Technical Overview

11 October 2016

runtime.io

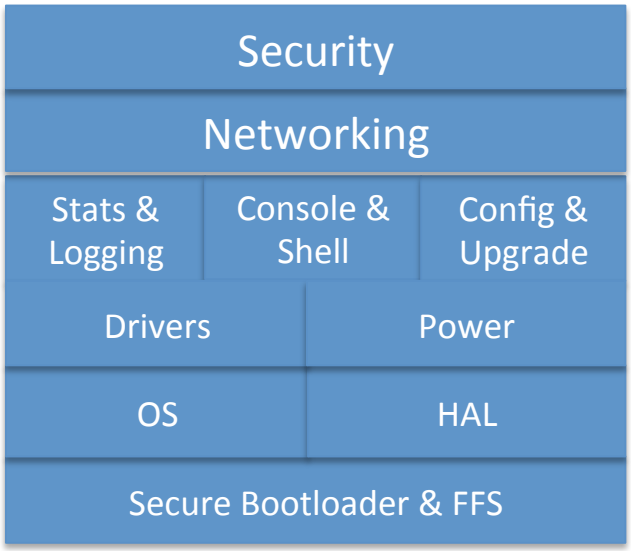
Problem and Context

First release of a successful connected product…

…. now make that repeatable please.

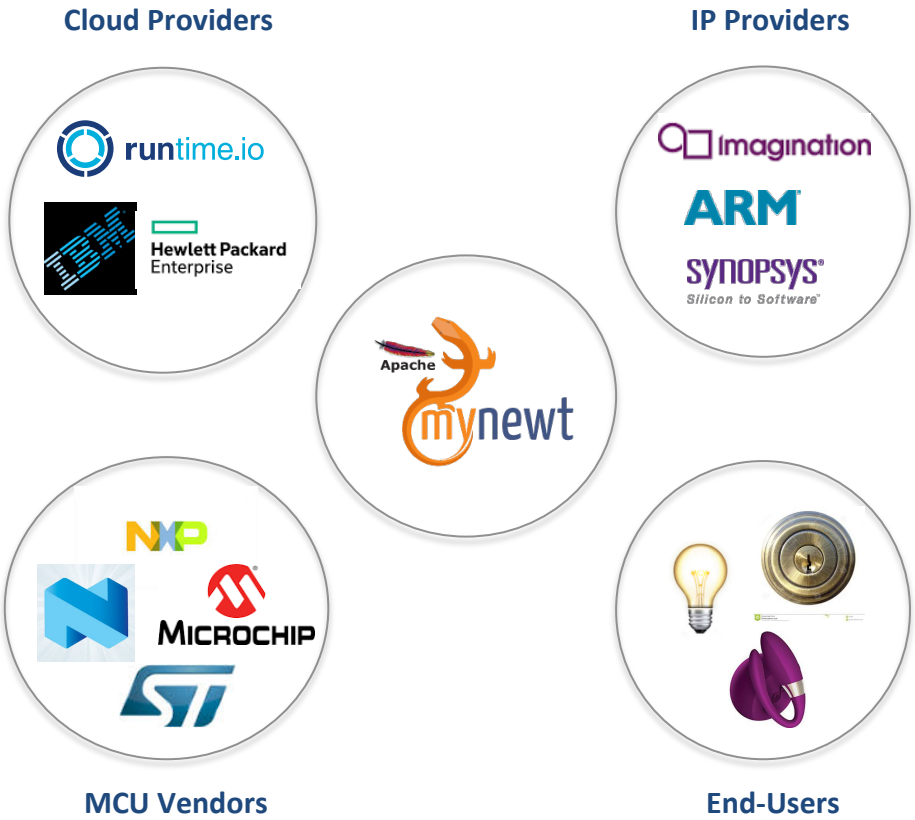
runtime.io

An Open Source OS for MCUs



- Open source networking stacks: Bluetooth Host and Controller
- Pre-emptive, tickless RTOS with power management
- Secure bootloader and Image Upgrade
- Flash FS and Access Mechanisms
- Build & Package Management
- Management Interfaces

A Community Driven OS



Why Apache Software Foundation?

- Liberal, BSD-style license
- Strong Licensing and IP policies
- Meritocracy
- Free to contribute, contributors control project direction.
- History of working with large organizations: IBM, Pivotal/ EMC, Microsoft.
- Many years experience managing large, complex projects (e.g., Apache, Hadoop, Subversion)

Community driven open source best way to maintain healthy ecosystem

RTOS

- Tickless operation: low power hooks
- Driver Interface
- Pre-emptive, multitasking RTOS
  - Strict priority-based scheduling
  - Up to 253 different priority levels
- Unified buffer management
- Resource utilization tracking and watchdog
- High-resolution timers
- Built-in tasks:
  - Idle

```c
#include <os/os.h>

#include <assert.h>

/* Task 1 */
#define TASK1_PRIO (1)
#define TASK1_STACK_SIZE    OS_STACK_ALIGN(1024)
struct os_task task1;
os_stack_t stack1[TASK1_STACK_SIZE];
static volatile int g_task1_loops;

void
task1_handler(void *arg)
{
    while (1) {
        ++g_task1_loops;

        /* Wait one second */
        os_time_delay(1000);
    }
}

int
main(int argc, char **argv)
{
    int rc;

    os_init();

    os_task_init(&task1, "task1", task1_handler, NULL,
            TASK1_PRIO, OS_WAIT_FOREVER, stack1, TASK1_STACK_SIZE);

    os_start();

    assert(0);

    return (0);
}
```

