2.6 kernel to u-boot device tree, so u-boot can turn it on

push it using APX and it works!

```
$ tegrarcm --bct broken.bct readbct # read device config
```

```
$ tegrarcm --bct broken.bct --bootloader u-boot-tegra.bin --loadaddr 0x108000
```

https://github.com/dpavlin/u-boot/tree/phj00-thinkpad-tablet

```
tegra20-ventana.dts |   46 +++++++++++++++++++++++++++++++---------------
 1 file changed, 31 insertions(+), 15 deletions(-)
```

Add one of supported usb ethernets: ASIX ASIX88179 LAN75XX LAN78XX MCS7830 RTL8152 SMSC95XX

Wait! Port changes?! It's not hard!

```
# diff of 2.6 kernel
static struct tegra_dc_mode
ventana_panel_modes[] = {
    {
-       .pclk = 72072000,
+       .pclk = 71500000,//72072000,
        .h_ref_to_sync = 11,
        .v_ref_to_sync = 1,
-       .h_sync_width = 58,
-       .v_sync_width = 4,
-       .h_back_porch = 58,
-       .v_back_porch = 4,
-       .h_active = 1366,
-       .v_active = 768,
-       .h_front_porch = 58,
-       .v_front_porch = 4,
+       .h_sync_width = 32,
+       .v_sync_width = 7,
+       .h_back_porch = 72,
+       .v_back_porch = 22,
+       .h_active = 1280,
+       .v_active = 800,
+       .h_front_porch = 48,
+       .v_front_porch = 3,
    },
 };
```

```
# u-boot dts diff
display-timings {
    timing@0 {
-     /* Seaboard has 1366x768 */
-     clock-frequency = <70600000>;
-     hactive = <1366>;
-     vactive = <768>;
-     hback-porch = <58>;
-     hfront-porch = <58>;
-     hsync-len = <58>;
-     vback-porch = <4>;
-     vfront-porch = <4>;
-     vsync-len = <4>;
+ /* XXX tegra_dc_mode ventana_panel_modes */
+     clock-frequency = <72072000>;
+     hactive = <1280>;
+     vactive = <800>;
+     hback-porch = <72>;
+     hfront-porch = <48>;
+     hsync-len = <32>;
+     vback-porch = <22>;
+     vfront-porch = <3>;
+     vsync-len = <7>;
      hsync-active = <1>;
    };
};
```

mainline (grate-driver) linux kernel and device tree

- start with ventana device tree (same as u-boot)
- examine 2.6 kernel diff and port changes to device tree
- simple-panel dts configuration didn't work, port it into kernel driver
- define gpio keys (active low/high testing)
    - use triggerhappy and shell to adjust brightness :-)
- added temperature nct1008, compass ak8975 and non-working accelerometer kxtf9

https://github.com/dpavlin/linux/tree/thinkpad-tablet-phj00

```
arch/arm/boot/dts/tegra20-ventana.dts |  154 ++++++++++++++--
arch/arm/configs/phj00_defconfig      |  313 ++++++++++++++++++++++++++++++++++
drivers/gpu/drm/panel/panel-simple.c  |   25 ++
3 files changed, 471 insertions(+), 21 deletions(-)
```

filesystem using nfsroot - optimize for fun!

It would be possible to test everything with usb storage (corruptions are real)

nfs ensures consistent filesystem (power loss, kernel ops)

Why? Modify files locally on nfs server or on the device!

nfsroot allows us to use qemu and chroot in nfs export to do fast package installation or similar

It also helps if you are testing on multiple devices -- filesystem is always same!

dhcp works for u-boot but not for kernel's ip=dhcp with dnsmasq

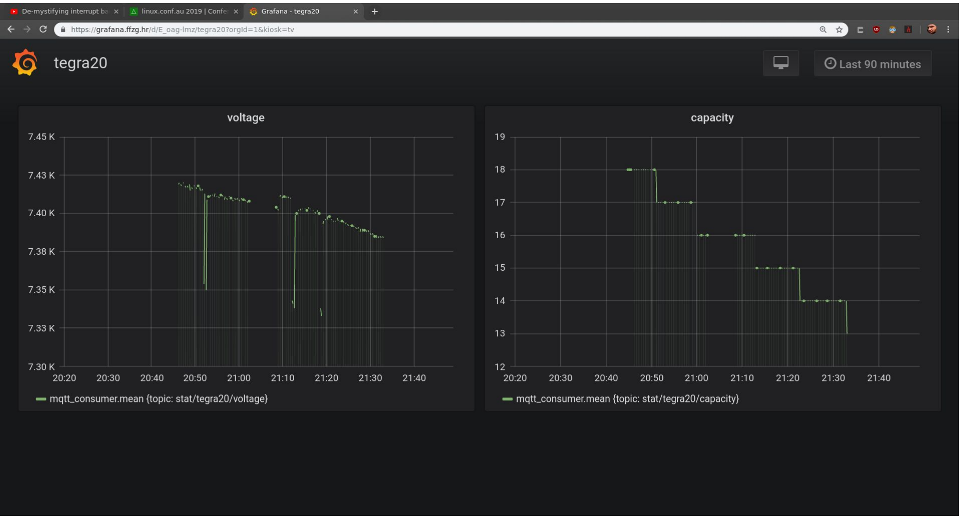
More info: https://saturn.ffzg.hr/rot13/index.cgi?u_boot

