Tech Talks LIVE Schedule – Presentation will begin shortly

Wireless Connectivity Tech Talks

Thursday, March 25th	Unboxing the BGM220 Explorer Kit
Wednesday, April 28th	**Uncover Sub-GHz and Proprietary Solution within Simplicity Studio v5**

Recording and slides will be posted to:
www.silabs.com/training

We will begin in 3:00

Speaker

노상영 (Young Noh)

Sr. Staff FAE, Korea

SILICON LABS

tech talks

WELCOME

Uncover Sub-GHz and Proprietary Solution within Simplicity Studio v5

Young Noh

Agenda

- High Level Overview
  - What is a Proprietary Wireless Application?
  - FLEX SDK: RAIL vs. Connect
- Silicon Labs Flex Gecko SoCs
- Proprietary Project in Simplicity Studio v5
- Demo: RAIL Application in Simplicity Studio v5

High Level Overview of EFR32 Proprietary Wireless

Typical Proprietary Wireless Solutions

- Smart Meters

- Home Automation and Security

- Garage Door Openers

- Public Infrastructure

- Agriculture

- Asset Tracking & ESL

When is Proprietary Wireless Appropriate?

When the application demands:

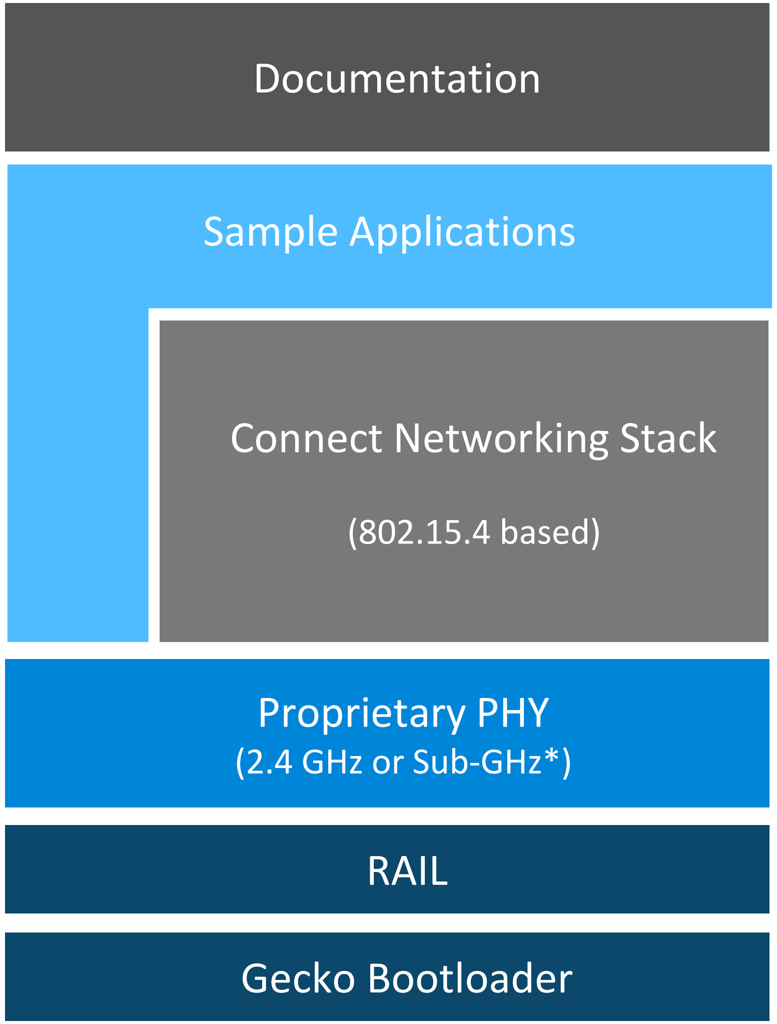
- + Backwards-compatibility with existing/legacy proprietary protocol(s)
- + High degree of protocol optimization
  - + For energy consumption
  - + For wireless range
  - + https://www.silabs.com/support/training/long-range-connectivity-using-proprietary-rf-solution
  - + https://www.silabs.com/support/training/sub-ghz-proprietary-and-connect-software-stack
- + Full control over the protocol

…at the expense of:

- – More difficult development, longer “time to market”
- – Incompatibility with existing/future non-proprietary infrastructures
- – Security holes that can remain hidden for a long time due to the difficulty of the analysis
  - – But once discovered, exploiting them is usually easy (high obfuscation, not necessarily high security)

