Připoj si JTAG
ke svému telefonu s Linuxem

Skrytý debug v USB portu Qualcommu (EUD)

Petr Hodina

OpenAlt 2026

O mně



Petr Hodina

Kontraktor

Embedded firmware

Přispěvatel do Mobile NixOS

Agenda

1. Apple: Lightning a JTAG
2. Proč JTAG na telefonu
3. EUD
4. OpenOCD
5. JTAG
6. ARM Debug Access Port
7. Omezení
8. Závěr

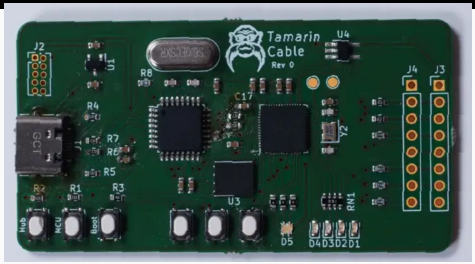
Kdo z vás někdy

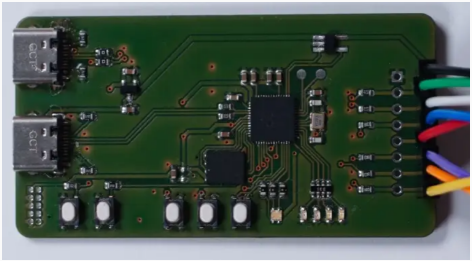
potřeboval ladit firmware

na telefonu?

Primáti na Lightningu

- Kanzi
- Kong
- Bonobo
- Tamarin





Lightning není jen USB



Tristar mluví SDQ (Apple ID Bus)

Správná sekvence ⇒ piny se přepnou na SWDIO / SWCLK



U Qualcommu stačí
obyčejný USB kabel.

Debug telefonu bez debuggeru

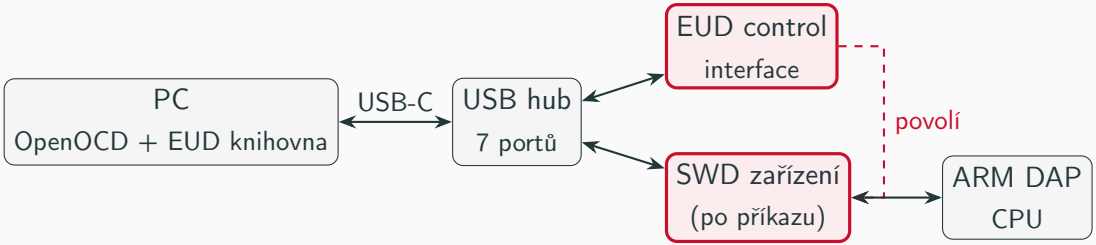
- Běžný postup: najít testpointy, **pájet** drátky, koupit externí JTAG/SWD debugger
- Typicky potřebujete **drahé IDE**, které daný debugger podporuje (J-Link, Lauterbach)
- Co potřebujeme: ladit **U-Boot**, TF-A, OP-TEE, kernel … i když systém ještě neběží
- Co nabízí Qualcomm: debug bridge zabudovaný v SoC, dostupný přes **USB-C**

Bez páječky, bez rozebírání telefonu.

EUD: Embedded USB Debug

- Hardwarový debug bridge uvnitř téměř každého SoC Qualcomm (přibližně od roku 2018)
- Napojený na CPU a také na DSP Hexagon
- **Nevyžaduje funkční systém ani USB stack** v telefonu, stačí zapnout pár registrů a USB PHY
- Qualcomm zveřejnil hostitelskou knihovnu až v únoru 2025: `github.com/quic/eud`

Jak to vypadá na USB sběrnici



PC uvidí **7portový hub**; obsazený je jen jeden port, *EUD control interface*.
Správnými USB příkazy se objeví druhé zařízení, které vystaví **SWD**.

Jak EUD zapnout

- Popsáno přímo v driveru `drivers/usb/misc/qcom_eud.c`: zápis do sysfs atributu `enable`
- EUD má dvě pojistky: jednu v **TrustZone** (secure svět, zápis přes SCM volání) a druhou v normálním světě (registr `CSR_EUD_EN`)
- Driver zapne obě, bez povolení v secure světě EUD nenaběhne
- Produkční zařízení: EUD je zakázané pojistkami (fuses), jde znovu povolit jen kryptograficky ověřenou **debug policy** podepsanou OEM
- Plně funguje na **Shift6mq** (bez secure bootu) a na **OnePlus 6** (běžný produkční telefon)

OpenOCD s podporou EUD

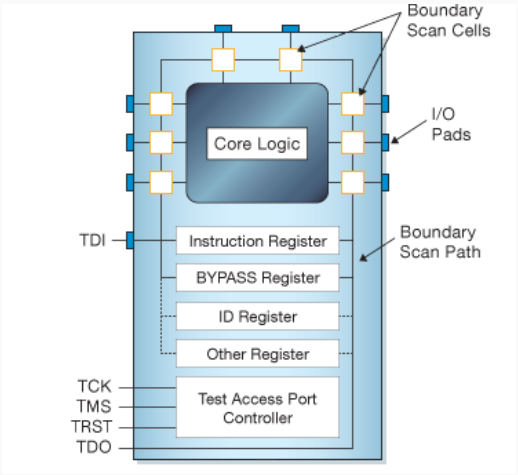
- Jde o fork OpenOCD: Qualcomm měl vlastní fork založený na OpenOCD 0.12.0, závislý na proprietární EUD knihovně
- Patche bylo nutné **portovat na nejnovější OpenOCD**: Linaro (Casey Connolly) je přeportovala na aktuální master, `github.com/linux-msm/openocd`
- Používá EUD knihovnu **bez NDA** (`quic/eud`)
- Konfigurace: `tcl/interface/eud.cfg` a `tcl/target/qualcomm/qcom.cfg`
- Otestováno na Snapdragonu 845

