Proprietary Flex SDK 3.7.6.0 GA
Gecko SDK Suite 4.4
February 26, 2025

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 operate 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):

3.7.6.0 GA released February 26, 2025.
3.7.5.0 GA released October 23, 2024.
3.7.4.0 GA released August 14, 2024.
3.7.3.0 GA released May 2, 2024.
3.7.2.0 GA released April 10, 2024.
3.7.1.0 GA released February 14, 2024.
3.7.0.0 GA released December 13, 2023.

((•)) Proprietary

RAIL APPS AND LIBRARY KEY FEATURES

- Connect OFDM support for some applications
- EFR32xG28 Proprietary 2.4 GHz 15.4 Standard PHY support
- Added hardware support: MG24 QFN40, EFRBG22-E, EFR32xG28 Explorer Kit

CONNECT APPS AND STACK KEY FEATURES

- SUN-FSK and SUN-OFDM support
- Long message reception and transmission

Compatibility and Use Notices

For information about security 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 9.40.1

- Using wine 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 12.2.1, provided with Simplicity Studio.

Contents

1   Connect Applications

1.1   New Items

Added in release 3.7.0.0

- EFR32xG28 Explorer Kit Support
- EFRBG22-E Support
- MG24 QFN40 Support

1.2   Improvements

Changed in release 3.7.1.0

- Connect NCP: Installed the bootloader interface component by default. It allows to perform firmware updates from the host.

Changed in release 3.7.0.0

- Gecko Bootloader is used instead of Legacy HAL

1.3   Fixed Issues

Fixed in release 3.7.0.0

ID #	Description	Workaround
1076409	OTA Bootloader is not working on Series2	

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"	
1139850	DMP instabilities with XG27	

1.5   Deprecated Items

Deprecated in release 3.7.0.0

None.

1.6   Removed Items

Removed in release 3.7.0.0

None.

2   Connect Stack

2.1   New Items

Added in release 3.7.0.0

- Added support of SUN-FSK and SUN-OFDM. Enable the transmission and reception of packets carrying up to 2033 bytes of payload. `EmberMessageLength` was extended to allow the message length to be coded on 16 bits.
Most of the packet's buffer are allocated dynamically except the radio RX fifo that is static and has to be extended to be able to contain at least one maximum length packet. It is the application responsibility to do so by implementing `RAILCb_Set-upRxFifo()` and calling `RAIL_SetRxFifo()`. Depending on the application configuration, RTOS task stack sizes, minimum heap size and CPC packet maximum length may also need to be increased.
- Added new API emberOfdmSetMcs() and emberOfdmGetMcs() that set and get the OFDM modulation and coding scheme (MCS).
- Updated the hardware abstraction layer.

2.2   Improvements

Changed in release 3.7.1.0

- Reduced Connect Serial Procotol RAM usage

Changed in release 3.7.0.0

None.

2.3   Fixed Issues

Fixed in release 3.7.4.0

ID #	Description
1301334	Fixed a memory leak when waking up from EM2. A buffer is allocated during the mac initialization and wake-up phases. It should only be allocated during initialization.
1334048	Fixed an error preventing the child table from being backed up.

Fixed in release 3.7.2.0

ID #	Description
1252147	Fixed a new usage of the assert function that caused a significant increase of our flash usage. The problem fixed came from the fact that the value of the *expression* parameter is saved in a string format. v3.7.0 made use of the standard assert function while the previous versions used an implementation from the legacy hardware abstraction layer. The *expression* parameter was then always forced to 0, and the assertion test was made prior to the function call. V3.7.2 reverts that change.
1121468	Fixed an issue that was causing RAIL Utility PA configuration to be ignored. It was hard-coded in the libraries.
1266682	Fixed an issue that was causing the range extender update period configuration to be ignored. It was hard-coded in the libraries.

Fixed in release 3.7.1.0

ID #	Description
1223893	Fixed Connect SUN PHY signaling in the radio configurator artifacts. The lack of a good signalization was preventing the lower MAC to detect which PHY profile was used.

Fixed in release 3.7.0.0

None.

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/gecko-software-development-kit.

ID #	Description	Workaround
	When running the RAIL Multiprotocol Library (used for example when running DMP Connect+BLE), IR Calibration is not performed because of a known issue in the RAIL Multiprotocol Library. As result, there is an RX sensitivity loss in the order of 3 or 4 dBm.	
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

Deprecated in release 3.7.0.0

None.

2.6 Removed Items

Removed in release 3.7.0.0

None.

