Konsulko Group

Linux+Zephyr: IoT made easy

IoT Explodes Everywhere

Konsulko Group

- "Sensors and actuators embedded in physical objects and linked through wired and wireless networks, often using the same Internet Protocol (IP) that connects the Internet" - Definition by McKinsey.
- Sounds just like embedded stuff we've been doing for decades! So what changed?
- Now there's a big market and the world is just 'ready'.
- Unfortunately there's no standard (and probably will not be one for many decades), fragmentation is running amok.

Linux IoT

Konsulko Group

- Around for many years and been doing IoT things with it before it had a cool name.
- All protocols have their reference implementation on Linux (AllJoyn, MQTT, Weave, XMPP, etc)
- Unfortunately Linux has gotten quite large
  - Minimum kernel for embedded target > 4MB compressed
  - Requires a few hundreds of GB of flash for a general purpose install.
- Not suitable for very small devices

Price is everything

Konsulko Group

- If we could run Linux on everything you wouldn't be in this presentation!
- Linux is secure (has years of scrutiny and professional security people go after every commit with a fine tooth comb).
- Linux has the full networking stack required.
- Linux has the I/O capabilities (every sensor/actuator driver)
- Unfortunately you can't run Linux on $0.5 part.

Price is everything (cont)

Konsulko Group

- Run IoT on a $0.5 part (ideally, we could do with $1)
- This is the sweet-spot for many applications
  - Put a $50 device off premises and get lost or stollen, you might get a bit upset -> end up not buying it.
  - Put a $1 device off premises and get lost or stollen -> meh.
- Turns out that we need an OS for our cheap devices that is leaner than Linux.
- Many options!

RTOS selection

Konsulko Group

- A big selection of choices, the big contenders
  - FreeRTOS
  - mbedOS
  - NuttX
  - magenta (LK)
  - None
  - Zephyr

FreeRTOS

Konsulko Group

- Dual License - GPLv2 with linking exception or commercial
- Sparse API - the most RTOS of old of the most
- No community development for the core (separate)
- Preemptive thread model, optional MPU protection
- Networking is an add-on - no high level frameworks
- Suited for people moving on from bare metal

mbedOS

Konsulko Group

- Apache 2, 6LoWPan under permissive binary license (etc).
- New mbed 5 RTOS (CMSIS-RTOS - RTX)
- Community involvement minimal (ARM focused).
- MINAR event based API
- Complete standard support (two different IP stacks)
- Tied to the mbed OS cloud API
- Most easy to get started, but not very open source IMO.

NuttX

Konsulko Group

- BSD 3 clause
- Most Linux API of them all.
- Has a large community but 80% is Mr. Nutt himself
- Quite POSIX compatible - easier to port Linux stuff
- IP stack but not much else IoT related
- Larger than the other options, most Linux like.

magenta (part of fuchsia)

Konsulko Group

- MIT license
- Littlekernel - Used on Android bootloaders.
- Just introduced, community is dubious
- Limited priority number, standard primitives.
- Networking stack - couldn’t figure this out :)
- Google project - significant infant mortality

None (is always an option)

- N/A
- N/A
- N/A
- N/A
- N/A
- Only for the hardcore

Zephyr

Konsulko Group

- Apache 2 (network stack Apache 2)
- Adequate RTOS API (and nano/micro option)
- Under Linux Foundation - true open source
- Networking stack - IP stack + IoT options (CoAP) + BLE
- Linux kconfig build system, feels right at home.
- Our selection.

IoT on Zephyr.

Konsulko Group

- CoAP, BLE, contiki + tinydtls
- Can be very small (smallest nanokernel example at 8K)
- Porting of IoT libraries possible
- You can accomplish quite a lot.
- What about Linux? Where does it fit in?

IoT on Zephyr (problems)

Konsulko Group

- Security - your IoT device has keys and passwords, how do you handle it being stolen by a malicious party?
- Convenience - How easy can you update the software on the devices? You might have dozen on your premises.
- Future Proofing - Not enough room in non-volatile storage for every IoT protocol. What happens if the company goes out of business? You have to change all the lightbulbs/security system/etc in such a case?
- Warring tribes - iPhone vs Android - it should work with my other devices too.

Solution: Linux Gateway

Konsulko Group

- Linux gateway and slave Zephyr IoT devices.
- Linux can run all IoT protocols (and with enough RAM at the same time)
- Future proof - Linux is easily updated, since point to do so.
- Software on the IoT devices? Should it be an IoT stack? Do we still need to update s/w on the IoT devices?

Intermission: Cheap IoT (1)

Konsulko Group

- Cheap MCU with on chip peripherals
  - ARM M core/PIC/ARM/x86/AVR
  - GPIOs
  - PWM
  - Serial
  - I2C
  - SPI
  - Networking (IEEE 802.15 or ZigBee or WiFi)…

Intermission: Cheap IoT (2)

Konsulko Group

- Analog glue
- Sensors on I2C/SPI bus
- Very price sensitive
- Speeds are usually low
- Power budget is small
- Less is more

What if there was no IoT (1)

Konsulko Group

- Linux has full I/O capabilities for IoT
- Problem is that the sensors are remote
- What if we got rid of the heavy weight IoT protocols?
- Zephyr devices are simply peripherals
- S/W load on Zephyr devices is the same for the same SoC no matter what different kind of sensor/IoT device it is
- Linux kernel interfaces mean all the IoT application are insulated from the underlying device details.

What if there was no IoT (2)

Konsulko Group

- Significant less attack surface with security problems:
  - The devices only contain enough keys to connect with the gateway and have no valuable information
  - There is much less software on the device
  - Both ends can be secured either with pre-programmed keys or using a smart-phone application on install time.
  - The devices are basically disposable.

Implementation (bare)

Konsulko Group

- Directly access the SoC resources
  - Registers
  - Busses
- + No software besides networking and configuration
- - Performance might be impacted
- - More power required (tradeoff)

Implementation (class drivers)

Konsulko Group

- Thin class layer
  - GPIO class for instance
  - Classes for every kind of peripheral
- - More S/W compared to barebones
- + Better performance
- + Less power required

Status

Konsulko Group

- Unfortunately no demo yet!
- Test bed is a beaglebone and an Intel Galileo
- Kernel driver provides remote GPIOs now, I2C, SPI to come
- Kernel drivers uses a user-space helper to bridge to the device over BLE.
- Pre-configured keys with DTLS
- WIP, hope to have a demo at ELC.

Thank you!

Konsulko Group

Questions?