Embedded Linux Conference
Europe

OpenIoTSummit
Europe

A Sockets API For LoRa

Andreas Färber,
SUSE Labs
afaerber@suse.com

THE LINUX FOUNDATION

About The Presenter

- Project Manager for arm64 architecture at SUSE Labs
- Involved in arm port of openSUSE Linux distribution
- Kernel maintainer for Realtek and Actions Semi arm SoCs
- Other kernel projects you might know:
  - Odroid-XU, Parallella, Spring Chromebook, GeekBox, …
  - STM32F4, FM4, XMC4500; S905, IAP140, MB86S71, RDA8810PL
- Background in virtualization technologies – QEMU

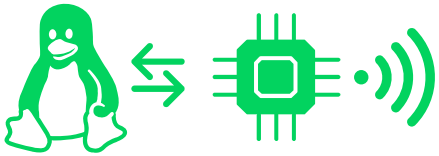
Why LoRa Technology?

- LoRa = **Long Range** – radio modulation by Semtech
  - https://archive.fosdem.org/2018/schedule/event/sdr_lora_aes/
- Low-Power Wide Area Network (**LPWAN**) with low throughput
- Unlicensed **sub-GHz** and 2.4 GHz ISM/SRD bands (U-LPWA)
- No dependency on network infrastructure providers
- Wide availability of HW – https://en.opensuse.org/HCL:LoRa
- … and just because it's possible!

Getting Started With LoRa Chipsets

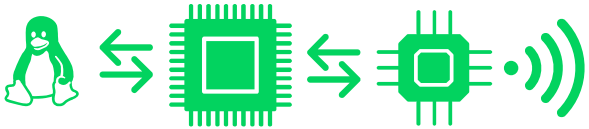
… and down the rabbithole it goes!

Types Of LoRa Radio Modules



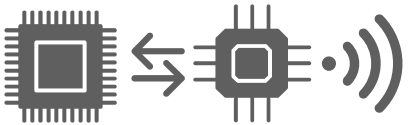
Plain transceiver

- SPI / UART / USB
- Linux host needs software MAC



MCU w/firmware + transceiver

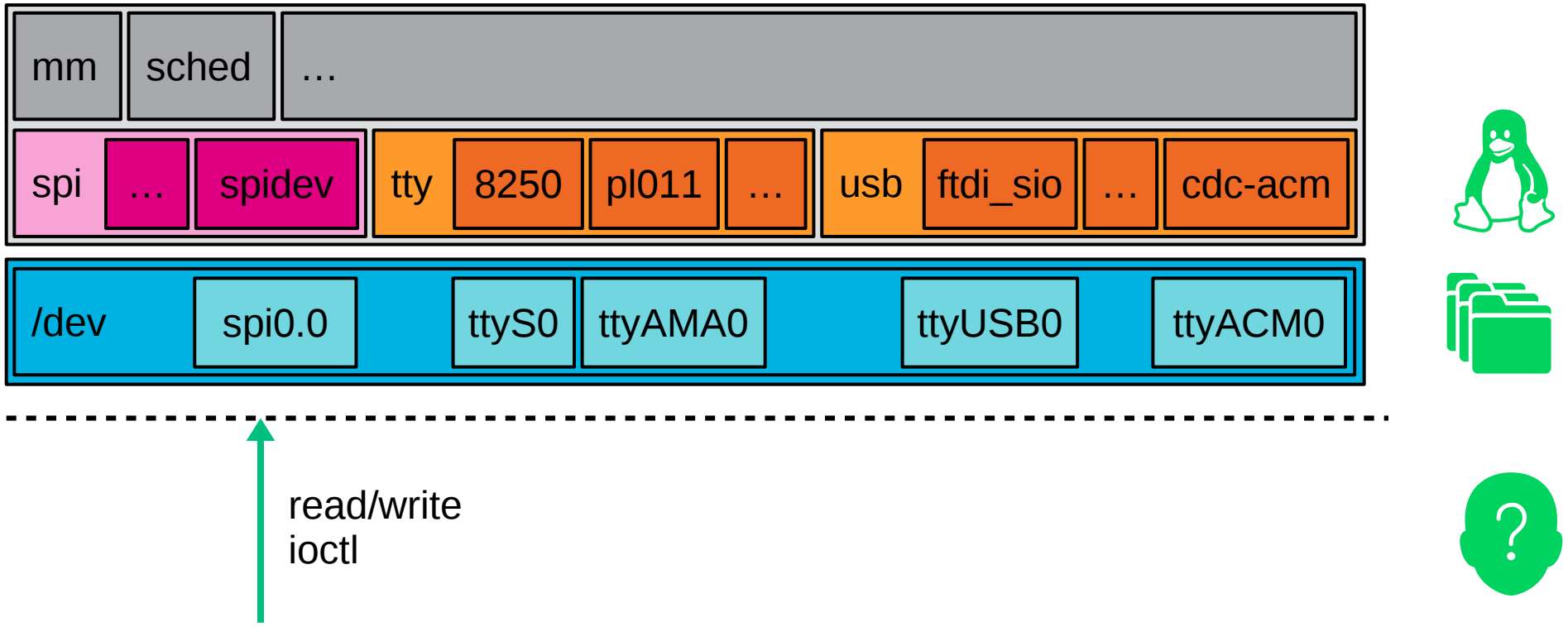
- UART / USB
- Firmware determines features exposed
- Optional certified MAC



