Tech Talks Schedule – Presentation will begin shortly

tech talks

Tuesday, April 19	**The Latest Bluetooth Low Energy Updates in GSDK 4.0**
Tuesday, May 3	Matter: Developing with Matter on the MG24
Tuesday, May 17	AI/ML: Bringing Intelligence to the Edge on the MG24
Tuesday, May 31	Matter: Securing your IoT devices
Tuesday, June 14	Wi-Fi: Coexistence with RS9116

We will begin in: 0:00

SILICON LABS | tech talks

Welcome

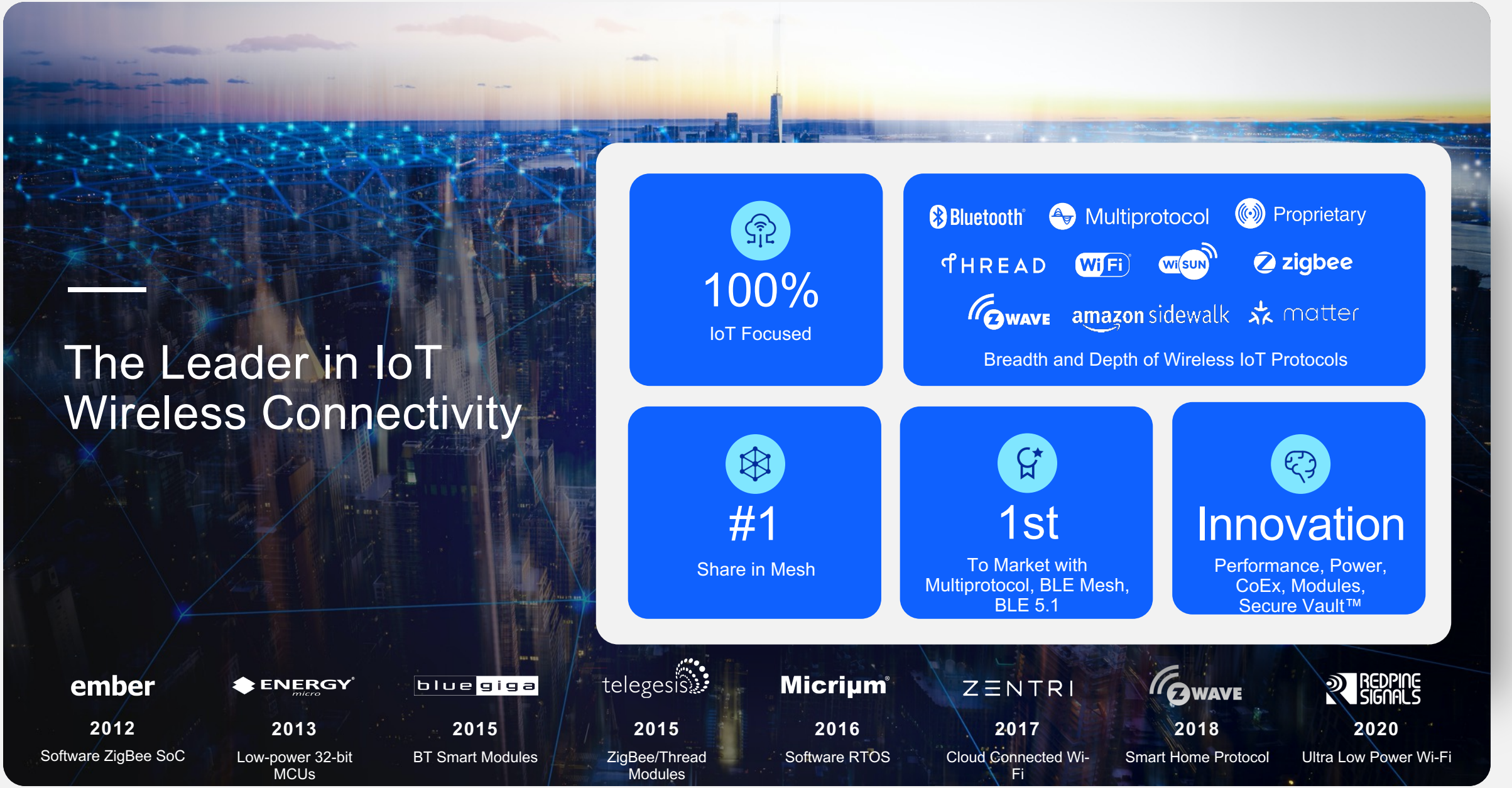
The Latest Bluetooth Low Energy Updates in GSDK 4.0

Kris Young

Agenda

- **Silicon Labs BT Introduction**
- **Latest GSDK updates**
  - Stack updates
    - Bluetooth 5.3 Qualification
    - New Advertiser API
  - SDK updates
    - Exporting NCP Host Examples
    - Dynamic GATT Support
    - PyBGAPI Examples
    - HCI over CPC
    - Apple Find My
  - Tooling updates
    - NCP Commander New Features (Dynamic GATT, Bluetooth Mesh)
    - Direction Finding Tool Suite
- **Demonstration: Using Bluetooth GATT Configurator GUI for NCP Host projects**

The Leader in IoT Wireless Connectivity



100%
IoT Focused

Bluetooth · Multiprotocol · Proprietary
THREAD · Wi Fi · Wi SUN · zigbee
Z-WAVE · amazon sidewalk · matter
Breadth and Depth of Wireless IoT Protocols

#1
Share in Mesh

1st
To Market with Multiprotocol, BLE Mesh, BLE 5.1

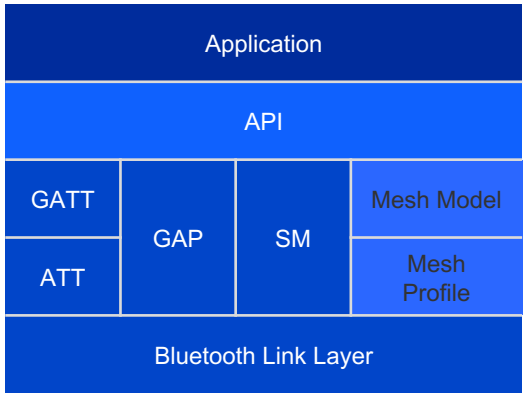
Innovation
Performance, Power, CoEx, Modules, Secure Vault™

ember	ENERGY micro	bluegiga	telegesis	Micrium	ZENTRI	Z-WAVE	REDPINE SIGNALS
2012	2013	2015	2015	2016	2017	2018	2020
Software ZigBee SoC	Low-power 32-bit MCUs	BT Smart Modules	ZigBee/Thread Modules	Software RTOS	Cloud Connected Wi-Fi	Smart Home Protocol	Ultra Low Power Wi-Fi