FLEX SDK

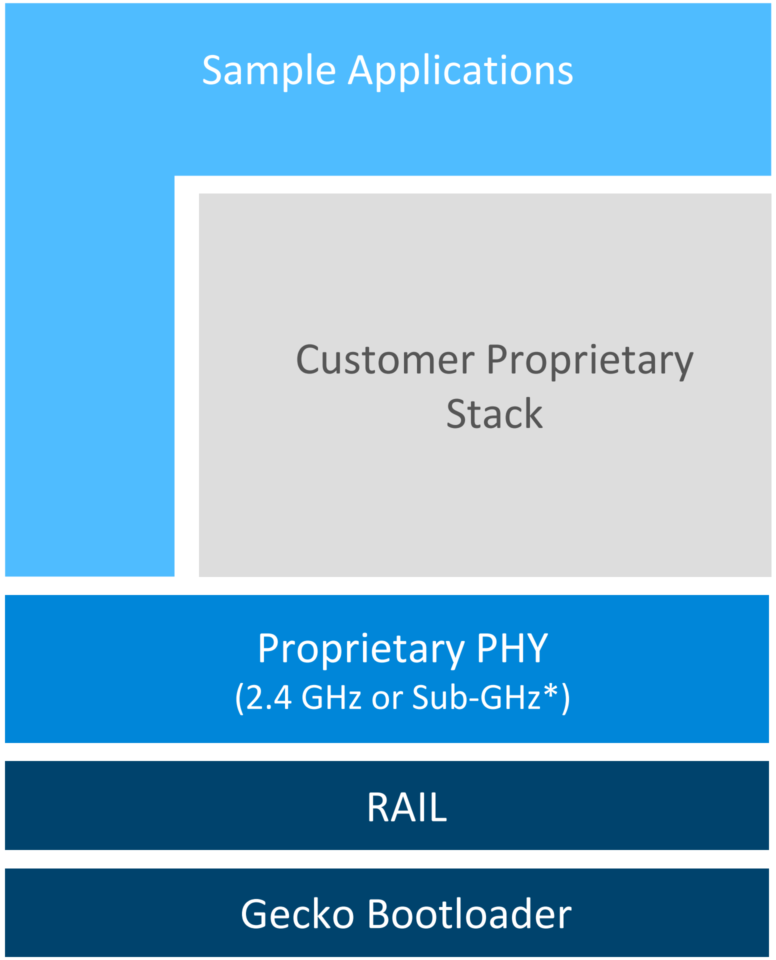
- Complete software development suite for proprietary wireless applications
- Common underlying software architecture with other wireless solutions
- Flexible, easy-to-use
  - Radio Abstraction Interface Layer (RAIL)
  - Connect Networking Stack
  - Sample applications
  - Extensive documentation
- Available through Simplicity Studio
  - Integrated with application builder, radio configurator, network analyzer and energy profiler



* Wireless Gecko Series 1

RAIL Software

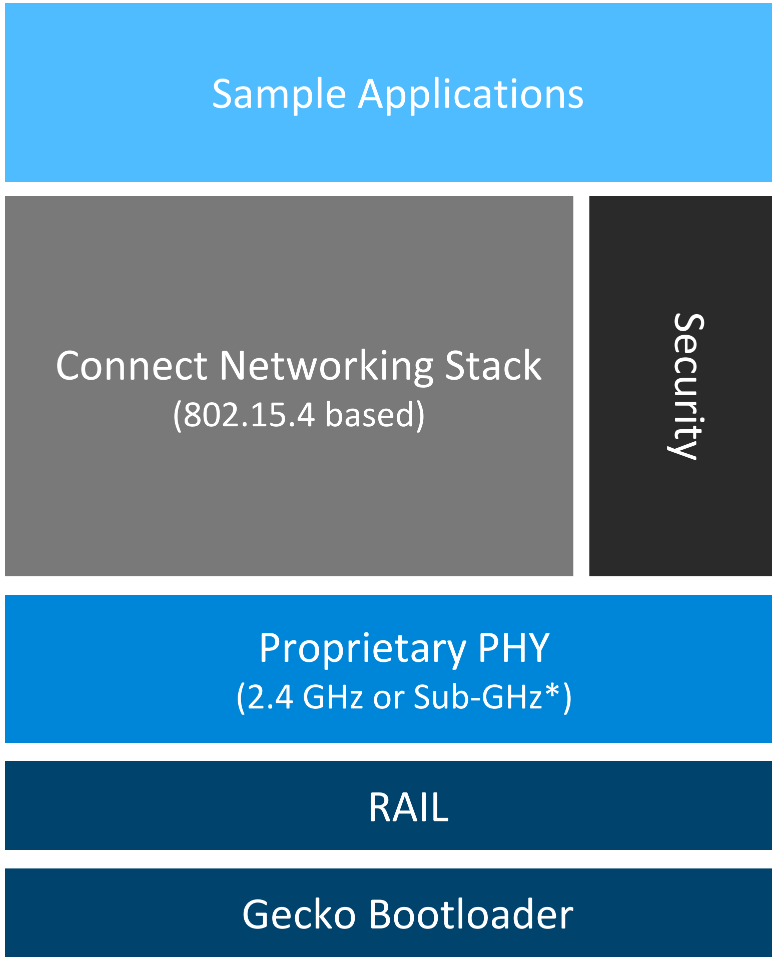
- Simplified Radio API
  - Common radio interface across SoCs
  - No need to learn complex low-level radio registers
- Lower development time
  - Easy migration of customer proprietary stack
  - Simplified radio testing
  - Quicker prototype boards bring up
- Complete software package
  - Integrated with Simplicity Studio via Flex SDK
  - Radio and RAIL test example applications
  - GCC and IAR toolchain support