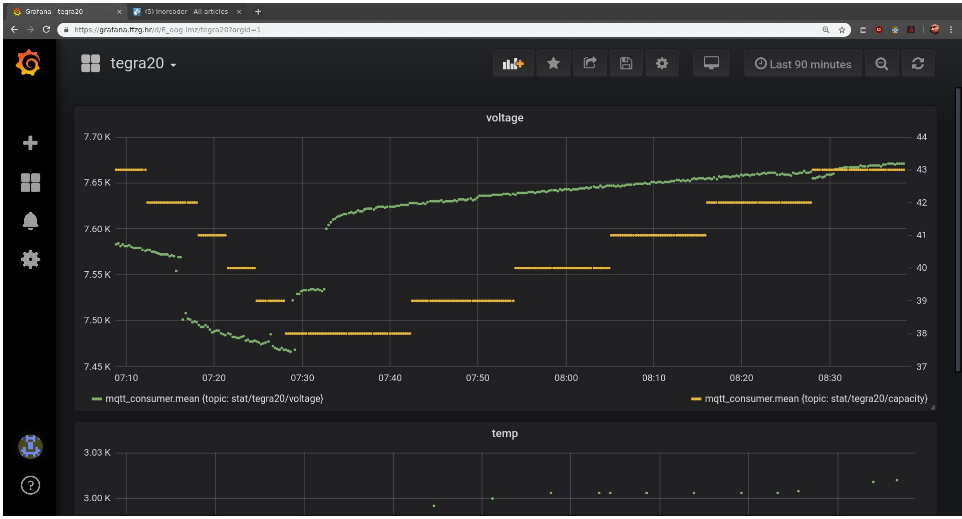
EC (8051 on i2c) and battery charging

Charging works when device is off, requires >1A, sensitive to drop below 5V - I used variable power supply set at 5.2V with 2A limit.

Based on battery state, it might be possible to keep battery charged using normal 500mA USB port, but only if the screen is not turned on (common on Android devices unfortunately)

Device that you can't keep powered on isn't very useful.




https://grafana.ffzg.hr/d/E_oag-lmz/tegra20?orgId=1
tegra20
Last 90 minutes
voltage
7.70 K
7.65 K
7.60 K
7.55 K
7.50 K
7.45 K
44
43
42
41
40
39
38
37
07:10
07:20
07:30
07:40
07:50
08:00
08:10
08:20
08:30
mqtt_consumer.mean {topic: stat/tegra20/voltage}
mqtt_consumer.mean {topic: stat/tegra20/capacity}
temp
3.03 K
3.00 K