Plain MCU + transceiver

- n/a – no fixed API
- Custom MCU code for sending / receiving
- Optional MAC

Accessing LoRa Hardware Today



Issues With LoRa Open Source Software Today

- No upstream community – per-vendor application forks
- Software license incompatibilities
- Use of spidev kernel module gets ugly in distros
- Hardware detection duplicated into applications

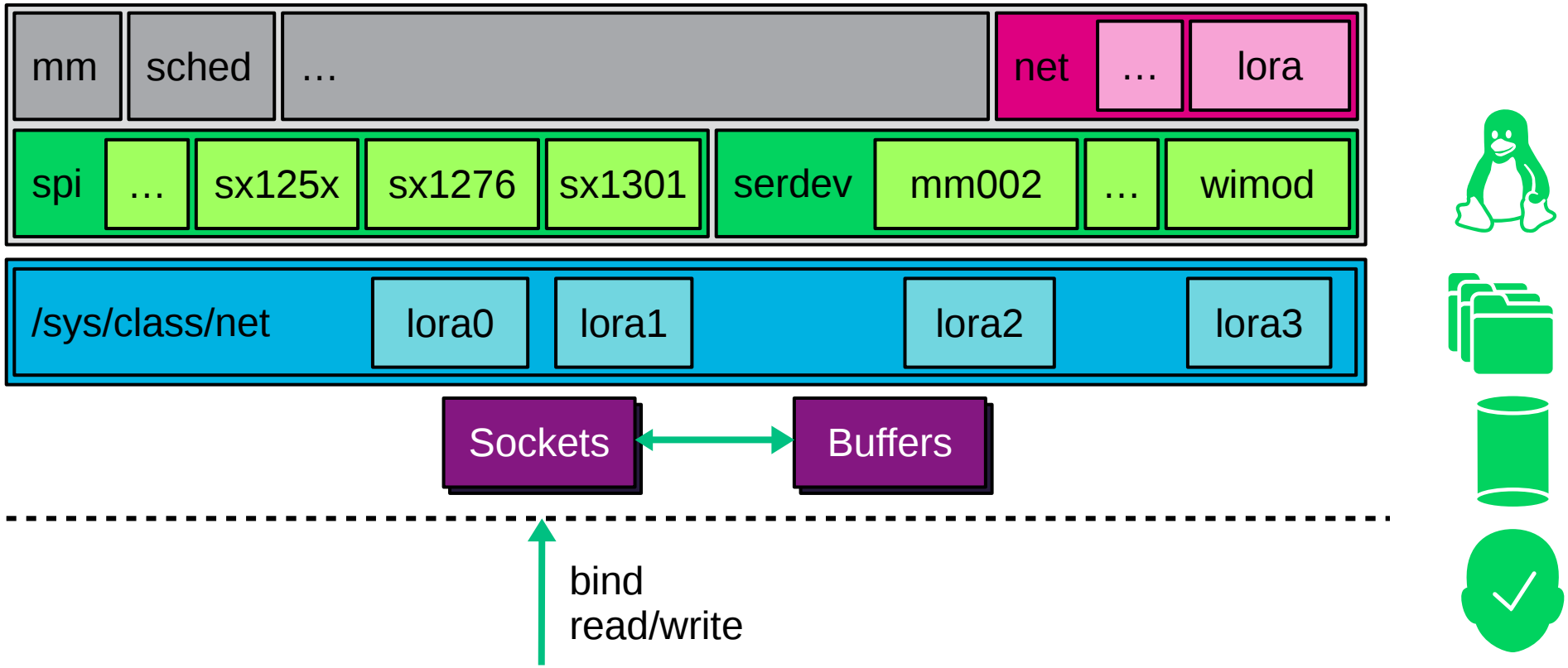
Idea: Move chipset drivers into mainline Linux kernel.
Encourage generic, community-maintained packet forwarders.

