# Tech Talks LIVE Schedule – Presentation will begin shortly

| | |
|---|---|
| Friday, March 26th | Unboxing the BGM220 Explorer Kit |
| **Friday, April 30th** | **Uncover Sub-GHz and Proprietary Solutions within Simplicity Studio v5** |

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

We will begin in 4:00

## Speaker

Vikram Pochampally

FAE, India

# WELCOME

## Uncover Sub-GHz and Proprietary Solutions within Simplicity Studio v5

Vikram Pochampally

## Agenda

- High Level Overview
  - What is a Proprietary Wireless Application?
  - Two EFR32 Application Development Paths: RAIL vs. Connect
  - Wi-SUN Introduction
- Silicon Labs Flex Gecko SoCs & Development Tools
- Proprietary Project in Simplicity Studio v5
- Demo: RAIL SimpleTRX 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/long-range-connectivity-using-proprietary-rf-solution)
  + [https://www.silabs.com/support/training/sub-ghz-proprietary-and-connect-software-stack](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”

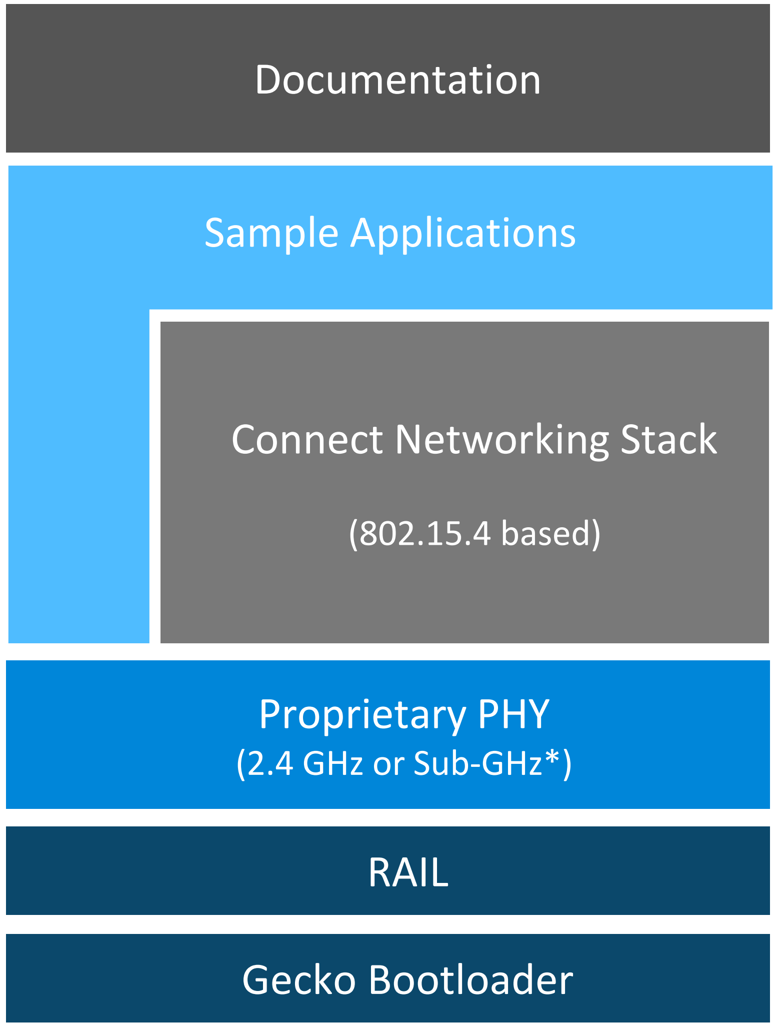
– Incompatibility with existing/future non-proprietary infrastructures

– 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

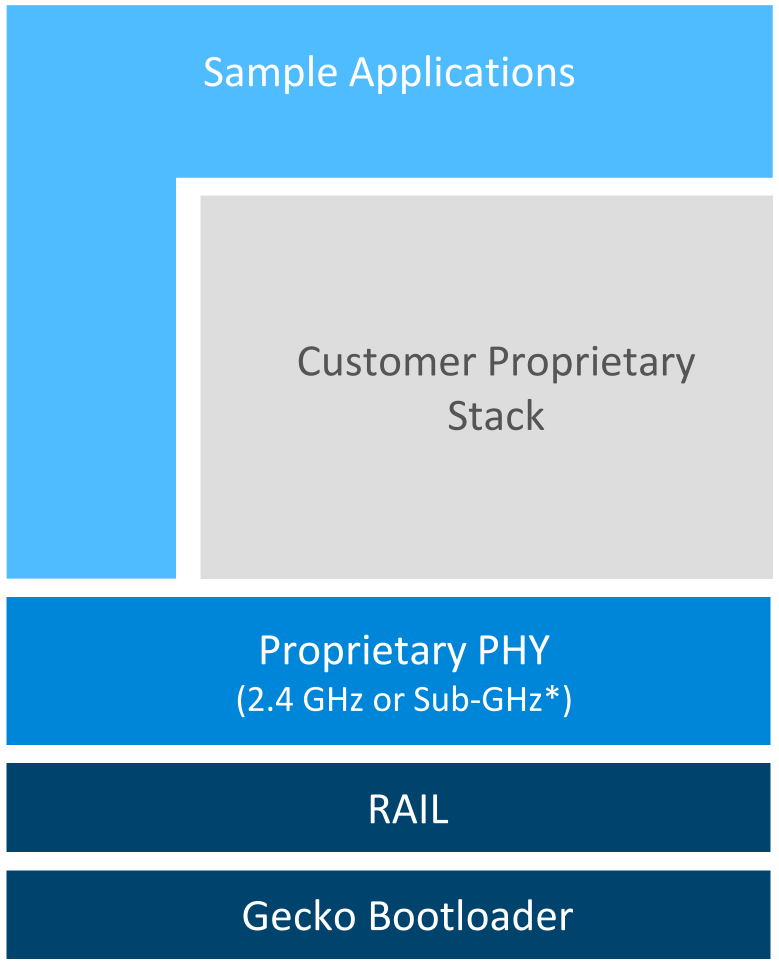
- Complete software development suite for proprietary wireless applications
- Common underlying software architecture with other wireless solutions
- Flexible, easy-to-use
  - Radio Abstraction Interface Layer (RAIL)
  - Connect Networking Stack
  - Sample applications
  - Extensive documentation
- Available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