Linux 2.6 drivers/power/EC_battery.c gives hints

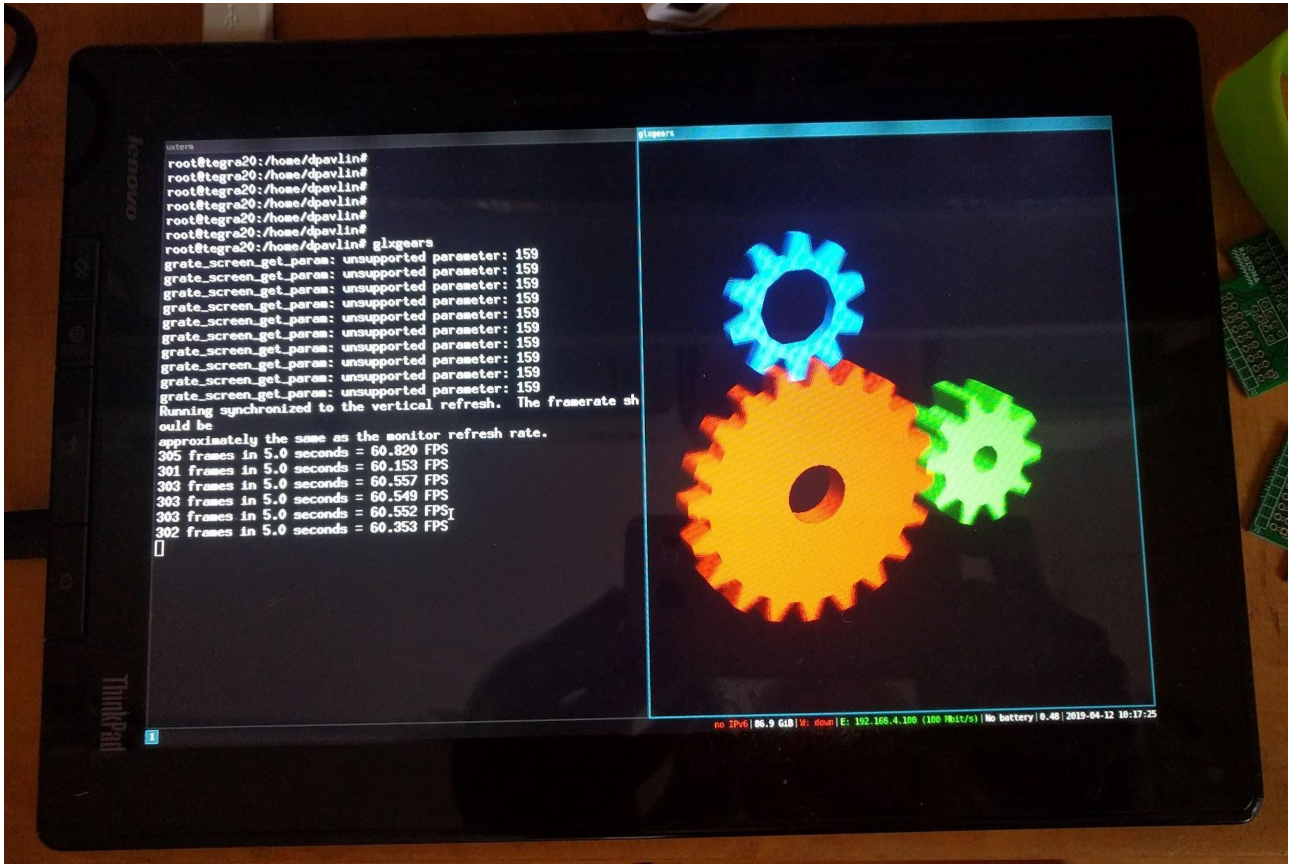
```
#define DOCK_ON     151 //GPIO_PS7

  //Dock in. report AC present
  if(gpio_get_value(DOCK_ON) == 1)
  {
    val->intval = 1;
    i2c_smbus_write_word_data(EC_Bat_device->client,0x5b,0x0001);
    return 0;
  }
```

How to tell EC that AC is plugged in start charging without writing kernel code?

```
# i2cset -y 5 0x58 0x5b 0x0001 w
```

grate driver - Tegra GPU/video acceleration


lenovo
uxterm
glxgears
1
no IPv6 | 86.9 GiB | W: down | E: 192.168.4.100 (100 Mbit/s) | No battery | 0.48 | 2019-04-12 10:17:25
ThinkPad

```
root@tegra20:/home/dpavlin#
root@tegra20:/home/dpavlin#
root@tegra20:/home/dpavlin#
root@tegra20:/home/dpavlin#
root@tegra20:/home/dpavlin#
root@tegra20:/home/dpavlin#
root@tegra20:/home/dpavlin# glxgears
grate_screen_get_param: unsupported parameter: 159
grate_screen_get_param: unsupported parameter: 159
grate_screen_get_param: unsupported parameter: 159
grate_screen_get_param: unsupported parameter: 159
grate_screen_get_param: unsupported parameter: 159
grate_screen_get_param: unsupported parameter: 159
grate_screen_get_param: unsupported parameter: 159
grate_screen_get_param: unsupported parameter: 159
grate_screen_get_param: unsupported parameter: 159
grate_screen_get_param: unsupported parameter: 159
Running synchronized to the vertical refresh.  The framerate sh
ould be
approximately the same as the monitor refresh rate.
305 frames in 5.0 seconds = 60.820 FPS
301 frames in 5.0 seconds = 60.153 FPS
303 frames in 5.0 seconds = 60.557 FPS
303 frames in 5.0 seconds = 60.549 FPS
303 frames in 5.0 seconds = 60.552 FPS
302 frames in 5.0 seconds = 60.353 FPS
```

https://github.com/grate-driver upstream has Ubuntu packages, rebuilt for Debian -- rebuild order important: libdrm, mesa, xorg-video-opentegra

