



Proprietary Flex SDK 3.5.5.0 GA

Gecko SDK Suite 4.2

January 24, 2024

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):

3.5.5.0 GA released January 24, 2024
3.5.4.0 GA released August 16, 2023
3.5.3.0 GA released May 3, 2023
3.5.2.0 GA released March 8, 2023
3.5.1.0 GA released February 1, 2023
3.5.0.0 GA released December 14, 2022



RAIL APPS AND LIBRARY KEY FEATURES

- FG25 Flex-RAIL GA support
- New Long Range PHYs support for 490 MHz and 915 MHz
- xG12 dynamic mode switching support in RAIL
- xG22 extended band support

CONNECT APPS AND STACK KEY FEATURES

- xG24 Connect support

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 9.20.4

- 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 10.3-2021.10, provided with Simplicity Studio.



Contents

1	Connect Applications.....	1
1.1	New Items.....	1
1.2	Improvements.....	1
1.3	Fixed Issues	1
1.4	Known Issues in the Current Release	1
1.5	Deprecated Items	1
1.6	Removed Items	1
2	Connect Stack.....	2
2.1	New Items.....	2
2.2	Improvements.....	2
2.3	Fixed Issues	2
2.4	Known Issues in the Current Release	2
2.5	Deprecated Items	2
2.6	Removed Items	2
3	RAIL Applications.....	3
3.1	New Items.....	3
3.2	Improvements.....	3
3.3	Fixed Issues	3
3.4	Known Issues in the Current Release	3
3.5	Deprecated Items	3
3.6	Removed Items	3
4	RAIL Library	4
4.1	New Items.....	4
4.2	Improvements.....	4
4.3	Fixed Issues	5
4.4	Known Issues in the Current Release	6
4.5	Deprecated Items	7
4.6	Removed Items	7
5	Using This Release.....	8
5.1	Installation and Use.....	8
5.2	Security Information.....	8
5.3	Support.....	9

1 Connect Applications

1.1 New Items

Added in release 3.5.0.0

- XG24 Support

1.2 Improvements

Changed in release 3.5.0.0

- OQPSK Long Range PHYs for XFG23

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

Added in release 3.5.0.0

- XG24 Support

2.2 Improvements

None

2.3 Fixed Issues

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
389462	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 Deprecatcd Items

None

2.6 Removed Items

None

3 RAIL Applications

3.1 New Items

Added in release 3.5.0.0

- XG25 Support
- RAIL SoC Mode Switch Application

3.2 Improvements

Changed in release 3.5.0.0

- RAIL SoC Long Preamble Duty Cycle support for XG24
- OQPSK Long Range PHYs for XFG23

3.3 Fixed Issues

Fixed in release 3.5.1.0

ID #	Description
	Mode Switch: MCS Rate selection fix for OFDM.

3.4 Known Issues in the Current Release

None

3.5 Deprecated Items

None

3.6 Removed Items

Removed in release 3.5.0.0

- RAIL SoC Long Preamble Duty Cycle (Legacy)
- RAIL SoC Light Standard
- RAIL SoC Switch Standard

4 RAIL Library

4.1 New Items

Added in release 3.5.2.0

- Added `RAIL_PacketTimeStamp_t::packetDurationUs` field, which is currently set only on EFR32xG25 for received OFDM packets.

Added in release 3.5.0.0

- Added HFXO temperature compensation in RAIL on platforms that support `RAIL_SUPPORTS_HFXO_COMPENSATION`. This feature can be configured with the new `RAIL_ConfigHFXOCompensation()` API. The user will also need to be sure to handle the new `RAIL_EVENT_THERMISTOR_DONE` event to trigger a call to `RAIL_CalibrateHFXO` to perform the compensation.
- Added options in "RAIL Utility, Protocol" component to control whether Z-Wave, 802.15.4 2.4 GHz and Sub-GHz, and Bluetooth LE are enabled so that the user can save space in their application by disabling unused protocols.
- Added a new API `RAIL_ZWAVE_PerformIrcal` to help perform an IR calibration across all the different PHYs used by a Z-Wave device.
- Added 40 MHz crystal support on EFR32xG24 devices to the "RAIL Utility, Built-in PHYs Across HFXO Frequencies" component.
- Added support for IEEE 802.15.4 fast RX channel switching with the new `RAIL_IEEE802154_ConfigRxChannelSwitching` API on supported platforms (see `RAIL_IEEE802154_SupportsRxChannelSwitching`). This feature allows us to simultaneously detect packets on any two 2.4 GHz 802.15.4 channels with a slight reduction in overall sensitivity of the PHY.
- Added a new Thermal Protection feature, on platforms that support `RAIL_SUPPORTS_THERMAL_PROTECTION`, to track temperature and prevent transmits when the chip is too hot.
- Added new table-based OFDM and FSK PAs for EFR32xG25 based devices. The output power of these can be modified through a new customer provided look-up table. Ask support or look for an updated app note on how to configure the values in this table for your board.
- Added support for the MGM240SA22VNA, BGM240SA22VNA, and BGM241SD22VNA modules and updated the configurations for the BGM240SB22VNA, MGM240SB22VNA, and the MGM240SD22VNA.