* Wireless Gecko Series 1

# RAIL Software

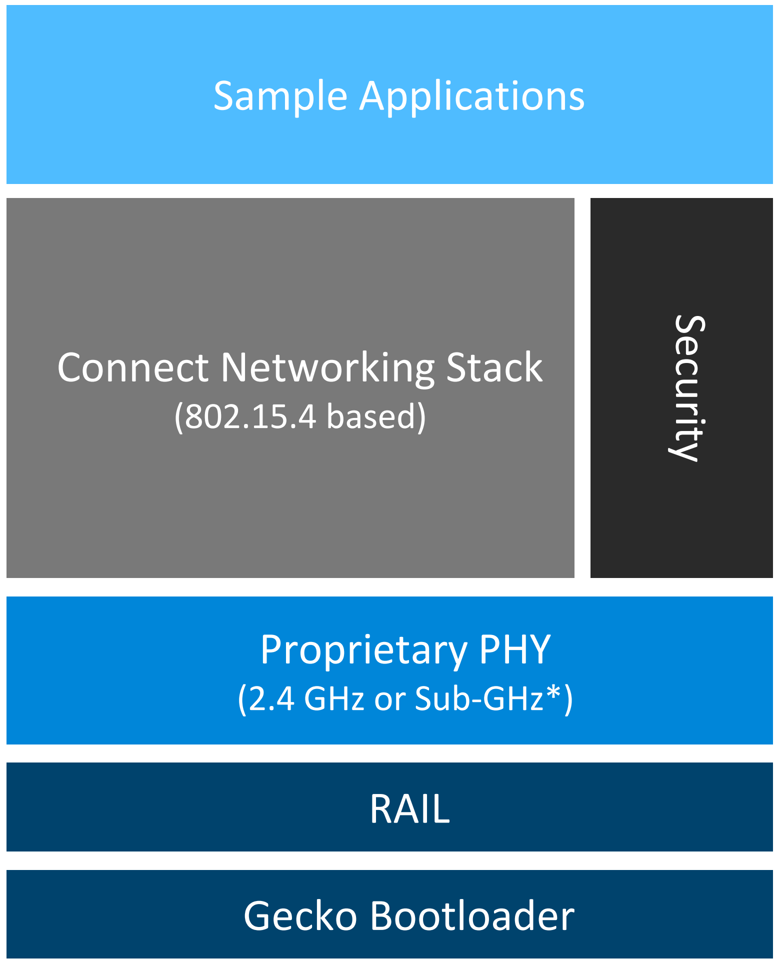
- Simplified Radio API
  - Common radio interface across SoCs
  - No need to learn complex low-level radio registers
- Lower development time
  - Easy migration of customer proprietary stack
  - Simplified radio testing
  - Quicker prototype boards bring up
- Complete software package
  - Integrated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



* Wireless Gecko Series 1

# Connect Networking Stack

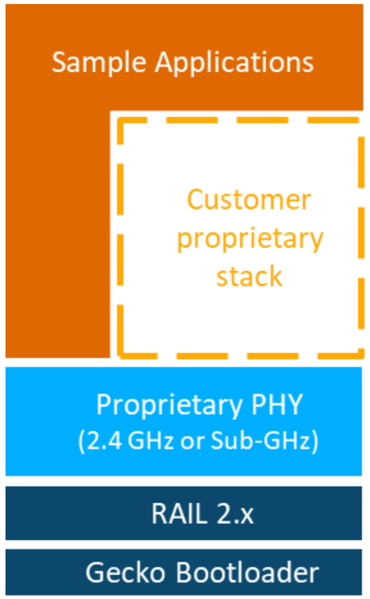
- Feature Rich Proprietary Wireless Networking Stack
  - Highly scalable
    - Up to 2K nodes in extended star mode
    - Up to 65K nodes in Direct Mode
  - Small stack footprint leaves room for application, OTA, etc.
  - Low power modes
  - Built-in security
- Faster Time To Market
  - Ready to use 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

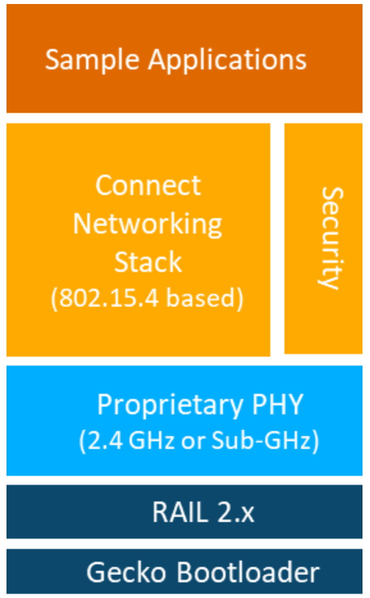
# RAIL API vs Connect Stack?

- 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

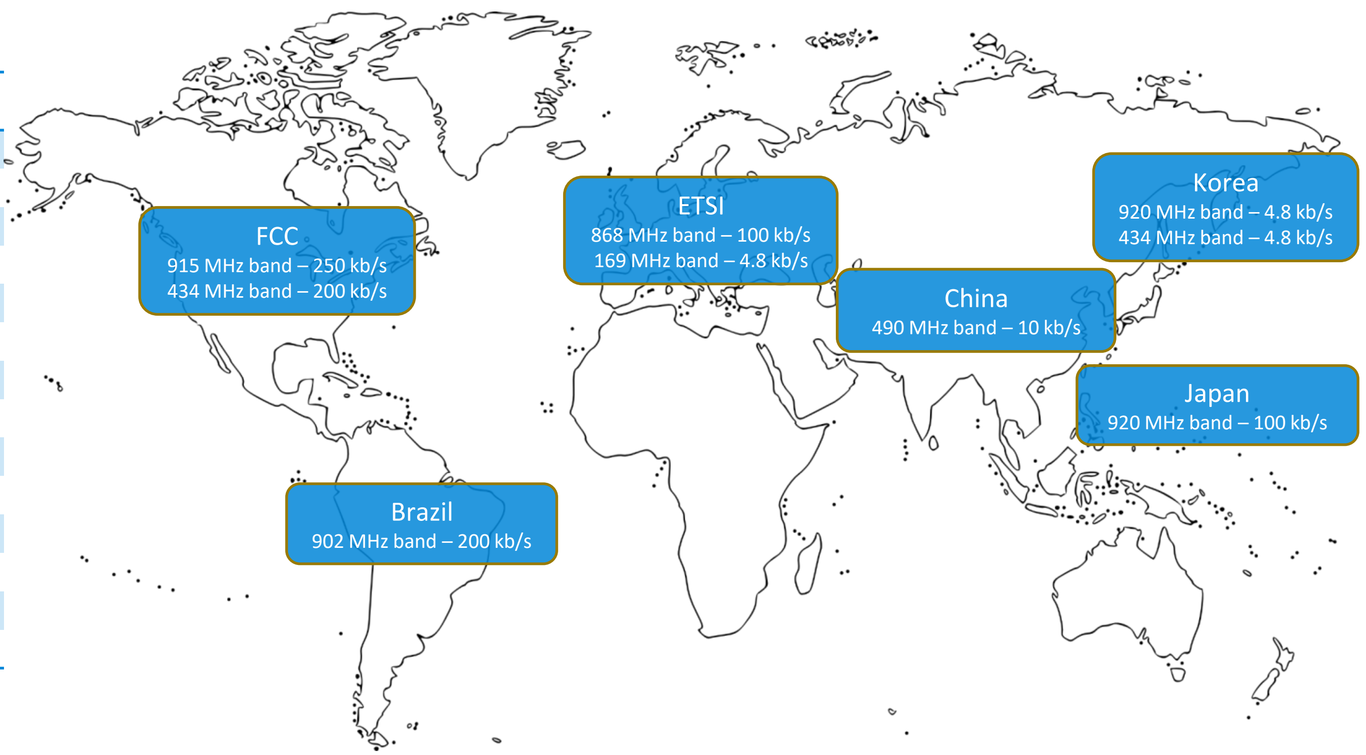


Sample Applications
Connect Networking Stack (802.15.4 based)
Security
Proprietary PHY (2.4 GHz or Sub-GHz)
RAIL 2.x
Gecko Bootloader

