AN0918.2: Wireless Gecko Series 1 to Series 2 Compatibility and Migration Guide

This porting guide is targeted at migrating an existing design from Wireless Gecko Series 1 to Wireless Gecko Series 2. Both hardware and software migration needs to be considered.

The core and peripherals of Wireless Gecko Series 2 devices are based on the existing MCU and Wireless Series 1 devices with better performance and lower current consumption.

This document will describe which aspects are enhanced in the peripherals common between Series 1 and Series 2. Details for all of the new peripherals of Series 2 can be found in the reference manual, and it is recommended to review the available example code for assistance and recommendations.

All peripherals in the Series 1 and Series 2 devices are described in general terms. Not all modules are present in all devices, and the feature set for each device might vary. Such differences, including pinout, are covered in the device-specific data sheets.

KEY POINTS

- Series 2 have commonalities and enhancements from Series 0 and Series 1 peripherals.
- Software and hardware migration must both be considered when porting from a Series 1 device to Wireless Gecko Series 2 device.
- Series 2 devices are software compatible with the existing Series 1 devices, so only minor changes are required for common peripherals.
- Refer to the example code for specific recommendations and assistance.

1. Device Compatibility

This application note supports multiple device families, and some functionality is different depending on the device.

MCU Series 0 consists of:
- EFM32G
- EFM32GG
- EFM32WG
- EFM32LG
- EFM32TG
- EFM32ZG
- EFM32HG

Wireless MCU Series 0 consists of:
- EZR32WG
- EZR32LG
- EZR32HG

MCU Series 1 consists of:
- EFM32JG1/EFM32JG12
- EFM32PG1/EFM32PG12
- EFM32GG11/EFM32GG12
- EFM32TG11

Wireless Gecko Series 1 consists of:
- EFR32BG1/EFR32BG12/EFR32BG13/EFR32BG14
- EFR32FG1/EFR32FG12/EFR32FG13/EFR32FG14
- EFR32MG1/EFR32MG12/EFR32MG13/EFR32MG14

Wireless Gecko Series 2 consists of:
- EFR32BG21/EFR32BG22
- EFR32FG22
- EFR32MG21/EFR32MG22

2. Compatibility Overview

Four factors must be considered when porting a design from Series 1 to Series 2: pin compatibility, hardware compatibility, software compatibility, and peripheral compatibility.

2.1 Pins and Hardware

Wireless Gecko Series 2 devices are not footprint compatible with Series 1 or Series 0.

More information on footprint and hardware compatibility between Series 1 and Series 2 can be found in 4. Hardware Migration.

2.2 Software and Peripherals

Software compatibility between Series 1 and Series 2 is maintained using emlib and emdrv, which are software abstraction layers built upon the CMSIS (Cortex Microcontroller Software Interface Standard) layer defined by Arm. These devices are not binary compatible, meaning code compiled for Series 1 will not work after being downloaded to Series 2. However, if the software is written using the emlib or emdrv modules, then the application code should not need to change in most cases when recompiling for the new Series 2 target.

Note: There are some small exceptions to full software compatibility across Series 1 and Series 2. For example, wake-up pins and GPIO drive strength are implemented slightly differently on these parts, so the emlib functions have changed slightly to accommodate these differences. Wherever possible, these details have been abstracted by emlib and emdrv. See 5.1 Peripheral Support Library (emlib) and Drivers (emdrv) for more information on compatibility between Series 1 and Series 2. Consult the [**SDK Documentation**] under the [**Getting Started**] tab in Simplicity Studio for more information on the emlib and emdrv modules.

The abstraction provided by emlib and emdrv simplifies peripheral initialization and usage. Version 5.9 and later (provided by Gecko SDK v2.7 and later) of emlib and emdrv support the following peripherals across Series 1 and Series 2:

Table 2.1. Support in emlib and emdrv for Series 1 and Series 2

emlib Peripheral Support							
ACMP	BURTC	CMU	CORE	CRYPTO[1]	DBG	EMU	EUART
GPCRC	GPIO	I2C	IADC	LDMA	LETIMER	MSC	PDM
PRS	RTCC	SYSTEM	TIMER	USART	WDOG	—	—

Note:
1. For CRYPTO, use the mbedTLS library.

emdrv Driver Support							
DMADRV	GPIOINTERRPUT	NVM3	RTCDRV	SLEEP	SPIDRV	TEMPDRV	UARTDRV
USTIMER	—	—	—	—	—	—	—

Because emlib and emdrv modules are common across Series 1 and Series 2, the look and feel of the software development experience is familiar. In other words, developers experienced with Series 1 devices will already know how to construct software and use peripherals on Series 2 devices. In addition, existing Series 1 designs can be quickly ported to new devices to take advantage of new capabilities available on Series 2 by utilizing the common code base between families.

More information on software migration can be found in 5. Software Migration, and more information on peripheral commonalities and differences can be found in 6. Peripherals Common Between Series 1 and Series 2.