A Complete Solution for Enabling Bluetooth Products

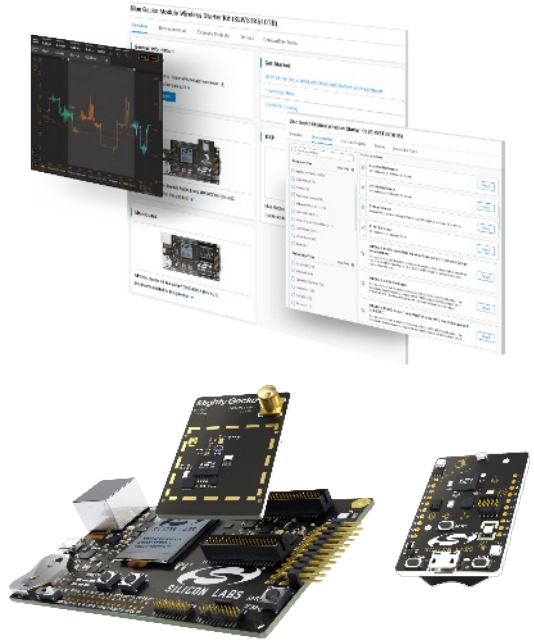
SoCS AND MODULES

Industry leading Bluetooth 5.1, 5.2 and 5.3 SoCs and pre-certified modules



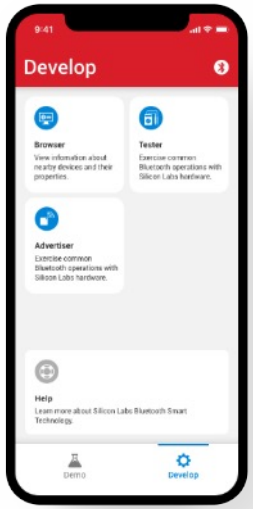
STACK SOFTWARE

In-house developed stacks with latest Bluetooth 5.3 and Bluetooth mesh features



DEVELOPMENT TOOLS

Advanced development hardware and software simplify development and speed time to market

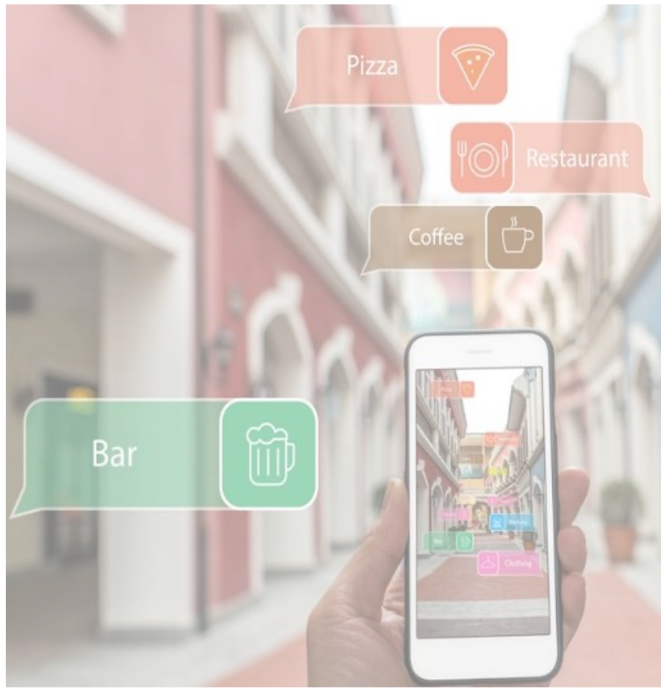


MOBILE APPLICATIONS

Reference applications and source code for iOS and Android

Phone interoperability test program

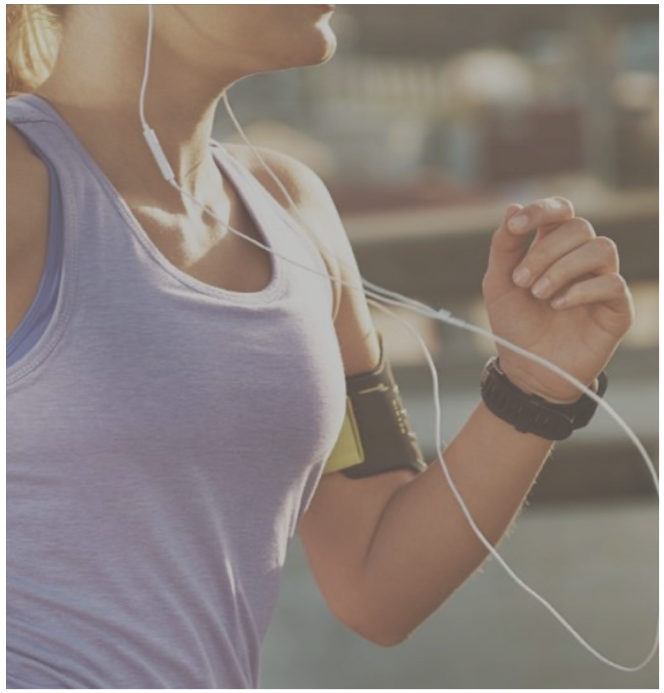
Supported Bluetooth Topologies



BEACONING

iBeacon, EddyStone and other beacon formats supported

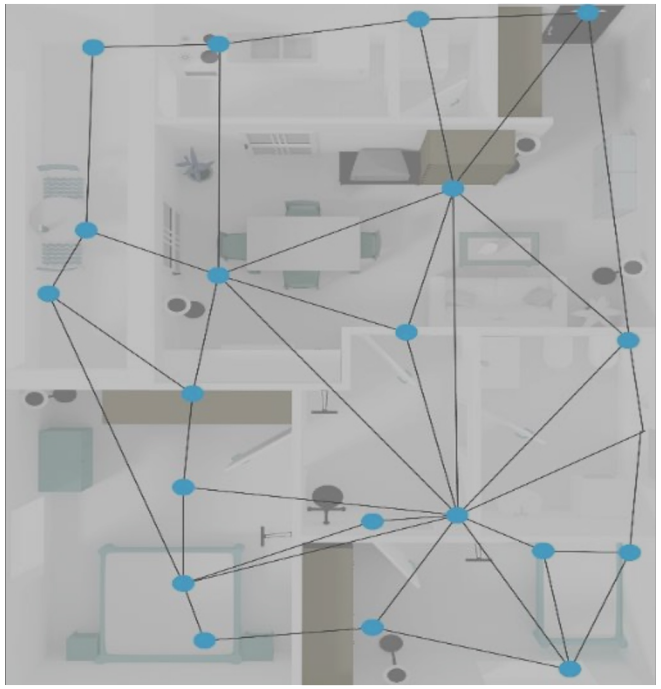
Support for advanced Bluetooth 5 beacon features