Připojení k DAPu

```
$ ./src/openocd -f tcl/interface/eud.cfg \
                -f tcl/target/qualcomm/qcom.cfg
Info : Using EUD 2.1.7
Info : SWD DPIDR 0x5ba02477
Info : QCOM.cpu0: hardware has 6 breakpoints, 4 watchpoints
Info : [QCOM.cpu0] Examination succeed
Info : [QCOM.cpu0] starting gdb server on 3333
QCOM.cpu0 halted in AArch64 state due to debug-request,
  current mode: EL0T
cpsr: 0x00000000 pc: 0xffff95cf9a4c
MMU: enabled, D-Cache: enabled, I-Cache: enabled
```

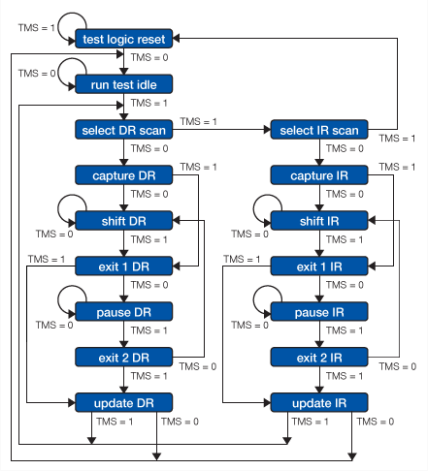
- Porty: gdb 3333, telnet 4444, tcl 6666
- Pak klasicky: `(gdb) target remote :3333`, `bt` při pádu U-Bootu

JTAG: přehled



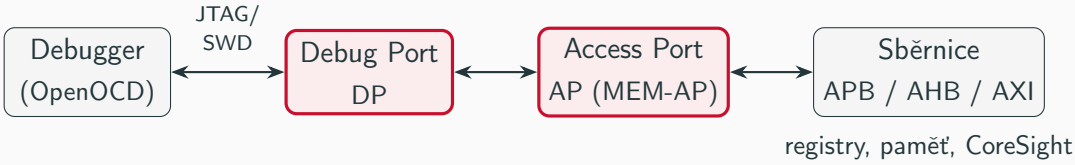
- Standard IEEE 1149.1, původně pro boundary scan desek
- Signály: **TCK**, **TMS**, **TDI**, **TDO**, volitelně TRST
- Čip má **TAP controller**, instrukční registr a datové registry (BYPASS, ID, ...)
- Instrukce vybere, který datový registr se připojí mezi TDI a TDO
- Stejné rozhraní využívá ARM DAP k přístupu do SoC

TAP: stavový automat



- 16 stavů, přechody řídí jen **TMS** při náběžné hraně TCK
- Pět taktů s TMS = 1 vždy vede do `test logic reset`
- Dvě větve: **IR scan** (instrukce) a **DR scan** (data)
- `shift`: bity se posouvají TDI → registr → TDO
- `update`: posunutá hodnota se uplatní

Co je DAP



- **DP** řeší inicializaci, reset a stav; má společnou část a část pro protokol (SWD nebo JTAG)
- **AP** zpřístupňuje zdroje SoC; MEM-AP čte a zapisuje paměťový prostor

Spojení obou světů

- EUD nám dá SWD místo JTAGu, ale nad ním je tentýž DAP: DP, AP, MEM-AP
- OpenOCD přečte DPIDR, prozkoumá AP a nastaví debug cíl
- Výsledek: `QCOM.cpu0`, breakpointy, watchpointy, gdb server
- Ladit lze U-Boot, TF-A nebo OP-TEE; na produkčním zařízení ale často jen do určité úrovně oprávnění

Na co si dát pozor

- Dostupnost: závisí na pojistkách a OEM debug policy
- Každý SoC má jiné **debug base** a **CTI base**; nové čipy chtějí další úpravy a kernel patche, které nejsou v upstreamu
- **SMP**: vhodné spíš pro **jednovláknové aplikace** (U-Boot); Linux na více jádrech je komplikovaný, bootujte s `maxcpus=1`
- Na hostu je podporovaný jen **USB2 hub**

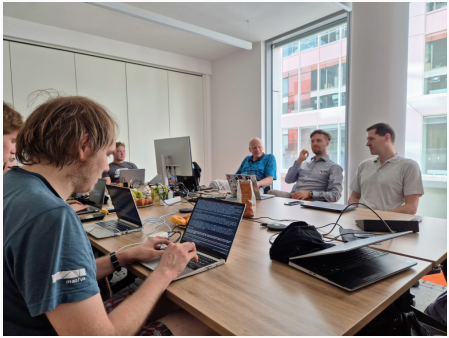
Shrnutí

1. **EUD** je JTAG/SWD zabudovaný v SoC Qualcommu a vyvedený přes USB
2. Stačí zapnout registry v **U-Bootu** nebo Linuxu a použít **OpenOCD**
3. Pod tím je standardní **ARM DAP**, takže platí vše, co znáte z jiných ARM
4. Žádná páječka, ale pozor na pojistky a debug policy

Zdroje:

`linaro.org/blog/hidden-jtag-qualcomm-snapdragon-usb`
`piolabs.com/blog/engineering/diving-into-arm-debug-access-port`

Děkuji!





Připojte se na Matrixu `#sdm845:ixit.cz`