3. System Overview

3.1 Core and Memory

This section compares the core and memory of Series 1 with Series 2.

Table 3.1. Core and Memory

Series 1	Series 2	Notes
Core		
Arm Cortex M3 and M4 with FPU	Arm Cortex M33	—
Debug Interface		
The 2-pin serial-wire debug (SWD) interface or a 4-pin Joint Test Action Group (JTAG) interface.	The 2-pin serial-wire debug (SWD) interface or a 4-pin Joint Test Action Group (JTAG) interface.	Details of the debug interface (SWD, JTAG, and ETM) on each device might vary; differences are covered in the device-specific data sheets.
DMA Controller (DMA)		
Linked DMA Controller (LDMA)	Linked DMA Controller (LDMA)	The number of channels on each device can vary; differences are covered in the device-specific data sheets. DMADRV can assist with family differences.
Flash Program Memory		
128 - 2048 kB	Up to 1024 kB	Series 2 flash words are 64 bits wide vs. 32 bits on Series 1 devices. Available flash memory among device family members can vary; differences are covered in the device-specific data sheets.
RAM Memory		
MCU Series 1: 32 - 512 kB Wireless Gecko Series 1: 16 - 256 kB	Up to 96 kB	Series 2 RAM supports ECC and may require wait states at higher operating frequencies. RAM features and capability can vary among devices; differences are covered in the device-specific data sheets and family reference manuals.

4. Hardware Migration

4.1 Pin Compatibility

Series 2 Wireless SoC devices are not pin compatible with any Series 0 or Series 1 SoC or MCU.

4.2 Series 1 Peripheral Migration

The following table lists Series 1 peripherals and, if they exist, their equivalents on Series 2 devices. The RF portions of Series 2 devices are not included in this comparison.

Table 4.1. Series 1 Peripheral Functionality

Series 1	Series 2	Notes
Analog Interfaces		
Analog to Digital Converter (ADC)	Incremental Analog to Digital Converter (IADC)	No change to system hardware is required.
Analog Port (APORT)	Analog Bus (ABUS)	No change to system hardware is required.
Capacitive Sense Module (CSEN)	—	This functionality is not available on Series 2.
Operational Amplifier (OPAMP)	—	This functionality is not available on Series 2.
Voltage Digital to Analog Converter (VDAC)	—	This functionality is not available on Series 2.
Current Digital to Analog Converter (IDAC)	—	This functionality is not available on Series 2.
Digital Pin Routing		
I/O ROUTE Registers	Digital Bus (DBUS)	DBUS is a switch matrix between peripheral resources and GPIO pins used for signal enabling and routing on Series 2 devices. No change to system hardware is required.
Energy Management		
RTCC Retention RAM	Backup RAM (BURAM)	No change to system hardware is required.
External Storage		
External Bus Interface (EBI)	—	This functionality is not available on Series 2.
Quad/Octal SPI Flash Interface (QSPI)	—	This functionality is not available on Series 2.
SDIO Host Controller (SDIO)	—	This functionality is not available on Series 2.
Serial Interfaces		
Universal Serial Bus Controller (USB)	—	This functionality is not available on Series 2.
Low Energy Universal Asynchronous Receiver/Transmitter (LEUART)	Enhanced Universal Asynchronous Receiver/Transmitter (EUART)	No change to system hardware is required.
Timers and Triggers		
Real Time Counter and Calendar (RTCC)	Real Time Clock with Capture (RTCC)	No change to system hardware is required.
Ultra Low Energy Timer/Counter (CRYOTIMER)	Backup Real Time Clock (BURTC)	No change to system hardware is required.
32-bit General Purpose Timer (WTIMER)	TIMER	Only TIMER0 supports 32-bit operation.
Pulse Counter (PCNT)	—	This functionality is not available on Series 2.
Low Energy Sensor Interface (LESENSE)	—	This functionality is not available on Series 2.

Series 1	Series 2	Notes
Security		
True Random Number Generator (TRNG) & CRYPTO	Secure Element (SE)	The SE implements CRYPTO and TRNG functionality on EFR32xG21. No hardware changes are required.
	Cryptographic Accelerator (CRYPTOACC)	The CRYPTOACC implements CRYPTO and TRNG functionality on EFR32xG22. No hardware changes are required.

5. Software Migration

Series 2 devices are software compatible with Series 1 devices, so only minor changes are required for peripherals that are common to Series 1 and Series 2 (especially when enhancements and new features are not used).

5.1 Peripheral Support Library (emlib) and Drivers (emdrv)

Initialization and basic peripheral functionality are handled using the emlib low-level peripheral support library, the API for which abstracts the register-level differences between Series 1 and Series 2 devices. Use of emlib simplifies the porting of firmware among devices in the same family and between Series 1 and Series 2.