3   RAIL Applications

3.1   New Items

Added in release 3.7.0.0

- EFR32xG28 Explorer Kit Support
- EFRBG22-E Support
- MG24 QFN40 Support

3.2   Improvements

Changed in release 3.7.0.0

- EFR32xG28 Proprietary 2.4 GHz 15.4 Standard PHY Support
- Connect OFDM PHY Support for
  - RAIL - SoC Simple TRX
  - RAIL - SoC Range Test

3.3   Fixed Issues

Fixed in release 3.7.0.0

None.

3.4   Known Issues in the Current Release

None.

3.5   Deprecated Items

Deprecated in release 3.7.0.0

None.

3.6   Removed Items

Removed in release 3.7.0.0

None.

4 RAIL Library

4.1 New Items

Added in release 3.7.4.0

None.

Added in release 3.7.1.0

- Added support for setting the Whitening and CRC initial values at runtime that are defined by the existing PHY's radio configuration.

Added in release 3.7.0.0

- Added support for a new assert, which will be thrown if the loaded PHY is not supported by the software-defined modem on the EFR32xG25.
- Added new RAIL_GetAutoAckFifo() API and allow NULL for RAIL_WriteAutoAckFifo() or RAIL_IEEE802154_WriteEnhAck() ackData parameter, which gives applications direct access to the AutoAck FIFO to construct Ack packets in pieces.
- Added support for antenna selection through the applicable RAIL_RxOptions_t and RAIL_TxOptions_t values when using OFDM on the EFR32xG25.
- Added a new "RAIL Utility, SFM Sequencer Image Selection" component to allow selection of modulations supported by EFR32xG25 software modem (SFM). These changes can save considerable flash space by reducing the set of modulations to just those that are needed.
- Added support for Sidewalk PHYs on the EFR32xG23 and EFR32xG28 chips.
- Added an assert RAIL_ASSERT_INVALID_XTAL_FREQUENCY on the EFR32xG1x and EFR32xG2x chips, to fire when a radio config that is loaded is incompatible with a device due to the defined crystal frequency of the config not matching with the device's crystal frequency.
- Added the RAIL_TX_REPEAT_OPTION_START_TO_START option to measure the delay between repeated transmits from the start of TX to start of TX instead of the default from end of TX to start of TX.
- Added support for GCC 12.2.1 and IAR 9.40.1 compilers.
- Added support for Fast Channel Switching PHYs on the EFR32xG24.
- Added support for RAIL_IEEE802154_SUPPORTS_G_MODESWITCH on the EFR32xG28.
- Added support for the IEE802154 2.4 GHz coherent PHYs via RAIL_SUPPORTS_IEEE802154_BAND_2P4 on the EFR32xG28.
- Added a new RAIL_RxOptions_t option to enable collision detection on the EFR32xG25. Once enabled, when a collision with a strong enough packet is detected, the demod will stop the current packet decoding and try to detect the preamble of the incoming packet.
- Added support of RAILTEST for the MGM240x modules.
- Added support for channel masks during Wi-SUN mode switch in RAILTEST application on the EFR32xG25.
- Added support of 802.15.4 IMM-ACK with OFDM and OQPSK modulations (FCS 4 bytes only) on the EFR32xG25.
- Updated the RAIL Library component to automatically include the "RAIL Utility, Built-in PHYs Across HFXO Frequencies" component to better support different HFXO frequencies on supported parts.

4.2 Improvements

Changed in release 3.7.4.0

- Reduced the delay between TX completion and the start of PA ramp down on the EFR32xG24.

Changed in release 3.7.0.0

- Updated the default PTI rate to 3200000 bps on the EFR32XG25.
- No longer include rail_chip_specific.h and rail_features.h (anything dependent on em_device.h) when SLI_LIBRARY_BUILD is defined. This will allow the user to build their radio code in a way that depends on RAIL generically but is not chip-specific. When doing this the code cannot depend on things that are inherently chip-specific and are still in those files like RAIL_RF_PATHS, RAIL_NUM_PA, or any of the compile-time RAIL_SUPPORTS_xxx defines. Code will need to call appropriate runtime APIs or split itself between the generic and chip specific portions and build them separately.
- To better support chip-agnostic builds RAIL_TxPowerMode_t is now a superset representing all possible PAs across all chips. Any code depending on the number of or consecutive ordering of chip-supported PAs will likely need to be updated.

- To better support chip-agnostic builds RAIL_CalValues_t and subordinate RAIL_IrCalValues_t have grown to encompass the super-set of fields needed on all chips, affecting all chips except EFR32xG25.

- To better support chip-agnostic builds RAIL_TransitionTime_t and therefore RAIL_StateTiming_t have grown to a superset type, affecting EFR32xG1.