# 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 |



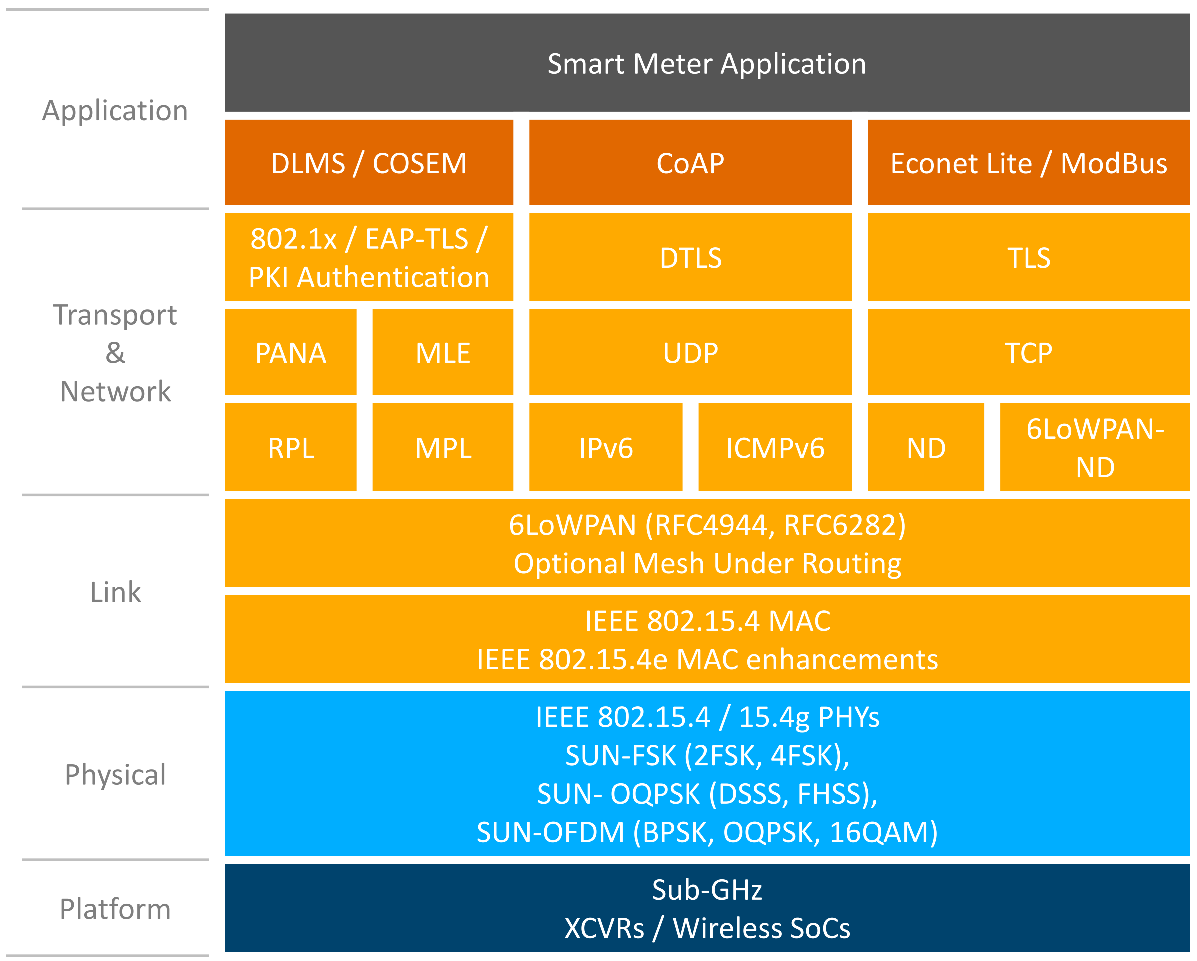
- Each RF configuration is tuned and tested by Silicon Labs for maximal performance
- Custom PHYs can be configured via Radio Configurator in Simplicity Studio

## Introduction to Wi-SUN

## Wi-SUN Overview

- What is Wi-SUN?
  - Is an alliance promoting open industry standards, interoperability testing and certification process for Wireless Smart Utility Networks
- What is a Wi-SUN Stack?
  - It is a complete OSI 7 layers stack implementation
  - It uses SUN-PHYs (also called **15.4g PHYs**) on **Sub-GHz bands** specific to various world regions
  - Based on IEEE/IETF standards **using 6LoWPAN & IPv6**
  - Implements strong **802.1x security** with EAP-TLS, PKI Authentication and Node 2 Node Key Management
  - Customer-defined application layer.  Common application layers used:
    - DLMS / COSEM, Econet Lite, ModBus TCP, CoAP, etc…
  - Different stack customizations called “profiles”:
    - FAN Profile (India), HAN Profile (Japan), etc

Generic Wi-SUN Stack Overview

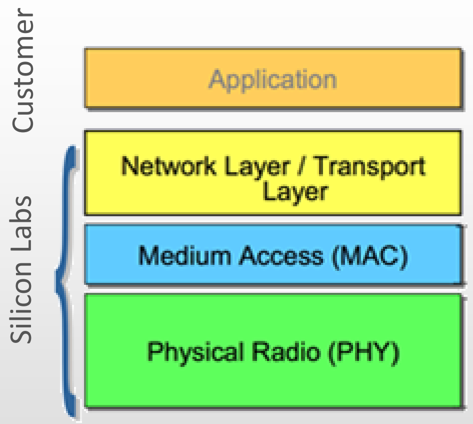

Application
Transport & Network
Link
Physical
Platform
Smart Meter Application
DLMS / COSEM
CoAP
Econet Lite / ModBus
802.1x / EAP-TLS / PKI Authentication
DTLS
TLS
PANA
MLE
UDP
TCP
RPL
MPL
IPv6
ICMPv6
ND
6LoWPAN-ND
6LoWPAN (RFC4944, RFC6282) Optional Mesh Under Routing
IEEE 802.15.4 MAC IEEE 802.15.4e MAC enhancements
IEEE 802.15.4 / 15.4g PHYs SUN-FSK (2FSK, 4FSK), SUN- OQPSK (DSSS, FHSS), SUN-OFDM (BPSK, OQPSK, 16QAM)
Sub-GHz XCVRs / Wireless SoCs

# Silicon Labs Wi-SUN Offering

## HARDWARE

EFR32xG12
Wi-SUN FSK PHYs

## WI-SUN STACK



Open-Source Solution
Based on ARM Nanostack
IPv6/6LoWPAN
Built-in Security

## TOOLS

Wi-SUN stack source code
Border Router as Demo
Command Line Interface
Wireshark Trace Export

## CERTIFICATION

PHY Certification
FAN 1.0 Certification

## Silicon Labs Wi-SUN FAN Stack

### Wi-SUN FAN Alpha Solution - Available

Selected customers only.

