RN-0.8.0-b2-incubating

New Features and Enhancements

1. Arduino Zero support

   Mynewt OS now supports two flavors of the Arduino Zero board - Zero and Zero PRO. Source code from the manufacturer is used for supporting the Atmel SAMD21 MCU used in these Arduino boards.
   For more on how to tell Mynewt to fetch such external files and get the boards blinking, try the tutorial: https://mynewt.apache.org/os/tutorials/arduino_zero/
   A bootloader is required for these boards which can be built using the "newt" build tool that comes with Mynewt.

2. Newt Tool Improvements

   Newt is a two-in-one command line tool for embedded contexts. It is both a source package management system and a build/debug/install system. A detailed introduction can be found here: https://mynewt.apache.org/newt/newt_intro/

   In this release, the user interface has been vastly improved to make it simpler to follow and use. Some examples of its updated simplicity are:
   - "newt new" - creates a new project and automatically creates a skeleton directory structure with some basic packages in them.
   - "newt test" - allows the user to execute unit tests for one or more or all packages.
   - "newt build" - allows the user to build an application for a specified target
   - "newt load" - allows the user to load a built image onto the board

3. Wall-clock Time

   In addition to relative time, either using OS time (which is a 1ms tick), or cputime (which allows higher resolution), Mynewt OS now has a concept of "real" / "wallclock" time. Wall-clock time is kept in a 64-bit value.

4. Support for all LE roles in BLE 4.2

   There are four GAP (Generic Access Profile) roles defined for devices operating over an LE physical transport - Broadcaster, Observer, Peripheral, and Central.

   Mynewt OS allows a device to be set up with multiple roles simultaneously. A device can be configured to act concurrently as a Central with one peer and as a Peripheral with another. There is no limit to the roles a device can take on concurrently, provided there is ample memory, CPU cycles, and power. A device playing the Central role also usually takes on the Observer role. A device that is a Peripheral usually plays the Broadcaster role. So technically, a Mynewt device can play all the four roles if a use case so requires.

   All mandatory features required in the host and controller parts in order to support all the four LE GAP roles (Broadcaster, Observer, Peripheral or Central) have been implemented.

5. Rich set of logs and statistics for BLE 4.2

   Numerous counters have been added in Mynewt's BLE stack (NimBLE) to collect and expose data about both the host and the controller activities.

   Controller stats include, but are not limited to:
   - packets received and sent with details
   - packet errors with error code and details (e.g. malformed packet, crc errors, bad ID etc.)
   - scheduling status and errors, if any

   Host stats include, but are not limited to:
   - command/response success and failure counts
   - errors and failures (e.g. MTU errors, read failures, write failures etc.)
   - timeouts
   - channel setup success/failure details

6. Random static address in BLE 4.2

   Mynewt NimBLE now supports random static addresses that are either burned into silicon or randomly generated to a new value at each power cycle.

7. New tutorials

   Several new lessons have been added here (https://mynewt.apache.org/os/tutorials/tutorials/) to help the user get familiar with Mynewt OS and its BLE 4.2 stack (NimBLE).