POINT-TO-POINT AND STAR

Bluetooth peripheral and central modes up to 32 connections and dual topology

Simultaneous peripheral and central operation

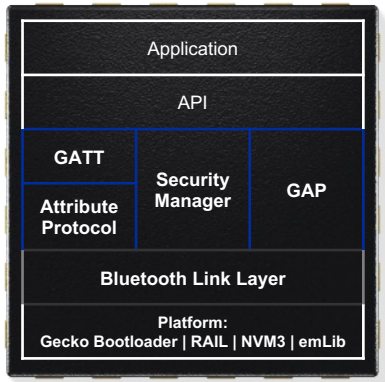


MESH

Bluetooth mesh for large device networks and many-to-many communications

Simultaneous Bluetooth LE and mesh use

Bluetooth LE and Bluetooth Mesh Software



A Bluetooth 5.3 compliant Bluetooth stack, with:

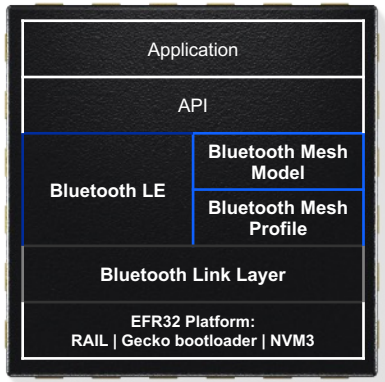
- Bluetooth 5.2 Dynamic TX power control
- Bluetooth 5.1 Direction Finding
- Bluetooth 5.0 standard features
- Bluetooth 4.x features

Packed with advanced functionality

- Multiple connections and advertisers
- Concurrent advertising, scanning and LE connections
- Optimized throughput and power consumption

Built on top of the common EFR32 software platform

- Gecko bootloader
- emLib for MCU peripherals and drivers
- NVM3 key/value pair data storage with wear leveling
- RAIL radio driver



A complete Bluetooth mesh profile, supporting:

- Proxy, relaying and friend nodes
- Bluetooth mesh low power nodes (LPN)
- Low latency communications down to 10ms per hop
- Large network support - up to 4096 nodes

A comprehensive Mesh Model application layer, with:

- Lighting models for On/Off, Dimming & color temperature
- Occupancy based lighting for commercial applications
- Scene, Sensor, Generic and Vendor models

Bluetooth LE support includes

- Beaconing for indoor positioning systems
- Scanning for asset tracking
- Phone connectivity
- Energy harvesting light switches

Bluetooth Stack Updates

Bluetooth Stack | Bluetooth 5.3 Qualification

- The Bluetooth stack is now **qualified against the Bluetooth 5.3** specification (released in July 2021)
- Bluetooth 5.3 has the following improvements
  - Periodic Advertising Enhancement
  - Connection Subrating
  - Channel Classification Enhancement
- These are **not yet implemented** in our stack

Bluetooth LE – Supported Features by Device

	Feature	xG24	xG22	xG21	xG13	xG12	xG1
Bluetooth 5.3	Connection Sub-rating						
	Periodic Advertising Enhancement						
	Channel Classification Enhancements						
Bluetooth 5.2	LE Audio (multiple subfeatures)						
	Dynamic TX power control	✓	✓				
	Enhanced ATT						
Bluetooth 5.1	Direction Finding (AoA and AoD)	✓	✓				
	GATT caching	✓	✓	✓	✓	✓	✓
	Adv. Channel Index Change						
	Periodic Adv. Sync Transfer						
	Control Length Extension						
Bluetooth 5.0	Higher Output Power	✓		✓	✓	✓	✓
	2M PHY	✓	✓	✓	✓	✓	
	LE Long Range	✓	✓	✓	✓		
	LE Advertising Extensions	✓	✓	✓	✓	✓	
	LE Periodic Advertising	✓	✓	✓	✓	✓	
	LE CSA#2	✓	✓	✓	✓	✓	✓
Bluetooth 4.2	LE Data Packet Length Extensions	✓	✓	✓	✓	✓	✓
	LE Privacy 1.2 (peripheral)	✓	✓	✓	✓	✓	✓
	LE Secure Connections	✓	✓	✓	✓	✓	✓
Bluetooth 4.1	LE Link Layer Topolgy	✓	✓	✓	✓	✓	✓