4.2 Improvements

Changed in release 3.5.2.0

- Added new `RAIL_ZWAVE_OPTION_PROMISCUOUS_BEAM_MODE` to trigger `RAIL_EVENT_ZWAVE_BEAM` on all beam frames.
- Added `RAIL_ZWAVE_GetBeamHomeIdHash()` to retrieve the beam frame's HomeIdHash when handling that event and made sure that the HomeIdHash byte is now present on PTI for Z-Wave beam frames even when NodeId does not match.

Changed in release 3.5.1.0

- Corrected the sign of the frequency error reported by `RAIL_GetRxFreqOffset()` when using OFDM on the EFR32xG25 to match how this was handled for other modulations (e.g. `Freq_error=current_freq-expected_freq`).
- The `RAIL_SetTune()` and `RAIL_GetTune()` functions now use the `CMU_HFXOCTuneSet()` and `CMU_HFXOCTuneGet()` functions respectively on EFR32xG2x and newer devices.

Changed in release 3.5.0.0

- The `RAIL_ConfigRfSenseSelectiveOokWakeupPhy()` will now return an error when run on the EFR32xG21 platform because this device cannot support the wakeup PHY.
- Updated the `pa_customer_curve_fits.py` helper script to accept floating point value for the maximum power argument, similar to the increment argument.
- Added support in "RAIL Utility, Coexistence" component for configuring priority options when directional priority is enabled but no static priority GPIO is defined.
- Broke up some EFR32xG12 802.15.4 dynamic FEC code to save code size for Zigbee and Bluetooth LE, which never need this functionality.
- Remove "RAIL Utility, Coexistence" component dependency from the RAIL Utility, Coulomb Counter component.
- The `RAIL_PrepareChannel()` function has been made dynamic multiprotocol safe and will no longer return an error if called when your protocol is inactive.

4.3 Fixed Issues

Fixed in release 3.5.3.0

ID #	Description
1058480	Fixed an RX FIFO corruption on EFR32xG25 that occurred when receiving/sending certain OFDM packets using FIFO mode.
1109993	Fixed an issue in the "RAIL Utility, Coexistence" component so that it simultaneously asserts request and priority if request and priority share the same GPIO port and polarity.
1118063	Fixed issue with recent RAIL_ZWAVE_OPTION_PROMISCUOUS_BEAM_MODE on EFR32xG13 and xG14 where the NodId of the promiscuous beam was not properly recorded for RAIL_ZWAVE_GetBeamNodId(), causing it to report 0xFF.
1126343	Fixed an issue on EFR32xG24 when using the IEEE 802.15.4 PHY where the radio could become stuck when doing an LBT transmit if a frame is received during the CCA check window.

Fixed in release 3.5.2.0

ID #	Description
747041	Fixed an issue on the EFR32xG23 and EFR32xG25 that could cause certain radio actions to delay for extended periods of time when the main core enters EM2 while the radio is still running.
1077623	Fixed an issue on EFR32ZG23 where multiple beam frames were lumped together on PTI as one large beam chain.
1090512	Fixed an issue in the "RAIL Utility, PA" component where certain functions would attempt to use the RAIL_TX_POWER_MODE_2P4GIG_HIGHEST macro even though they didn't support it. Previously this resulted in undefined behavior but will now correctly error.
1090728	Fixed a possible RAIL_ASSERT_FAILED_UNEXPECTED_STATE_RX_FIFO issue on EFR32xG12 with RAIL_IEEE802154_G_OPTION_GB868 enabled for a FEC-capable PHY which can happen when aborting a packet at frame detection, for instance by idling the radio.
1092769	Fixed an issue when using Dynamic Multiprotocol and BLE Coded PHYs where a transmit could underflow depending on what protocol was active when the PHY and syncword were loaded.
1103966	Fixed an unexpected Rx packet abort on the EFR32xG25 when using the Wi-SUN OFDM option4 MCS0 PHY.
1105134	Fixed an issue when switching certain PHYs that could cause the first received packet to be reported as RAIL_RX_PACKET_READY_CRC_ERROR instead of RAIL_RX_PACKET_READY_SUCCESS. This issue could potentially impact EFR32xG22 and newer chips.
1109574	Fixed an issue on EFR32xG22 and newer chips where a radio sequencer assert could cause the application to hang in an ISR rather than report the assert via RAILCb_AssertFailed().

Fixed in release 3.5.1.0

ID #	Description
1077611	Fixed an issue on the EFR32xG25 that would cause a 40 μ s porch before an OFDM TX.
1082274	Fixed an issue on the EFR32xG22, EFR32xG23, EFR32xG24, and EFR32xG25 chips that could cause the chip to lock up if the application attempted to re-enter EM2 within ~ 10 μ s after wake-up and hit a <0.5 μ s timing window. If hit, this lockup required a power on reset to restore normal operation to the chip.

