Proprietary Flex SDK 3.3.2.0 GA
Gecko SDK Suite 4.0
March 9, 2022

The Proprietary Flex SDK is a complete software development suite for proprietary wireless applications. Per its namesake, Flex offers two implementation options.

The first uses Silicon Labs RAIL (Radio Abstraction Interface Layer), an intuitive and easily-customizable radio interface layer designed to support both proprietary and standards-based wireless protocols.

The second uses Silicon Labs Connect, an IEEE 802.15.4-based networking stack designed for customizable broad-based proprietary wireless networking solutions that require low power consumption and operates in either the sub-GHz or 2.4 GHz frequency bands. The solution is targeted towards simple network topologies.

The Flex SDK is supplied with extensive documentation and sample applications. All examples are provided in source code within the Flex SDK sample applications.

These release notes cover SDK version(s):

3.3.2.0 GA released March 9, 2022
3.3.1.0 GA released January 26, 2022
3.3.0.0 GA released December 15, 2021

Proprietary

RAIL APPS AND LIBRARY KEY FEATURES

- Added support for EFR32xG24 parts
- Added support for Z-Wave on EFR32xG23 parts
- Added support for new ZGM230 modules.
- Updated the default PA curves for EFR32xG23 parts to be more accurate on Silicon Labs radio boards.

Compatibility and Use Notices

For information aboutsecurity updates and notices, see the Security chapter of the Gecko Platform Release notes installed with this SDK or on the TECH DOCS tab on https://www.silabs.com/developers/flex-sdk-connect-networking-stack. Silicon Labs also strongly recommends that you subscribe to Security Advisories for up-to-date information. For instructions, or if you are new to the Silicon Labs Flex SDK, see Using This Release.

Compatible Compilers:

IAR Embedded Workbench for ARM (IAR-EWARM) version 8.50.9

- Usingwine to build with the IarBuild.exe command line utility or IAR Embedded Workbench GUI on macOS or Linux could result in incorrect files being used due to collisions in wine's hashing algorithm for generating short file names.
- Customers on macOS or Linux are advised not to build with IAR outside of Simplicity Studio. Customers who do should carefully verify that the correct files are being used.

GCC (The GNU Compiler Collection) version 10.2.1, provided with Simplicity Studio.

Contents

1 Connect Applications

1.1 New Items

Added in release 4.0.0.0

EFR32xG23 Support

BRD4002A Support

1.2 Improvements

None

1.3 Fixed Issues

None

1.4 Known Issues in the Current Release

Issues in bold were added since the previous release. If you have missed a release, recent release notes are available on the TECH DOCS tab on https://www.silabs.com/developers/flex-sdk-connect-networking-stack.

ID #	Description	Workaround
652925	EFR32XG21 is not supported for "Flex (Connect) - SoC Light Example DMP" and "Flex (Connect) - SoC Switch Example"	

1.5 Deprecated Items

None

1.6 Removed Items

None

2   Connect Stack

2.1   New Items

None

2.2   Improvements

None

2.3   Fixed Issues

Fixed in release 4.0.0.0

ID #	Description
754930	Fixed an issue that was causing emberStopTxStream to fail when the parent support plugin is not installed.

2.4   Known Issues in the Current Release

Issues in bold were added since the previous release. If you have missed a release, recent release notes are available on the TECH DOCS tab on https://www.silabs.com/developers/flex-sdk-connect-networking-stack.

ID #	Description	Workaround
501561	In the Legacy HAL component, the PA configuration is hard-coded regardless of the user or board settings.	Until this is changed to properly pull from the configuration header, the file ember-phy.c in the user's project will need to be modified by hand to reflect the desired PA mode, voltage, and ramp time.
711804	Connecting multiple devices simultaneously may fail with a timeout error.	

