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):

3.5.1.0 GA released February 1, 2023
3.5.0.0 GA released December 14, 2022

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

- 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 10.3-2021.10, provided with Simplicity Studio.
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.

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>652925</td>
<td>EFR32XG21 is not supported for &quot;Flex (Connect) - SoC Light Example DMP&quot; and &quot;Flex (Connect) - SoC Switch Example&quot;</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>501561</td>
<td>In the Legacy HAL component, the PA configuration is hard-coded regardless of the user or board settings.</td>
</tr>
<tr>
<td>711804</td>
<td>Connecting multiple devices simultaneously may fail with a timeout error.</td>
</tr>
</tbody>
</table>

2.5 Deprecated Items

None

2.6 Removed Items

None
3 RAIL Applications

3.1 New Items

Added in release 3.5.0.0

- XG25 Support
- EFF01 + 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

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mode Switch: MCS Rate selection fix for OFDM.</td>
</tr>
</tbody>
</table>

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 3.5.0.0**

- 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 support for the BGM220SC23HNA2 module.

- 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 3.5.1.0**

- 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 multi-protocol safe and will no longer return an error if called when your protocol is inactive.

4.3 Fixed Issues

**Fixed in release 3.5.1.0**

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1077611</td>
<td>Fixed an issue on the EFR32xG25 that would cause a 40 µs porch before an OFDM TX.</td>
</tr>
</tbody>
</table>
### ID # | Description
--- | ---
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 µs after wake-up and hit a <0.5 µs 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

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>843708</td>
<td>Moved function declarations from rail_features.h to rail.h to avoid a convoluted include dependency order.</td>
</tr>
<tr>
<td>844325</td>
<td>Fixed RAIL_SetTxFifo() to properly return 0 (error) rather than 4096 for an undersized FIFO.</td>
</tr>
<tr>
<td>845608</td>
<td>Fixed an issue with the RAIL_ConfigSyncWords API when using certain underlying demodulator hardware on EFR32xG2x parts.</td>
</tr>
<tr>
<td>851150</td>
<td>Fixed an issue on EFR32xG2 series devices where the radio would trigger RAIL_ASSERT_SEQUENCER_FAULT when PTI is used and GPIO configuration is locked. GPIO configuration can only be locked when PTI is disabled. See RAIL_EnablePti() for further information.</td>
</tr>
<tr>
<td>857267</td>
<td>Fixed an issue when using the &quot;RAIL Utility, Coexistence&quot; component with TX abort, the signal identifier feature and DMP.</td>
</tr>
<tr>
<td>1015152</td>
<td>Fixed an issue on EFR32xG2x devices where RAIL_EVENT_RX_FIFO_ALMOST_FULL or RAIL_EVENT_TX_FIFO_ALMOST_EMPTY could trigger improperly when the event is enabled or the FIFO is reset.</td>
</tr>
<tr>
<td>1017609</td>
<td>Fixed an issue where PTI appended information could be corrupted when RAIL_RX_OPTION_TRACK_ABORTED_FRAMES is in effect when RAIL_IDLE_FORCE_SHUTDOWN or RAIL_IDLE_FORCE_SHUTDOWN_CLEAR_FLAGS is used. Also clarified that RAIL_RX_OPTION_TRACK_ABORTED_FRAMES is not useful with coded PHYs.</td>
</tr>
<tr>
<td>1019590</td>
<td>Fixed an issue when using the &quot;RAIL Utility, Coexistence&quot; component with BLE where the sl_bt_system_get_counters() function would always return 0 for GRANT denied counts.</td>
</tr>
<tr>
<td>1019794</td>
<td>Eliminated compiler warning in &quot;RAIL Utility, Initialization&quot; component when few of its features are enabled.</td>
</tr>
<tr>
<td>1023016</td>
<td>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 RAIL_ConfigRxDutyCycle with large off time values.</td>
</tr>
<tr>
<td>1029740</td>
<td>Fixed issue where RAIL_GetRssi()/RAIL_GetRssiAlt() could return a &quot;stale&quot; RSSI value (the value was from previous RX state instead of the current one) if called quickly upon entering receive.</td>
</tr>
<tr>
<td>1040814</td>
<td>Added support to the &quot;RAIL Utility, Coexistence&quot; component for configuring the coexistence request priority on sync detect when using BLE.</td>
</tr>
<tr>
<td>1056207</td>
<td>Fixed an issue with IQ sampling when using the &quot;RAIL Utility, AoX&quot; component with only 0 or 1 antennas selected.</td>
</tr>
<tr>
<td>1062712</td>
<td>Fixed an issue where the &quot;RAIL Utility, Coexistence&quot; component would not always update request states correctly, which could lead to missed events triggered by new requests.</td>
</tr>
<tr>
<td>1062940</td>
<td>Prevented the &quot;RAIL Utility, Coexistence&quot; component from aborting BLE transmits when SL_RAIL_UTIL_COEX_BLE_TX_ABORT is disabled.</td>
</tr>
<tr>
<td>1063152</td>
<td>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.</td>
</tr>
</tbody>
</table>
### 4.4 Known Issues in the Current Release

Issues in bold were added since the previous release.

<table>
<thead>
<tr>
<th>ID #</th>
<th>Description</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>641705</td>
<td>Infinite receive operations where the frame’s fixed length is set to 0 are not working correctly on the EFR32xG23 series chips.</td>
<td></td>
</tr>
</tbody>
</table>
| 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.

<table>
<thead>
<tr>
<th>Wrapped Key</th>
<th>Exportable / Non-Exportable</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thread Master Key</td>
<td>Exportable</td>
<td>Must be exportable to form the TLVs</td>
</tr>
<tr>
<td>PSKc</td>
<td>Exportable</td>
<td>Must be exportable to form the TLVs</td>
</tr>
<tr>
<td>Key Encryption Key</td>
<td>Exportable</td>
<td>Must be exportable to form the TLVs</td>
</tr>
<tr>
<td>MLE Key</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>Temporary MLE Key</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>MAC Previous Key</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>MAC Current Key</td>
<td>Non-Exportable</td>
<td></td>
</tr>
<tr>
<td>MAC Next Key</td>
<td>Non-Exportable</td>
<td></td>
</tr>
</tbody>
</table>

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.

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!