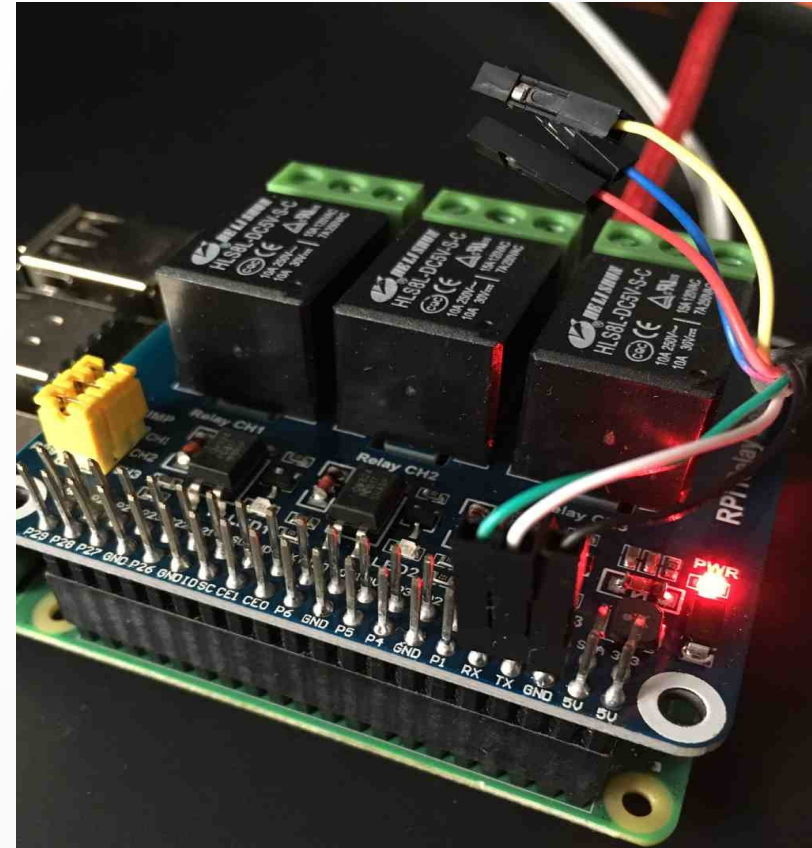


# Device Trees Should Be Modular

Heinrich Schuchardt, 2020-05-26  
CC-BY-SA-4.0

# Systems Are Modular

- Baseboard + SoC module
- Hats (Raspberry),  
Capes (BeagleBoard)
- Memory modules



# Devices Are Hot Pluggable

- Even CPUs and Memory can be removed and added at runtime

Linux: [Documentation/core-api/memory-hotplug.rst](#)

Linux: [Documentation/core-api/cpu\\_hotplug.rst](#)

# Is There A Master Device?

- With protocols like CCIX there can be multiple bus masters.
- So in future modular systems it is hard to say whose requirements dictate the “one” device tree.
- Do we need separate device trees per bus master?

# Configurable IO Routing

- SoCs may have more supported IOs than connected pins.
- The routing of GPIOs is configurable, e.g. Raspberry GPIO 18 may be connected to input register, output register, PWM, SPI, PCM Audio.
- PCI-e bifurcation is used to connect multiple devices to one PCI-e connector.

# Device Configurability

- Many devices contain their own firmware or even FPGA.
- Device firmware changes may make the device appear completely different to the outside world.
- How about
  - Changing functions according to license status
  - Changing CPU ISA via microcode

# Hardware or Software?

```
vdd_soc: sd0 {  
    regulator-name = "VDD_SOC";  
    regulator-min-microvolt = <1000000>;  
    regulator-max-microvolt = <1170000>;  
    regulator-enable-ramp-delay = <146>;  
    regulator-disable-ramp-delay = <4080>;  
    regulator-ramp-delay = <27500>;  
    regulator-ramp-delay-scale = <300>;  
    regulator-always-on;  
    regulator-boot-on;  
    maxim,active-fps-source = <MAX77620_FPS_SRC_1>;  
    maxim,active-fps-power-up-slot = <1>;  
    maxim,active-fps-power-down-slot = <6>;  
};
```



# DT Nodes as Communication Area

- Necessary properties, e.g.
  - `cpu-release-addr` in CPU node for CPUs with `spin-table enable-method`
  - Available memory
- Optional properties, e.g.
  - GRUB passing `'bootargs'`, `'linux,initrd-start'`, `'linux,initrd-end'` to Linux



# DT in the Software Stack

- Some devices should only be available in secure world
  - SPI flash for UEFI variables

# Conclusion

- Device trees are a mishmash of
  - Hardware description
  - Hardware configuration
  - Software configuration
  - Communication area

# Areas for Improvement

- Separation between hardware and software properties
- Support for hardware modularity
- Software layer specific device trees
- Multiple masters (CPUs, Accelerators)
- Ownership

# Could Device Trees Be Modular?