* Wireless Gecko Series 1

Connect Networking Stack

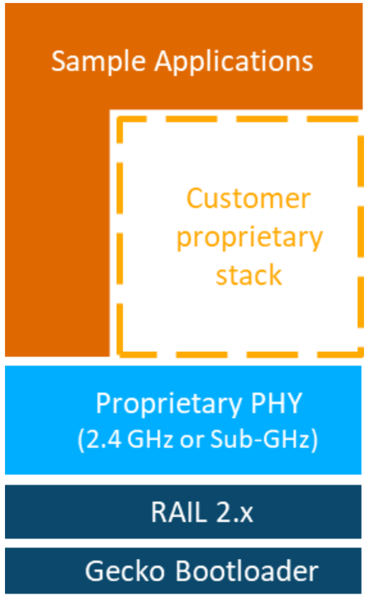
- Feature Rich Proprietary Wireless Networking Stack
  - Highly scalable
    - Up to 2K nodes in extended star mode
    - Up to 65K nodes in Direct Mode
  - Small stack footprint leaves room for application, OTA, etc.
  - Low power modes
  - Built-in security
- Faster Time To Market
  - Ready to use and customizable PHYs
  - Application Builder, Network Analyzer and Energy Profiler
  - GCC and IAR compiler support
  - Sample applications
- Field Upgradable via OTA
  - Over-the-air firmware updates (unicast & multicast)



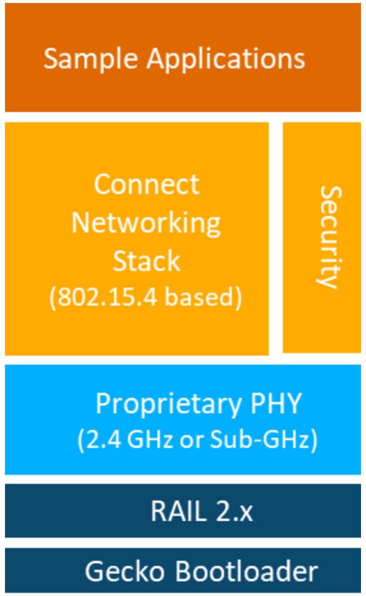
* Wireless Gecko Series 1

RAIL API vs Connect Stack?

- RAIL – What is it?
  - Radio API, examples, documentation and tools for PHY level access
- RAIL – When to use?
  - To develop low level PHY layer based applications
  - To port an existing wireless protocol to EFR32
  - To port your own networking stack to EFR32 on top of PHY layer
- RAIL – What does it require?
  - Medium to advanced PHY level wireless expertise


Sample Applications
Customer proprietary stack
Proprietary PHY (2.4 GHz or Sub-GHz)
RAIL 2.x
Gecko Bootloader

