

Linux Installation auf Surface RT

Von Elektromüll zum Linux-Tablet?

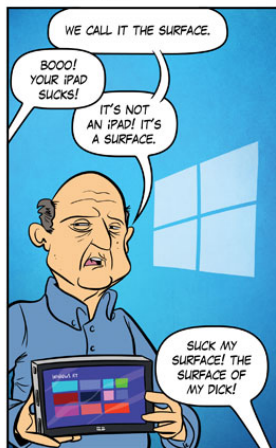
Kyeong Ro Lee

Leipziger Linux User Group

24.09.2024

Surface Tension?

"Surface Tension"

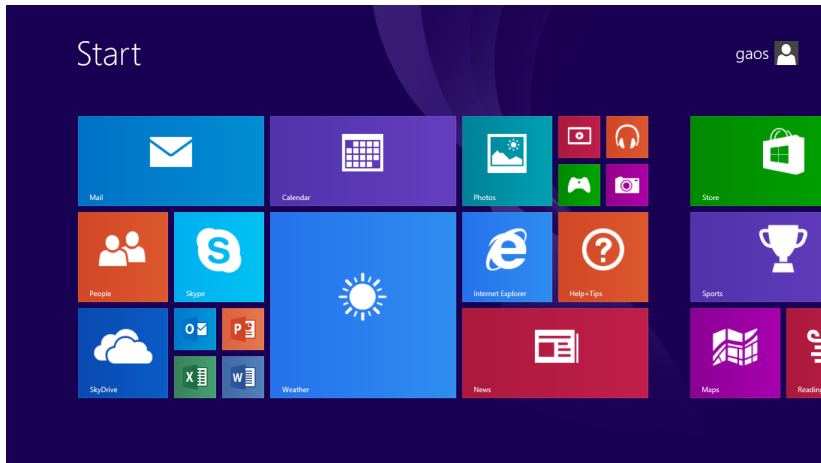


HIJINKS ENSUE by Joel Watson

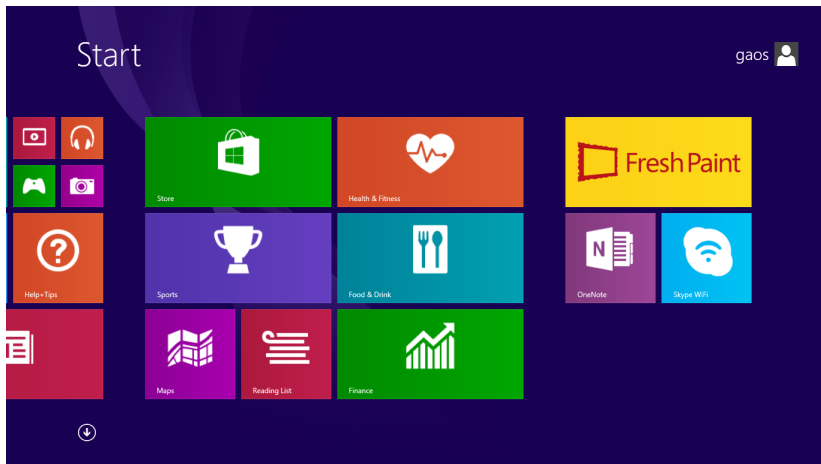
© 2012 Joel Watson www.hijinksensue.com

- Nvidia Tegra 3 SoC (4x ARM Cortex-A9)
- 2GB RAM
- 32/64GB eMMC
- 1x microSD
- 1x USB2.0 (Type-A)
- Gehäuse aus Magnesium