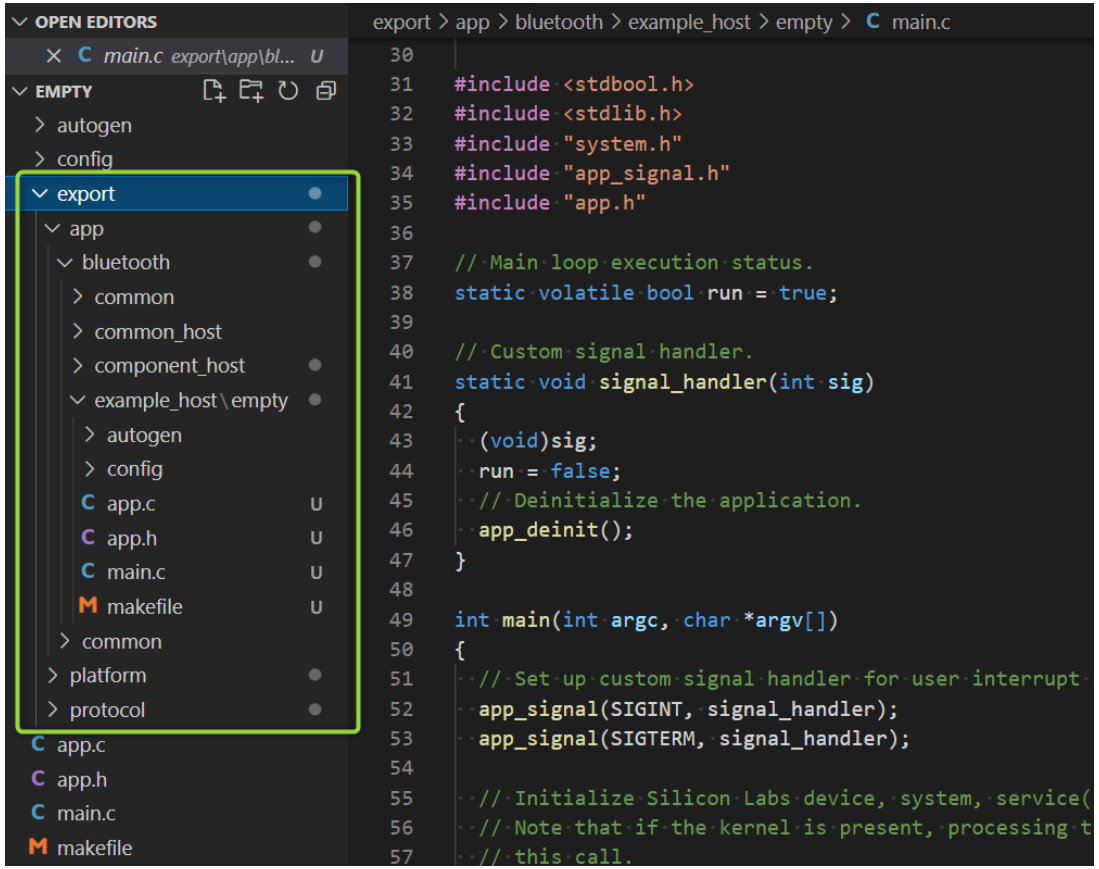
Bluetooth Stack | New Advertiser API

- In Bluetooth **SDK v3.2** there is only **one API class** for all advertisement commands
- In Bluetooth **SDK v3.3** advertiser APIs are split into **three classes**
  - **Legacy advertiser**
  - **Extended advertiser**
  - **Periodic advertiser**
  - (there will be even more in future)
- The old **Advertiser class** is still in use but is now **deprecated**.
  - Existing applications to be migrated to the new API
- New API: **sl_bt_xxx_advertiser_generate_data()**
  - Generates data from device name, TX power, advertised services, etc.
  - Not automatic anymore! (except if the old API class is used)

Bluetooth SDK Updates

Bluetooth SDK | Exporting Host Examples

- **SDK v3.2:**
  - Host applications must be **built in SDK folder due to relative paths**
  - No easy way to build without the SDK folders
- **SDK v.3.3:**
  - '**make export**' generates a folder that copies all the SDK content needed for the sample app in one folder
  - **Export folder can be moved** anywhere
  - Similar to Studio workflow where relevant files are copied from the SDK folder to the project
  - OS dependent export copies files only for the given OS:
    - **make export OS=win**
    - **make export OS=posix**



Bluetooth SDK | Dynamic GATT Support

- **Dynamic GATT is preferred in NCP host application**
- **Use “NCP” project instead of “NCP – empty”**
- **SDK v3.2**
  - To use the Dynamic GATT database feature, one must use the dynamic GATT API in the application
  - E.g. *sl_bt_gattdb_add_uuid16_characteristic(session, service, 2, 0, 0, '002a', 2, 12, ‘example char')*
- **SDK v3.3**
  - **gatt_db.c/.h is interpreted** by the application and the corresponding dynamic **GATT APIs are automatically called**
  - gatt_db.c/.h can be **generated from the .btconf file**, which can be edited by the **GATT Configurator**
  - To generate gatt_db.c/.h outside of studio, use `` `make gattdb` ``
    - Needs python and jinja2 package!!

Bluetooth SDK | PyBGAPI Examples

- PyBGAPI development can now be started with sample apps
- https://github.com/SiliconLabs/pybgapi-examples
  - Provides basic examples for PyBGAPI
  - Includes latest .xapi file
  - Implements BluetoothApp class
    - Implements init and event query code
    - **Application extends BluetoothApp class**

```
class App(BluetoothApp):
    """ Application derived from generic BluetoothApp. """
    def event_handler(self, evt):
        """ Override default event handler of the parent class. """
        # This event indicates the device has started and the radio is ready.
        # Do not call any stack command before receiving this boot event!
        if evt == "bt_evt_system_boot":
            self.adv_handle = None
            self.gattdb_init()
            self.adv_start()
```

Bluetooth SDK | PyBGAPI Examples

- **PyBGAPI can now be used also with Bluetooth Mesh**
  - Provides btmesh_empty example
  - Includes also .xapi file for the Bluetooth Mesh
  - Implements BtMeshApp class
    - Implements init and event query code with internal event handler
    - Uses both BLE and BT Mesh apis
    - Handles BLE and BT Mesh events in the same handler unlike the C application example
  - Application extends BtMeshApp class
    - Only the user event handler is implemented as on example

```
class App(BtMeshApp):
    """ Application derived from BtMeshApp. """
    def event_handler(self, evt):
        """ Override default event handler of the parent class. """
        # This event indicates the device has started and the radio is ready.
        # Do not call any stack command before receiving this boot event!
        if evt == "btmesh_evt_node_initialized":
            if not evt.provisioned:
                self.lib.btmesh.node.start_unprov_beaconing(PB_ADV | PB_GATT)
```

Bluetooth SDK | PyBGAPI Use Cases

Produce easily portable host applications

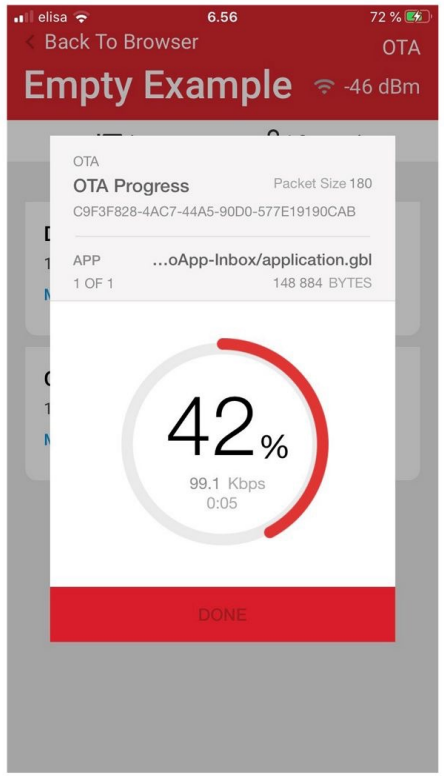
- Can run the same application without modification on any platform that supports Python 3 (Mac, PC, Raspberry Pi, etc.)