The emlib modules are found under the Simplicity Studio installation path. The default location on Windows is:

```
C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\vX.Y\platform\emlib
```

While emlib handles initialization of and low-level data movement to and from peripherals, emdrv provides driver-level functionality, such as sending or receiving strings of serial data or instantiating one-shot or periodic timer events. Each emdrv component has an API that is identical across all EFM32 and EFR32 devices and is optimized for speed and power by using DMA and running in the lowest allowable energy mode. Because it maintains API consistency, emdrv does not take advantage of enhancements that are made to common peripherals on new devices. Although they are unavailable through emdrv, these features are generally accessible through emlib's initialization functions and can often simplify software development or reduce energy use.

The emdrv modules are found under the Simplicity Studio installation path. The default location on Windows is:

```
C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\vX.Y\platform\emdrv
```

Developing software with the emlib and emdrv libraries is highly recommended because they provide software compatibility across devices. API availability on and differences between Series 1 and Series 2 are discussed below.

Table 5.1. Software Migration Checklist

Series 1	Series 2	Notes
API for DCDC power configurations in `em_emu.c`.	API for DCDC power configuration remains in `em_emu.c` for EFR32xG22. It is unused on EFR32xG21, which does not have a DCDC converter.	—
API for HFXO startup in `em_cmu.c`. **emlib:** Common API for HFXO startup initialization.	API for HFXO startup in `em_cmu.c`. **emlib:** Common API for HFXO startup initialization.	Configure the HFXO for safe crystal startup with: `void CMU_HFXOInit(const CMU_HFXOInit_TypeDef *hfxoInit)`
API for LFXO startup in `em_cmu.c`. **emlib:** Common API for LFXO startup initialization.	API for LFXO startup in `em_cmu.c`. **emlib:** Common API for LFXO startup initialization.	Configure the LFXO for safe crystal startup with: `void CMU_LFXOInit(const CMU_LFXOInit_TypeDef *lfxoInit)`
API for HFRCO band selection in `em_cmu.c` on Series 1 devices. **emlib:** `void CMU_HFRCOBandSet(CMU_HFRCOFreq_TypeDef setFreq)`	API for HFRCODPLL band selection in `em_cmu.c` on Series 2 devices. **emlib:** `void CMU_HFRCODPLLBandSet(CMU_HFRCODPLLFreq_TypeDef freq)`	The HFRCO on Series 1 devices is replaced with the HFRCODPLL on Series 2 devices. The HFRCODPLL API in `em_cmu.c` implements an enumerated type to specify one of several predefined frequency bands that are a superset of those supported on Series 1 devices. For example, `CMU_HFRCOBandSet(cmuHFRCOFreq_38M0Hz)` is be replaced with `CMUHFRCODPLLBandSet(cmuHFRCODPLLFreq_38M0Hz)` on a Series 2 device. Although the register interface is similar, code that tunes the HFRCO to operate at a frequency other than one of the factory-calibrated presets must be rewritten for use with the HFRCODPLL. Alternatively, the DPLL API in `em_cmu.c` can be used to run at a customer-designated frequency on Series 2 devices without the need calibrate for iterative tuning, as is the case Series 1 devices.

Series 1	Series 2	Notes
Peripheral-specific ROUTEPEN/ROUTELOCn registers are used to enable signals and route them to physical pins on Series 1 devices. **emlib:** There is no emlib API to enable and route peripheral signals to physical pins. The ROUTEPEN and ROUTELOC definitions of digital peripherals can be found in corresponding device header files. For example, to enable SCL and SDA for I2C0 and route them to the location 15 pins: `I2C0->ROUTEPEN = I2C_ROUTEPEN_SDAPEN	I2C_ROUTEPEN_SCLPEN;I2C0->ROUTELOC0 = (I2C0->ROUTELOC0 & (~_I2C_ROUTELOC0_SDALOC_MASK))	I2C_ROUTELOC0_SDALOC_LOC4;I2C0->ROUTELOC0 = (I2C0->ROUTELOC0 & (~_I2C_ROUTELOC0_SCLLOC_MASK))	I2C_ROUTELOC0_SCLLOC_LOC4;`	GPIO registers are used to enable the DBUS signals for digital peripherals and route them to specific pins on Series 2 devices. **emlib:** There is no emlib API to enable and route peripheral signals to physical pins. To pin enable and route digital peripheral, first set the required port/pin bits of GPIO_x_yROUTE register, where x is the peripheral name and y is the resource name. Then set GPIO_x_ROUTEEN register, where x is the peripheral name. For example, to enable SCL and SDA for I2C0 and route them to pins PA5 and PA6: `GPIO->I2CROUTE[0].SCLROUTE = (GPIO->I2CROUTE[0].SCLROUTE & ~_GPIO_I2C_SCLROUTE_MASK)	(gpioPortA << _GPIO_I2C_SCLROUTE_PORT_SHIFT	(5 << _GPIO_I2C_SCLROUTE_PIN_SHIFT));` `GPIO->I2CROUTE[0].SDAROUTE = (GPIO->I2CROUTE[0].SDAROUTE & ~_GPIO_I2C_SDAROUTE_MASK)	(gpioPortA << _GPIO_I2C_SDAROUTE_PORT_SHIFT	(6 << _GPIO_I2C_SDAROUTE_PIN_SHIFT));` `GPIO->I2CROUTE[0].ROUTEEN = GPIO_I2C_ROUTEEN_SDAPEN	GPIO_I2C_ROUTEEN_SCLPEN;`	Peripheral route enable, port select, and pin select registers can be found in the GPIO section of the reference manual.