**Wi-SUN FAN Software Stack**

- **Source code available**
- Based on Arm open-source stack (Nanostack)
- Wi-SUN CLI sample application source code
- Run with Micrium OS kernel in GSDK 3.1
- **Border router demo provided as binary file**

**Hardware & PHY**

- WSTK + EFR32xG12 Radio boards

**Development Environment & Tools**

- Software hosted in private GitHub repositories
- Simplicity Studio v5 (UC/UP integration)
- PTI with Wireshark export capability

**Documentation**

- Bring-up Guide and API documentation on GitHub

### Wi-SUN FAN GA Solution - 2021 Q2

Publicly available.

**Wi-SUN FAN Software Stack**

- **Certified FAN 1.0 Stack** available in the GSDK 3.2
- Several sample applications available in the GSDK 3.2
- Run with Micrium OS or FreeRTOS (CMSIS-RTOS V2 layer)
- **Border router demo provided as binary file**

**Hardware & PHY**

- WSTK + EFR32xG12 Radio boards
- **Certified Wi-SUN FSK PHYs**

**Development Environment & Tools**

- Simplicity Studio v5 (UC/UP integration)
- PTI with Wireshark export capability

**Documentation**

- Wi-SUN FAN stack documentation on docs.silabs.com
- Application notes, Whiter papers…

## Silicon Labs Flex Gecko SoCs & Development Tools Overview

# Flex Gecko SoC Portfolio

## Flex Gecko SoCs

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

**EFR32FG22 –** 512kB / 32 kB **(ARM Cortex-M33 with TrustZone)**
(QFN32, QFN40)

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

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

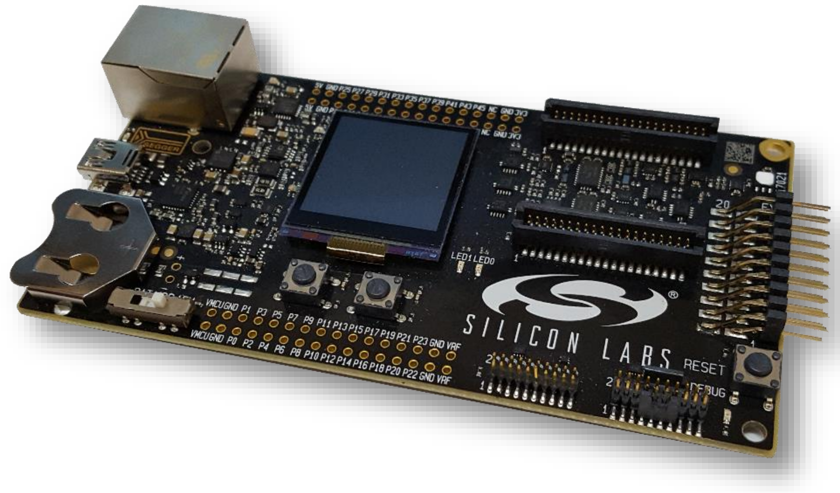
**EFR32FG1 -** 256 kB / 32 kB
(QFN48, QFN32)

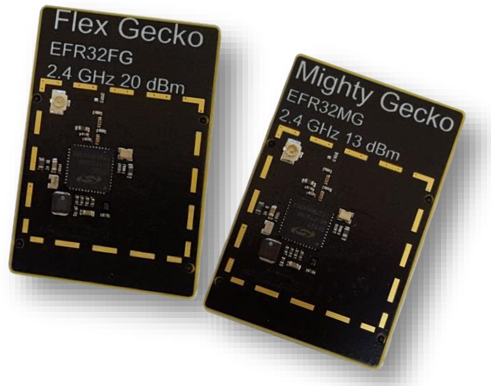
**EFR32FG14 -** 256 kB / 32 kB
(QFN48, QFN32)