Mobile app emulation

- Use python to quickly implement a client with similar functionality to an intended mobile app
- Allows decoupling of device firmware development from mobile app software development

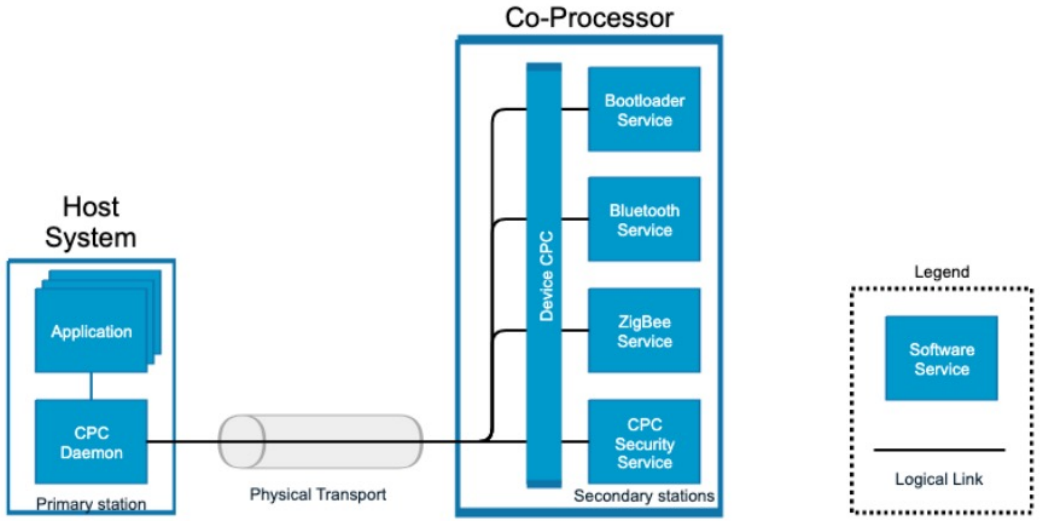
Automated hardware/software testing

- Python can be used to create a test application (either automated or interactive) to connect to your Bluetooth device (or mesh network) and exercise all of its features



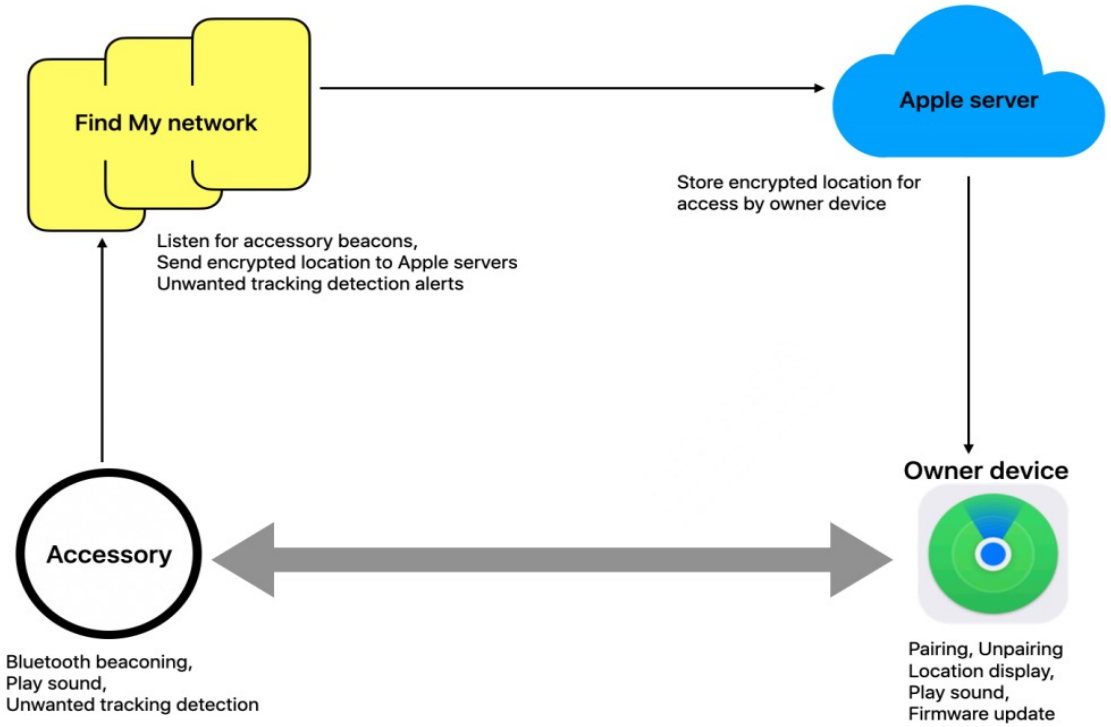
Bluetooth SDK | HCI over CPC

- **CPC** (Co-Processor Communication) is a **common transport protocol** for all NCP/RCP communication
  - It **tunnels** any kind of data for different protocol stacks / services
  - Multiple stacks / services can communicate over the **same UART interface at the same time**
- On the host side **CPC daemon dispatches the different packets** to different host applications
- Right now, it is important **in DMP (Dynamic Multi-Protocol) use cases to tunnel multiple interfaces (e.g. HCI)** via a common channel



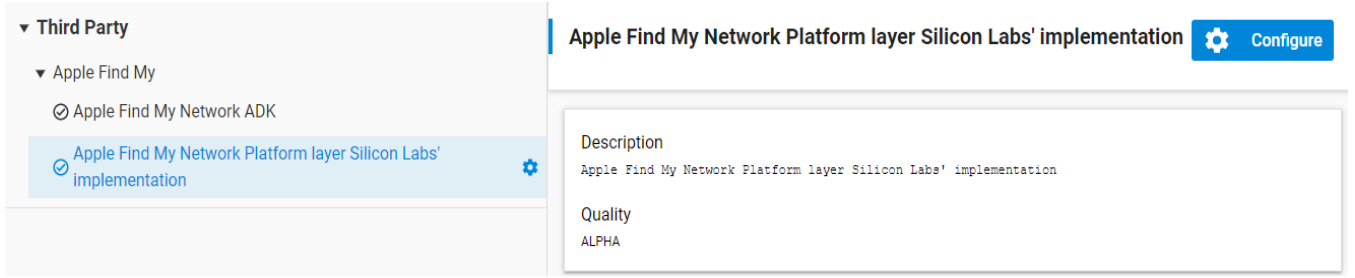
Bluetooth SDK | Apple Find My

- Find My Network is a service provided by Apple that **helps to track and locate compatible accessories** with the application called Find My App
- **Any iOS device can pick up** the signal of the accessory and **communicate its location secretly** (end-to-end encryption) to the owner
  - This needs **strong security** and application level **pairing**
- Most of the time it serves only as a **beacon**
- Accessories can implement optional / HW dependent features:
  - Playing sound → helps to find nearby
  - NFC
  - Motion detection → unwanted tracking



