AN1379: EFR32xG22 to EFR32xG27 Compatibility and Migration Guide

This porting guide is for users migrating an existing design based on EFR32BG22, EFR32FG22, or EFR32MG22 to EFR32BG27 or EFR32MG27. Migration entails both hardware and software considerations.

This application note covers the MCU compatibility and migration between EFR32xG22 and EFR32xG27. Radio and security considerations are not covered in this application note.

KEY POINTS

- EFR32xG27 maintains a high degree of pin and feature compatibility with EFR32xG22 and provides additional or enhanced features.
- Migrating designs from EFR32xG22 to EFR32xG27 may require hardware and software changes depending on the feature set and package type.
- Datasheets for each device detail specific performance metrics that may differ between EFR32xG22 and EFR32xG27.

1. Introduction

This application note provides guidance on migrating from the EFR32xG22 family, which consists of the following:

- EFR32BG22
- EFR32MG22
- EFR32FG22

...to the EFR32xG27 family, which consists of the following:

- EFR32BG27
- EFR32MG27

This application note highlights the microcontroller-level differences between the EFR32xG22 and the EFR32xG27. Certain orderable part numbers (OPNs) of EFR32xG27 are designed to be both pin- and hardware-compatible, as well as largely software-compatible, with EFR32xG22 devices, thus requiring only minor changes when porting designs. Other OPNs of EFR32xG27 require a specific hardware configuration that differs from EFR32xG22.

Four factors must be considered when porting a design from EFR32xG22 to EFR32xG27: pin-compatibility, external hardware compatibility, peripheral compatibility, and software compatibility.

1.1 Pins and External Hardware

Certain EFR32xG27 devices are pin-compatiable with EFR32xG22 devices in the same package. More information on the footprints, packages, and external hardware compatibility between EFR32xG22 devices and EFR32xG27 devices can be found in 2. Pin Compatibility and 3. External Hardware Migration.

1.2 Peripherals and Software

The EFR32xG27 peripherals are a superset of the EFR32xG22, except with the replacement of the Enhanced Universal Asynchronous Receiver Transmitter (EUART) with the Enhanced Universal Synchronous/Asynchronous Receiver Transmitter (EUSART). In addition to its new and enhanced features, EFR32xG27 includes all of the hardware resources available on EFR32xG22.

Similarly, all of the software features in the SDKs that support EFR32xG22 are available on EFR32xG27; however, existing binaries built for the EFR32xG22 cannot be directly flashed onto the EFR32xG27 or vice versa. 4. Peripheral Compatibility and 5. Software Compatibility covers these areas in greater detail.

2. Pin Compatibility

Pin compatibility between EFR32xG22 and EFR32xG27 depends on the specific package and the device feature set. Table 2.1 Pin Compatibility Overview on page 3 provides an overview of the pin compatibility between EFR32xG22 and EFR32xG27. Two devices are pin-compatible if both the pin definitions and the package footprints between the two devices are the same. For information on the OPNs in a specific package and their feature sets, refer to the device-specific data sheet.

Table 2.1. Pin Compatibility Overview

Package	DC-DC Configuration	EFR32xG22 OPNs	EFR32xG27 OPNs	Pin-Compatible
QFN32	Buck	EFR32BG22C112F352GM32 EFR32BG22C222F352GM32 EFR32BG22C224F512GM32 EFR32BG22C224F512IM32 EFR32FG22C121F512GM32 EFR32MG22C224F512IM32	EFR32BG27C140F768IM32 EFR32MG27C140F768IM32	Yes
TQFN32	Buck	EFR32BG22C222F352GN32 EFR32BG22C224F512GN32 EFR32MG22C224F512GN32	N/A	No[1]
QFN32	Boost	N/A	EFR32BG27C230F768IM32	No
QFN40	Buck	EFR32BG22C222F352GM40 EFR32BG22C224F512GM40 EFR32BG22C224F512IM40 EFR32FG22C121F512GM40 EFR32MG22C224F512IM40	EFR32BG27C140F768IM40 EFR32MG27C140F768IM40	Yes
QFN40	Boost	N/A	EFR32BG27C230F768IM40	No
WLCSP39	Buck/Boost	N/A	EFR32BG27C320F768GJ39	No

Note:

1. While the EFR32xG27 buck DC-DC QFN32 is footprint compatible with the EFR32xG22 buck DC-DC TQFN32, its Z-height (package thickness) is greater.

3. External Hardware Migration

The external hardware components — including crystals, debugger, and power supply components — for EFR32xG27 devices and EFR32xG22 devices are largely compatible, with minor configuration changes needed for certain feature sets on EFR32xG27 devices.

The EFR32xG27 has the same crystal oscillators as the EFR32xG22, and also share the same specifications for these oscillators. Furthermore, for EFR32xG27 devices that are pin compatiable with the relevant EFR32xG22 devices, the debug capabilities and the debug connection layout are the same between them. The electrical specifications of the debug interface are also identical, and the same debug signals can be connected to a debug connector using the same footprint as in an EFR32xG22 design. Therefore, any external hardware component differences between EFR32xG22 and EFR32xG27 are going to concern the device's power supply. 3.1 Power Supply Compatibility provides an overview of certain power supply configuration changes that may need to be considered when porting a board design from EFR32xG22 to EFR32xG27.