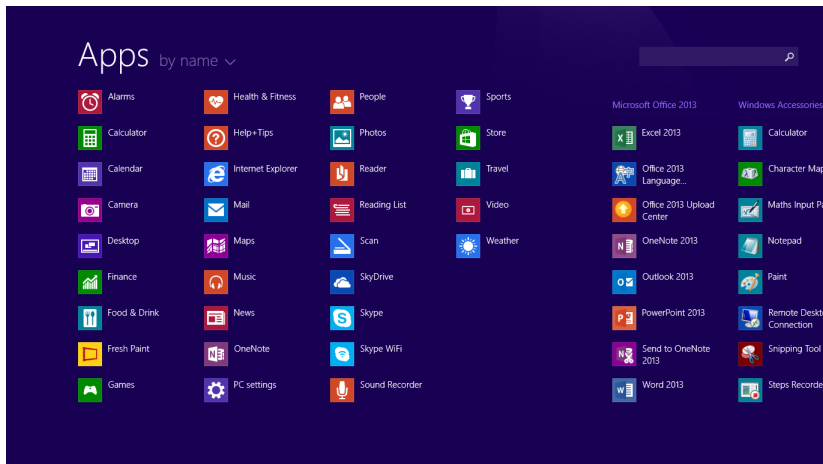
Probleme mit Windows RT



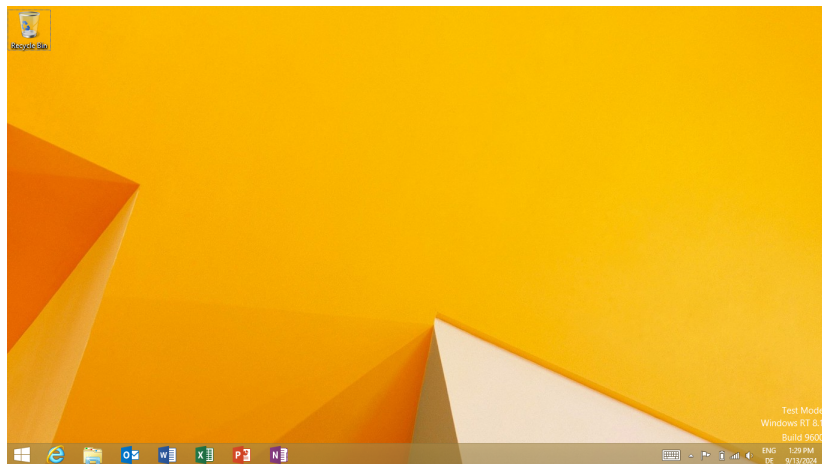
Probleme mit Windows RT



Probleme mit Windows RT

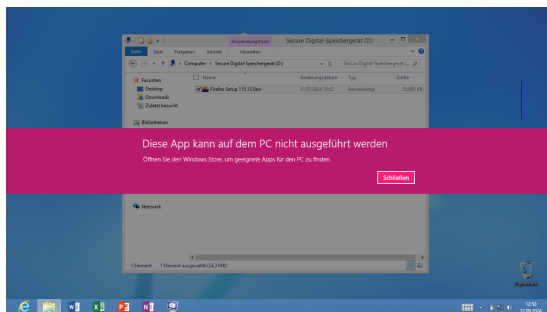


Probleme mit Windows RT



Probleme mit Windows RT

- ARMv7 ohne x86-Kompatibilität

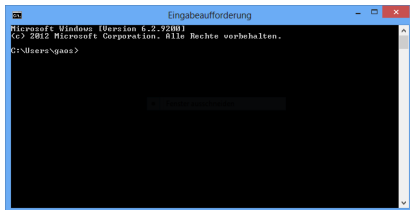


- Windows Store Apps ausschließlich (Ausnahme: MS Office) - siehe <https://openrt.gitbook.io/open-surfacert/common/windows-rt/jailbreak-exploits>
- ... aber Windows Store geschlossen!
- kein moderner Webbrowser verfügbar!

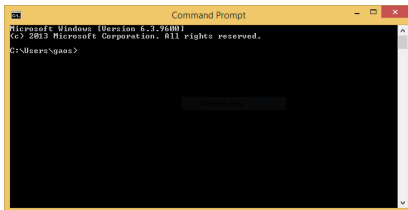
Step 0: “Upgrade” Windows!

- Windows 8 RT – Windows 8.1 RT
- ... aber Update-Server aus!
- Recovery Media archiviert

<https://windows-rt-devices.gitbook.io/windows/miscellaneous/bare-metal-recovery>



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\gaos>
```

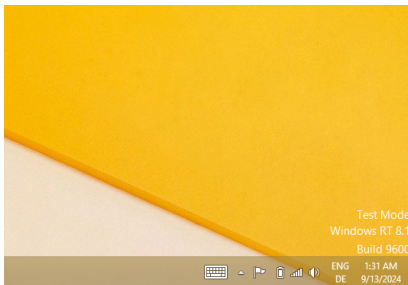
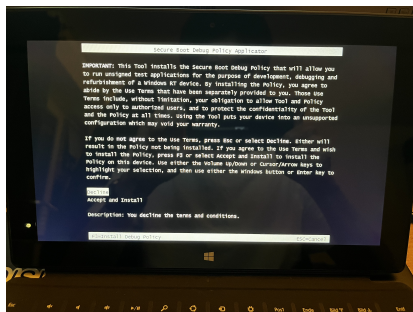