Bluetooth SDK | Apple Find My

- The specification is done by Apple and **MFI license is needed** to access it
- Customer must get **Authentication tokens from Apple** to be able to “pair” with the accessory
  - Configuration interface is only exposed after the **Apple pairing process**: without tokens it is not usable with Apple devices
- **SDK**:
  - **sample app / Software component** based on the ADK provided by Apple
  - MFI access **needs to be verified before exposing** to customer
  - Device compatibility: **all EFR devices with at least 48 kByte RAM**
  - **core functionality** is implemented, the optional features area provided as “weak” functions

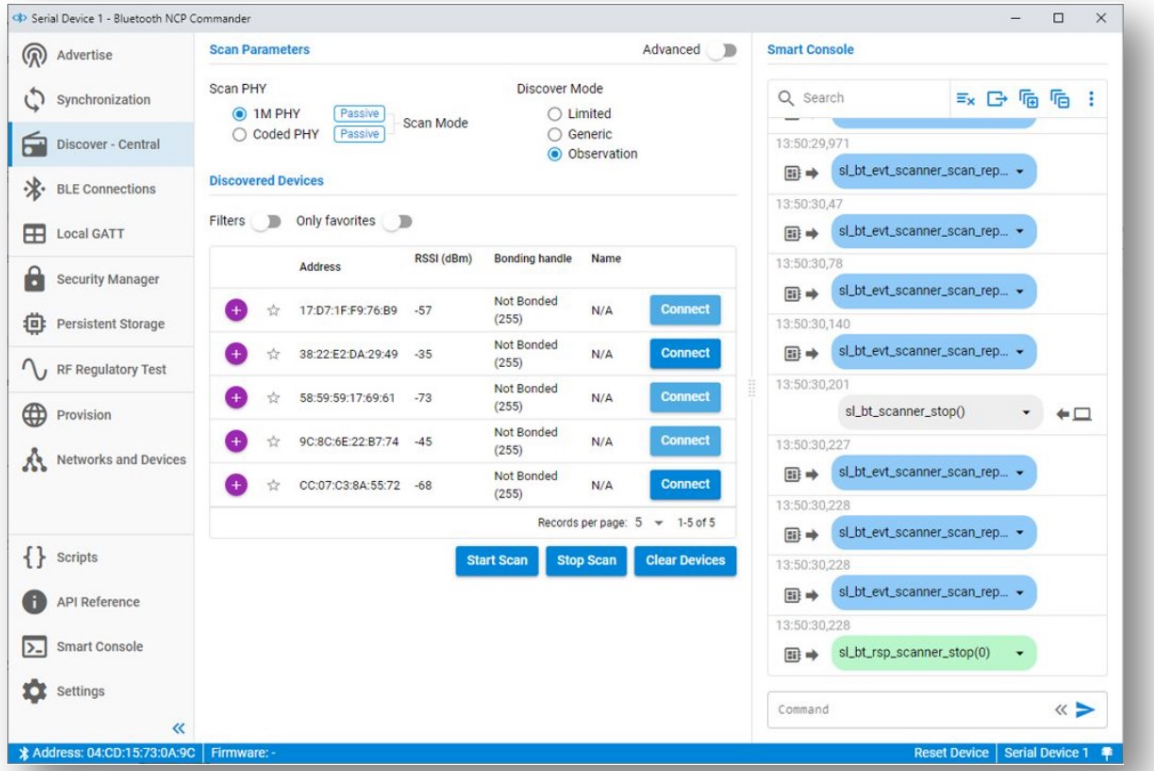


- **Tools:**
  - There is a “hardcoded pairing” mode which can be used for testing without the need of a token (note: it won’t work with any Apple devices). We have a PyBGAPI script which can activate it via an NCP target
  - Token_tool.py is a script to convert the token and UUID provided by Apple into an .s37 file

Tool Updates

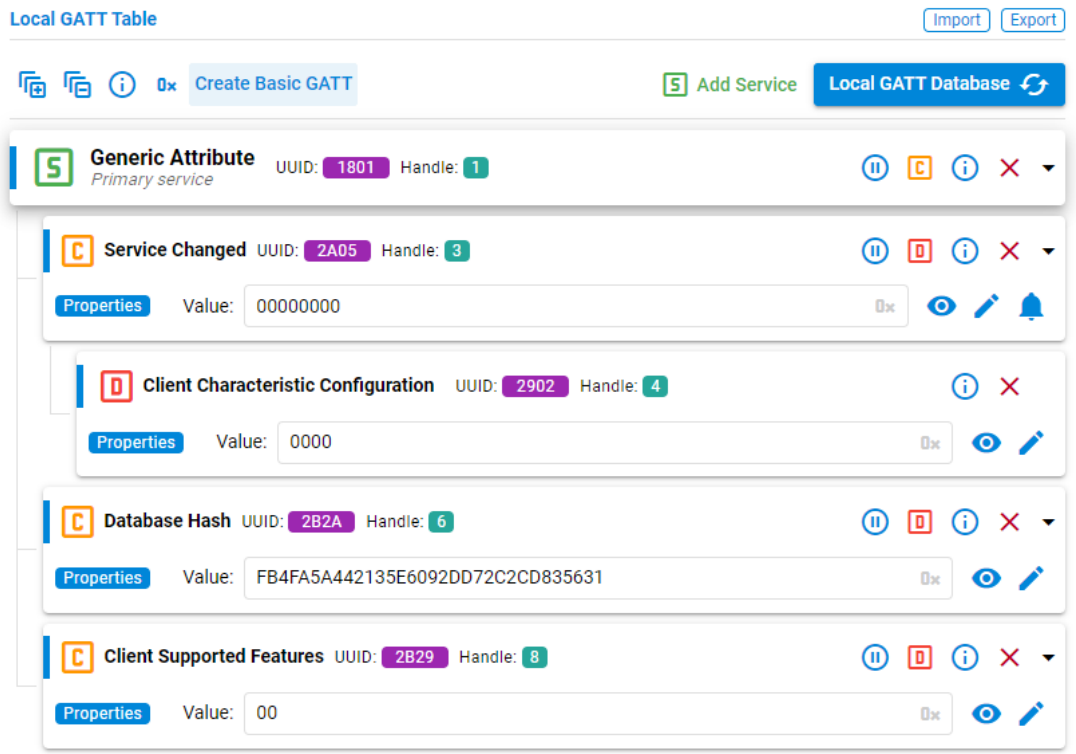
Tools | NCP Commander

- **Control the NCP (Network Co-Processor) target intuitively through a graphical user interface and learn the inner workings of the Bluetooth API**
- **Launch commands effortlessly through the smart console with built-in documentation and intellisense**
- **Perform the most common BLE functions (advertising, scanning, connections)**
- **Two versions:**
  - “Bluetooth NCP Commander” that is a Simplicity Studio integrated version
    - WSTK UART connection and also WSTK Ethernet VCOM
  - “Bluetooth NCP Commander Standalone” that is an independent application
    - Any UART interface (not just WSTK)
    - Specific BGAPI version files can be downloaded from Settings -> API Settings