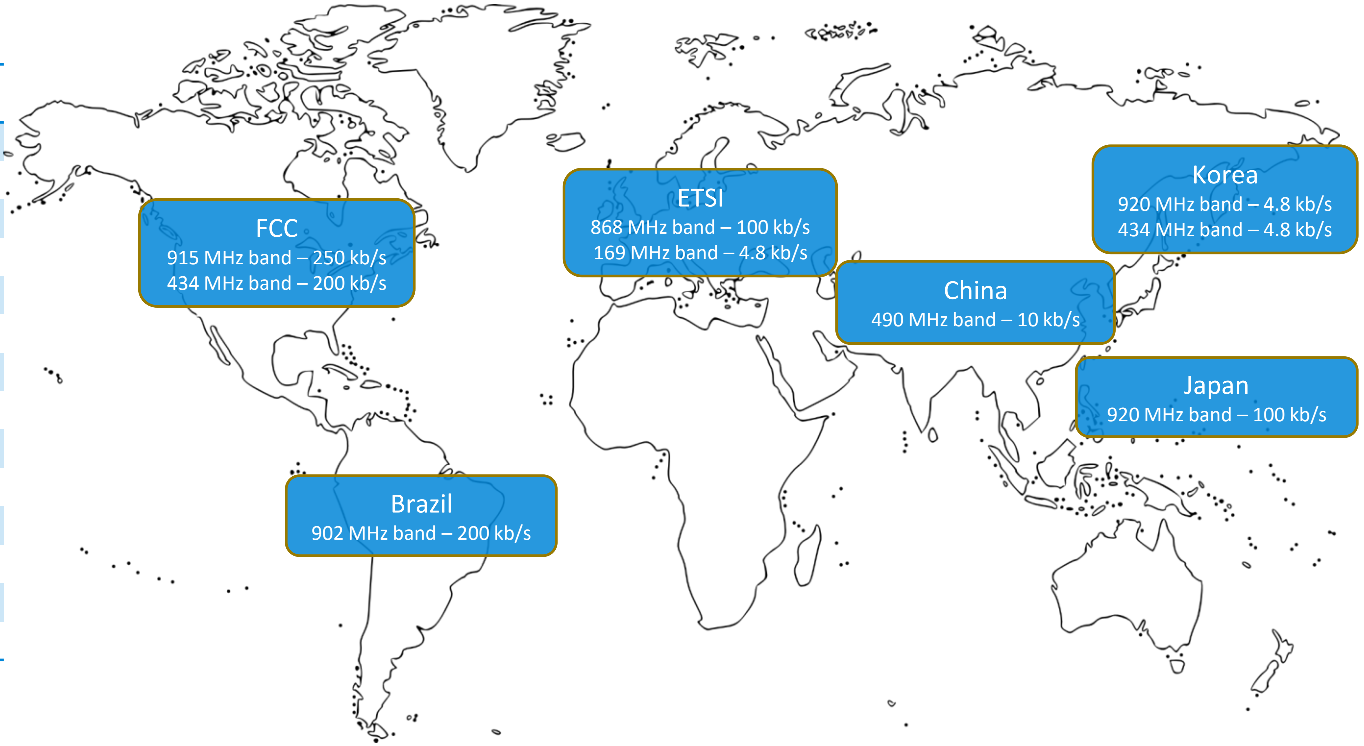
- Connect Stack– What is it?
  - Network layer library, examples, documentation and tools
- Connect – When to use?
  - You need star, extended star or direct mode network topology up to 2K devices
  - You need a network layer stack with low power support, security, FHSS and OTA
  - OR you want to implement your own network layer on top of a standard 802.15.4 MAC
- Connect – What does it require?
  - Basic to none wireless experience


Sample Applications
Connect Networking Stack (802.15.4 based)
Security
Proprietary PHY (2.4 GHz or Sub-GHz)
RAIL 2.x
Gecko Bootloader

Connect Networking Stack : Worldwide PHYs

- Regional standards compliant

PHY profile	Frequency	Modulation	Bitrate
2.4GHz 802.15.4	2.4 GHz	OQPSK	250 kbps
DSSS 915-100	915 MHz	OQPSK	100 kbps (0.8Mcps)
DSSS 915-250	915 MHz	OQPSK	250 kbps (2Mcps)
915-500	915 MHz	2GFSK	500 kbps
DSSS 915-500	915 MHz	OQPSK	500 kbps
China 490	490 MHz	2GFSK	10 kbps
Europe 169	169 MHz	2GFSK	4.8 kbps
Europe 868	863 MHz	2GFSK	100 kbps
Japan 915	920 MHz	2GFSK	100 kbps
Korea 424	424 MHz	2GFSK	4.8 kbps
Korea 447	447 MHz	2GFSK	4.8 kbps
Korea 915	917 MHz	2GFSK	4.8 kbps
US FCC 434	434 MHz	2GFSK	200 kbps
US FCC 902, Brazil 902	902 MHz	2GFSK	200 kbps



- Each RF configuration is tuned and tested by Silicon Labs for maximal performance
- Custom PHYs can be configured via Radio Configurator in Simplicity Studio

Silicon Labs Flex Gecko SoCs

Flex Gecko SoC Portfolio

Flex Gecko SoCs

- ARM® Cortex®-M4
- Multiband Proprietary
  (2.4 GHz, sub-GHz)
- Up to +20 dBm TX Power

EFR32FG12 - 1 MB / 256 kB
(QFN48, QFN68, BGA125)