- *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*

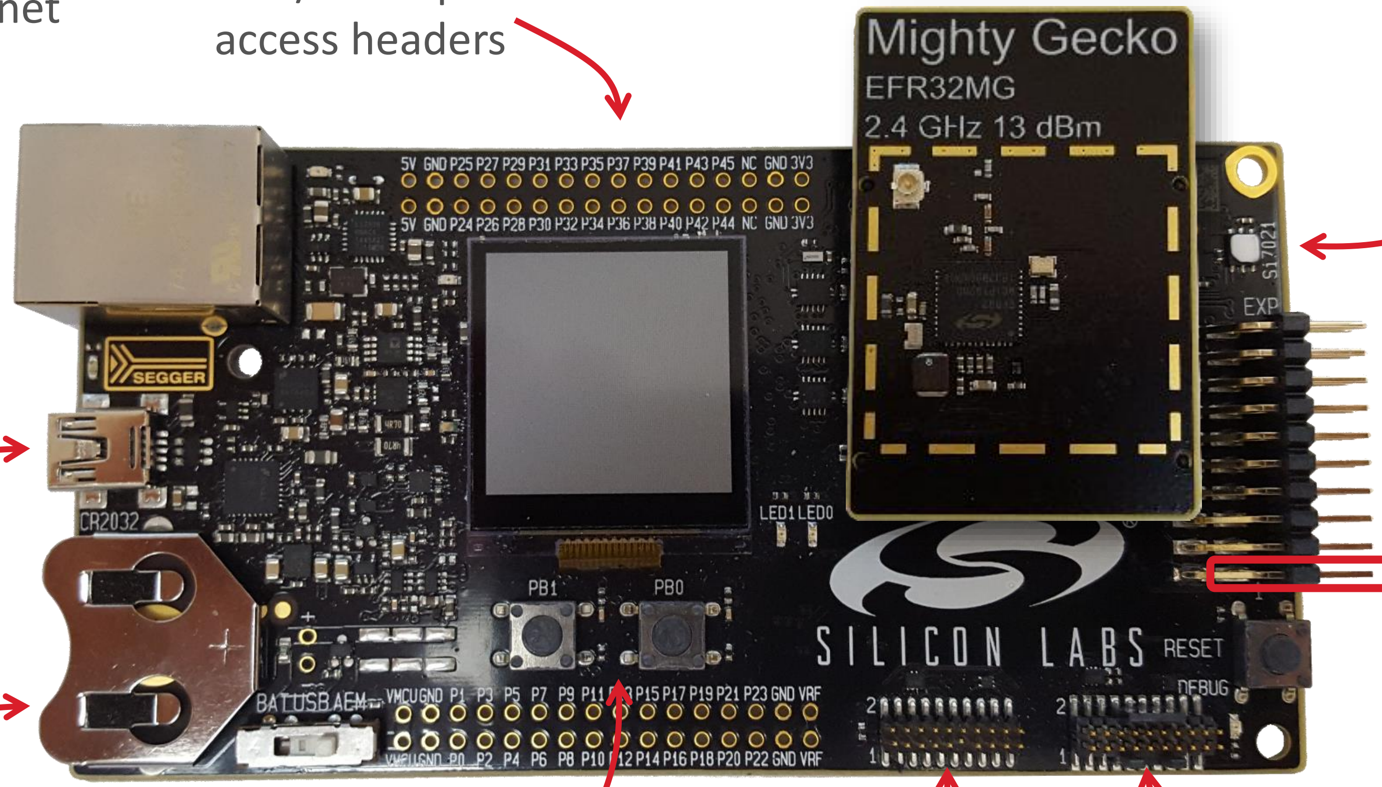
## Hardware Tools

- Built upon the WSTK platform (Wireless Starter Kit)
  - Common motherboard for all EFR32 variants
- Variety of different Radio boards
  - Mighty Gecko (2.4GHz/Sub-GHz Mesh Networking)
  - Flex Gecko (2.4GHz/Sub-GHz Proprietary Protocols)
  - Multiple boards for different output power levels (10.5dBm, 13dBm, 19.5dBm)
- Easy copy-and-paste reference designs
- On-board SEGGER debugger
- Supports enhanced development w/ EFR32 software tools
  - AEM, PTI, peripherals via board support packages, etc.
- Works seamlessly with RAIL & Connect



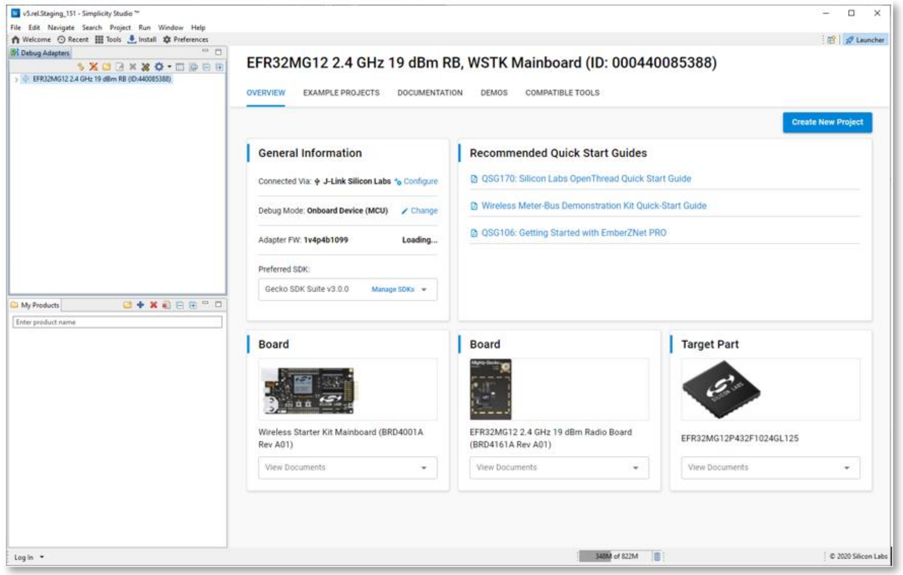


# Hardware Tools (Continued)


Onboard USB and Ethernet
J-Link Debugger
Packet Trace
SoC/Radio pin access headers
EFR32 Radio Module
Si7021 Humidity and Temperature Sensor
EXP header for expansion boards
USB serial-port
Advanced Energy
Monitoring
To AA Battery Holder
CR2032 Battery Holder
Power Source Selector Switch
Ultra-low-power 128x128 pixel memory LCD, buttons and LEDs
Simplicity header: Serial-port, packet trace and Advanced Energy Monitoring
ARM Coresight 19-pin trace/debug header
Mighty Gecko
EFR32MG
2.4 GHz 13 dBm
SILICON LABS
SEGGER
RESET
PB1
PB0
LED1 LED0
EXP
CR2032
DEBUG

# Software Tools

- **Simplicity Studio v5**
  - Complete SW suite makes development fast, easy, and efficient
  - One-click access to design tools, documentation, example projects, support resources
  - Provides hardware configuration, network analysis, real-time energy debugging tool, a high-powered IDE, and links to resources (see next slide)
- Flex SDK (installed within Simplicity Studio)
  - RAIL examples, Connect stack and Radio Configurator
- Gecko Platform (EMLIB, EMDRV, NVM3, mbedTLS, etc.) – inclunding RAIL library
- Simplicity Commander (standalone, GUI & CLI)
  - Installed alongside Simplicity Studio
  - Flash write/erase, debug access lock/unlock, NVM token management, scriptable manufacturing support, etc.



EFR32MG12 2.4 GHz 19 dBm RB, WSTK Mainboard (ID: 000440085388)

General Information

Connected Via: J-Link Silicon Labs Configure

Debug Mode: Onboard Device (MCU) Change

Adapter FW: 1v4p4b1099 Loading...

Preferred SDK: Gecko SDK Suite v3.0.0 Manage SDKs

Recommended Quick Start Guides

- QSG170: Silicon Labs OpenThread Quick Start Guide
- Wireless Meter-Bus Demonstration Kit Quick-Start Guide
- QSG106: Getting Started with EmberZNet PRO

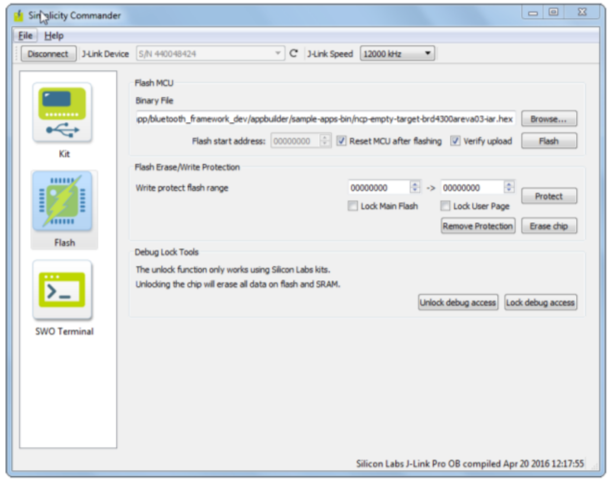
Board: Wireless Starter Kit Mainboard (BRD4001A Rev A01)

Board: EFR32MG12 2.4 GHz 19 dBm Radio Board (BRD4161A Rev A01)

Target Part: EFR32MG12P432F1024GL125

348M of 822M

© 2020 Silicon Labs



J-Link Device: S/N 440048424 J-Link Speed: 12000 kHz

Flash MCU

Binary File: ıpp/bluetooth_framework_dev/appbuilder/sample-apps-bin/ncp-empty-target-brd4300areva03-iar.hex

Flash start address: 00000000 [✓ Reset MCU after flashing] [✓ Verify upload]