Collecting Requirements

- Shall expose equivalent chipset features as before
- Shall allow implementation of proprietary protocols
- Shall allow reuse of protocols layered on top
- Shall fit all Semtech chipsets and many third-party modules

Idea: Sockets seem an intriguing approach for LoRaWAN. Similarities to Wifi and IEEE 802.15.4 may help users.

Andreas In Wonderland – Sockets (Proposed)



Semtech SX1272 f. / SX1276 ff. Transceivers

- Single channel
- Two modes: FSK/OOK and LoRa (switchable via Sleep mode)
- State machine for RX vs. TX (switchable via Standby)
- SPI register interface
- 256 byte RAM data buffer (LoRa) / 64 byte FIFO (FSK)

Semtech SX1261 f. / SX1268 Transceivers

- Single channel
- Two modes: LoRa and FSK
- State machine for RX vs. TX (switchable via Standby RC/XOSC)
- SPI command interface, indirect register interface
- 256 byte RAM data buffer

Semtech SX1280 f. Transceivers

- Single channel
- Multiple modes: LoRa, FLRC, FSK, BLE and Ranging
- State machine for RX vs. TX (switchable via Standby RC/XOSC)
- UART and SPI command interface, indirect register interface
- 256 byte RAM data buffer

Semtech SX1301 / SX1308 Concentrators

- Multi-channel
- IF0-7 LoRa channels, IF8 LoRa uplink channel, IF9 FSK channel
- Two radio transceivers (SPI/ADC) – SX1255 / SX1257 f.
- SPI register interface – no documentation, only reference code
- 1024 byte data buffer
- Firmware blobs for calibration and operation

Socket Addressing For Radios

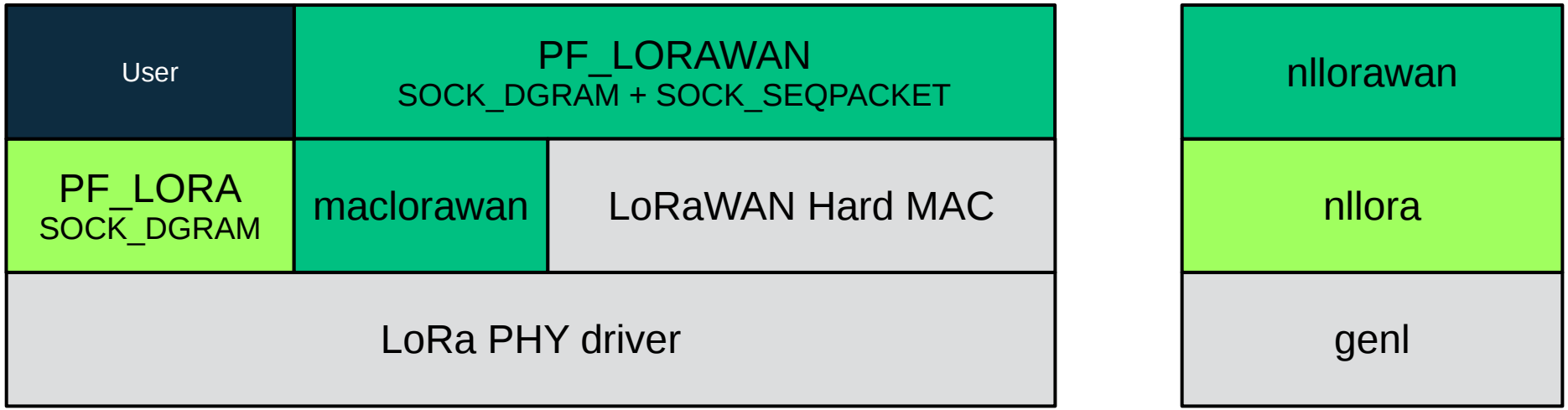
- Transmission is broadcast
  - Addressing only at MAC layer
- Preamble may serve to recognize frame start, not “metadata”
- Optional filtering by Sync Word

Idea: Define address as radio properties that allow reception. (An alternative following later.)