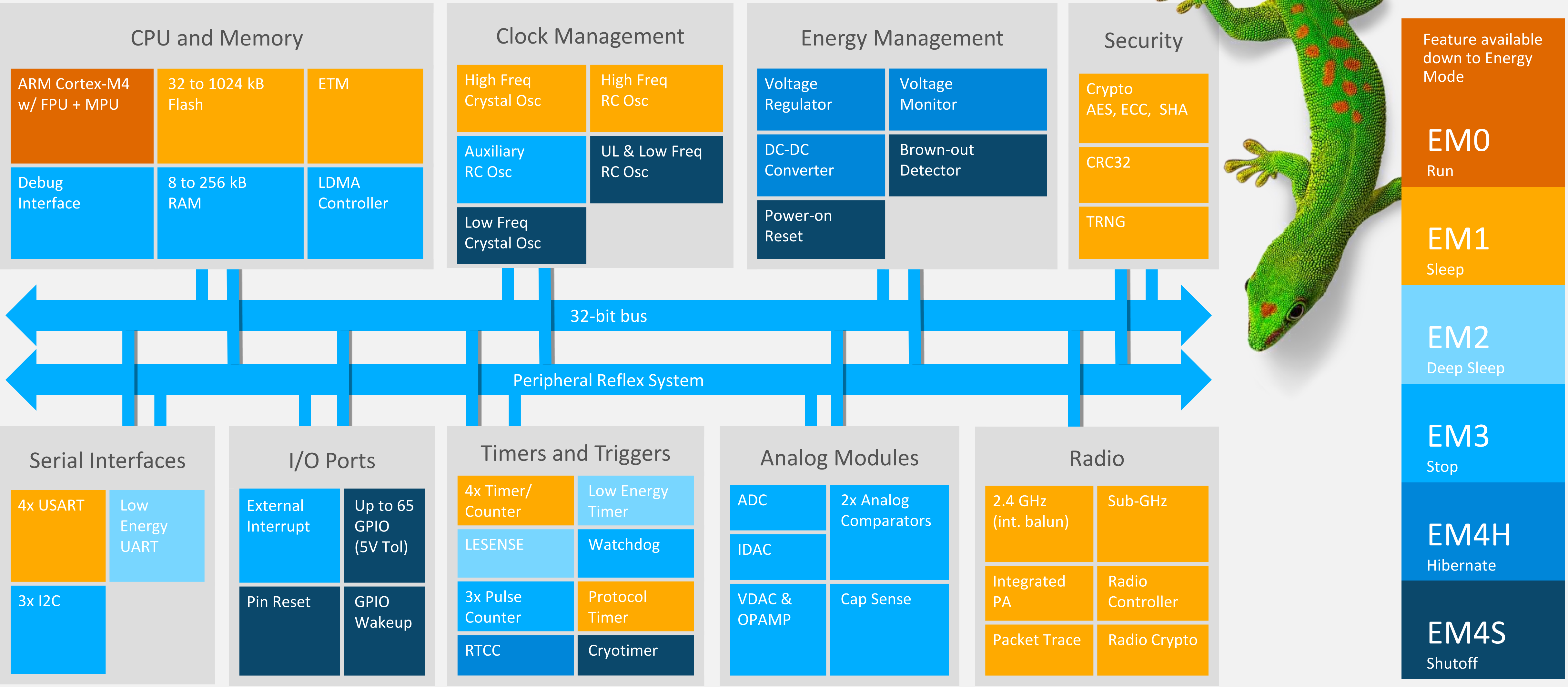
EFR32FG13 - 512 kB / 64 kB
(QFN48, QFN32)

EFR32FG1 - 256 kB / 32 kB
(QFN48, QFN32)

EFR32FG14 - 256 kB / 32 kB
(QFN48, QFN32)

- *Blue Gecko variants available for multiprotocol Bluetooth + Proprietary (2.4 GHz / Sub-GHz) use cases*
- *Flex SDK for 2.4GHz and Sub-GHz Proprietary wireless is* ***NOT supported*** *on MGM and BGM modules*

Flex Gecko Block Diagram (EFR32FG1x)



CPU and Memory
- ARM Cortex-M4 w/ FPU + MPU
- 32 to 1024 kB Flash
- ETM
- Debug Interface
- 8 to 256 kB RAM
- LDMA Controller

Clock Management
- High Freq Crystal Osc
- High Freq RC Osc
- Auxiliary RC Osc
- UL & Low Freq RC Osc
- Low Freq Crystal Osc

Energy Management
- Voltage Regulator
- Voltage Monitor
- DC-DC Converter
- Brown-out Detector
- Power-on Reset

Security
- Crypto AES, ECC, SHA
- CRC32
- TRNG

32-bit bus

Peripheral Reflex System

Serial Interfaces
- 4x USART
- Low Energy UART
- 3x I2C

I/O Ports
- External Interrupt
- Up to 65 GPIO (5V Tol)
- Pin Reset
- GPIO Wakeup

Timers and Triggers
- 4x Timer/Counter
- Low Energy Timer
- LESENSE
- Watchdog
- 3x Pulse Counter
- Protocol Timer
- RTCC
- Cryotimer

Analog Modules
- ADC
- 2x Analog Comparators
- IDAC
- VDAC & OPAMP
- Cap Sense

Radio
- 2.4 GHz (int. balun)
- Sub-GHz
- Integrated PA
- Radio Controller
- Packet Trace
- Radio Crypto

Feature available down to Energy Mode
- EM0 Run
- EM1 Sleep
- EM2 Deep Sleep
- EM3 Stop
- EM4H Hibernate
- EM4S Shutoff

Note: this is a superset diagram for the FG1x platform and not all features are supported on every SoC variant. Consult product data sheet for a detailed list of supported features.

