Improving stability for TCPM using boards that are not self-powered

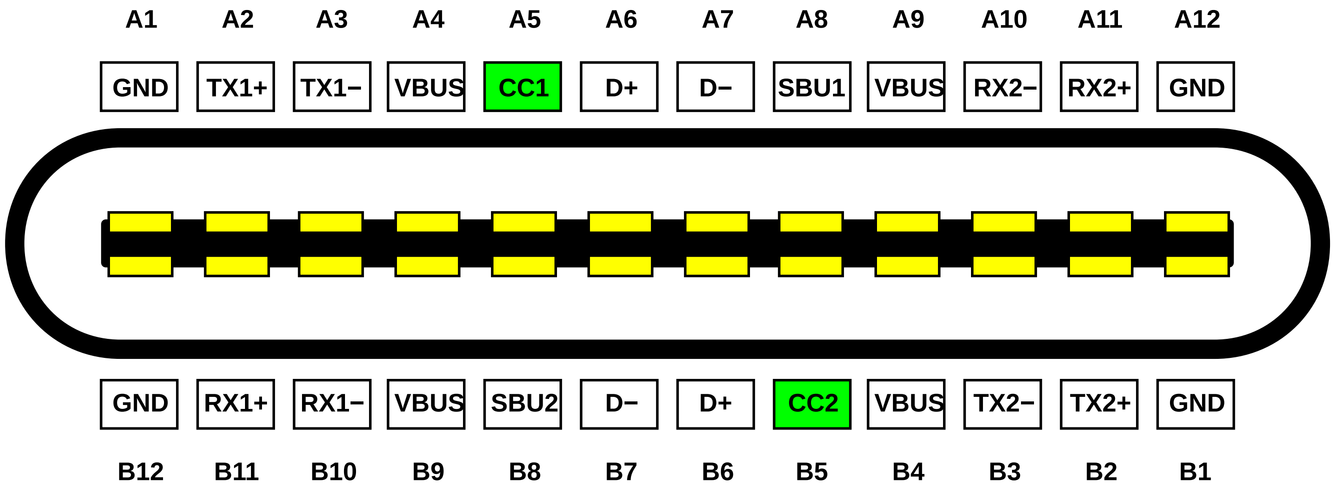
Sebastian Reichel

東京 2025 LINUX PLUMBERS CONFERENCE
TOKYO, JAPAN / DEC. 11-13, 2025

COLLABORA

Open First

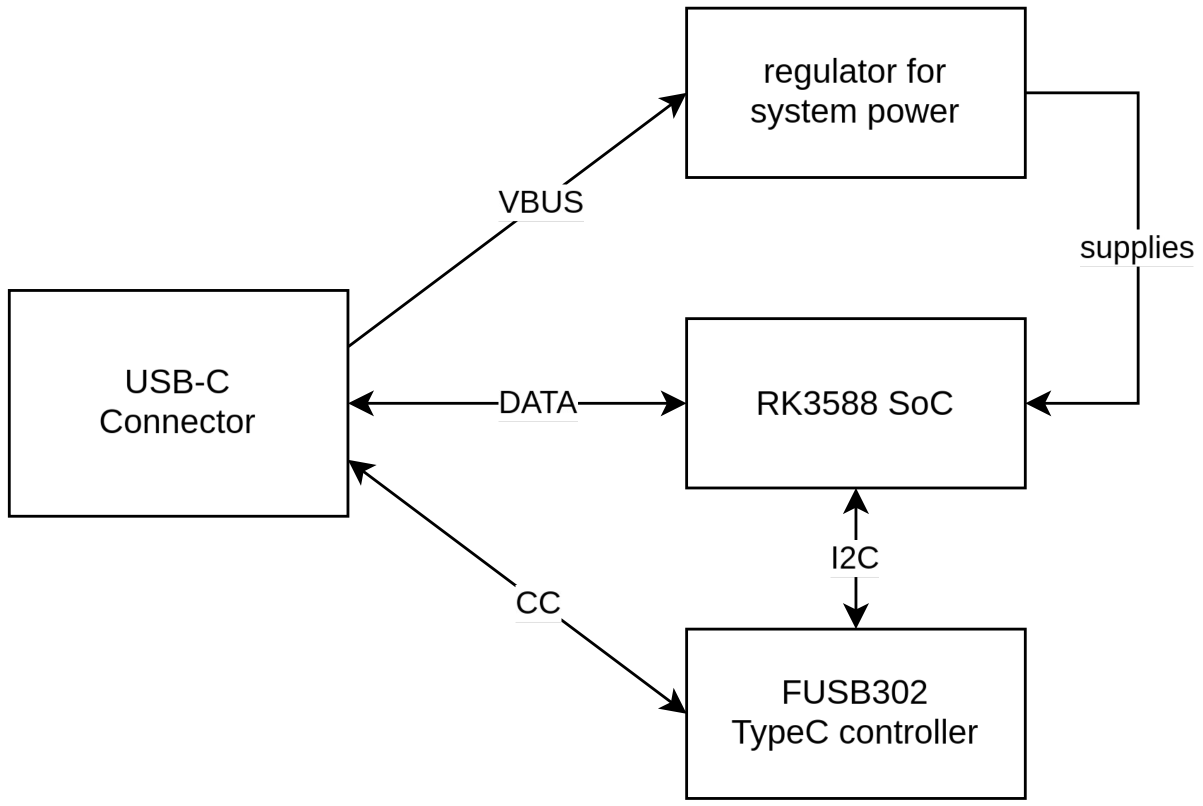
USB-C Power Delivery (PD)



