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

This porting guide is targeted at migrating an existing design from 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.
- The 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:
- EFM32 Gecko (EFM32G)
- EFM32 Giant Gecko (EFM32GG)
- EFM32 Wonder Gecko (EFM32WG)
- EFM32 Leopard Gecko (EFM32LG)
- EFM32 Tiny Gecko (EFM32TG)
- EFM32 Zero Gecko (EFM32ZG)
- EFM32 Happy Gecko (EFM32HG)

Wireless MCU Series 0 consists of:
- EZR32 Wonder Gecko (EZR32WG)
- EZR32 Leopard Gecko (EZR32LG)
- EZR32 Happy Gecko (EZR32HG)

MCU Series 1 consists of:
- EFM32 Jade Gecko (EFM32JG1/EFM32JG12)
- EFM32 Pearl Gecko (EFM32PG1/EFM32PG12)
- EFM32 Giant Gecko 11 (EFM32GG11)
- EFM32 Giant Gecko 12 (EFM32GG12)
- EFM32 Tiny Gecko 11 (EFM32TG11)

Wireless SoC Series 1 consists of:
- EFR32 Blue Gecko (EFR32BG1/EFR32BG12/EFR32BG13/EFR32BG14)
- EFR32 Flex Gecko (EFR32FG1/EFR32FG12/EFR32FG13/EFR32FG14)
- EFR32 Mighty Gecko (EFR32MG1/EFR32MG12/EFR32MG13/EFR32MG14)

Wireless Gecko Series 2 consists of:
- EFR32 Blue Gecko (EFR32BG21)
- EFR32 Mighty Gecko (EFR32MG21)

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 is maintained using emlib and emdrv, which are software libraries 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 away by the emlib and emdrv modules. See 5.1 Peripheral Support Library (emlib) and energyAware 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 emlib and emdrv modules provide abstraction layers that make peripheral initialization and usage simple and easy. Version 5.6.0 or above of the emlib and emdrv modules support the following peripherals across Series 1 and Series 2:

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

Peripherals Supported by emlib							
ACMP	AES[1]	BURTC	CMU	CORE	CRYPTO[1]	DBG	EMU
GPCRC	GPIO	I2C	INT	LDMA	LETIMER	MSC	PRS
RTCC	SYSTEM	TIMER	USART	WDOG	—	—	—
Note: 1. For AES and CRYPTO, use the mbedTLS library.							

The emdrv Modules							
DMADRV	EZRADIODRV	GPIOINTERR-PUT	NVM3	RTCDRV	SLEEP	SPIDRV	TEMPDRV
UARTDRV	USTIMER	—	—	—	—	—	—

Since the 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 the Series 1 products will already know how to construct software and use peripherals on the Series 2 products. In addition, existing Series 1 designs can be quickly ported to new products to take advantage of new capabilities available on the Series 2 by utilizing the common code base between the families.

More information on software migration can be found in 5. Software Migration, and more information on the 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.	Debug interface (SWD, JTAG, and ETM) for each device might vary, such differences are covered in the device-specific data sheets.
DMA Controller (DMA)		
Linked DMA Controller (LDMA)	Linked DMA Controller (LDMA)	Number of DMA channels for each device might vary, such differences are covered in the device-specific data sheets. The dmadrv module can also be used to assist with family differences.
Flash Program Memory		
128 - 1024 kB	Up to 1024 kB	Flash word for Series 2 devices is 64 bits. Flash program memory for each device might vary, such differences are covered in the device-specific data sheets.
RAM Memory		
MCU Series 1: 32 - 256 kB Wireless SoC Series 1: 16 - 256 kB	Up to 96 kB	Series 2 has a ECC RAM with wait states at 80MHz. RAM memory for each device might vary, such differences are covered in the device-specific data sheets.

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 Peripherals Migration

The following table describes how to migrate Series 1 peripherals that are not available in Series 2. The RF portions of Series 2are not included in this comparison.

