Creating Bluetooth®-based IoT Solutions with Zephyr™ OS

Johan Hedberg
Intel Corporation
Bluetooth® low energy technology

- Also known as BLE or Bluetooth Smart
- Introduced in 2010 with Bluetooth 4.0
- 2.4 GHz, slightly different radio modulation than Bluetooth Classic
- 100m range, 1Mbps bandwidth
- Years of battery life on a coin-cell battery
- Controllers come in single- & dual-mode variants
- Perfect for IoT use-cases
Zephyr™ OS Bluetooth® Stack

- Bluetooth 4.2 compliant, 5.0 in progress
- Almost complete low energy feature set
  - All mandatory features
  - Most optional features
- Bluetooth Classic (BR/EDR)
- Host-Controller separation through HCI
- Native Controller support
Bluetooth® host stack architecture

- GAP (Generic Access Profile)
  - Peripheral & Central
  - Observer & Broadcaster
- IPSP for IPv6 over Bluetooth LE
- Clean HCI driver abstraction
  - Standard physical transport drivers (UART, SPI, etc.)
  - Virtual driver for native Controller support
- Verified with multiple popular controllers
- Highly configurable
  - Features, buffer sizes/counts, etc.
Host stack runtime view

ACL Data → rx_thread → bt_recv()

HCI Event → tx_thread → bt_send()

bt_hci_cmd_send() → bt_conn_send()
Host stack runtime, Controller-side thread

- ACL Data
- HCI Event
  - bthci_cmd_send()
  - bt_conn_send()
- rx_thread
  - bt_recv()
- tx_thread
  - bt_send()
- HCI Driver
Network buffer integration

- Common network buffer API: net_buf
- Easy encoding & decoding
- Fragmentation
- (Near) zero-copy
- Compatible with kernel objects like FIFOs
- Cross-layer, e.g. to/from controller
- Cross-subsystem, e.g. to/from IP stack
Configuring Bluetooth® Host Features

- HCI driver
- Features
  - GAP/GATT roles
  - Security (pairing & signing)
- Buffer sizes & counts
- Stack sizes
- Max number of paired devices & connections
  - Can be even 0
- Debug options
Creating a Bluetooth® application

- Initialize the stack
  - bt_enable()
- Register GATT service database
  - bt_gatt_register(services)
- Advertise and let others connect
  - bt_le_adv_start(parameters)
- Notify of value changes
  - bt_gatt_notify(parameters)
- Many samples available
  - samples/bluetooth/*
Development tools

- QEMU* support
  - Integration with BlueZ on a Linux* host
  - HCI tracing
  - GDB

- Real devices
  - Bluetooth Monitor Protocol over console UART
  - Interleaved log messages & HCI data
  - Decoded using btmon from BlueZ

*Other names and brands may be claimed as the property of others.
LE Controller implementation

- Contributed by Nordic Semiconductor
- Available since Zephyr 1.6
- LE Link Layer
- As many instances of connected LE roles as RAM & configuration permits
- nRF5x radios supported
- Radio abstraction
- Exposes HCI to the Host stack
Possible configuration options

**Controller-only**
- UART/SPI/USB
- Raw HCI API
- Controller

**Host-only**
- Bluetooth® app
- Host stack
- HCI Driver

**Combined Host & Controller**
- Bluetooth app
- Host stack
- Controller
### Possible configuration options - details

<table>
<thead>
<tr>
<th>Controller-only</th>
<th>Host-only</th>
<th>Combined Host &amp; Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw HCI API</td>
<td>Bluetooth® API</td>
<td>Bluetooth API</td>
</tr>
<tr>
<td>UART, USB, SPI</td>
<td>HCI transport driver</td>
<td>Virtual HCI driver</td>
</tr>
<tr>
<td>Arduino 101* (nRF51)</td>
<td>UART, SPI</td>
<td>nRF52-based boards</td>
</tr>
<tr>
<td>Carbon (nRF51)</td>
<td>Arduino 101® (Quark SE)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Carbon (Cortex M4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QEMU*</td>
<td></td>
</tr>
</tbody>
</table>

*Other names and brands may be claimed as the property of others.
Bluetooth® BR/EDR support

- Bluetooth Classic
- Generic Access Profile (GAP)
  - Device discovery, pairing, connection creation
- Data transfer (L2CAP & RFCOMM)
- Service Discovery (SDP)
- Hands-Free Profile (HFP)
- Advanced Audio Distribution Profile (A2DP)
- Audio/Video Remote Control Profile (AVRCP)
Future

- Work on upcoming specifications
- More Bluetooth® 5.0 features
- Bluetooth Mesh
- LE Link Layer support for more radios (non-Nordic)
- Better net_buf integration for Link Layer
- Link Layer Privacy
- Vendor HCI specification
Get involved!

- www.zephyrproject.org
- Mailing list: devel@lists.zephyrproject.org
- IRC: #zephyrproject, #zephyr-bt @freenode.net
- Code: gerrit.zephyrproject.org, bluetooth branch
- Issue tracking: jira.zephyrproject.org
Questions?