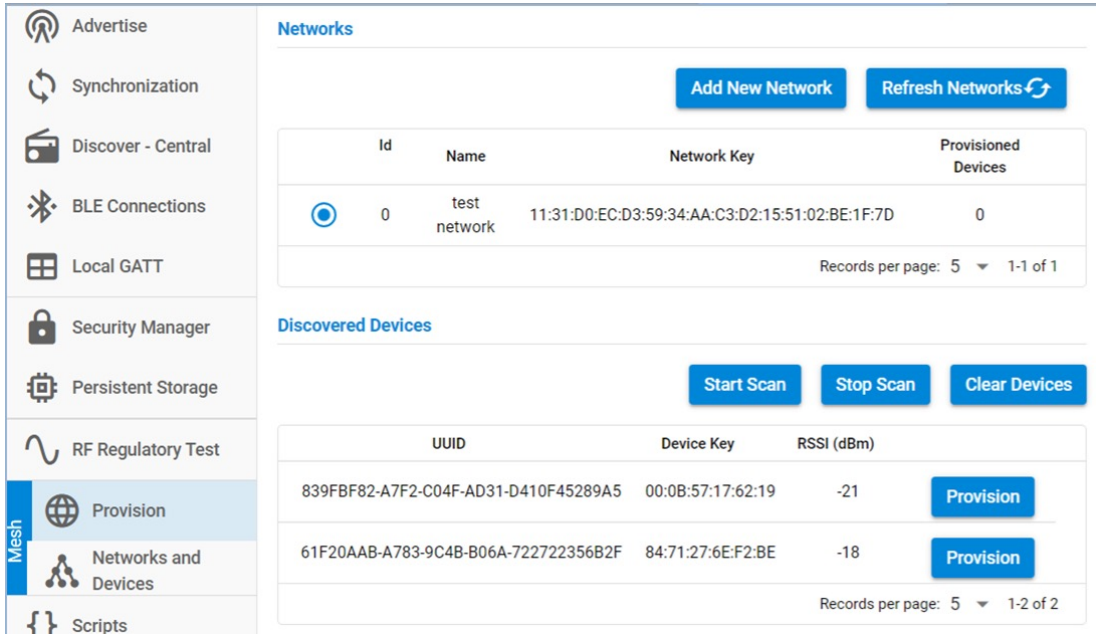
Tools | NCP Commander New Features

- **Periodic advertisements**
  - Periodic advertisements can be edited, configured and started on the GUI
- **Periodic advertisement synchronization**
  - GUI for synchronizations
- **Local GATT database read-out and creation via dynamic GATT API**



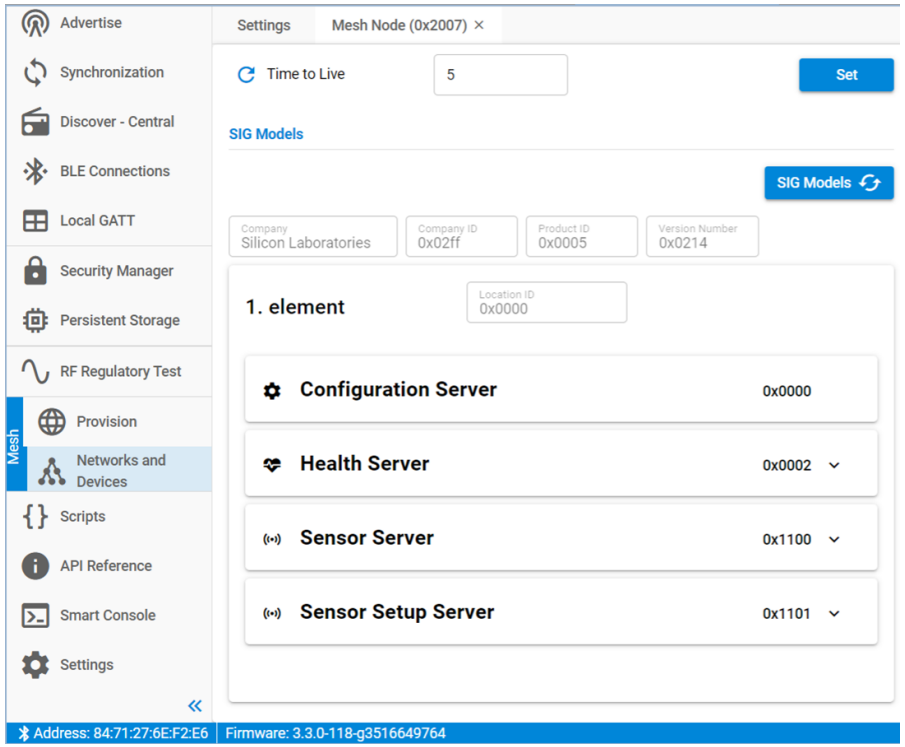
Tools | NCP Commander – New Bluetooth Mesh features

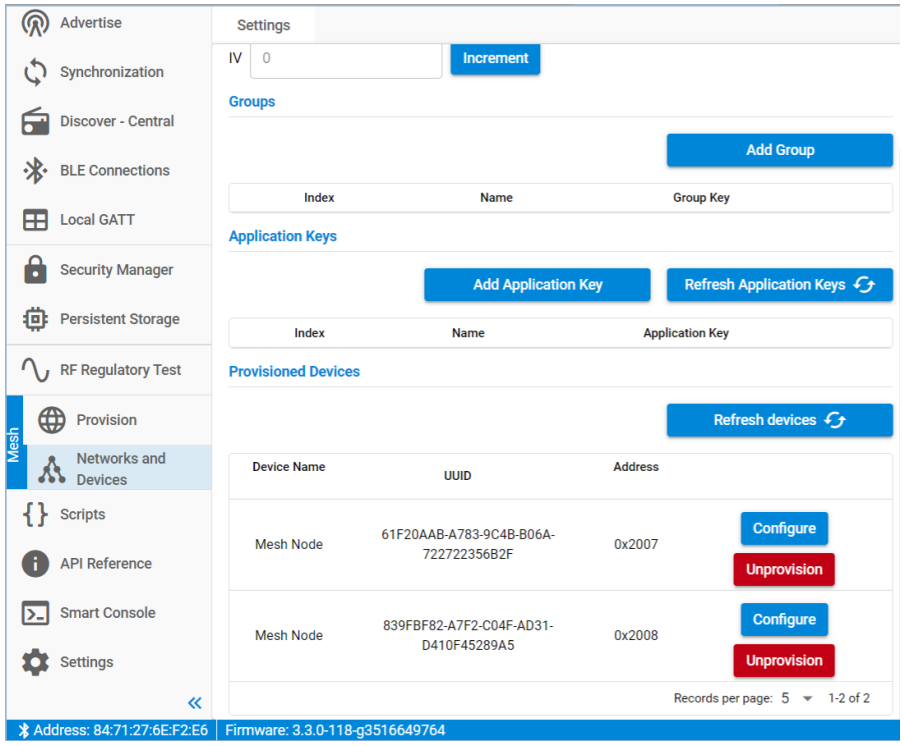
- Bluetooth NCP Commander supports now also Bluetooth Mesh features
- Flash the latest “Bluetooth Mesh – NCP Empty” –NCP target application into your device to use the features
- You can issue Bluetooth Mesh commands manually in the command box of Smart Console or use the host provisioner feature from the left menu
- You can use the feature to provision and configure mesh nodes and to manage mesh networks rather than using a Bluetooth Mesh mobile application