Flash Erase/Write Protection

Write protect flash range 00000000 -> 00000000

Lock Main Flash Lock User Page

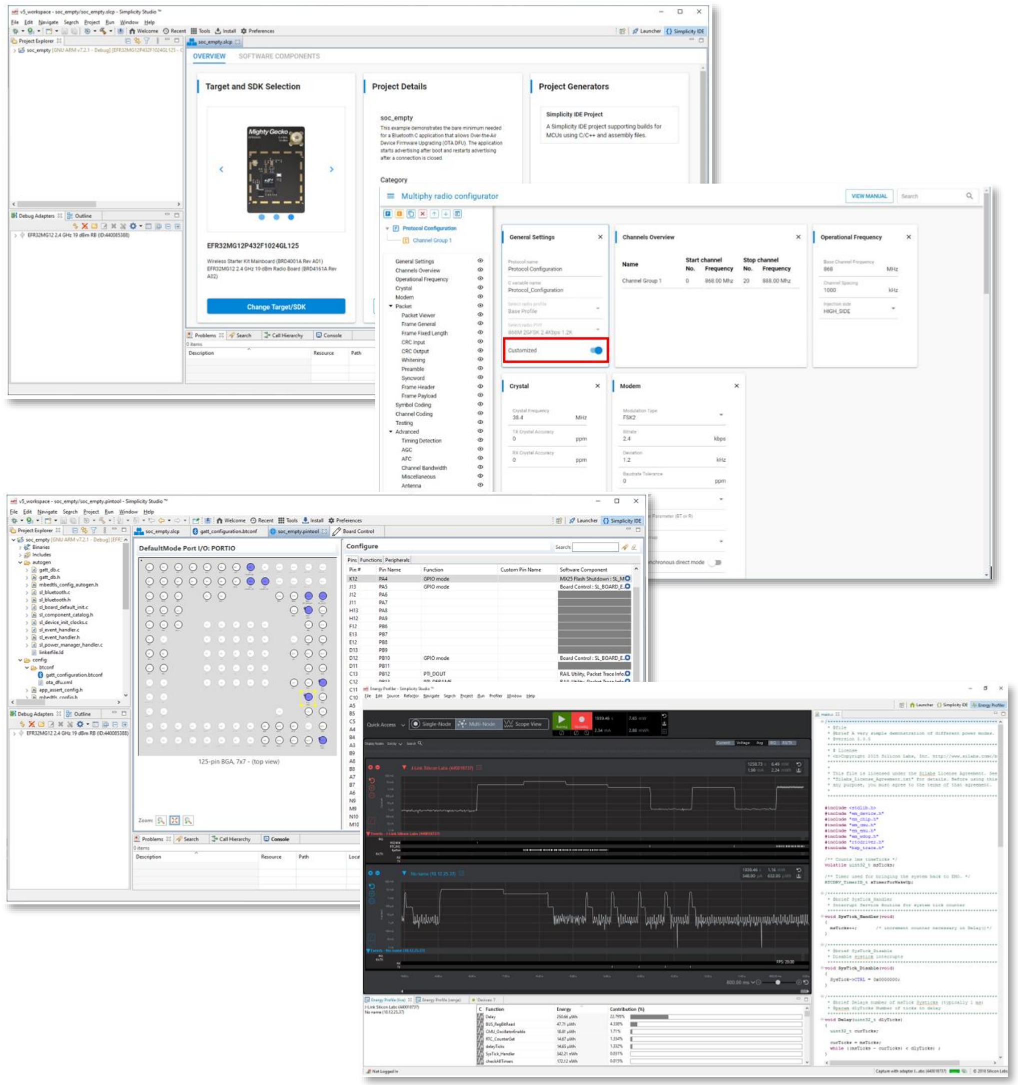
Debug Lock Tools

The unlock function only works using Silicon Labs kits.
Unlocking the chip will erase all data on flash and SRAM.

Silicon Labs J-Link Pro OB compiled Apr 20 2016 12:17:55

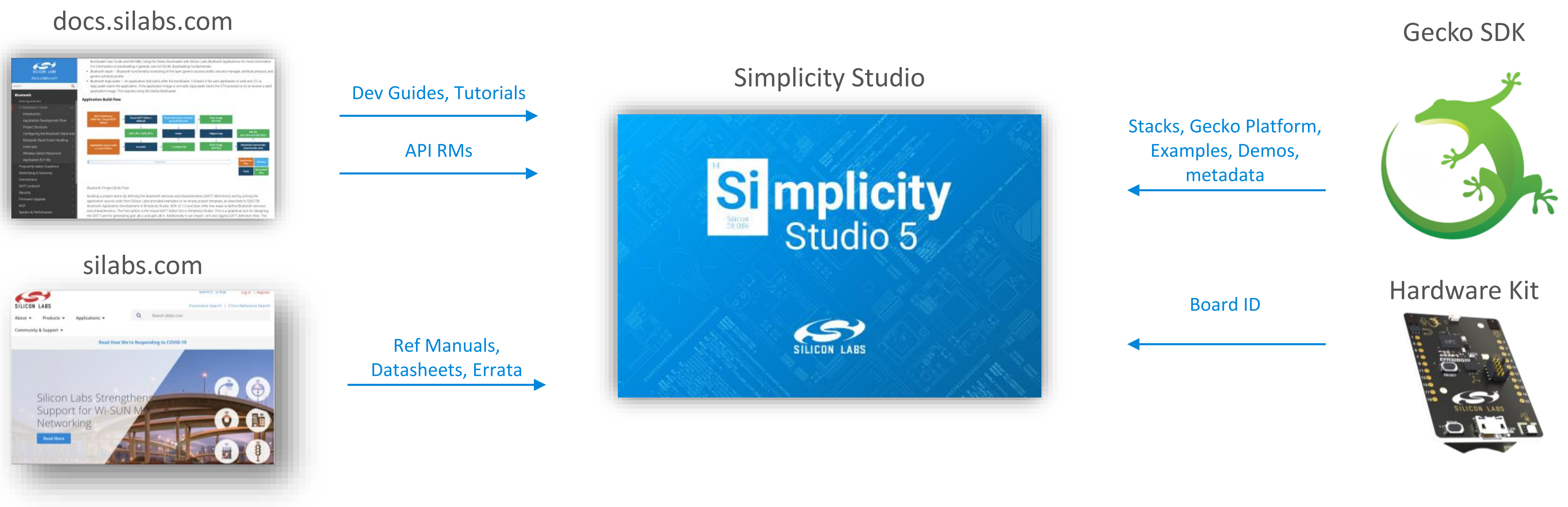
# Software Tools (Continued)

- Also available within Simplicity Studio
  - Eclipse based IDE
  - Project Configurator
    - Component-based modular feature management
  - Radio Configurator
    - PHY & frame format definition
  - Pin Tool
  - Device Console (command line tool)
  - Multi-Node Energy Profiler
  - Network Analyzer
- References:
  - Simplicity Studio® 5 User's Guide
  - Developing with Project Configurator
  - Proprietary Radio Configurator
  - QSG168: Proprietary Flex SDK v3.x Quick Start Guide