FG22: Energy Efficient Proprietary Wireless

Optimized

Secure SoCs for Proprietary Wireless Applications

Radio

+6 dBm TX
2.4 GHz: GFSK, GMSK, OQPSK, DSSS
-101.2 dBm RX (2GFSK, 500 kbps)
-102.3 dBm RX (O-QPSK DSSS)

Ultra-Low Power

3.4 mA TX (radio)
2.5 mA RX (radio)
1.2 µA EM2 with 8 kB RAM
0.5 µA w/ RTC in EM4

World Class Software

Flex SDK
RAIL
Radio Configurator
Connect (802.15.4 based)

Compact Size

5x5 QFN40 (26 GPIO)
4x4 QFN32 (18 GPIO)

ARM Cortex-M33 with TrustZone

38.4 MHz
FPU and DSP
512 kB of flash
32kB RAM

Peripherals Fit for Purpose

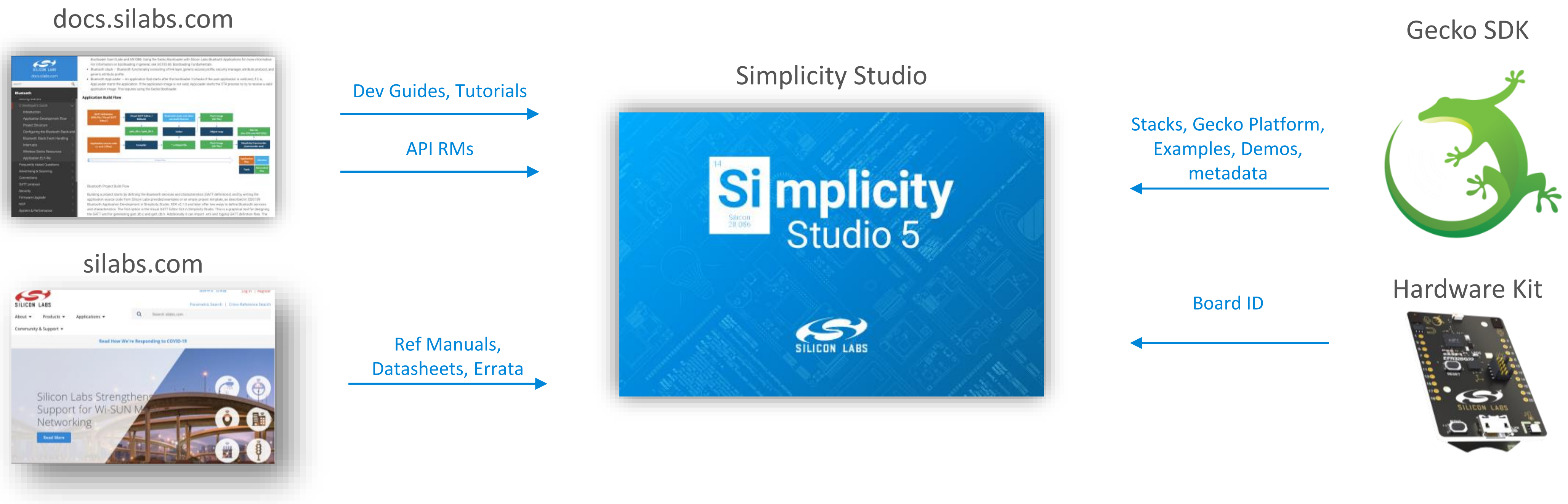
2x USART, 1x UART 2x I2C, PDM and GPIO
12-bit SAR ADC (1 Msps)
Built-in temperature sensor +/- 1.5 ^{o}C
RFSENSE with selective OOK mode

Security

AES128/256,SHA-1, SHA-2 (256-bit)
ECC (up to 256-bit), ECDSA and ECDH
True Random Number Generator (TRNG)
Secure boot with RTSL
Secure debug with lock/unlock