Table 4.1. Series 1 Only Peripherals Migration

Series 1	Series 2	Notes
Analog Interfaces		
Analog to Digital Converter (ADC)	Incremental Analog to Digital Converter (IADC)	Hardware change is not required.
Analog Port (APORT)	Analog Bus (ABUS)	Hardware change is not required.
Capacitive Sense Module (CSEN)	—	This functionality is not available on EFR32xG21.
Operational Amplifier (OPAMP)	—	This functionality is not available on EFR32xG21.
Voltage Digital to Analog Converter (VDAC)	—	This functionality is not available on EFR32xG21.
Current Digital to Analog Converter (IDAC)	—	This functionality is not available on EFR32xG21.
Digital Interface		
I/O ROUTE Registers	Digital Bus (DBUS)	DBUS is a switch matrix between peripheral resources and GPIO pins which is used for pin enabling and routing of digital peripherals in Series 2. Hardware change is not required.
Energy Mnagement		
—	Backup RAM (BURAM)	—
I/O Ports		
External Bus Interface (EBI)	—	This functionality is not available on EFR32xG21.
Serial Interfaces		
Universal Serial Bus Controller (USB)	—	This functionality is not available on EFR32xG21.
Low Energy Universal Asynchronous Receiver/Transmitter (LEUART)	—	This functionality is not available on EFR32xG21.
Timers and Triggers		
Real Time Counter and Calendar (RTCC)	Real Time Clock with Capture (RTCC)	Hardware change is not required.
Ultra Low Energy Timer/Counter (CRYOTIMER)	Backup Real Time Clock (BURTC)	Hardware change is not required.
32 bit General Purpose Timer (WTIMER)	TIMER	TIMER0 is supports 32 bit operation.
Pulse Counter (PCNT)	—	This functionality is not available on EFR32xG21.
Low Energy Sensor Interface (LESENSE)	—	This functionality is not available on EFR32xG21.
Security		
True Random Number Generator (TRNG) & CRYPTO	Secure Element (SE)	CRYPTO operations and TRNG functionality are incorporated in SE for Series 2. Hardware change is not required.

5. Software Migration

The Series 2 devices are software compatible with the existing 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 energyAware Drivers (emdrv)

The Peripheral Support Library (emlib) abstracts the differences between Series 1 and Series 2 through the API, so software migration becomes transparent to the firmware author.

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
```

The energyAware Drivers Library (emdrv) is optimized for speed and power consumption while maintaining API compatibility between different product families.

The available peripherals' 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
```

It is highly recommended to develop peripherals' software with emlib or emdrv since it provides software compatibility across Series 1 and Series 2.

Table 5.1. Software Migration Checklist

Series 1	Series 2	Notes
API for DCDC power configurations in em_emu.c.	No API for DCDC power configurations for EFR32xG21.	—
API for HFXO startup in em_cmu.c (emlib). **emlib:** Common API for HFXO startup initialization.	API for HFXO startup in em_cmu.c (emlib). **emlib:** Common API for HFXO startup initialization.	Call below function to configure HFXO to ensure safe startup for the given crystal. `void CMU_HFXOInit(const CMU_HFXOInit_TypeDef *hfxoInit)`
API for LFXO startup in em_cmu.c (emlib). **emlib:** Common API for LFXO startup initialization.	API for LFXO startup in em_cmu.c (emlib). **emlib:** Common API for LFXO startup initialization.	Call below function to optimize startup time and power consumption for a given low frequency crystal. `void CMU_LFXOInit(const CMU_LFXOInit_TypeDef *lfxoInit)`
API for HFRCO band select in em_cmu.c (emlib). Call below function to change HFRCO frequency band in Series 1. **emlib:** `void CMU_HFRCOBandSet(CMU_HFRCOFreq_TypeDef setFreq)`	API for HFRCODPLL band select in em_cmu.c (emlib).Call below function to change HFRCODPLL frequency band in Series 2. **emlib:** `void CMU_HFRCODPLLBandSet(CMU_HFRCODPLLFreq_TypeDef freq)`	Use HFRCODPLL as HFRCO for series 2. Calibrate HFRCODPLL of Series 2 at startup to match withSeries 1 HFRCO frequency band. Otherwise manual adjustment for HFRCODPLL frequency dependent parameters are required.

Series 1	Series 2	Notes
ROUTEPEN/ROUTELOCn registers are used for pin enable/routing of digital peripherals on Series 1 . **emlib:** No emlib API for pin enable and routing. The ROUTEPEN and ROUTELOC definitions of digital peripherals can be found in corresponding emlib header files. For example, to enable pins at location 15 and route the SCL and SDA pins for I2C0: `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 for DBUS pin enable/routing of digitial peripherals on Series 2. **emlib:** No emlib API for pin enable and routing. 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 route the SCL and SDA pins for I2C0: Set `GPIO_I2C0_ROUTEEN`, `GPIO_I2C0_SCLROUTE`, and `GPIO_I2C0_SDAROUTE` registers.	Peripheral route enable, port select, and pin select registers can be found in the GPIO section of the reference manual.
APORT for pin enable and routing of analog peripherals on Series 1. **emlib:** No emlib API for pin routing. APORT definitions of analog peripherals can be found in corresponding emlib header files. For example, to use APORT BUS1X channel 6 for ADC input: `ADC_InitSingle_TypeDef singleInit = ADC_INITSINGLE_DEFAULT; singleInit.posSel = adcPosSelAPORT1XCH6;`	ABUS for pin enable and routing of analog peripherals on Series 2. **emlib:** No emlib API for pin routing. Peripheral allocation to a bus is determined by GPIO_ABUSxALLOC register, where x is the port. For example, to allocate ACMP0 to BBUS even/odd: Set the bits in `GPIO_BBUSALLOC` register accordingly.	There are three analog buses on the EFR32xG21: 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. Bus allocation registers can be found in the GPIO section of the reference manual.