- To better support chip-agnostic builds RAIL_FIFO_ALIGNMENT is now universally 32-bit, affecting EFR32xG1x and EFR32xG21, but is still only enforced on chips actually requiring that alignment.

- Updated the default power curves for the 10dBm High Power PA on EFR32xG24.

4.3   Fixed Issues

Fixed in release 3.7.6.0

ID #	Description
1315679	Fixed a multiprotocol issue where a custom sync word set for one channel could override another channel's sync word after a DMP switch back to that protocol. This change reverts earlier fix attempts by documenting this limitation: if multiple protocols share the same radio configuration. The setting of custom sync words in any of these protocols should be avoided, as they can again leak into another protocol sharing the same radio configuration.
1377955	Fixed a DMP issue where the device would incorrectly switch to a background receive task with insufficient time to switch back to an upcoming high priority scheduled task.
1380153	Fixed an issue where calling \ref RAIL_Idle() could prevent any future reception on the EFR32xG25 for OFDM-FSK concurrent PHYs.

Fixed in release 3.7.4.0

ID #	Description
1271542	Fixed an issue on EFR32xG21 where an inopportunely-timed RAIL_StopTx() using RAIL_STOP_MODE_ACTIVE can hang a subsequent CSMA/LBT transmit during its clear-channel assessment (CCA).
1306597	Fixed a CCA issue on EFR32xG28 where, in some cases, the channel could be reported as busy even though it was free.

Fixed in release 3.7.2.0

ID #	Description
1248013	Fixed an issue with IEEE 802.15.4 High Data Rate (HDR) packets such that PTI now properly indicates they have a 2-byte 802.15.4 PHY header.
1255347	Fixed an issue with RAIL_SetTxFifo() setup that could trip RAIL_ASSERT_FAILED_UNEXPECTED_STATE_TX_FIFO when the prior TX FIFO was larger and contained more data than the new, smaller TX FIFO can hold. Any data in the old TX FIFO should have been ignored.
1271435	Fixed an issue where RAIL_WriteTxFifo() could write outside the TX FIFO when called without resetting the FIFO in dynamic multiprotocol applications.

Fixed in release 3.7.1.0

ID #	Description
1241800	Added the missing pa_dbm_mapping_table.py script to the release package. This script is used to help create power tables on the EFR32xG25 part.
1242723	Fixed an issue to exit from the critical section while performing multiprotocol PA operations on the EFR32xG25.
1243727	Improved the CCA fiability on the EFR32xg23, EFR32xg25 and EFR32xg28 chips.

Fixed in release 3.7.0.0

ID #	Description
1079816	Fixed a race condition on the EFR32xG22 and later during RX channel hopping or duty-cycling where frame detection occurring close to when a hop should happen could leave the radio stuck in reception but not receiving anything, with the only remedy being to idle the radio.
1088439	Fixed an issue which would cause the incorrect antenna to be reported for a received packet on the EFR32xG25 when using OFDM and antenna diversity.
1153679	Fixed an issue in "RAIL Utility, Coexistence" component on the EFR32xG24 where a GRANT signal pulse less than 100us might result in the radio not being properly placed in hold off after GRANT is deasserted.
1156980	Fixed an issue with channel hopping on the EFR32xG22 and later where use of RAIL_RX_CHANNEL_HOPPING_OPTION_RSSI_THRESHOLD can prevent the timed RX channel hopping modes including RX duty-cycling from timing out properly.
1175684	Fixed an issue with the RAIL_IDLE form of RAIL_Idle() and RAIL_STOP_MODE_PENDING form RAIL_StopTx() during the LBT/CSMA phase of a transmit which could previously hang. Pending LBT/CSMA and scheduled transmits are now stopped or idled with RAIL_EVENT_TX_BLOCKED triggered, except for idle mode RAIL_IDLE_FORCE_SHUTDOWN_CLEAR_FLAGS.
1183040	Updated all PHYs built into RAIL on Series 2 platforms using the latest radio calculator to reduce observed high reference spurs.
1184982	Fixed an issue with RAIL_StartAverageRssi() that could cause execution to hang in interrupt context just before raising RAIL_EVENT_RSSI_AVERAGE_DONE. This was primarily an issue on the EFR32xG21 platform. Note that it is possible, though unlikely, that RAIL_GetAverageRssi() might still return RAIL_RSSI_INVALID after the average RSSI period has finished.
1184982	Fixed an issue with RAIL_StartAverageRssi() that caused it to mistakenly idle the radio in the newly-activated protocol if a dynamic protocol switch occurred during the averaging operation of the suspended protocol.
1188083	Fixed an issue where RAIL_Idle() could hang waiting for the radio to idle when using IEEE 802.15.4 fast RX channel switching.
1190187	Fixed an issue where idling the radio before a Scheduled Receive window ends could cause a subsequent packet that should be silently filtered and rolled back to instead be received with RAIL_RX_PACKET_READY_CRC_ERROR.
1201506	Fixed an issue in multiprotocol applications where the wrong sync word would be used if two protocols used the same radio configuration and only one of those protocols set a custom sync word with RAIL_ConfigSyncWords() API.