The APORT is used to specify analog peripheral inputs and output on Series 1 devices.
 	
emlib:
 	
An analog input or output, when selectable, is specifed by the appropriate initializer function for a given peripheral.
 	
APORT definitions for selectable analog inputs and outputs can be found in the corresponding peripheral's emlib header file.
 	
For example, to use APORT BUS1X channel 6 as an ADC input:
`ADC_InitSingle_TypeDef singleInit = ADC_INITSINGLE_DEFAULT;`
`singleInit.posSel = adcPosSelAPORT1XCH6;`
	The ABUS is used to specify analog peripheral inputs and output on Series 2 devices.
	
	emlib:
	
	Bus allocation to analog peripherals is managed by fields in the GPIO_ABUSxALLOC registers. Selection of a specific port and pin is managed by the peripheral because it is possible for more than one bus to be assigned to a single peripheral.
	
	For example, to allow ACMP0 to use PD3 as its positive input:
	`GPIO->CDBUSALLOC = GPIO_CDBUSALLOC_CDODD0_ACMP0;ACMP_ChannelSet(ACMP0, acmpInputVREFDIV1V25, acmpInputPD3);`
		There are three analog buses on the EFR32xG21 and EFR32xG22: one dedicated to Port A (ABUSA), one dedicated to port B (ABUSB), and one that serves both ports C and D (ABUSCD). Up to two analog peripherals may be given access to an ABUS at any one time and the even/odd pins of each bus are configured independently. This means that a single bus may have up to four different analog peripherals connected to it and using (different) pins simultaneously.
		
		Bus allocation registers are found in the GPIO section of the reference manual.

Series 1	Series 2	Notes

APIs for GPIO in `em_gpio.c` (emlib) and `GPIONT` (emdrv).
 	
emlib and emdrv:
Common APIs for GPIO pin and external interrupt configuration and selection.
	APIs for GPIO in `em_gpio.c` (emlib) and `GPIONT` (emdrv).
	
	emlib and emdrv:
	Common APIs for GPIO pin and external interrupt configuration and selection.
		The emlib code to configure PC6 (interrupt source 6) as a falling edge interrupt is the same on both Series 1 and Series 2:
		
		`GPIO_PinModeSet(gpioPortC, 6, gpioModeInputPull, 1);`
		
		`GPIO_ExtIntConfig(gpioPortC, 6, 6, false, true, true);`
		
		`GPIOINT` (emdrv) is a callback dispatcher and, apart from clearing interrupt flags, is not dependent on the underlying hardware for configuration.

On Series 1 devices, Gecko bootloader contains two stages (first stage bootloader and main bootloader). To enable bootloader functionality, Gecko Bootloader must be configured and programmed into the dedicated bootloader region of flash, if applicable. The main bootloader (second stage) is upgradable through the first stage bootloader.	Gecko Bootloader is also used on Series 2 devices. To enable bootloader functionality, the second stage (main bootloader) must be configured and programmed into the first 16 KB of flash. The main bootloader is upgradable through the Secure Loader, firmware maintained by Silicon Labs and pre-programmed into flash at the factory.	See application note, AN1218: Series 2 Secure Boot with RTSL and UG266: Gecko Bootloader User's Guide for more information.
APIs for IDAC and VDAC in `em_idac.c` and `em_vdac.c`, respectively (both in emlib).	—	Neither the IDAC nor the VDAC is currently present on Series 2 devices.
API for VMON in `em_emu.c` (emlib).	—	VMON is not present on Series 2 devices. On EFR32xG21, the VSENSE feature of the ACMP can be used to monitor the AVDD and DVDD supplies. For more customized monitoring (or, in the case of EFR32xG22, which does not have the ACMP module), the IADC's window comparison unit can generate interrupts in response to sample results that are less than or greater than specific threshold values.
Cryptographic operations on Series 1 are implemented using the mbedTLS library and accelerated in hardware by the CRYPTO and TRNG modules.	Cryptographic operations on Series 2 are implemented using the mbedTLS library and accelerated in hardware by the SE (EFR32xG21) or CRYPTOACC (EFR32xG22) modules.	The mbedTLS library is found under the Simplicity Studio installation path. The default location on Windows is: `C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\vX.Y\util\third_party\mbedtls`