LoRa Socket Address (Proposed)

- Network interface index
- Radio frequency
- Spreading Factor
- Bandwidth
- Sync Word (1 Byte)

LoRa Socket Layers (Proposed)



LoRaWAN Socket Address (Proposed)

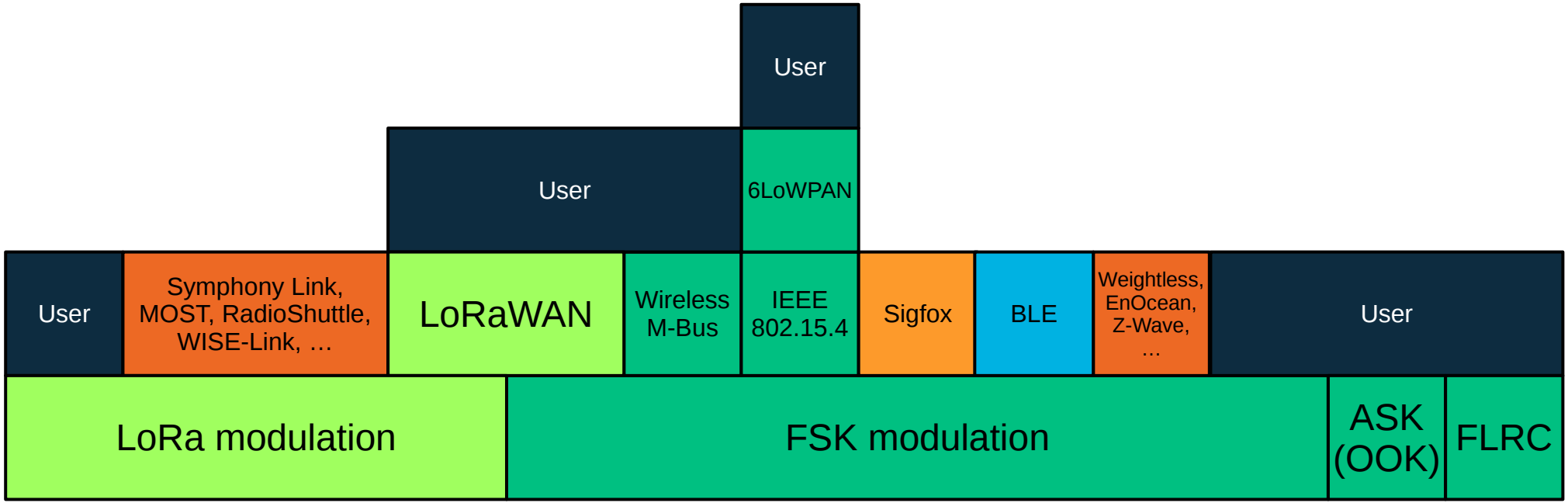
- Network interface index
- Data Rate
  - LoRa: channel frequency, SF, bandwidth
  - FSK: channel frequency, bandwidth
- Port

Implies a fixed LoRa / FSK Sync Word respectively.

Listening Can Be Hard

- Packets can be **transmitted** with different modes and settings
- Sockets require to **receive** whenever we're not transmitting
  - How to detect and handle conflicting settings for reception?
  - When socket is opened, all settings need to have been initialized
- There's no unified frame format field to **detect** MAC protocols
  - Need to try to parse incoming frames for each protocol

Protocol Layers Around LoRa



Frequency-Shift Keying (FSK)

- Address: frequency, sync word (multi-byte), Gauss …

- Also found in: nRF24L01+, CC1120, MRF89XAM8A, SP1ML

On/Off-Keying (Amplitude-Shift Keying)

- Address: frequency, …

- Also found in: CC1120, MRF89XAM8A

Fighting Pollution: Unified Radio Sockets?

- Can we avoid a socket address for each modulation?
- Use generic **PF_PACKET** + SOCK_DGRAM + htons(ETH_P_…)?
  - Would not allow radio configuration via socket address
  - Would still allow SOCK_RAW for Software Defined Radio
  - How could we switch modes or detect conflicts? Socket options?

Related: Bluetooth LE Support

- Semtech SX128x: alternative mode
- AppconWireless RF1276TS, Laird RM1xx: separate antenna
- Kernel appears to rely on **HCI** – what to do about raw PDUs?