Apache mynewt

runtime.io

RTOS - Event Driven Model

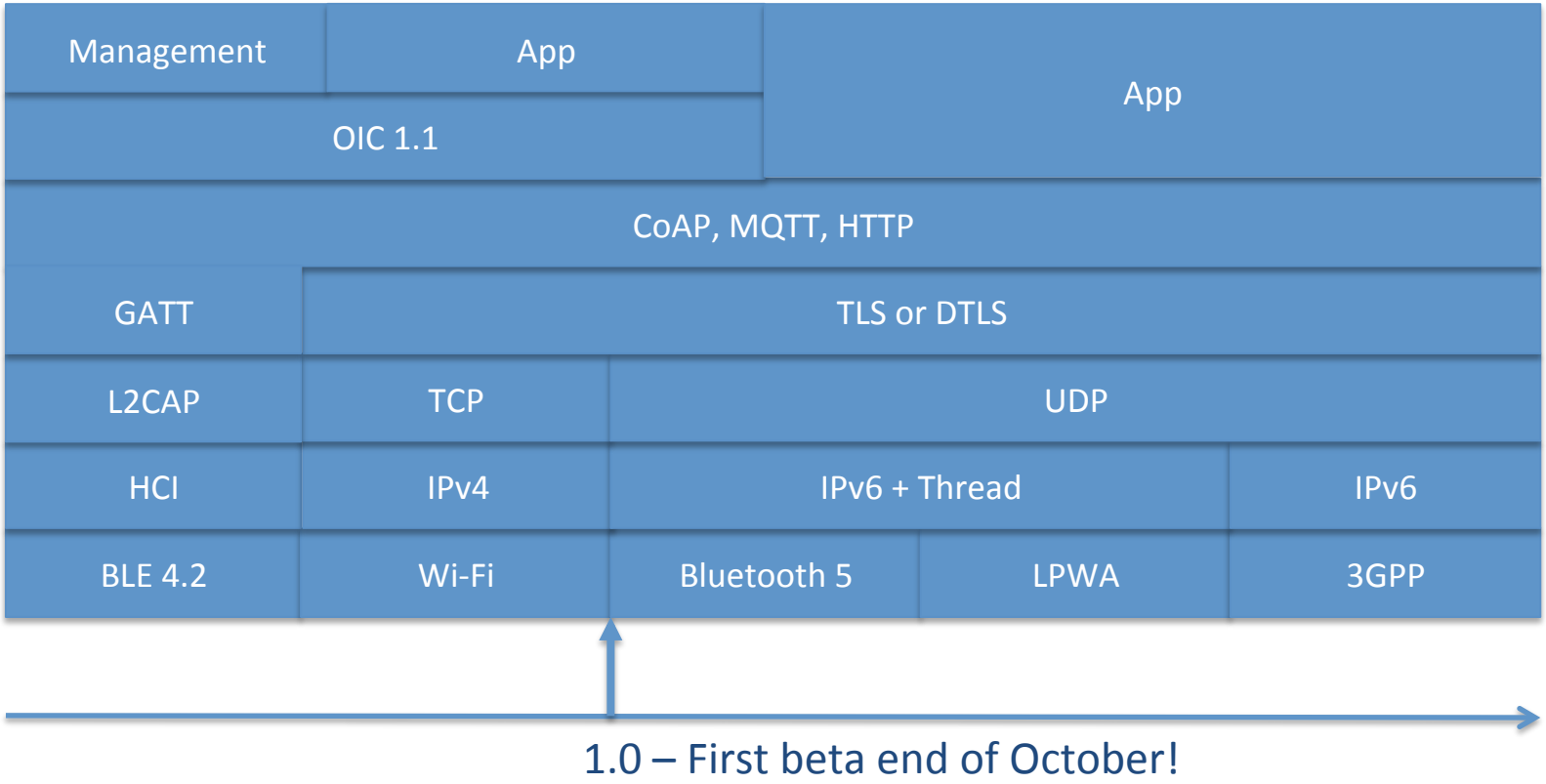
- Event Queues provide a mechanism for “mostly-sleeping” asynchronous tasks
- Wake-up on:
  - Message from another Task
  - Timer
  - I/O state change
  - Incoming packet
  - Watchdog
- Perform operations:
  - Send an alert
  - Respond to a request
  - Schedule a wakeup
- Go back to sleep

```c
struct os_eventq task1_evq;
struct os_eventq task2_evq;

#define OS_EVENT_T_PING (OS_EVENT_PERUSER)
#define OS_EVENT_T_PONG (OS_EVENT_PERUSER + 1)

void
task1_handler(void *arg)
{
    struct os_event *ev;
    struct os_event ping_ev;

    ping_ev.ev_type = OS_EVENT_T_PING;
    ping_ev.ev_arg = NULL;

    os_eventq_put(&task2_evq, &ping_ev);

    while (1) {
        ev = os_eventq_get(&task1_evq);
        assert(ev->ev_type == OS_EVENT_T_PONG);

        os_eventq_put(&task2_evq, &ping_ev);

        ++g_task1_loops;

        /* Wait one second */
        os_time_delay(1000);
    }
}

void
task2_handler(void *arg)
{
    struct os_event *ev;
    struct os_event pong_ev;

    pong_ev.ev_type = OS_EVENT_T_PONG;
    pong_ev.ev_arg = NULL;

    while (1) {
        ev = os_eventq_get(&task2_evq);
        assert(ev->ev_type == OS_EVENT_T_PING);

        os_eventq_put(&task1_evq, &pong_ev);

        ++g_task2_loops;
    }
}
```