2.5   Deprecated Items

None

2.6   Removed Items

None

3 RAIL Applications

3.1 New Items

Added in release 4.0.2.0

- EFR32xG24 Support

Added in release 4.0.0.0

- EFR32xG23 Support
- BRD4002A Support
- PSM Support for EFR32XG23, demonstrated by *Flex (RAIL) - Long Preamble Duty Cycle*
- Wi-SUN FSK Support for:
  - *Flex (RAIL) - Range Test*
  - *Flex (RAIL) - Simple TRX*

3.2 Improvements

Changed in release 4.0.0.0

The *Range Test* Applications can be used purely via CLI (LCD and Buttons can be uninstalled)

3.3 Fixed Issues

None

3.4 Known Issues in the Current Release

None

3.5 Deprecated Items

None

3.6 Removed Items

None

4 RAIL Library

4.1 New Items

Added in release 4.0.2.0

- Added support for the EFR32xG24 platform.
- Added a new **RAIL Utility, Built-in PHYs Across HFXO Frequencies** component to enable the built-in PHYs to operate with either 38.4 MHz or 39 MHz crystals on EFR32xG24 devices.
- Added support for additional IEEE 802.15.4 CCA modes on platforms that support signal detection (EFR32xG24). See the RAIL_IEEE802154_ConfigCcaMode API for more details.
- Added support for IEEE 802.15.4 and BLE Signal Identifier hardware on the EFR32xG24. See the new RAIL_IEEE802154_ConfigSignalIdentifier and RAIL_BLE_ConfigSignalIdentifier functions for more information. Note that this feature has only been tested to Alpha quality for this release.

Added in release 4.0.0.0

- Added support for the ZGM230SA27HGN, ZGM230SA27HNN, and ZGM230SB27HGN modules.
- Added RAIL_GetTxPacketsRemaining() API for use when handling one of the RAIL_EVENTS_TX_COMPLETION to get a sense of how many transmits remain in a RAIL_SetNextTxRepeat() sequence.
- Added PA curves for HP, MP, LP and LLP modes on all EFR32xG23 radio boards.
- Added RAIL_PA_BAND_COUNT to count RAIL_PaBand_t.
- Added a new RAIL_RxDataSource_t to capture direct mode data on supported devices.
- Added a new RAIL_IEEE802154_Config2p4GHzRadioCustom1 API to configure an alternate IEEE 802.15.4 PHY with slightly different performance characteristics for the EFR32xG12 and EFR32xG13 parts. Use this API if instructed by Silicon Labs for your use case.

4.2 Improvements

Changed in release 4.0.2.0

- BLE and 802.15.4 built-in PHYs are now exposed as public symbols in header files. Do not modify these without explicit instruction by Silicon Labs.
- Previously selecting an invalid CCA mode using RAIL_IEEE802154_ConfigCcaMode would fail silently and continue to use the RSSI based-CCA mode. Now RAIL_IEEE802154_ConfigCcaMode will return an error if an invalid CCA mode is selected.

Changed in release 4.0.1.0

- Restricted the SL_RAIL_UTIL_PA_RAMP_TIME_US to 10 µs on some EFR32 modules to match the certification conditions.
- Updated the Z-Wave PHYs for the EFR32xG23 to prevent a sensitivity degradation on the R2 (9.6 kbps) PHY.

Changed in release 4.0.0.0

- Added IEEE802.15.4 Coexistence and FEM PHYs to EFR32xG12 and EFR32xG13 based modules.
- Updated IEEE802.15.4 FEM PHYs on EFR32xG12 and EFR32xG13 based modules for improved performance.
- Updated all header files to have extern "C" when being built with C++ for compatibility.
- Made the RAIL_EnableRxDutyCycle() API safe to call in a multiprotocol application.
- To save both flash and RAM, moved information formerly contained in RAIL_Config_t::protocol, RAIL_Config_t::scheduler, and RAIL_Config_t::buffer internal to RAIL and sized appropriately for single vs. multiprotocol. RAIL multiprotocol now provides two internal state buffers for two protocols by default. An application that needs more must now call RAIL_AddStateBuffer3() or RAIL_AddStateBuffer4() to add a 3rd and 4th buffer, respectively. Otherwise RAIL_Init() will fail when trying to initialize a 3rd or 4th protocol.
- "RAIL Utility, Coexistence" component GPIO interrupt numbers are now chosen at runtime to avoid conflicts.