So, which devices are supported currently?

i2c devices

0-001a	wm8903	✔	audio
0-001c	al3000a_ls	✘	light
1-003a	nvhdcp1	✘	?
1-0050	tegra_edid	✘	fake
2-0050	phj00_lcd	✘	fake
2-0058	EC_Battery	✘☚	charging
3-003c	mt9p111	✘	camera
3-003d	mt9d115	✘	camera
4-000c	akm8975	✔☚	compass
4-000f	kxtf9	✘	accel
4-0034	tps6586x	✔	power
4-004c	nct1008	✔☚	temp

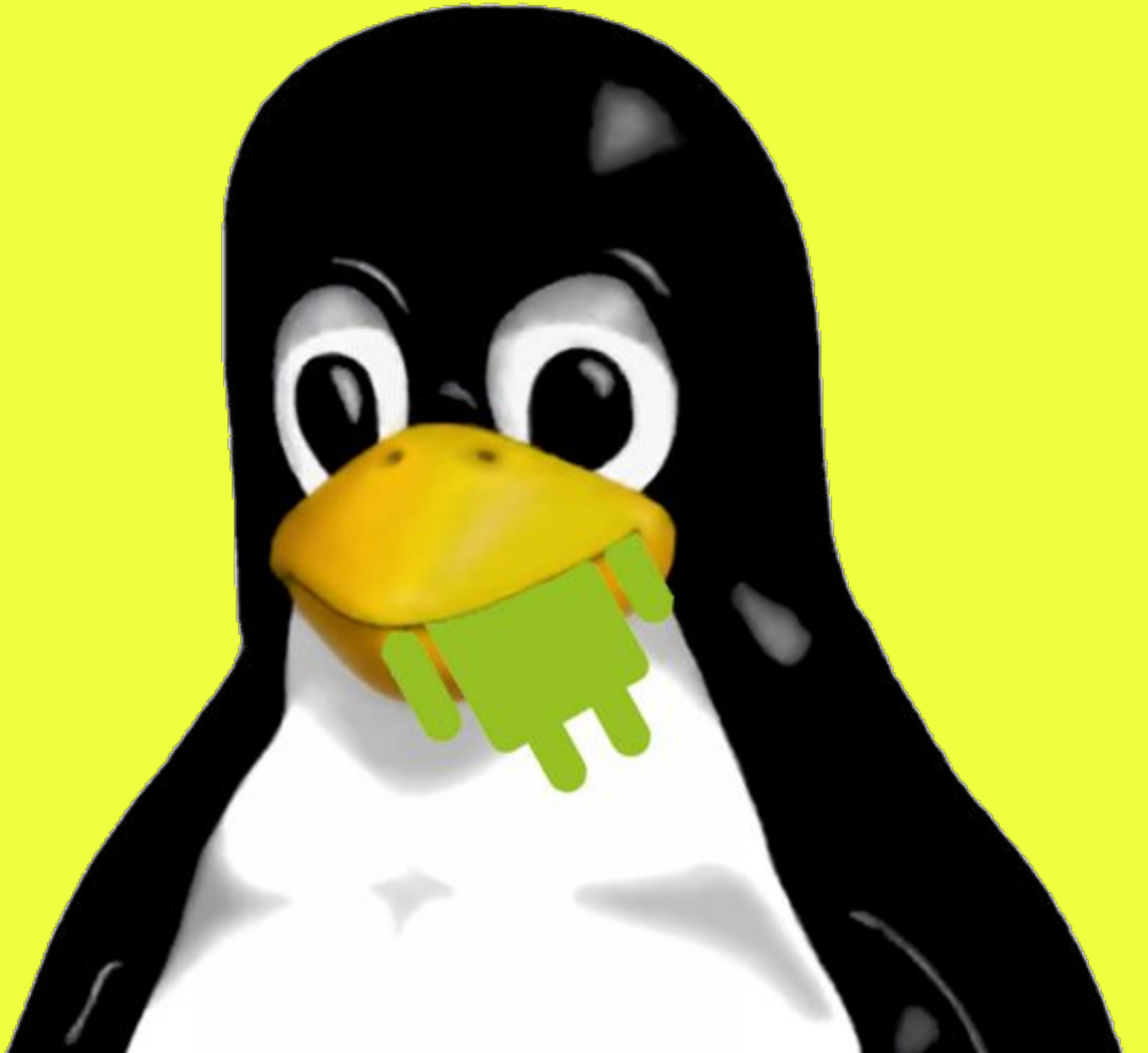
other devices

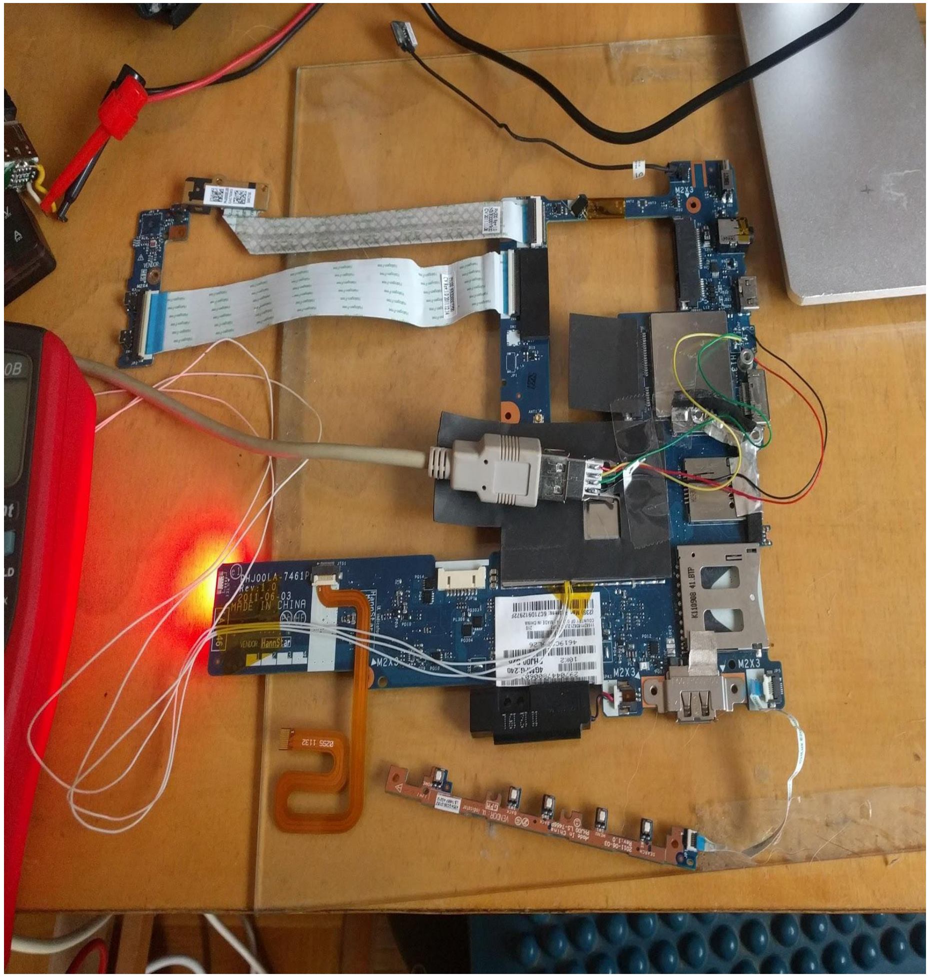
display	✔☚
hdmi	?
NTrig spi touch	✘
keys (gpio)	✔☚
vibrator (gpio)	✘
proximity (gpio)	✔☚
bcm4329 wifi (usb)	✔☚
modem (usb)	✔
mmc	✔
sdcard	?

Was it worth the effort?

YES!

apt update





Future work:

SPI enablement (tegra pins are configured, but kernel doesn't init SPI) -- surface3_spi might work for a touchscreen if I managed to persuade it to use device tree instead of ACPI

EC need more work (and proper kernel driver) - wifi disappears after reboot, charging requires i2cset

Cameras are unsupported (and v4l kernel bindings are changing right now, so I don't know how to implement them)

Real Linux distro on the device is still more useful to me than unsupported Android!

If you enjoy something like this, it was!

https://saturn.ffzg.hr/rot13/index.cgi?lenovo_thinkpad_tablet

Related work for other ARM devices

- Armbian - best choice of Linux distro for ARM boards https://www.armbian.com/
- Mainline Linux on Motorola Droid 4 [OMAP] https://archive.fosdem.org/2018/schedule/event/hwenablement_mainline_linux_on_motorola_droid_4/
- Maemo Leste - A Debian/Devuan based mobile hacker OS [OMAP] https://fosdem.org/2019/schedule/event/maemo_leste_mobile/
- postmarketOS [vendor kernels, Alpine] https://postmarketos.org/
    - Samsung Galaxy Tab 10.1 (another Tegra20 device) https://github.com/Decatf/linux

Hopefully this will motivate you to revive your old Android devices!