Apache mynewt

runtime.io

Apache Mynewt Connectivity Layer (Runtime's View)



1.0 – First beta end of October!

Standards-based IoT communication

runtime.io

Highlights: Apache Mynewt Networking

Bluetooth 4.2

- Nordic nRF51 and nRF52 support
- Simultaneous Central and Peripheral modes
  - Supports up to 32 connections
- Combined (host + controller), host-only and controller-only mode
- Compatible with BlueZ
- 40% less code versus licensed binaries in peripheral-only mode

Wi-Fi

- Abstracted interface to Wi-Fi supplicants, and socket layer
- LWIP integrated to provide native-IP support.
- Support for WINC1500...more soon!

Highlights: Challenges Managing Connected Products

Cross-platform Support

- Well-defined drivers and HAL interfaces
- Build and package management system optimized to understand BSP and link options

Software Upgrade

- Build tool creates signed images
- Image download over Serial, BLE and Wi-Fi
- Bootloader verifies SHA-256/RSA/ECC-DSA signature

Debugging

- Consistent logging and statistics infrastructure
- Core dumps
- Kernel-level support: sanity, stack guards, memory tracking

Power Mgmt

- Hard sleep and wakeup support (low RAM states)
- Tickless 'idle' and driver suspend
- Networking stack sleep management

Build, Package, and Project Management: newt tool

- Composable System
  - Open-source project collaboration
  - Maintaining private code trees
  - Enforced source code layout

- Install and Upgrade
- Unified method for #includes and #defines
- Dependencies and APIs

- Build
  - Configuration
  - Multiple targets
  - Source code layout
    - 3rd party SDKs

- System configuration (a la Device Tree)
- Toolchains
- Target management
- SDK compilation rules

- Package Management
  - Versioning and stability

- Go Small or Go BIG
- VCS Versioning
- Versioning Scheme (major, minor, rev)
- Tracking branches

Build and Package Management: newt tool (continued)

- Artifacts
  - Debugger maintenance
  - Generation of flash images, upgradable images

- bin/ directory, with object files
- Multiple targets stored simultaneously
- Compiler definitions, map files

- Introspection
  - Size
  - Packages
  - Versions

- Display dependencies
- Search for functionality
- Versions installed – tracking branches

- Enforced Hierarchy
  - HW: MCU, BSP
  - APP

- BSP + App = BUILD
- BSP -> MCU definition

- System Definition
  - Split images for upgrade
  - RAM locations
  - Flash

- Linker sections defined by system
- Tie-in with flash layout

Highlight: Apache Mynewt Security

Provisioning →

- Unique device identification
- Certificate Management
- Prevent counterfeiting

Upgrade →

- Signed firmware images (newt tool)
- Secure bootloader

Communications →

- Leverage either BLE or DTLS security
- RBAC for commands based upon identity

Data and Tamper →

- Encrypted flash storage
- TPM/Smart Card aware
- Support hardware key access

Designed for security from the ground-up

Hardware Platform Support

Nordic Semiconductor nRF51/52

- Suitable for single-chip / host BLE designs
- Controller-only operation with host processor
- Cost-effective nRF51, powerful nRF52

ST Micro STM32F/L34

- Host processor, popular for BLE + Wi-Fi
- Offload processor for DSP processing
- Extensive, quality peripheral support
- L* series provides reduced power consumption

Atmel SAMD/L21

- Extensive low power modes and operation
- Rich set of peripherals
- Community-supported: Arduino Zero series

More Coming Soon!

Cross platform support provides flexibility and price leverage

What's Next?

- More boards(!) and processors(!)
  - PIC32MZ (MIPS - underway), NXP FRDM-K64, NXP KW41Z, STM32 L4
- Wireless
  - Bluetooth 5 and Bluetooth Mesh
  - LPWA
  - Improved Wi-Fi
- Wired: Ethernet
- Sensor APIs and Sensor Management
- You Decide!

Apache mynewt

runtime.io

- More information: http://mynewt.apache.org/
- Join the development, subscribe to dev@ list.
- Contributors welcome!

THANK YOU