API for GPIO is in em_gpio.c (emlib) and gpiointerrup.c (emdrv).
 	
emlib and emdrv:
Common API for GPIO configuration.
	API for GPIO is in em_gpio.c (emlib) and gpiointerrup.c (emdrv).
	
	emlib and emdrv:
	Common API for GPIO configuration.
		Call below functions to configure PC6 (interrupt source 6) and PF6 (interrupt source 4) as interrupt sources without conflict.
		
		`GPIO_PinModeSet(gpioPortC, 6, gpioModeInputPull, 1);`
		
		`GPIO_ExtIntConfig(gpioPortC, 6, 6, false, true, true);`
		
		`GPIO_PinModeSet(gpioPortF, 6, gpioModeInputPull, 1);`
		
		`GPIO_ExtIntConfig(gpioPortF, 6, 4, false, true, true);`

On Series 1 devices, Gecko bootloader contains two stages (first stage bootloader and main stage bootloader). To enable bootloader functionality, Gecko Bootloader must be configured and programmed into the dedicated bootloader area(if applicable). The main bootloader is upgradable through the first stage bootloader.	On EFR32xG21, Gecko Bootloader contains second stage (main stage bootloader). To enable bootloader functionality, the second stage of the bootloader must be configured and programmed into the first 16KB of flash. The main bootloader is upgradable through the hardware peripheral Secure Element.	See UG266 for more information.

Series 1	Series 2	Notes
API for DAC in em_dac.c (emlib). API for VDAC in em_vdac.c (emlib).	—	DAC and VDAC are not present on EFR32xG21.
API for VCMP in em_vcmp.c (emlib). API for VMON in em_emu.c (emlib).	—	VCMP and VMON are not present on EFR32xG21 On EFR32xG21, ACMP has a low-power voltage scaler which allows it to function like VCMP. On EFR32xG21, It is recommended to use ACMP with voltage scalar or IADC for early warning voltage monitoring .
AES and cryptographic operations in Series 1 are handled by mbedTLS library. **emlib:** Common API for AES and CRYPTO.	AES and cryptographic operations in Series 2 are handled by mbedTLS library. **emlib:** Common API for AES and CRYPTO.	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`
API for RTCC in em_rtcc.c (emlib) and rtcdrv (emdrv). **emlib and emdrv:** Common API for RTCC.	API for RTCC in em_rtcc.c (emlib) and rtcdrc (emdrv) **emlib and emdrv:** Common API for RTCC.	The em_rtcc.c and rtcdrv provides common RTCC APIs for Series 1 and Series 2.
API for LDMA in em_ldma.c (emlib) and dmadrv (emdrv). **emlib and emdrv:** Common API for LDMA.	API for LDMA in em_ldma.c (emlib) and dmadrv (emdrv). **emlib and emdrv:** Common API for LDMA.	The em_ldma.c and dmadrv provides common LDMA APIs for Series 1 and Series 2.

6. Peripherals Common Between Series 1 and Series 2

6.1 Core, Memory, and Bus System

The major changes are the switch to a Cortex-M33 core and the addition of secure element

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 provieds 128 bytes of low power RAM that is retained in EM4.
—	Flash Lock	Series 2 devices doesn't have Lock Bits Page, unlike Series 0 and Series 1. Series 2 devices can disable the ability to program or erase the corresponding pages by writing to bits in MSC_PAGELOCKn register which can be set by CPU and cleared only with the device reset. The user data page may be locked by setting UDLOCKBIT in MSC_MISCLOCKWORD which can be cleared only by CPU.
—	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 on Secure Debug Unlock for more infomration.
Enhancements		
Cortex-M3 or Cortex-M4	Cortex-M33	Cortex-M33 includes TrustZone, implemented in the SMU. For more information on Cortex-M33, see Arm's documentation.

6.2 Clock Management Unit (CMU)

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

Table 6.2. CMU

Series 1	Series 2	Notes
New Features		
—	FSRCO (20 MHz)	The Fast Startup RC Oscillator is a fixed frequency, low energy oscillator with a short start up time. This oscillator is used after reset
—	PLL for HFRCO enables arbitrary clock frequency generation	See Reference Manual, for more information on 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	—
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
DBGCLK	TRACECLK	—
Three clocks (HFPERCLK, HFPERBCLK, and HFPERCCLK) drive all high-frequency peripherals	Peripherals are asynchronous to the main system clock (HCLK), peripheral clocks requested by peripheral EN bit	—
HFPERCLK	PCLK, LSPCLK, EM01GRPA	EM01GRPA supports synchronous operation of timer and IADC. LSPCLK saves power for peripherals that do not need the highest frequency clock available
LFA, LFE	EM23GRPA, EM4GRPA	Seperate clock selection for EM23 and EM4 peripherals. So EM23 peripherals can use 32kHz clock, while EM4 peripherals use 1kHz clock