Series 1	Series 2	Notes
APIs for RTCC in `em_rtcc.c` (emlib), `RTCDRV` (emdrv), and `SLEEPTIMER` (platform services).	APIs for RTCC in `em_rtcc.c` (emlib), `RTCDRV` (emdrv), and `SLEEPTIMER` (platform services).	Low-level management of the RTCC can be implemented with the API in `em_rtcc.c`. `RTCDRV` provides low-frequency (millisecond-level) one-shot and periodic timers by abstracting the RTCC hardware but is now deprecated in favor of the `SLEEPTIMER` platform service. `SLEEPTIMER`implements the same functionality as `RTCDRV`. It maintains a continuous 64-bit tick counter upon driver initialization and can return a UNIX-style time stamp (seconds elapsed since January 1, 1970, 00:00:00). Functions are provided for conversion between the UNIX time stamp and ZigBee and NTP time formats. All three APIs are available on Series 1 and Series 2 devices.
API for LDMA in `em_ldma.c` (emlib) and `DMADRV` (emdrv). **emlib and emdrv:** Common APIs for configuring and managing the LDMA and transfers.	API for LDMA in `em_ldma.c` (emlib) and `DMADRV` (emdrv). **emlib and emdrv:** Common APIs for configuring and managing the LDMA and transfers.	The `em_ldma.c` and `DMADRV` APIs are common across Series 1 and Series 2 devices.

6. Peripherals Common Between Series 1 and Series 2

6.1 Core, Memory, and Bus System

The major changes are the switch to the Cortex-M33 core and, on some devices, the addition of the Secure Element (SE).

Table 6.1. Core, Memory, and Bus System

Series 1	Series 2	Notes
New Features		
—	Secure Memory	Secure memory prevents secure addresses from being accessed by unauthorized code or peripherals.
—	Backup RAM	Backup RAM provides 128 bytes of low power RAM that is retained in EM4.
—	Flash Lock	Unlike Series 0 and 1, Series 2 devices do not have a lock bits page in flash. Instead, Series 2 devices have a register-based flash locking mechanism where, as on earlier devices, each bit corresponds to a single flash page. The MSC_PAGELOCKn registers are cleared after reset. Writing a given bit to 1 can be done one time after reset by the CPU or the Secure Element and prevents the designated flash page from being programmed or erased. Like other flash pages, the user data page can be locked via a one time write that sets the UDLOCK bit in the MSC_MISCLOCKWORD register. When the Secure Element is present on a device, only it is allowed to write to the user data page, which, by consequence, is locked upon reset.
—	Secure Debug Unlock	The Secure Element provides a secure debug unlock function that allows users to grant debug access to locked devices on a device by device basis. See the application note, AN1190: EFR32xG21-22, for more information.
Enhancements		
Cortex-M3 or Cortex-M4 (ARMv7-M)	Cortex-M33 (ARMv8-M)	Inclusion of the single-precision Floating Point (FPU) and Memory Protection (MPU) Units effectively makes the Cortex-M33 a superset of the Cortex-M4. ARM's TrustZone security provides dedicated hardware resources to permit the execution of secure and non-secure code. Refer to Arm's architecture, core, and security documentation for more details.

6.2 Clock Management Unit (CMU)

The major changes are the on demand oscillators, synchronous registers, and new oscillator names.

Table 6.2. CMU

Series 1	Series 2	Notes
New Features		
—	FSRCO (20 MHz)	The Fast Start RC Oscillator is a fixed frequency, low energy oscillator with a short start up time. This oscillator is used after reset.
—	Digital PLL for the HFRCO enables arbitrary clock frequency generation	See the device-specific reference manual for more information on the DPLL.
Enhancements		
1 MHz – 72 MHz HFRCO (1, 2, 4, 7, 13, 16, 19, 26, 32, 38, 48, 56, 64 and 72 MHz), HFRCO is 19 MHz after reset.	1 MHz – 80 MHz HFRCODPLL	The HFRCO on Series 2 devices has selectable tuning bands that are a superset of those available on Series 1 devices. The DPLL is available for use in EM0/1 and can be set to arbitrary frequencies as permitted by the M and N dividers.
1 MHz – 50 MHz AUXHFRCO (1, 2, 4, 7, 13, 16, 19, 26, 32, 38, 48 and 50 MHz), AUXHFRCO is 19 MHz after reset	1 MHz – 38 MHz HFRCOEM23	Oscillator is enabled on demand in energy modes EM2 and EM3 (EFR32xG21). Not all Series 2 devices include the HFRCOEM23.
DBGCLK	TRACECLK	—
Three clocks (HFPERCLK, HFPERBCLK, and HFPERCCLK) drive all high-frequency peripherals.	Peripheral clocks (PCLK) are asynchronous to the main system clock (HCLK).	Peripheral clocks are enabled by specific bits in the CMU_CLKENn registers. On EFR32xG21, peripheral clocks are requested by setting the EN bit in the peripheral's EN register.
HFPERCLK	PCLK, LSPCLK, EM01GRPA, EM01GRPB	PCLK is the standard peripheral clock. EM01GRPA and EM01GRPB are selectable by certain peripherals to permit asynchronous operation relative to the (PCLK). LSPCLK permits operation at lower frequency in order to reduce energy use for peripherals that do not need the highest frequency clock available.
LFA, LFE	EM23GRPA, EM4GRPA	Seperate clocks selectable by peripherals capable of operation in EM2/3 or EM4. This allows peripherals that can operate in EM2/3 to use the LFXO or LFRCO as a clock source, while peripherals capable of operation in EM4 can use the ULFRCO.