```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\gaos>
```

Step 1: Jailbreak

- Windows 8.1 RT ohne “Jailbreak Killing” Update
- Golden Keys / Longhorn

<https://windows-rt-devices.gitbook.io/windows/jailbreaks/golden-keys-longhorn>

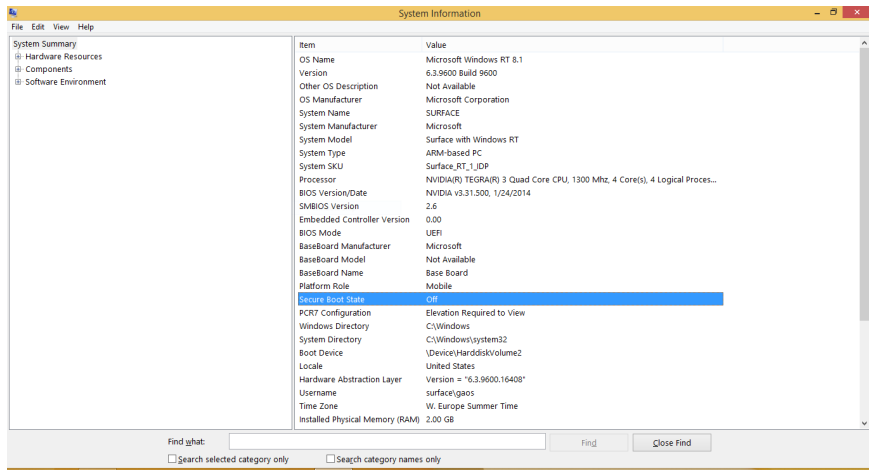


Step 2: Secure Boot deaktivieren

- Yahallo deaktiviert SecureBoot dauerhaft
- UEFI v3.30.500 – v3.31.500 mithilfe Tegra Jailbreak USB
<https://windows-rt-devices.gitbook.io/windows/tools/tegra-jailbreak-usb>



Step 2: Secure Boot deaktivieren



The screenshot shows the Windows System Information application. The 'System Summary' pane on the left is expanded to show 'Hardware Resources', 'Components', and 'Software Environment'. The main pane displays a list of system information items and their values. The 'Secure Boot State' is highlighted in blue and is set to 'Off'.

Item	Value
OS Name	Microsoft Windows RT 8.1
Version	6.3.9600 Build 9600
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	SURFACE
System Manufacturer	Microsoft
System Model	Surface with Windows RT
System Type	ARM-based PC
System SKU	Surface_RT_1_JDP
Processor	NVIDIA(R) TEGRA(R) 3 Quad Core CPU, 1300 Mhz, 4 Core(s), 4 Logical Proces...
BIOS Version/Date	NVIDIA v3.31.500, 1/24/2014
SMBIOS Version	2.6
Embedded Controller Version	0.00
BIOS Mode	UEFI
BaseBoard Manufacturer	Microsoft
BaseBoard Model	Not Available
BaseBoard Name	Base Board
Platform Role	Mobile
Secure Boot State	Off
PCR7 Configuration	Elevation Required to View
Windows Directory	C:\Windows
System Directory	C:\Windows\system32
Boot Device	\Device\Harddiskvolume2
Locale	United States
Hardware Abstraction Layer	Version = "6.3.9600.16408"
Username	surface\gaos
Time Zone	W. Europe Summer Time
Installed Physical Memory (RAM)	2.00 GB

Find what: Find Close Find

Search selected category only Search category names only

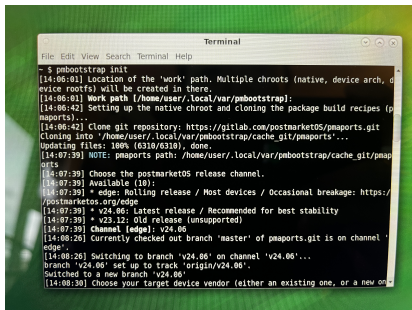
Step 3: Linux installieren (UEFI)

- Allgemeine Infos (Kernel, Bootloader und rootfs):
<https://openrt.gitbook.io/open-surfacert>
- Welche Distribution? Ubuntu? Raspberry Pi OS?
<https://wiki.ubuntu.com/ARM/SurfaceRT>
- OOTB-funktionierend? Spezieller Kernel erforderlich?

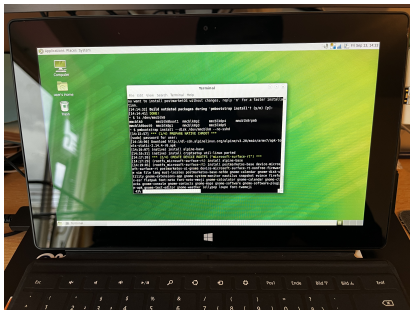
Step 3: Linux installieren (UEFI)

- <https://postmarketos.org>
- basiert auf Alpine Linux (musl libc, busybox, OpenRC etc)
- geeignet für Handys, Tablets und 2-in-1s.
- OOTB-Image für Surface RT

[https://wiki.postmarketos.org/wiki/Microsoft_Surface_RT_\(microsoft-surface-rt\)](https://wiki.postmarketos.org/wiki/Microsoft_Surface_RT_(microsoft-surface-rt))