- A new RAIL API RAIL_GetSchedulerStatusAlt will now return more descriptive radio scheduler events as well as the RAIL_Status_t of the RAIL API invoked by the radio scheduler. As a part of the new API, new RAIL_SchedulerStatus_t events have been added while retaining the previous ones for backwards compatibility. Note that the underlying values of the existing RAIL_SchedulerStatus_t events may have changed.
- Updated the RAIL_IEEE802154_Config2p4GHzRadio*Fem PHYs on the EFR32xG12 and EFR32xG13 devices to improve performance.
- The example CSV files referred to in *AN1127: Power Amplifier Power Conversion Functions in RAIL 2.x* are updated with realistic values.

4.3   Fixed Issues

Fixed in release 4.0.2.0

ID #	Description
759793	Fixed an issue with BLE long-range reception on EFR32xG21 that corrupted packet data and tripped RAIL_ASSERT_FAILED_UNEXPECTED_STATE_RX_FIFO.
774883	Updated power curves for ZGM230SA27HGN, ZGM230SA27HNN, ZGM230SB27HGN modules to provide more accurate output powers at the lower and higher end of the dBm range.
777290	The PA auto mode configuration is fixed to use both HP and LP PA modes on the 10dBm EFR32xG24 chips.
777427	Fixed support for using the signal identifier CCA modes simultaneously with a user-enabled signal identifier trigger event.
812938	Fixed the RAIL_RX_CHANNEL_HOPPING_MODE_MULTI_SENSE mode on EFR32xG22 and EFR32xG23 to properly stay active after detecting Timing and Preamble when using the standard demodulator.

Fixed in release 4.0.1.0

ID #	Description
764234	Changed Quuppa channel 2 frequency from 2480 MHz to 2403 MHz.
773178	Fixed a compiler warning in "RAIL Utility, Callbacks" component when app_assert component ignores asserts.

Fixed in release 4.0.0.0

ID #	Description
646980	An attempt to use an unsupported built-in radio channel configuration, e.g., on a module that does not support that protocol or configuration, will now trip RAIL_ASSERT_FAILED_INVALID_CHANNEL_CONFIG rather than returning success and ignoring the configuration.
671651	Fixed timing problems with certain State_Transitions or Rx_Channel_Hopping delay values on the EFR32xG22 and newer parts.
682739	Fixed an issue with the BLE coded PHY's modulation index on the EFR32xG21 parts that could cause deviation measurements to fail.
714271	Fixed an issue where RAIL_IEEE802154_Config2p4GHzRadio*() and RAIL_IEEE802154_ConfigGB*Radio() functions were improperly clearing or setting certain RAIL_IEEE802154_EOptions_t. Also documented that these functions still implicitly clear or set certain RAIL_IEEE802154_GOptions_t suitable for that configuration.
716369	Fixed an issue where incorrect radio transition times were being applied at higher temperatures when using the high power PA on EFR32xG22 parts.
723098	Fixed RAIL_SetFixedLength(handle, RAIL_SETFIXEDLENGTH_INVALID) to restore dynamic frame length operation if the current PHY was originally configured for that.
738931	Fixed an issue with the BLE Coded PHYs on the EFR32xG22 device that could cause some packets to be improperly sent and not trigger a RAIL_EVENT_TX_PACKET_SENT event.
739594	Fix an issue with the RX_IQDATA_FILTLSB RAIL_RxDataSource_t on EFR32xG23 parts where the data did not properly saturate and was instead just the lower 16 bits of IQ sample data.

ID #	Description
744323	Fixed an issue when using BLE AoX where non-AoX packets were transmitted on an undefined antenna. They will now always use the first antenna in the configured RAIL_BLE_AoxConfig_t::antArrayAddr pattern.
745528	Fixed some incorrect RAIL_RxPacketInfo_t::filterMask values for 802.15.4 ACKs when promiscuous, or when the PanId coordinator received a packet with only source PanId and no destination address.
753860	Fixed an issue when running IR Calibration on the EFR32xG23 (RAIL_CalibrateIrAlt) where we could compute a completely invalid IRCAL value for certain PHYs and chips.
754219	Increase maximum BLE coex request window setting, SL_RAIL_UTIL_COEX_REQ_WINDOW, in "RAIL Utility, Coexistence" component from 255 to 5000.