6.3 Energy Management

6.3.1 Energy Management Unit (EMU)

The major change is the new single EM4 energy mode.

Table 6.3. EMU

Series 1	Series 2	Notes
Enhancements		
EM4 is separated into hibernate (EM4H) and shutdown (EM4S) modes with certain peripherals available in EM4H not available in EM4S for further reduction in energy use.	Single EM4 Mode	EM4 with one of the low frequency clocks running has much lower current draw on Series 2 devices than does the equivalent peripheral running in EM4H on Series 1 devices. EM4 mode with clocks shutdown on Series 2 devices is comparable to EM4S on Series 1 devices.
Limitations		
—	DCDC availability is device-dependent	EFR32xG22 devices have the DCDC converter as a standard feature. EFR32xG21 devices do not have a DCDC converter.

6.3.2 Reset Management Unit (RMU)

RMU functionality has been folded into the EMU on Series 2 devices. Many reset sources can now be individually enabled or disabled.

Table 6.4. RMU

Series 1	Series 2	Notes
New Features		
Brown-out Detector (BOD) resets cannot be masked.	All BODs can be individually enabled and masked.	—
Enhancement		
BODs for analog unregulated supply (AVDD), digital unregulated supply (DVDD), and regulated digital supply (DECOUPLE).	BODs for AVDD, IOVDD, DECOUPLE (under and over voltage), DVDD in EM0/EM1, and DVDD in EM2/EM3/EM4.	—

6.4 Serial Interfaces

6.4.1 Inter-Integrated Circuit Interface (I2C)

There are no major changes to the I2C module. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

Table 6.5. I2C

Series 1	Series 2	Notes
Enhancements		
—	I2C0 is available in EM0/EM1 and can be available EM2/3. I2C1 in available only EM0/EM1.	—

6.4.2 Universal Synchronous Asynchronous Receiver/Transmitter (USART)

There are no major changes to the USART module. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

6.4.3 Enhanced Universal Asynchronous Receiver/Transmitter (EUART)

The EUART on Series 2 devices replaces the Low-Energy UART (LEUART) on Series 1 devices. It maintains the core functionality of the LEUART and adds features for high-frequency operation found in the USART. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

Table 6.6. EUART

Series 1	Series 2	Notes
Enhancements		
—	Low-frequency EM2 operation at 9600 baud from 32.768 kHz. High-frequency operation in EM0/1 at programmable baud rates derived from PCLK ÷ 4/6/8/16.	As with the LEUART, low-frequency EUART operation imposes certain limitations (e.g. no receive oversampling).
—	FIFOs increased from 2 to 4 words (8/9 bits).	—
—	IrDA support	—
—	Autobaud in EM0/1 upon reception of 0x55 (standard) or 0x00 (IrDA).	—
—	Hardware flow control.	—

6.4.4 Pulse Density Modulation (PDM) Interface

Apart from the addition of a second pair of channels for a total of four, PDM functionality is effectively unchanged between Series 1 and Series 2. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

6.5 I/O Ports

6.5.1 General Purpose Input/Output (GPIO)

Major changes to GPIO are the additions of DBUS pin routing, the relocation of all pin routing and enabling registers from peripherals to the GPIO register space, and the removal of 5 V tolerant inputs. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

Table 6.7. GPIO

Series 1	Series 2	Notes
New Features		
—	DBUS	DBUS is an any-to-any switch matrix that routes digital peripherals to pins. See the reference manual for the device in question more information.
Limitations		
—	Port C and port D only permit state retention in EM2/EM3 and do not support low-frequency peripheral operation.	Only port A and port B pins can be used in EM2/3 for low-frequency peripherals, such as the EUART, or GPIO inputs intended to serve as wake-up interrupts.
5 V tolerant pins.	Inputs are not 5 V tolerant. All I/O pins support a maximum input voltage of IOVDD + 0.3 V.	—

6.6 Timers and Triggers

6.6.1 Timer/Counter (TIMER)

While some Series 1 devices implement both 32- and 16-bit wide timers named WTIMER and TIMER, respectively, Series 2 simply uses TIMER for all module instances, regardless of width. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

Table 6.8. TIMER

Series 1	Series 2	Notes
Enhancements		
Separate 32-bit (WTIMER) and 16-bit (TIMER) modules.	Single TIMER module regardless of width.	By current convention, TIMER0 is 32 bits wide and all other instances are 16 bits wide.
Limitation		
Timers are synchronized to HFPERCLK.	Timers use the EM01GRPA clock and can be synchronous or asynchronous to the HCLK/PCLK.	—