Starting a Proprietary Project in Simplicity Studio v5

Simplicity Studio v5



docs.silabs.com

silabs.com

Dev Guides, Tutorials

API RMs

Ref Manuals, Datasheets, Errata

Simplicity Studio

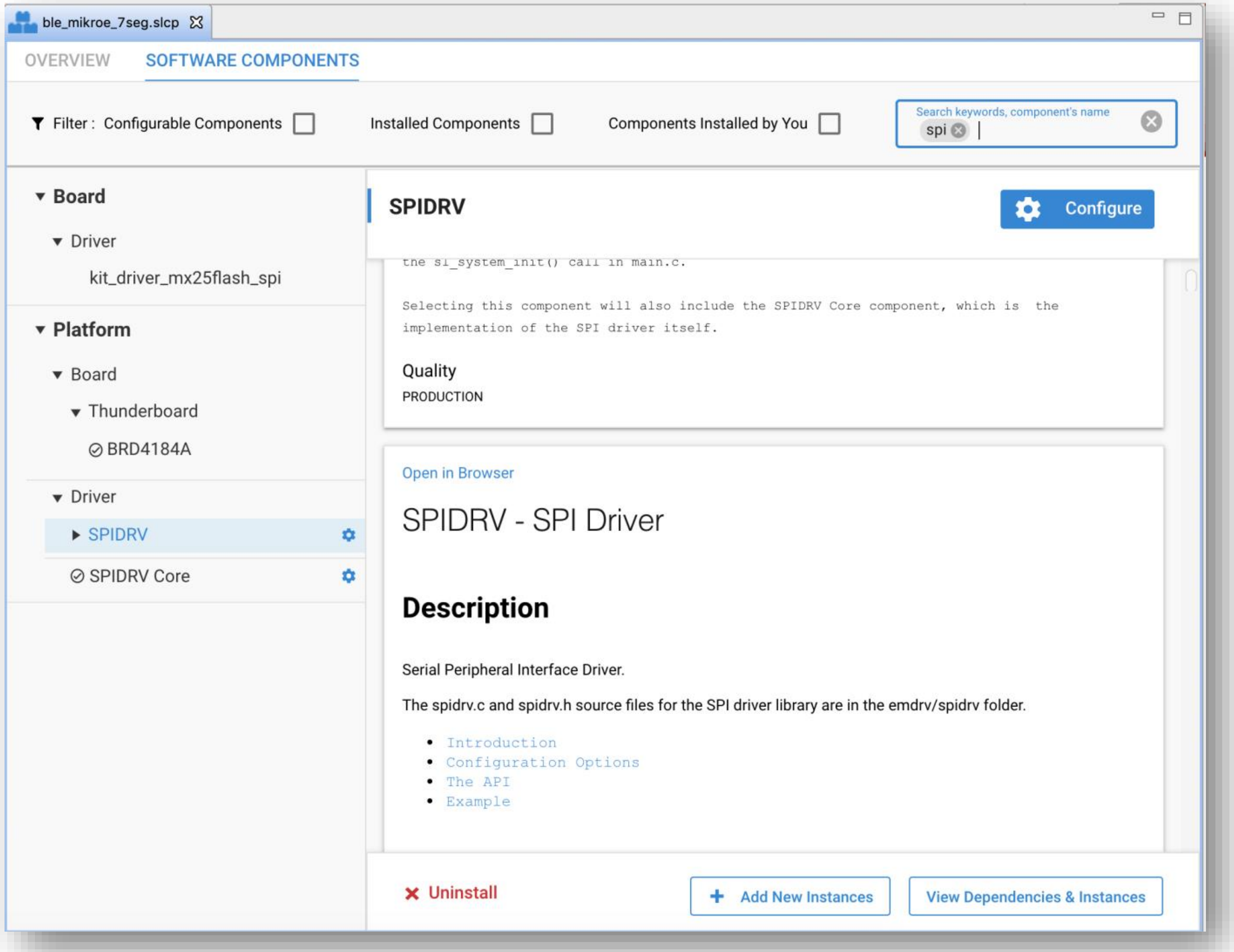
Stacks, Gecko Platform, Examples, Demos, metadata

Gecko SDK

Board ID

Hardware Kit

Simplicity Studio 5 – New Project Configuration Tools



ble_mikroe_7seg.slcp ☒

OVERVIEW SOFTWARE COMPONENTS

Filter : Configurable Components □ Installed Components □ Components Installed by You □

Search keywords, component's name: spi

- Board
  - Driver
    - kit_driver_mx25flash_spi
- Platform
  - Board
    - Thunderboard
      - BRD4184A
  - Driver
    - SPIDRV
    - SPIDRV Core

SPIDRV Configure

```
the sl_system_init() call in main.c.

Selecting this component will also include the SPIDRV Core component, which is  the
implementation of the SPI driver itself.
```

Quality

PRODUCTION

Open in Browser

SPIDRV - SPI Driver

Description

Serial Peripheral Interface Driver.

The spidrv.c and spidrv.h source files for the SPI driver library are in the emdrv/spidrv folder.