4.4   Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
	Using direct mode (or IQ) functionality on EFR32xG23 requires a specifically set radio configuration that is not yet supported by the radio configurator. For these requirements, reach out to technical support who could provide that configuration based on your specification	
641705	Infinite receive operations where the frame's fixed length is set to 0 are not working correctly on the EFR32xG23 series chips.	
732659	On EFR32xG23: • Wi-SUN FSK mode 1a exhibits a PER floor with frequency offsets around ± 8 to 10 KHz • Wi-SUN FSK mode 1b exhibits a PER floor with frequency offsets around ± 18 to 20 KHz	
819544	**Rx duty cycle mode does not work reliably on the EFR32xG24 platform.**	
818707	**BLE CTE timings are sometimes slightly off when using the EFR32xG24 with a 38.4MHz crystal.**	
820912	**The Tx Abort feature in the IEEE 802.15.4 coexistence implementation does not work if enabled in the component.**	**Manually define SL_RAIL_UTIL_COEX_TX_ABORT globally if you want to ensure this feature is properly enabled.**

4.5   Deprecated Items

None

4.6   Removed Items

None

5   Using This Release

This release contains the following

- Radio Abstraction Interface Layer (RAIL) stack library
- Connect Stack Library
- RAIL and Connect Sample Applications
- RAIL and Connect Components and Application Framework

This SDK depends on Gecko Platform. The Gecko Platform code provides functionality that supports protocol plugins and APIs in the form of drivers and other lower layer features that interact directly with Silicon Labs chips and modules. Gecko Platform components include EMLIB, EMDRV, RAIL Library, NVM3, and mbedTLS. Gecko Platform release notes are available through Simplicity Studio's Documentation tab.

For more information about the Flex SDK v3.x see UG103.13: RAIL Fundamentals and UG103.12: Silicon Labs Connect Fundamentals. If you are a first time user, see QSG168: Proprietary Flex SDK v3.x Quick Start Guide.

5.1   Installation and Use

The Proprietary Flex SDK is provided as part of the Gecko SDK (GSDK), the suite of Silicon Labs SDKs. To quickly get started with the GSDK, install Simplicity Studio 5, which will set up your development environment and walk you through GSDK installation. Simplicity Studio 5 includes everything needed for IoT product development with Silicon Labs devices, including a resource and project launcher, software configuration tools, full IDE with GNU toolchain, and analysis tools. Installation instructions are provided in the online Simplicity Studio 5 User's Guide.

Alternatively, Gecko SDK may be installed manually by downloading or cloning the latest from GitHub. See https://github.com/SiliconLabs/gecko_sdk for more information.

Simplicity Studio installs the GSDK by default in:

- (Windows): C:\Users\<NAME>\SimplicityStudio\SDKs\gecko_sdk
- (MacOS): /Users/<NAME>/SimplicityStudio/SDKs/gecko_sdk

Documentation specific to the SDK version is installed with the SDK. Additional information can often be found in the knowledge base articles (KBAs). API references and other information about this and earlier releases is available on https://docs.silabs.com/.

5.2   Security Information

Secure Vault Integration

This version of the stack does not integrate Secure Vault Key Management.

Security Advisories

To subscribe to Security Advisories, log in to the Silicon Labs customer portal, then select **Account Home**. Click **HOME** to go to the portal home page and then click the **Manage Notifications** tile. Make sure that 'Software/Security Advisory Notices & Product Change Notices (PCNs)' is checked, and that you are subscribed at minimum for your platform and protocol. Click **Save** to save any changes.

5.3   Support

Development Kit customers are eligible for training and technical support. Use the Silicon Labs Flex web page to obtain information about all Silicon Labs Thread products and services, and to sign up for product support.

You can contact Silicon Laboratories support at http://www.silabs.com/support.

Simplicity Studio

One-click access to MCU and wireless
tools, documentation, software,
source code libraries & more. Available
for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

SILICON LABS

www.silabs.com