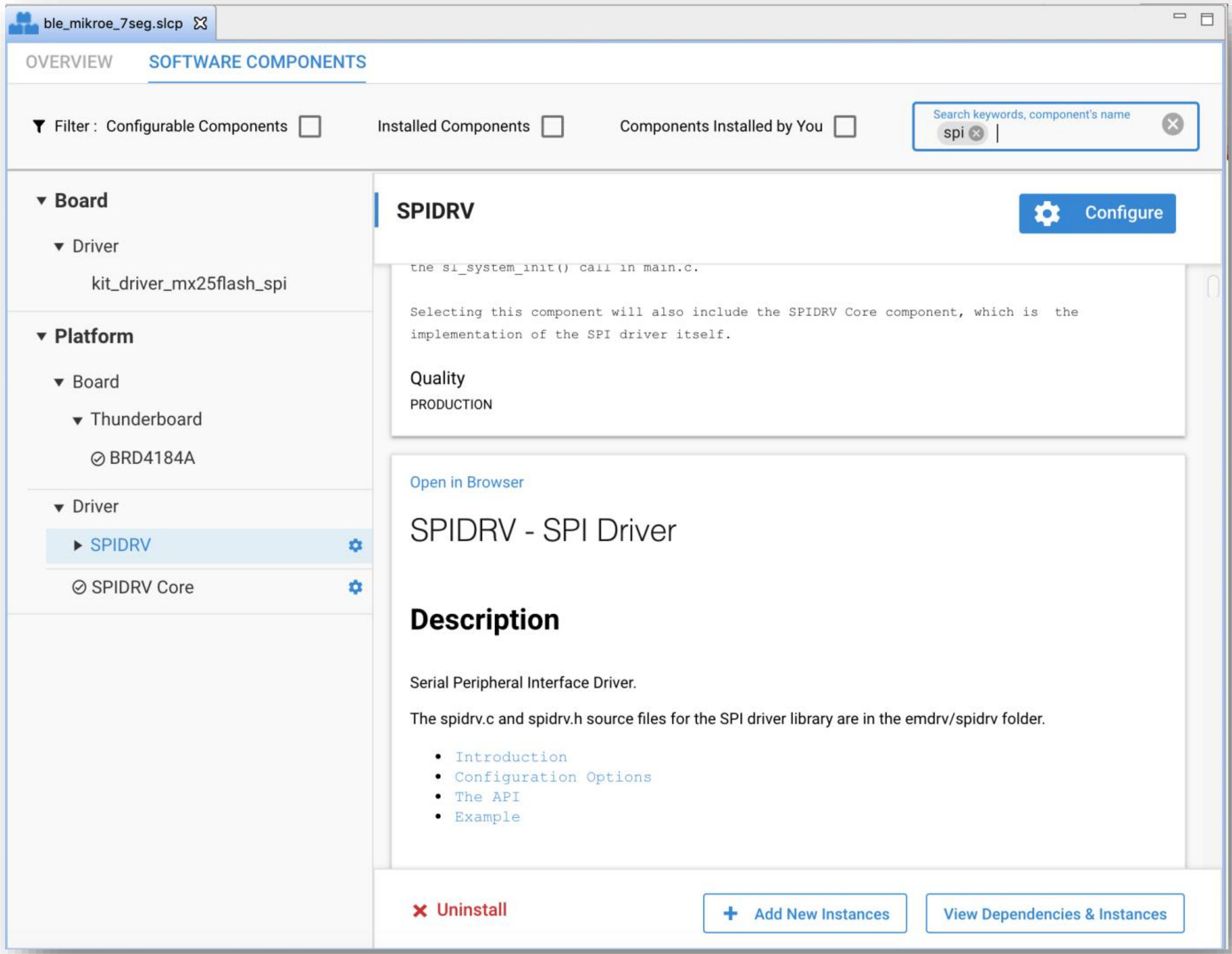


## Starting a Proprietary Project in Simplicity Studio5

# Simplicity Studio v5


docs.silabs.com
silabs.com
Dev Guides, Tutorials
API RMs
Ref Manuals, Datasheets, Errata
Simplicity Studio
Simplicity Studio 5
SILICON LABS
Stacks, Gecko Platform, Examples, Demos, metadata
Gecko SDK
Board ID
Hardware Kit
Silicon Labs Strengthens Support for Wi-SUN Networking

# Simplicity Studio 5 – New Project Configuration Tools



SPIDRV

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

SPIDRV - SPI Driver

Description

Serial Peripheral Interface Driver.

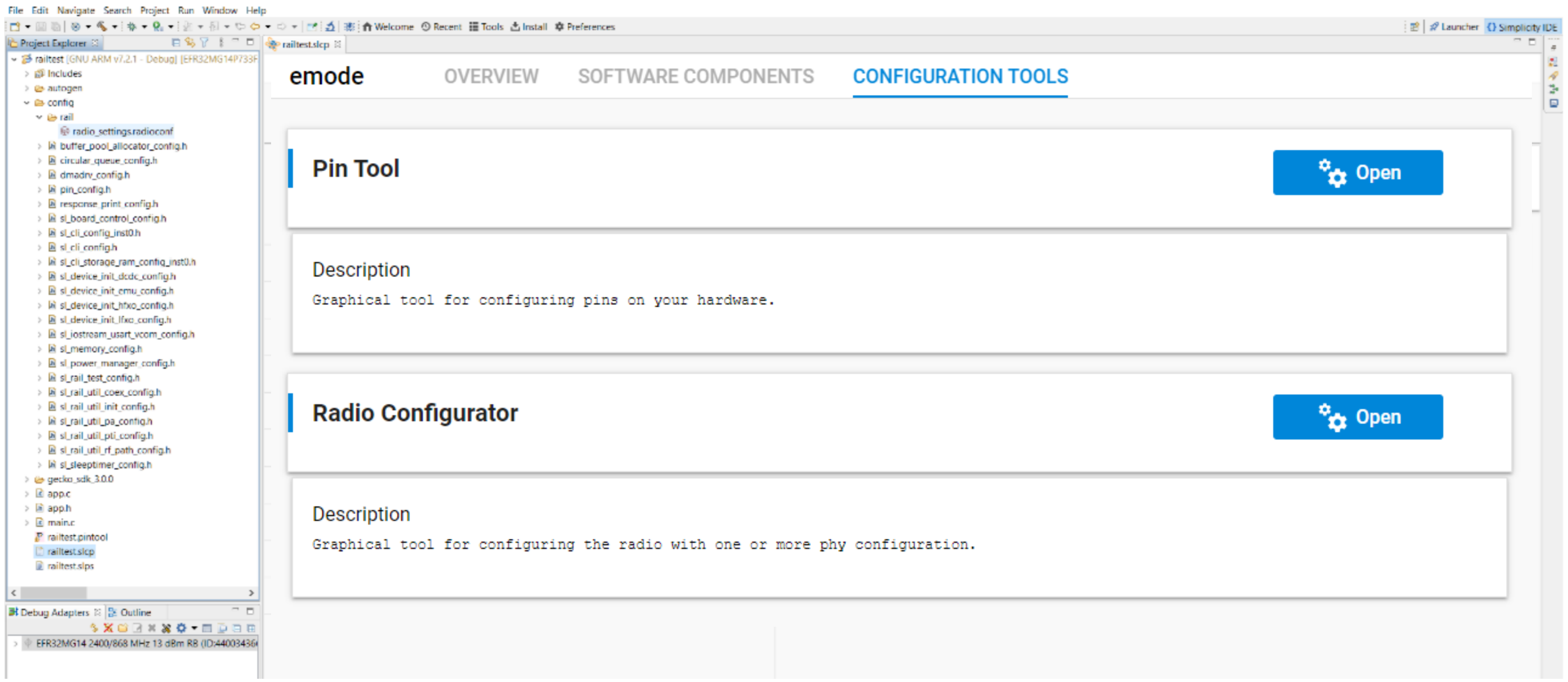
The spidrv.c and spidrv.h source files for the SPI driver library are in the emdrv/spidrv folder.