```
Terminal
File Edit View Search Terminal Help
- $ pmbootstrap init
[14:06:01] Location of the 'work' path. Multiple chroots (native, device arch, d
device rootfs) will be created in there.
[14:06:01] Work path [/home/user/.local/var/pmbootstrap]:
[14:06:42] Setting up the native chroot and cloning the package build recipes (p
pmaports)...
[14:06:42] Clone git repository: https://gitlab.com/postmarketOS/pmaports.git
Cloning into '/home/user/.local/var/pmbootstrap/cache_git/pmaports'...
Updating files: 100% (6310/6310), done.
[14:07:39] NOTE: pmaports path: /home/user/.local/var/pmbootstrap/cache_git/pmap
orts
[14:07:39] Choose the postmarketOS release channel.
[14:07:39] Available (10):
[14:07:39] * edge: Rolling release / Most devices / Occasional breakage: https://
/postmarketos.org/edge
[14:07:39] * v24.06: Latest release / Recommended for best stability
[14:07:39] * v23.12: Old release (unsupported)
[14:07:39] Channel [edge]: v24.06
[14:08:26] Currently checked out branch 'master' of pmaports.git is on channel '
edge'.
[14:08:26] Switching to branch 'v24.06' on channel 'v24.06'...
branch 'v24.06' set up to track 'origin/v24.06'.
Switched to a new branch 'v24.06'
[14:08:30] Choose your target device vendor (either an existing one, or a new on
```



- Katastrophale Leistung (kein L2-Cache, keine Spannungssteuerung) - UEFI
- Vom Schlafmodus wacht WLAN nicht wieder auf
[https://wiki.postmarketos.org/wiki/Microsoft_Surface_RT_\(microsoft-surface-rt\)](https://wiki.postmarketos.org/wiki/Microsoft_Surface_RT_(microsoft-surface-rt))
- Gerät schaltet nicht aus (PostmarketOS v24.06)
- manuell ausschalten wie IBM PC/AT!
- Kamera funktioniert nicht
- Bluetooth funktioniert nicht (kein freier Treiber?)

APX Boot als Lösung?

- <https://openrt.gitbook.io/open-surface-rt/surface-rt/linux/booting/apx-boot>
- UEFI komplett umleiten
- De-brick Modus
- L2-Cache und Spannungssteuerung verfügbar

Kernel manuell kompilieren

- Linux v6.6.22 mit Patches
<https://gitlab.com/grate-driver/linux>
- flex, bison, gcc-arm-linux-gnu (Fedora/RedHat) oder gcc-arm-linux-gnueabi (Debian/Ubuntu)
- Infos: <https://openrt.gitbook.io/open-surfacert/surface-rt/linux/kernel/configure-and-build>
e.g.

```
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi- -j $(nproc) (Debian/Ubuntu)
```

oder

```
$ make ARCH=arm CROSS_COMPILE=arm-linux-gnu- -j $(nproc) (Fedora/RedHat)
```

- Das U-Boot (v2024.04)

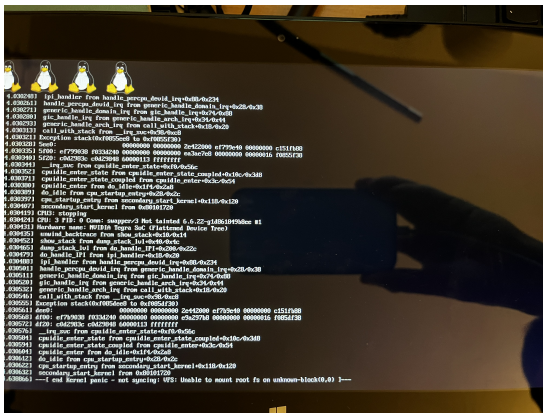
```
$ git clone https://source.denx.de/u-boot/u-boot.git -b v2024.04 --depth=1
```
- swig, python3-devel (Fedora/RedHat) oder python3-dev (Debian/Ubuntu)
- Infos: <https://openrt.gitbook.io/open-surfacert/surface-rt/linux/booting/apx-boot/uboot>

Jump Start! (WIP)

- python3-pyusb
- USB-A auf USB-A Kabel
- Fusee Gelee launcher
(https://github.com/tofurky/tegra30_debrick)
- BootConfigTable vom Surface RT kopieren
- Fusee Gelee Exploit senden

Jump Start! WIP

- Manuell kompilierter Kernel bootet nicht
- Linux binar v6.4: <https://openrt.gitbook.io/open-surfacert/surface-rt/linux/kernel/prebuilt-binaries>
- Kernel startet, aber findet kein rootfs (/dev/mmcblk0p2)!