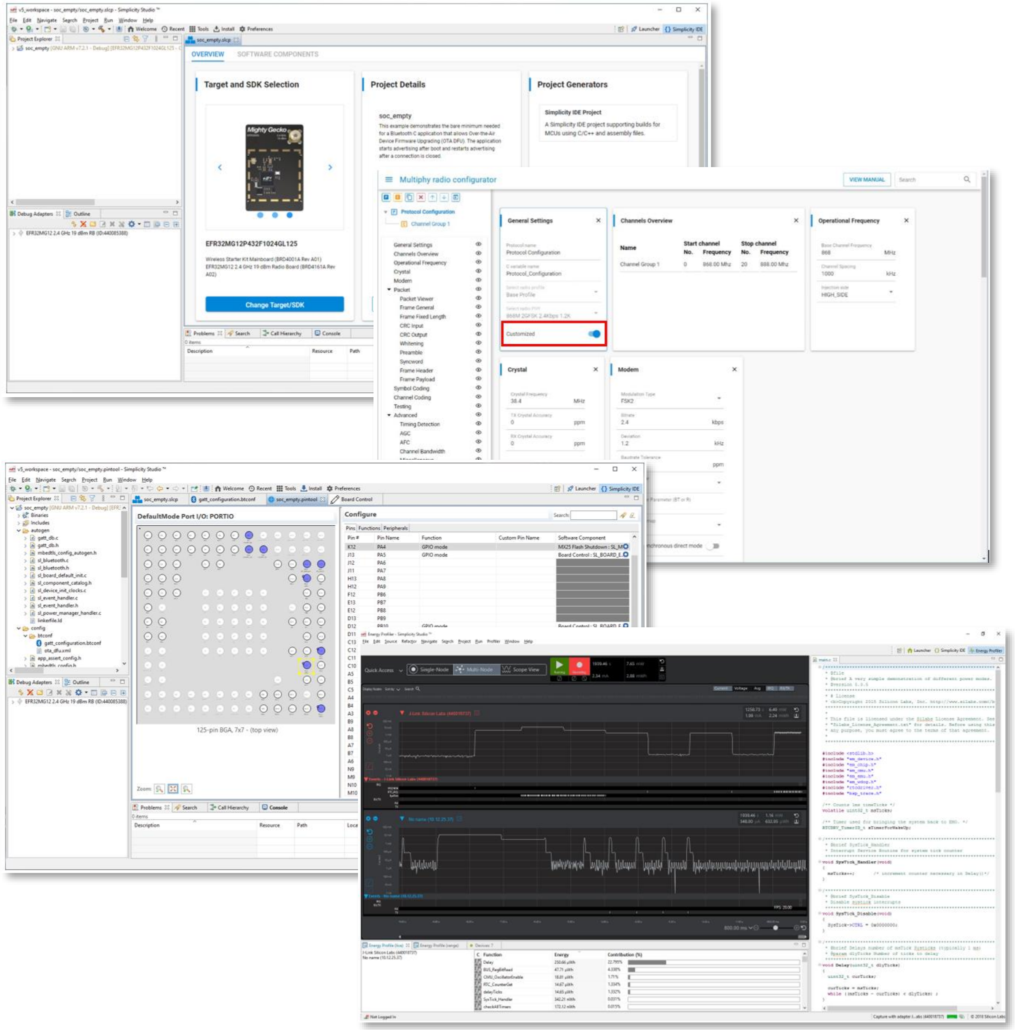
- Introduction
- Configuration Options
- The API
- Example

✕ Uninstall + Add New Instances View Dependencies & Instances

- Software component-based project configuration
  - Search and filter to discover and find software components
  - Automatically pull in dependencies and initialization code
  - All settings saved in source code (C header files)
  - Error checking and alerts
  - Easily manage all project source via git or other SCM tools
  - Managed migrations to future component and SDK versions
  - Simplified transition from Silicon Labs dev kits to custom HW
- Graphical pin configuration
- Redesigned Bluetooth Configurator
- Redesigned Radio Configurator
- 3rd party IDEs with support for iterative development
  - IAR Embedded Workbench
  - GNU makefiles as a build option

Tools and API

Tools



+

RAIL API

- Transmit/Receive
- Automatic State Transitions
  - E.g. automatically go to rx after tx
- Frame Buffering
  - Maintains buffer for both tx and rx
- Timekeeping, Timestamping and Timers
- Scheduled Transmit
- Scheduled Receive
- CCA with Retransmission
  - Supports CSMA/CA and LBT, but doesn’t support CCA without retransmission
- Address Filtering
  - With two fixed offset, max 4B address or 802.15.4 addressing
- Auto ACK
  - Preconfigured ACK packet automatically transmitted on every packet that passed all filtering or 802.15.4 ACK

Demo: RAIL Application in Simplicity Studio v5

Support Documentation

- Proprietary Flex SDK v3.x Quick Start Guide -- QSG168
- RAIL Fundamentals -- UG103.13
- Connect Fundamentals -- UG103.12
- Multiprotocol Fundamentals -- UG103.16
- Dynamic Multiprotocol User's Guide -- UG305
- Simplicity Studio® 5 User's Guide
- EFR32 Migration Guide for Proprietary Applications -- AN1244
- About the Connect v3.x User's Guide -- UG435.01
- Building Low Power Networks with the Silicon Labs Connect Stack v3.x -- AN1252
- Silicon Labs Connect API Reference Guide
- EFR32 Radio Configurator Guide for Simplicity Studio 5 -- AN1253
- RAILtest User's Guide -- UG409
- EFR32 RF Evaluation Guide -- AN972
- Silicon Labs RAIL API Reference Guide
- https://www.silabs.com/support/training/rail
- RAIL Tutorials

SILICON LABS

tech talks

Q&A

Facebook	Twitter	Community

SILICON LABS

tech talks

THANK YOU

Recording and slides will be posted to:
www.silabs.com/training