- Introduction
- Configuration Options
- The API
- Example

- 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

# Radio Configuration Flow: Radio Configurator

Once an EFR32-based project that uses Proprietary protocol (either a project in Flex SDK, or a DMP project) has been created in Simplicity Studio (as described in *QSG168: Silicon Labs Flex SDK v3.x Getting Started Guide*) an .slcp project file is created and an *Overview* tab is opened. Next to it, in the *Software Components* tab, the Radio Configurator can be accessed under the *Advanced Configurators* group. (For some examples, the Radio Configurator might open on project creation). All the radio configurator settings are stored at config/rail/ in the *radio_settings.radioconf* file.



```
railtest [GNU ARM v7.2.1 - Debug] [EFR32MG14P733F
  Includes
  autogen
  config
    rail
      radio_settings.radioconf
    buffer_pool_allocator_config.h
    circular_queue_config.h
    dmadrv_config.h
    pin_config.h
    response_print_config.h
    sl_board_control_config.h
    sl_cli_config_inst0.h
    sl_cli_config.h
    sl_cli_storage_ram_config_inst0.h
    sl_device_init_dcdc_config.h
    sl_device_init_emu_config.h
    sl_device_init_hfxo_config.h
    sl_device_init_lfxo_config.h
    sl_iostream_usart_vcom_config.h
    sl_memory_config.h
    sl_power_manager_config.h
    sl_rail_test_config.h
    sl_rail_util_coex_config.h
    sl_rail_util_init_config.h
    sl_rail_util_pa_config.h
    sl_rail_util_pti_config.h
    sl_rail_util_rf_path_config.h
    sl_sleeptimer_config.h
  gecko_sdk_3.0.0
  app.c
  app.h
  main.c
  railtest.pintool
  railtest.slcp
  railtest.slps
```

EFR32MG14 2400/868 MHz 13 dBm RB (ID:44003436

**Pin Tool**

Description

`Graphical tool for configuring pins on your hardware.`

**Radio Configurator**

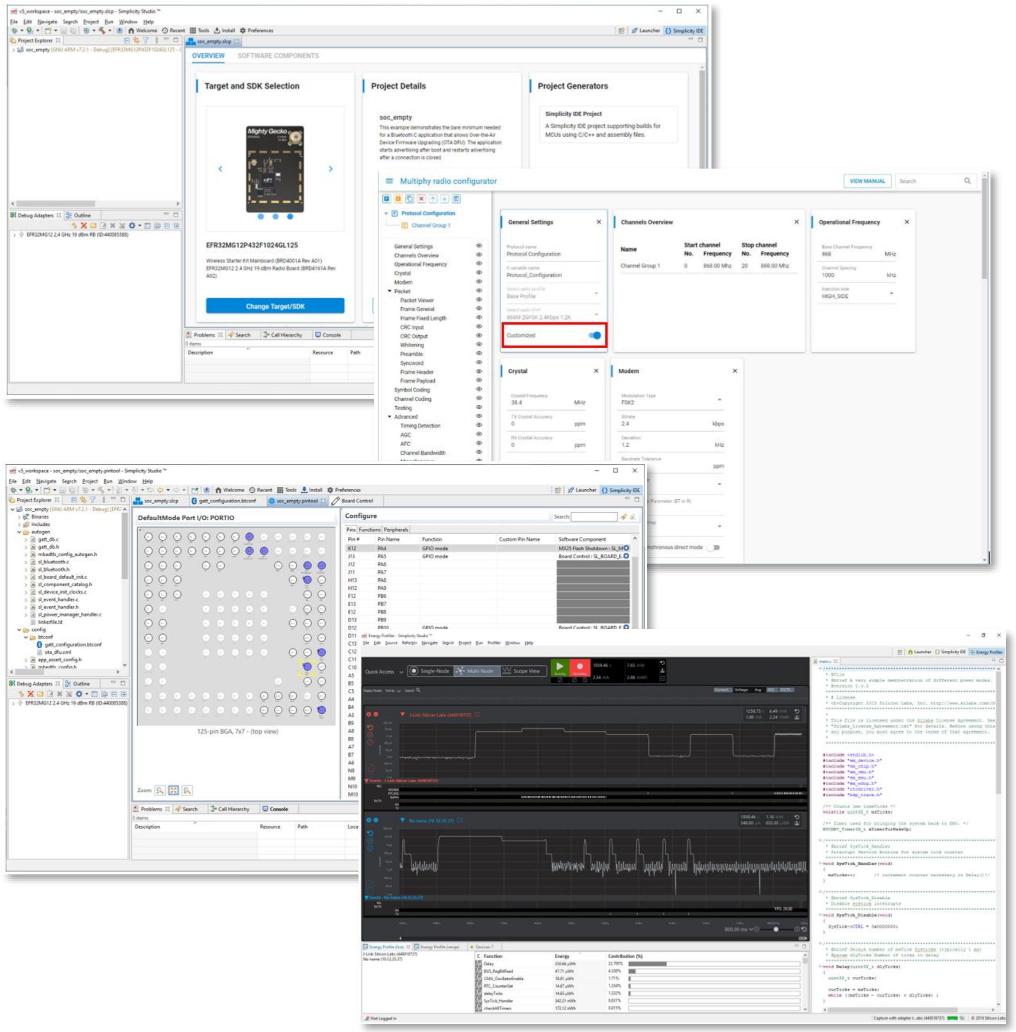
Description

`Graphical tool for configuring the radio with one or more phy configuration.`

*Borrowed from [AN1253: EFR32 Radio Configurator Guide for Simplicity Studio 5]*

# 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 SimpleTRX Application in Simplicity Studio v5

Blue Gecko
Wizard Gecko
Mighty Gecko
Mighty Gecko
SILICON LABS
SILICON LABS
Blue Gecko
Zen Gecko
Wireless Gecko

## 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 |
|---|---|---|

# THANK YOU

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