```
4.030280 | ipi_handler from handle_percpu_devid_irq+0x00-0x234
4.030281 | handle_percpu_devid_irq from generic_handle_domain_irq+0x28-0x30
4.030281 | generic_handle_domain_irq from gic_handle_irq+0x74-0x0
4.030280 | gic_handle_irq from generic_handle_arch_irq+0x34-0x41
4.030282 | generic_handle_arch_irq from call_with_stack+0x18-0x20
4.030313 | call_with_stack from _irq_svc+0x0-0x0
4.030323 | Exception stack(0xffff0000 to 0xffff0000)
4.030320 | back: 00000000 00000000 2c422000 c1770c40 00000000 c151f400
4.030325 | sframe: c1770c30 f0334240 00000000 00000000 aa3ac70b 00000000 00000010 f025f730
4.030340 | sframe: f062920c 0a220000 00000113 ffffffff
4.030341 | _irq_svc from cpuidle_enter_state+0x10-0x21c
4.030342 | cpuidle_enter_state from cpuidle_enter_state_completed+0x10c-0x340
4.030371 | cpuidle_enter_state_completed from cpuidle_enter+0x3c-0x4
4.030380 | cpuidle_enter from do_idle+0x1f4-0x2ad
4.030389 | do_idle from cpu_startup_entry+0x28-0x0c
4.030392 | cpu_startup_entry from secondary_start_kernel+0x11b-0x120
4.030402 | secondary_start_kernel from 0x00101720
4.030410 | CPU1: stoping
4.030421 | CPU: 3 Fib: 0 Conn: 0sagepcc/3 Net: tainted 6.6.22-g1861945d0m #1
4.030421 | hardware name: RK3308 Tegrn SoC (Platform Device Tree)
4.030425 | unbind_backtraces from show_stack+0x10-0x14
4.030452 | show_stack from dump_stack+0x40-0x4c
4.030462 | dump_stack_lol from do_handle_irq+0x0-0x2c
4.030473 | do_handle_irq from ipi_handler+0x1b-0x20
4.030480 | ipi_handler from handle_percpu_devid_irq+0x00-0x234
4.030501 | handle_percpu_devid_irq from generic_handle_domain_irq+0x28-0x30
4.030511 | generic_handle_domain_irq from gic_handle_irq+0x74-0x0
4.030520 | gic_handle_irq from generic_handle_arch_irq+0x34-0x41
4.030521 | generic_handle_arch_irq from call_with_stack+0x18-0x20
4.030546 | call_with_stack from _irq_svc+0x0-0x0
4.030551 | Exception stack(0xffff0000 to 0xffff0000)
4.030561 | sframe: c1770c30 f0334240 00000000 2c422000 c1770c40 00000000 c151f400
4.030562 | sframe: f062920c 0a220000 00000113 ffffffff
4.030572 | sframe: c862903c c0420000 60000113 ffffffff
4.030576 | _irq_svc from cpuidle_enter_state+0x10-0x21c
4.030584 | cpuidle_enter_state from cpuidle_enter_state_completed+0x10c-0x340
4.030594 | cpuidle_enter_state_completed from cpuidle_enter+0x3c-0x4
4.030604 | cpuidle_enter from do_idle+0x1f4-0x2ad
4.030612 | do_idle from cpu_startup_entry+0x28-0x0c
4.030621 | cpu_startup_entry from secondary_start_kernel+0x11b-0x120
4.030621 | secondary_start_kernel from 0x00101720
4.030661 | — I end kernel panic - not syncing: UFS: Unable to mount root fs on unknown-block(0,0) !—
```

... und sagte Steve Jobs:

*The only problem with Microsoft is, they just have no taste ... in the sense that they don't think of original ideas and they don't bring much culture into their products.
... I have a problem with the fact that they just make really third-grade products.*