6.6.2 Low Energy Timer (LETIMER)

LETIMER functionality is enhanced in several aspects. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

Table 6.9. TIMER

Series 1	Series 2	Notes
Enhancements		
16- bit down counter.	24-bit down counter.	By current convention, TIMER0 is 32 bits wide and all other instances are 16 bits wide.
—	8-bit prescaler.	—
Limitation		
Only one comparator (LETIMER_COMP1) available if LETIMER_COMP0 is used as the counter top value.	Dedicated LETIMER_TOP register permits use of LETIMER_COMP[1:0] comparators.	—

6.6.3 Real Time Clock with Capture (RTCC)

The Real Time Counter and Calendar (RTCC) on Series 1 has been replaced by the Real Time Clock with Capture (RTCC) on Series 2. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

Table 6.10. RTCC

Series 1	Series 2	Notes
Limitations		
Hardware calendar mode.	No calendar mode.	—
128 bytes of state retention RAM	State retention RAM is no longer part of RTCC.	The BURAM provides 128 bytes of low-power RAM that is retained in EM4.
Operates down to EM4H.	The RTCC is only available down to EM3 on Series 2 devices.	Use the BURTC for low-power timekeeping in EM4.
Oscillator failure detection logic resides in the RTCC.	Oscillator failure detection logic resides in the LFXO.	Set the LFXO_CTRL_FAILDETEM4WUEN bit to enable wake-up from EM4 upon LFXO failure.

6.6.4 Peripheral Reflex System (PRS)

The major change to the PRS is the addition of four high-speed synchronous channels explicitly for synchronizing IADC and TIMER events. Like other Series 2 peripheral modules, a hardware revision register has been added along with the peripheral register set/clear/toggle address aliases.

Table 6.11. PRS

Series 1	Series 2	Notes
New Features		
12 synchronous/asynchronous channels.	12 asynchronous and 4 synchronous channels dedicated to the IADC and TIMERs.	Synchronous channels support synchronous operation of the IADC with timer triggers on Series 2.
OR with the previous channel and AND with the next channel logic operation.	Expanded set of logic operations that take the previous and current channels as inputs.	—

6.6.5 Watchdog Timer (WDOG)

WDOG functionality is effectively unchanged between Series 1 and Series 2. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

6.7 Analog Interfaces

6.7.1 Analog Comparator (ACMP)

ACMP functionality is effectively unchanged between Series 1 and Series 2. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

6.7.2 Incremental Analog to Digital Converter (IADC)

The ADC has been replaced by the IADC, which has significant functional enhancements and differences. See the application note, AN1189: Incremental Analog to Digital Converter (IADC), and the device reference manual for more information. Some of the notable differences are listed below. Like all Series 2 peripheral modules, dedicated enable and hardware revision registers have been added and the individual interrupt flag set/clear registers have been removed since their functionality is duplicated by the peripheral register set/clear/toggle address aliases.

Table 6.12. IADC

Series 1	Series 2	Notes
New Features		
—	Two sets of converter setup, scaling, and scheduling registers can be independently selected for single conversions or multi-channel scans.	Configuration changes require a 5 µs warm-up delay.
Enhancements		
Traditional SAR ADC.	The Incremental ADC combines SAR and Delta-Sigma techniques.	Operates with 12-bit resolution and achieves 11 ENOB at 1 Msps with 2x and 12 ENOB at 555 ksps with 4x oversampling.
Uniform input gain.	Selectable 0.5x, 1x, 2x, 3x, or 4x input gain.	—
Limitations		
ADC_CLK can range from 32 kHz to 16 MHz.	ADC_CLK can range from 32 kHz to 10 MHz.	Maximum ADC_CLK frequency is reduced to 5 MHz, 3.3 MHz and 2.5 MHz for 2x, 3x and 4x input gain, respectively.