4.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

ID #	Description	Workaround
1335868	On EFR32xG28, when large data is written and read back using rxFifoManualRead API, it fails to read even though the data is ready to be read.	On EFR32xG28: If customers want to use the packet length up to 2058 bytes, the value of BUFFER_POOL_ALLOCATOR_BUFFER_SIZE_MAX needs to be set to 2102 for device_sdid_235 in RAIL - SoC RAILtest application.
	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.	

ID #	Description	Workaround
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	

4.5   Deprecated Items

Note:

- The RAIL 2.x API is planned for deprecation in the 25Q2-GA release (June 2025). At that time, the new RAIL 3.0 API will be released for all supported chips along with a RAIL 2.x compatibility layer and migration guide.

- The goal of this new API is to evolve what we currently have to get rid of some unused features, add better support for concurrent listening use cases, and simplify channel and PA configurations.

- We hope the migration is straight forward and simple for the majority of customers, but there may be manual help needed in some cases which we will do our best to document to ease this transition.

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

When deployed to Secure Vault High devices, sensitive keys are protected using the Secure Vault Key Management functionality. The following table shows the protected keys and their storage protection characteristics.

Wrapped Key	Exportable / Non-Exportable	Notes
Thread Master Key	Exportable	Must be exportable to form the TLVs
PSKc	Exportable	Must be exportable to form the TLVs
Key Encryption Key	Exportable	Must be exportable to form the TLVs
MLE Key	Non-Exportable	
Temporary MLE Key	Non-Exportable	
MAC Previous Key	Non-Exportable	
MAC Current Key	Non-Exportable	
MAC Next Key	Non-Exportable	

Wrapped keys that are marked as "Non-Exportable" can be used but cannot be viewed or shared at runtime.
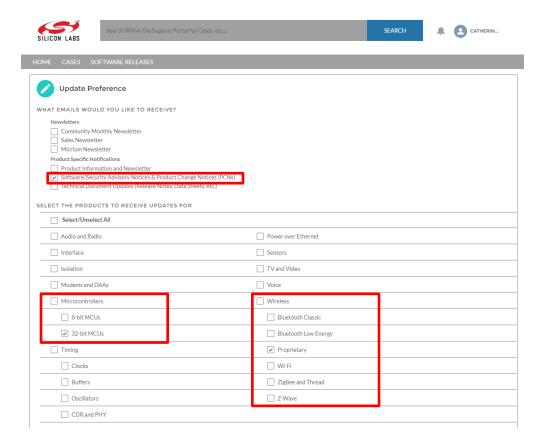
Wrapped keys that are marked as "Exportable" can be used or shared at runtime but remain encrypted while stored in flash.

For more information on Secure Vault Key Management functionality, see AN1271: Secure Key Storage.

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.
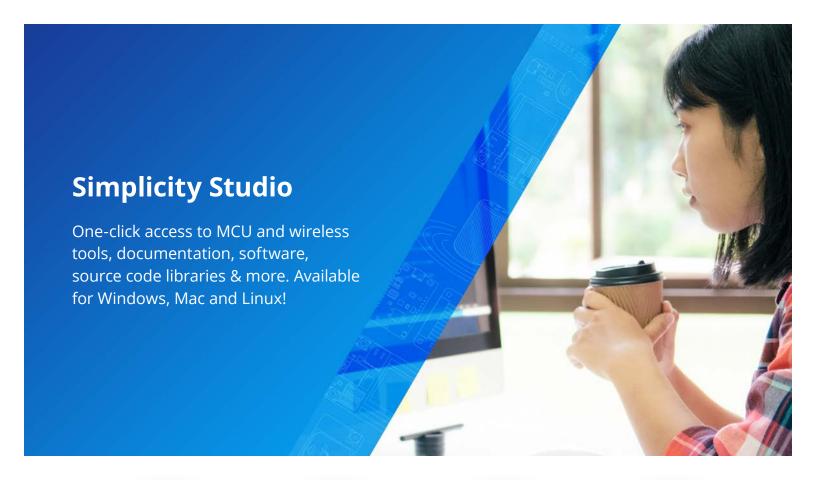
The following figure is an example:

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