Tools | NCP Commander for Bluetooth Mesh

- Provision devices
- Configure models
- Manage keys (network, application, device)
- Manage groups

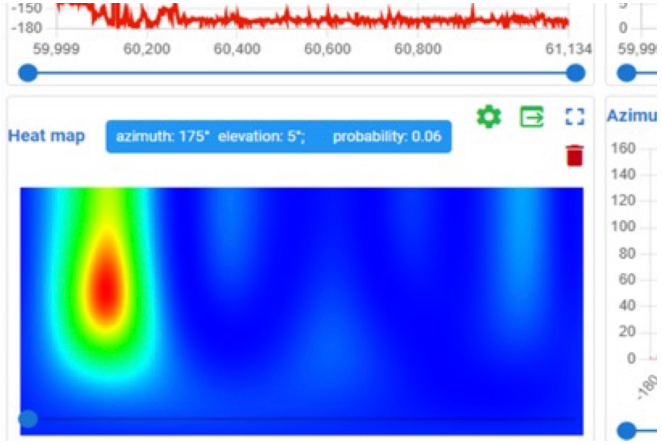


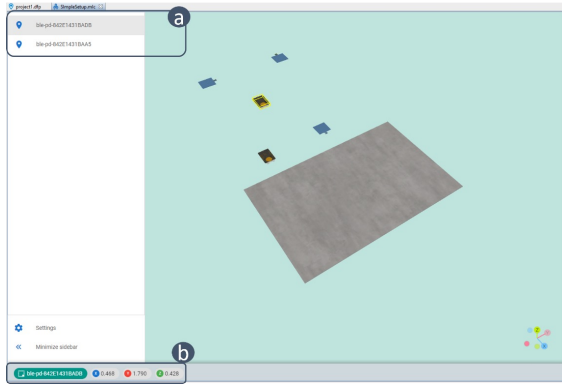


SILICON LABS

Tools | Direction Finding Tool suite

- **AoA Analyzer**
  - Multi-estimator support (enables comparison of performance)
  - Record/playback
  - Pseudo spectrum
- **Positioning Tool**
  - Demonstrates multi-locator position estimation in 3D
- **Direction Finding Projects**
  - Stores configurations for locators and multi-locators with topology





SILICON LABS

Demonstration: Using Bluetooth GATT Configurator GUI for NCP Host projects

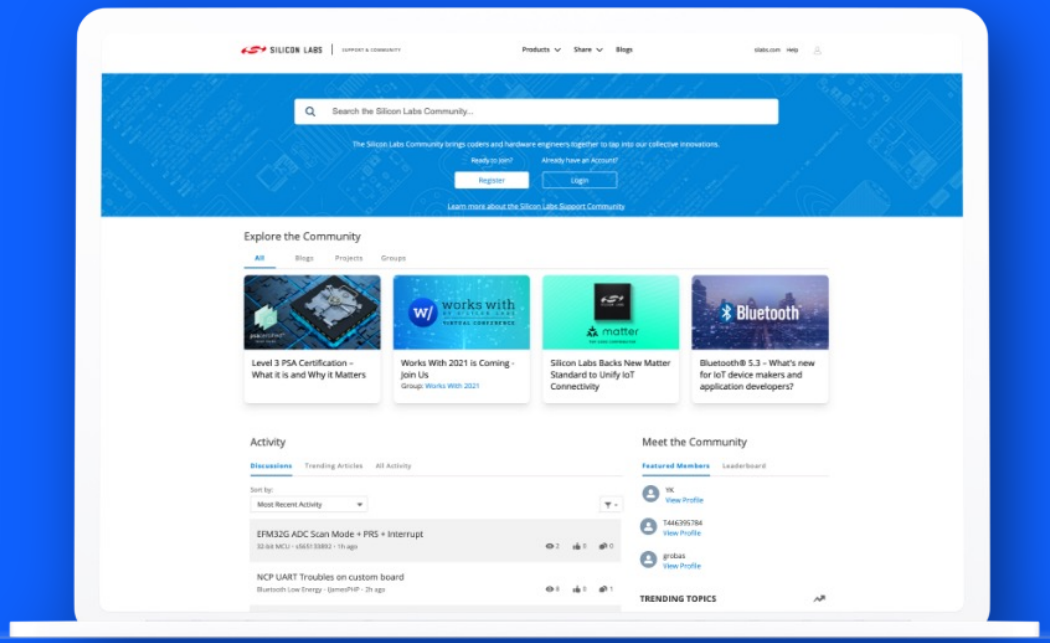
References

- Bluetooth NCP Commander User Guide: https://docs.silabs.com/simplicity-studio-5-users-guide/latest/ss-5-users-guide-tools-bluetooth-ncp-commander/
- Documentation on Bluetooth Direction Finding Tool Suite: https://docs.silabs.com/simplicity-studio-5-users-guide/latest/ss-5-users-guide-direction-finding-tools/
- AN1259: Using the v3.x Silicon Labs Bluetooth Stack in Network Co-Processor Mode https://www.silabs.com/documents/public/application-notes/an1259-bt-ncp-mode-sdk-v3x.pdf


SILICON LABS
tech talks
Thank You


SILICON LABS
tech talks
Q&A

Continue Discussion in Our Community!



How to Navigate:

- “Products” to troubleshooting forums
- “Applications” to discuss IoT
- “Share” to view example projects and existing groups
- “Blogs” to view and discuss thoughts from our specialists

community.silabs.com

tech talks

WEBINAR

Developing with Matter on the MG24

MAY 3 | 10AM CDT