7. Revision History

Revision 0.2

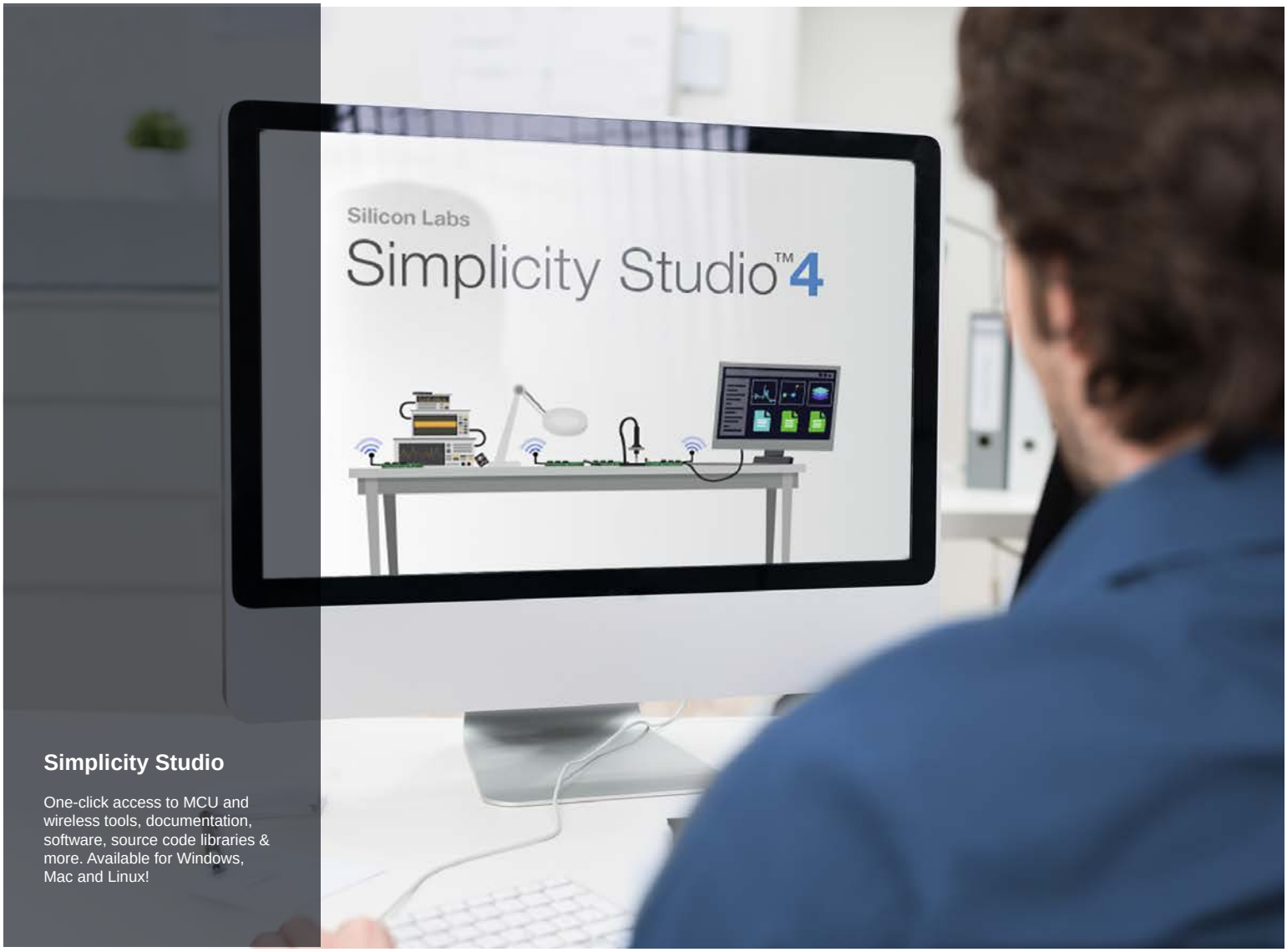
February, 2020

- EFR32xG22 devices added to 1. Device Compatibility.
- 2.2 Software and Peripherals updated to reflect new modules on and versions of emlib and emdrv that add support for EFR32xG22.
- Requirement for RAM wait states made conditional in Table 3.1 Core and Memory on page 4.
- RTCC Retention RAM and EUART added to 4.2 Series 1 Peripheral Migration
- I/O Ports section in Table 4.1 Series 1 Peripheral Functionality on page 6 changed to External Storage and expanded to include QSPI and SDIO.
- CRYPTOACC added to Security section in Table 4.1 Series 1 Peripheral Functionality on page 6.
- Updates to the Notes throughout 6. Peripherals Common Between Series 1 and Series 2.
- Differences in user data page lock functionality added to 6.1 Core, Memory, and Bus System.
- DCDC support added on EFR32xG22 in 6.3.1 Energy Management Unit (EMU).
- 6.4.3 Enhanced Universal Asynchronous Receiver/Transmitter (EUART) added under 6.4 Serial Interfaces.
- 6.4.4 Pulse Density Modulation (PDM) Interface added under 6.4 Serial Interfaces.
- 6.5.1 General Purpose Input/Output (GPIO) updated to reflect the type of functionality supported by ports A/B vs. ports C/D in EM2/3.
- 6.6.2 Low Energy Timer (LETIMER) updated with Series 2 enhancements relative to Series 1.
- Oscillator failure detection differences explained in 6.6.3 Real Time Clock with Capture (RTCC).
- Expanded logical operations functionality added to 6.6.4 Peripheral Reflex System (PRS).
- Input gain selection added to 6.7.2 Incremental Analog to Digital Converter (IADC).

Revision 0.1

February, 2019

- Initial revision.



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 and Community
community.silabs.com

SILICON LABS

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

http://www.silabs.com