Fixed in release 3.5.0.0

ID #	Description
843708	Moved function declarations from rail_features.h to rail.h to avoid a convoluted include dependency order.
844325	Fixed RAIL_SetTxFifo() to properly return 0 (error) rather than 4096 for an undersized FIFO.
845608	Fixed an issue with the RAIL_ConfigSyncWords API when using certain underlying demodulator hardware on EFR32xG2x parts.

ID #	Description
851150	Fixed an issue on EFR32xG2 series devices where the radio would trigger <code>RAIL_ASSERT_SEQUENCER_FAULT</code> when PTI is used and GPIO configuration is locked. GPIO configuration can only be locked when PTI is disabled. See <code>RAIL_EnablePti()</code> for further information.
857267	Fixed an issue when using the "RAIL Utility, Coexistence" component with TX abort, the signal identifier feature and DMP.
1015152	Fixed an issue on EFR32xG2x devices where <code>RAIL_EVENT_RX_FIFO_ALMOST_FULL</code> or <code>RAIL_EVENT_TX_FIFO_ALMOST_EMPTY</code> could trigger improperly when the event is enabled or the FIFO is reset.
1017609	Fixed an issue where PTI appended information could be corrupted when <code>RAIL_RX_OPTION_TRACK_ABORTED_FRAMES</code> is in effect when <code>RAIL_IDLE_FORCE_SHUTDOWN</code> or <code>RAIL_IDLE_FORCE_SHUTDOWN_CLEAR_FLAGS</code> is used. Also clarified that <code>RAIL_RX_OPTION_TRACK_ABORTED_FRAMES</code> is not useful with coded PHYs.
1019590	Fixed an issue when using the "RAIL Utility, Coexistence" component with BLE where the <code>sl_bt_system_get_counters()</code> function would always return 0 for GRANT denied counts.
1019794	Eliminated compiler warning in "RAIL Utility, Initialization" component when few of its features are enabled.
1023016	Fixed an issue on EFR32xG22 and newer chips where waits in between radio activity would consume slightly more power than necessary after the first 13 ms. This was especially noticeable when using <code>RAIL_ConfigRxDutyCycle</code> with large off time values.
1029740	Fixed issue where <code>RAIL_GetRssi()</code> / <code>RAIL_GetRssiAlt()</code> could return a "stale" RSSI value (the value was from previous RX state instead of the current one) if called quickly upon entering receive.
1040814	Added support to the "RAIL Utility, Coexistence" component for configuring the coexistence request priority on sync detect when using BLE.
1056207	Fixed an issue with IQ sampling when using the "RAIL Utility, AoX" component with only 0 or 1 antennas selected.
1062712	Fixed an issue where the "RAIL Utility, Coexistence" component would not always update request states correctly, which could lead to missed events triggered by new requests.
1062940	Prevented the "RAIL Utility, Coexistence" component from aborting BLE transmits when <code>SL_RAIL_UTIL_COEX_BLE_TX_ABORT</code> is disabled.
1063152	Fixed an issue where radio reception would not be fully cleaned up when a receive error occurred with receive state transitions set to idle on error but transmit on success, a configuration mostly associated with BLE. On the EFR32xG24 this could cause a SYNTH calibration to not be properly restored and eventually cause the radio to stop working.

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	

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

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.

Update Preference

WHAT EMAILS WOULD YOU LIKE TO RECEIVE?

Newsletters

- ☐ Community Monthly Newsletter
- ☐ Sales Newsletter
- ☐ Micrium Newsletter

Product Specific Notifications

- ☐ Product Information and Newsletter
- ☒ Software/Security Advisory Notices & Product Change Notices (PCNs)

SELECT THE PRODUCTS TO RECEIVE UPDATES FOR

☐ Select/Unselect All	☐ Power over Ethernet
☐ Audio and Radio	☐ Sensors
☐ Interface	☐ TV and Video
☐ Isolation	☐ Wire
☐ Modems and DMAs	☐ Wireless
☒ Microcontrollers	☐ Bluetooth Classic
☐ 8-bit MCUs	☐ Bluetooth Low Energy
☒ 32-bit MCUs	☒ Proprietary
☐ Timing	☐ WiFi
☐ Clocks	☐ ZigBee and Thread
☐ Buffers	☐ Z-Wave
☐ Oscillators	
☐ CDR and PHY	

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

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project

Trademark Information

Silicon Laboratories Inc.[®], Silicon Laboratories[®], Silicon Labs[®], SiLabs[®] and the Silicon Labs logo[®], Bluegiga[®], Bluegiga Logo[®], EFM[®], EFM32[®], EFR, Ember[®], Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Redpine Signals[®], WiSeConnect, n-Link, ThreadArch[®], EZLink[®], EZRadio[®], EZRadioPRO[®], Gecko[®], Gecko OS, Gecko OS Studio, Precision32[®], Simplicity Studio[®], Telegesis, the Telegesis Logo[®], USBXpress[®], Zentri, the Zentri logo and Zentri DMS, Z-Wave[®], and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



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

www.silabs.com