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 splits in EM4H (Hibernate) and EM4S (Shutoff) energy modes	Single EM4 Mode	EM4 mode with clock running is much lower power than EM4 hibernate mode and EM4 mode with clocks shutdown is about the same as EM4 shutdown mode
Limitations		
—	No DCDC converter	—

6.3.2 Reset Management Unit (RMU)

The major change is configurable reset levels for Watchdog, Lockup, Pin, and System reset requests.

Table 6.4. RMU

Series 1	Series 2	Notes
New Features		
BOD reset cannot be masked	All BODs can be individually enabled and masked	—
Enhancement		
Brown-out Detection (BOD) on Analog Unregulated Power Domain (AVDD), Digital Unregulated Power Domain (DVDD) and Regulated Digital Domain (DECOUPLE)	Brown-out Detection (BOD) on AVDD, IOVDD0, IOVDD1, 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 for the I2C module. Added revision register and enable register. There are some changes to register access modes from Series 1 to Series 2.

Table 6.5. I2C

Series 1	Series 2	Notes
Enhancements		
—	I2C0 available in EM0/EM1 and EM2/3, I2C1 available in EM0/EM1	—

6.4.2 Universal Synchronous Asynchronous Receiver/Transmitter (USART)

There are no major changes for the USART module.

6.5 I/O Ports

6.5.1 General Purpose Input/Output (GPIO)

The major change is the addition of the DBUS, and the lack of 5 V tolerant pins.

Table 6.6. GPIO

Series 1	Series 2	Notes
New Features		
—	DBUS	DBUS is an any-to-any switch matrix that routes digital peripherals to pins See reference manual for more information
Limitations		
—	Port C and Port D hold state in EM2/EM3	—
5 V tolerant pins.	No 5 V tolerant pins. All IO pins have a max voltage of IOVDD + 0.3 V	—

6.6 Timers and Triggers

6.6.1 Timer/Counter (TIMER)

The major change is TIMER0 supports 32-bit operation.

Table 6.7. TIMER

Series 1	Series 2	Notes
Enhancements		
Some timers are 16 bit and some (WTIMER) are 32-bit	TIMER0 support 32-bit operation and TIMER 2/3/4 support 16-bit operation	—
Limitation		
Timers are synchronous	Timers use GRPA clock and can be synchronized	

6.6.2 Real Time Clock with Capture (RTCC)

The Real Time Counter and Calendar (RTCC) in Series 1 has been replaced by the Real Time Clock with Capture (RTCC) in Series 2.

Table 6.8. 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	BURAM provides 128 bytes of low power RAM that is retained in EM4
Operates down to EM4H	RTCC is only available in EM0 to EM3	In Series 2, RTCC is for EM23 and BURTC is for EM4 as BURTC has reduced EM4 current consumption

6.6.3 Low Energy Timer (LETIMER)

There are no major changes to LETIMER.

6.6.4 Peripheral Reflex System (PRS)

The major change is the number of PRS channels, and dedicated synchronous and asynchronous signals.

Table 6.9. PRS

Series 1	Series 2	Notes
New Features		
12 Sync/Async signals	12 Async and 4 sync channels	Synchronous channel support synchronous operation of IADC with timer trigger in Series 2

6.6.5 Watchdog Timer (WDOG)

A few minor changes for the WDOG module include the addition of a revision register and an enable register, and some changes in the register access modes from Series 1 to Series 2.

6.7 Analog Interfaces

6.7.1 Analog Comparator (ACMP)

A few minor changes for the ACMP module include the addition of a revision register and an enable register, and some changes in the register access modes from Series 1 to Series 2.

6.7.2 Incremental Analog to Digital Converter (IADC)

The ADC peripheral has been replaced by the IADC peripheral, which has significant differences. See AN1189 and the reference manual for more information.

Table 6.10. IADC

Series 1	Series 2	Notes
New Features		
—	Multiple (2) configurations can be specified and selected during channel scans	Changing the configuration requires warming up the ADC again, and requires a 5 us delay
Enhancements		
Traditional SAR ADC	Incremental ADC is combination of SAR and Delta-Sigma	EFR32MG21 supports 12 bit resolution at 1 Msps
Limitations		
ADC_CLK from 32 kHz to 16 MHz	ADC_CLK from 32 kHz to 10 MHz	For analog gain of 2x, 3x and 4x, the max ADC_CLK is 5 MHz, 3.3 MHz and 2.5 MHz respectively

7. Revision History

Revision 0.01

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 logo]

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

http://www.silabs.com