Remote-update adventures

RAUC, Yocto, Barebox

Meetup Paris Embedded #12
2019-03-28 – EPITA, Paris
Patrick Boettcher YAISE

About me

- Kernel developer since 2004 – (media drivers)
- Embedded C++ and C (mostly Linux and RTOS)
- Freelancer with YAISE

Overview

1- Intro / Starting position

2- Remote update-strategy

3- Remote upgrade-strategy

Initial project requirements

- industrialized LORA-gateway (LoRaWan over 4G)
- special housing and connectors
- selected Phytec phyCORE-i.MX 6UL
  - 512MB RAM
  - 512MB NAND
  - ARM Cortex A7 – up to 900MHz
- Phytec provides a customizable Yocto distribution

Phytec's Yocto Distro

- based on morty (yocto 2.2)
- using barebox as bootloader
- device-tree well handled (barebox passes it to the kernel)
- no “standard” upgrade mechanism in this release

System configuration

- barebox (with built-in-device-tree)
- barebox-env
- kernel
- dtb
- rootfs (~490MB)

- system is accessible via remote-ssh-tunnel (initiated by the system)

What happened?

- June 2018: first 100 devices have been delivered with the application-layer running nicely
- September 2018: a software update requires update of the base-system – *of course*
  - tried to do it with opkg/ipk – not satisfying (to be retried)
  - decision to study upgrade possibilities
- October 2018: Decision-time (customer pays):
  - 1st define general update strategy
  - 2nd define strategy to update the 100 installed

Overview

1- Intro / Starting position

2- Remote update-strategy

3- Remote upgrade-strategy

Basic facts after study - 1

Stick close to what exists and
do not re-invent the wheel, but what does exist?

- Partitioning (UBI) and filesystem (UBIFS) ***cannot*** be modified from Linux’s user-space running on UBI
- Putting processes “somewhere else” to change UBI is complex, barebox can of course do it
- Barebox has bootchooser
- Barebox has “state”-partition

Basic facts after study - 2

- Barebox has a filesystem inside its environment
- Barebox is self-updatable
- Barebox environment can be accessed from Linux’s user-space
- Robust Auto-Update Controller (RAUC) with signed bundles exists and works
- Newer versions of Phytec’s Distro use all this, but do not support Morty

New partitioning

- barebox (with built-in-device-tree)
- barebox-env

- rescue-initramfs (46M)
- rescue-dtb (512K)

- system0/1: kernel (9M), dtb (512K), rootfs (210M)

Reconfiguring barebox

- Add bootchooser-functionality
- Add state-functionality and to device-tree (stored in eeprom)
- Add format and flash-scripts to defaultenv
- Add boot-scripts to defaultenv
- Build barebox-target-tools to rootfs

- `barebox_2017.04.0-phy3.bbappend / state-eeprom.dts`

RAUC - bundle

- Add meta-rauc
  - provides “bundle-class”
  - provides user-space tools and config-base

- In your layer add a bundle-file and a .bbappend for certs and config

- bitbake <bundle-file-target>

`meli-bundle.bb`

RAUC – bootchooser - demo

Overview

1- Intro / Starting position

2- Remote update-strategy

3- Remote upgrade-strategy

Blind upgrade strategy

- How to convert a remote system from a single-system-installation to dual-installation?

- Simple:
  - Copy new-barebox, kernel, dtb, rootfs to the rootfs (scp)
  - Instrument barebox from userspace (bareboxenv)
  - Reboot
  - Meditation during 3 minutes
  - ssh-reverse-tunnel is apprearing

Blind time - 1

- user-space
  - injects a script to barebox which is executed at boot
  - reboot
- in old-barebox:
  - mount rootfs, update-barebox if image present
  - reset

Blind time - 2

- in new-barebox:
  - script mounts rootfs, copies images if present
  - copy some specific config-files to backup-dir
  - repartition
  - flash rootfs, kernel, dtb
  - resets state, bootchooser
  - reset
- in new-linux
  - first-boot: config-files are extracted from barebox-env
  - everything works – ssh-tunnel appears

Success

- All remotely deployed devices have been upgraded successfully to date

Thanks.

p@yai.se

@PatBoeFra

https://github.com/pboettch

https://yai.se/