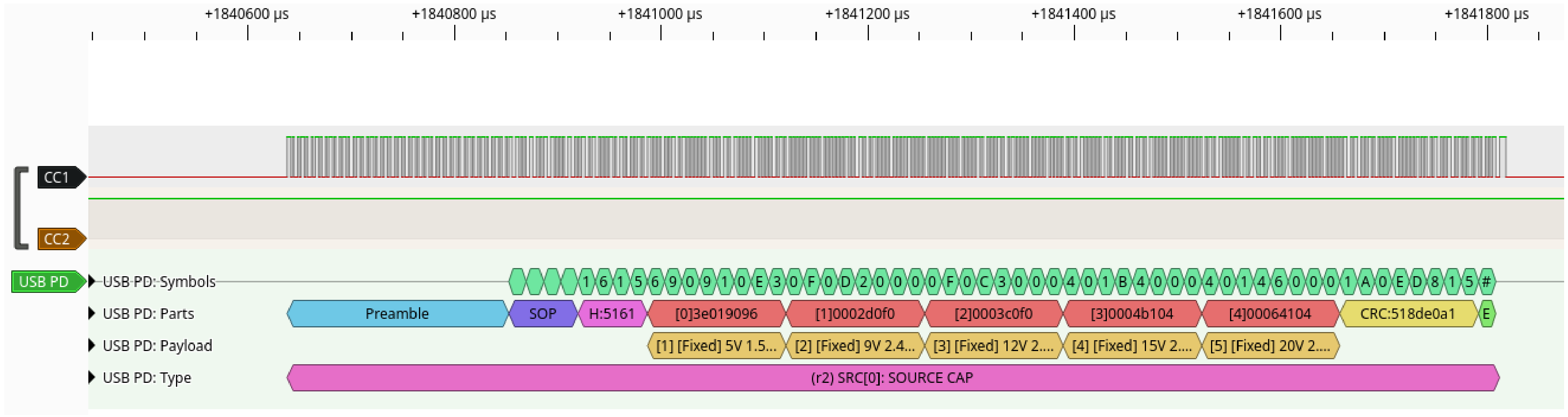
USB PD implementations

- Handled by Port Controller
- Option 1: autonomously handled
  - USB Type-C Connector System Software Interface (UCSI)
  - (this setup is usually found on laptops)
- **Option 2: non-autonomously**
  - e.g. USB Type-C Port Controller Manager (TCPM)

Radxa ROCK 5B USB-C setup



USB PD Source Capability



```
USB PD: Symbols   1 6 1 5 6 9 0 9 1 0 E 3 0 F 0 D 2 0 0 0 0 F 0 C 3 0 0 0 4 0 1 B 4 0 0 0 4 0 1 4 6 0 0 0 1 A 0 E D 8 1 5 #
USB PD: Parts     Preamble | SOP | H:5161 | [0]3e019096 | [1]0002d0f0 | [2]0003c0f0 | [3]0004b104 | [4]00064104 | CRC:518de0a1 | E
USB PD: Payload   [1] [Fixed] 5V 1.5... | [2] [Fixed] 9V 2.4... | [3] [Fixed] 12V 2.... | [4] [Fixed] 15V 2.... | [5] [Fixed] 20V 2....
USB PD: Type      (r2) SRC[0]: SOURCE CAP
```

COLLABORA

Open First

USB PD Initialization

- Source Capability message is send multiple times
- After 5 sec source assumes device is not PD compliant
- If source receives PD message later: Error State
- Error recovery: Hard Reset
- Hard Reset involves disabling VBUS!

Radxa ROCK 5B USB-C issue

- Boot process is relatively slow
  - DRAM Init, Trusted FW setup, U-Boot SPL, U-Boot, Load compressed kernel, decompress it, jump to it, boot to fusb302 driver probe
- specification compliant source will result in boot loop

Solutions?

- Modify hardware design to use an autonomous controller
  - Too late, hardware is already on the market
- Make booting the kernel fast enough?
  - Unrealistic in lots of scenarios (network boot, debugging)
- Handle USB-C controller in firmware
  - firmware is right now SoC specific, but PD controller is board specific
  - requires sharing I2C interface between firmware and kernel
  - requires new protocol between firmware and kernel to share USB-C information

Solutions?

- Add TCPM to bootloader (U-Boot)
  - Landed upstream by now
  - U-Boot initializes PD
  - Linux re-initializes by doing a soft-reset (which does not remove VBUS)

COLLABORA

So problem fixed, all good?

- Unfortunately no
- A lot of USB-PD source equipment does not strictly follow the spec
  - For example many do not start to send source capability messages after a soft-reset
- TCPM strictly follows the state machine described in the specification
  - This may result in hard reset being requested by the kernel; effectively a hara-kiri operation

Kernel Workarounds

- I introduced SNK_WAIT_CAPABILITIES_TIMEOUT, which diverts from the spec (122968f8dda8)
- Print error on potential hara-kiri operation (876483a5a5bd)

Ideas?

- Avoid hard reset and (ignore potentially broken USB-PD)?
  - Requires better detection of affected devices, currently relying on *self-powered;* property not being set for the connector DT node
- Hand over state from bootloader?
  - State machines are not exactly the same...
- ...

Thank you!