Test Setup For LoRa Kernel Drivers

- arm, arm64 and mips Single Board Computers
  - Using Device Tree Overlay where possible
- openSUSE Tumbleweed + Kernel:HEAD repo
  - Tricks for insmod: https://github.com/afaerber/lora-modules
- Relevant chipsets being tested before pushing to linux-lora.git
  - Limitations: 868 MHz and 433 MHz (EU), donated hardware
- Idea: interoperability testing, both locally and across radio link

Action Plan

- Working towards RFC v2 – need to complete regmap adoption
  - Staging branch to be archived and squashed into series
- On top: LoRaWAN soft MAC patch series by Jian-Hong Pan
  - https://www.slideshare.net/chienhungpan/lorawan-class-module-and-subsystem
- Validate / evolve socket design – needs testing and feedback
- Merge into mainline kernel, enable in openSUSE Tumbleweed

Credits

Industry Contributors – Code

Industry Supporters – Hardware

Competing LPWAN Technologies

Other U-LPWAN: Sigfox

- Frequency: Unlicensed sub-GHz SRD/ISM bands
- MTU: 12 bytes uplink, 8 bytes downlink

- Why care? Found in Nemeus MM002-L**S** modules
  - How to expose? Device? PF_SIGFOX? lora0 + sigfox0?
  - How to interact with LoRa sockets?

Other LPWAN: NB-IoT

- Frequency: Licensed 3GPP bands
- MTU: 1500 bytes
- Two modes: UDP and non-IP
- SIM card needed

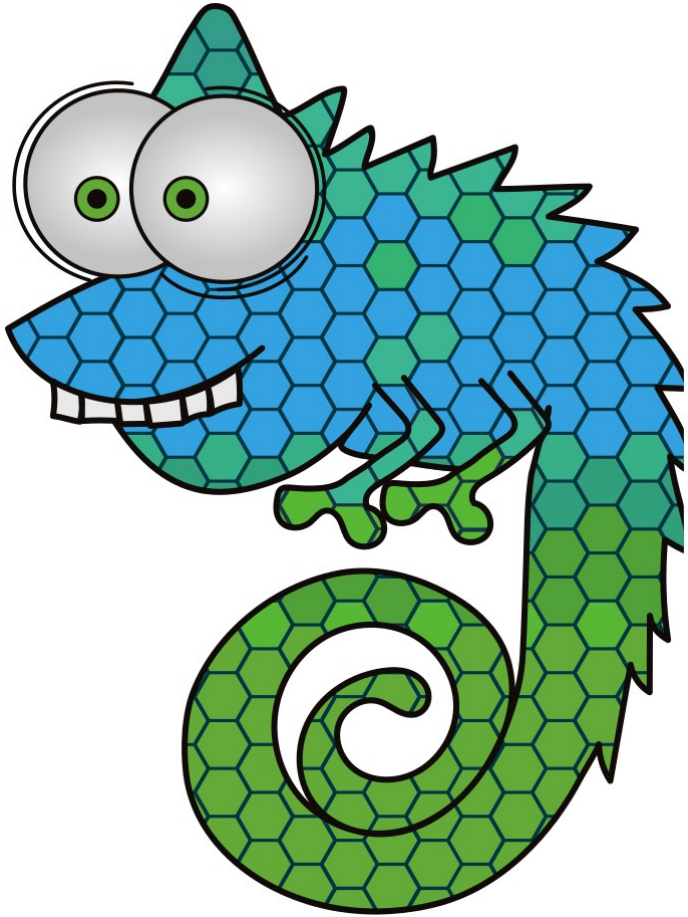
How to handle in Linux?

Conclusions

Resources

- RFC patch series: https://patchwork.ozlabs.org/cover/937545/
- Testing hints: https://github.com/afaerber/lora-modules
- Staging tree with lora-next branch: https://git.kernel.org/pub/scm/linux/kernel/git/afaerber/linux-lora.git/
- Chipset overview and links to SBC expansion boards: https://en.opensuse.org/HCL:LoRa

Questions? Feedback?



Join Us at www.opensuse.org

Backup

Radio Modulation Types Of Other Technologies

- MIOTY: Lfour: BPSK; TS-UNB: GMSK; DD-UNB: BFSK
- Sigfox: D-BPSK and GFSK
- Weightless-P: GMSK BT=0.3 or OQPSK
- Wireless M-Bus: 4GFSK

- Bluetooth LE: GFSK (2.4 GHz)

License

General Disclaimer

Credits

Template
Richard Brown
rbrown@opensuse.org

Design & Inspiration
openSUSE Design Team
http://opensuse.github.io/branding-guidelines/

Embedded Linux Conference Europe

OpenIoTSummit Europe

THE LINUX FOUNDATION