3.1 Power Supply Compatibility

The power supply compatibility between EFR32xG22 devices and EFR32xG27 devices is dependent on the feature set of the device specifically, with particular regard to the capabilities of the DC-DC converter present on the device.

DC-DC Compatibility

EFR32xG22 devices feature a buck DC-DC converter that supports an input voltage range from 1.8 V to 3.8 V and a nominal output voltage of 1.8 V.

The DC-DC converter on EFR32xG27 devices supports either buck or boost operation depending on the device's feature set and hardware configuration. The buck converter on EFR32xG27 has the same specifications as the buck converter on EFR32xG22. The boost converter, which is only found on specific OPNs of the EFR32xG27, supports an input voltage ranging from 0.8 V to 1.7 V (TBD) and has a nominal output voltage of 1.8 V.

OPNs supporting boost converter are listed below:

Table 3.1. Boost DC-DC Devices

Orderable Part Number (OPN)	Package
EFR32BG27C320F768GJ39	WLCSP39
EFR32BG27C230F768IM40	QFN32 Boost DC-DC
EFR32BG27C230F768IM32	QFN40 Boost DC-DC

DC-DC Configuration

On EFR32xG27 devices where the DC-DC converter operates in buck mode or is not used at all, the following power supply pin dependencies must be observed:

- VREGVDD & DVDD
  - In systems using the DC-DC converter, DVDD (the buck converter output) should be connected with VREGSW via the recommended L_{DCDC} and C_{DCDC} configuration, and should not be driven by an off-chip regulator
  - In systems not using the DC-DC converter, DVDD must be shorted to VREGVDD on the PCB (VREGVDD = DVDD)
- DVDD ≥ DECOUPLE
- PAVDD ≥ RFVDD
- AVDD, IOVDD: No dependency with each other or any other supply pin
- VBAT, BOOST_EN: Tie to VSS (WLCSP package only)

On EFR32xG27 devices where the DC-DC converter operates in boost mode, the following power supply pin dependencies must be observed:

- VBAT: DCDC converter input. Connect to recommended supply via L_{DCDC}
- DVDD: DVDD is the boost converter output and should be bypassed to ground with the recommended C_{DCDC}. DVDD should not be driven by an off-chip regulator
- VREGVDD: Tie directly to DVDD (WLCSP package only)
- DVDD ≥ DECOUPLE
- PAVDD ≥ RFVDD
- AVDD, IOVDD: No dependency with each other or any other supply pin

4. Peripheral Compatibility

EFR32xG27 is a superset of the EFR32xG22. All peripherals present on the EFR32xG22 are present on the EFR32xG27. Some peripherals have small enhancements, and there are new peripherals that are not present on the EFR32xG22.

4.1 Analog Peripherals

The EFR32xG27 includes one analog comparator (ACMP). There is no ACMP on the EFR32xG22. On the incremental analog to digital converter (IADC) on EFR32xG27, the VBAT supply voltage can be selected as an input to measure the battery supply in boost mode. No other differences exist between the EFR32xG22 and EFR32xG27 with regards to analog peripherals.

4.2 CMU

Functionally, the CMU is mostly unchanged between EFR32xG22 and EFR32xG27. In particular, the enabling and selection of clocks are performed in the same way on both devices. However, new clock branches have been added to accommodate the new peripherals on EFR32xG27. Additionally, there are differences in the allowable maximum frequencies for certain clock branches. These changes and differences are discussed in the following subsections.

4.2.1 Peripheral Clocks

The new peripherals on EFR32xG27 reside on specific clock tree branches and have their own bus clock enable bits described in Table 4.1 Peripheral Clock Additions on page 6.

Table 4.1. Peripheral Clock Additions

Peripheral	Clock Branch	Register Enable Bit
ACMP	LSPCLK	CMU_CLKEN1_ACMP0
ETAMPDET	PCLK	CMU_CLKEN1_ETAMPDET
EUSART0	LSPCLK	CMU_CLKEN1_EUSART0
	EUSART0CLK	CMU_EUSART0CLKCTRL_CLKSEL

The EUART on EFR32xG22 has been replaced with the EUSART on EFR32xG27, resulting in the CMU changes listed in Table 4.2 Peripheral Clock Deletions on page 6.

Table 4.2. Peripheral Clock Deletions

Peripheral	Clock Branch	Register Enable Bit
EUART	LSPCLK	CMU_CLKEN0_EUART0
	EUARTCLK	CMU_EUART0CLKCTRL_CLKSEL

Note: Although EUSART0 on EFR32xG27 effectively replaces the functionality provided by EUART0 on EFR32xG22, it does not use the same module clock enable bit nor does its clock control register reside at the same offset in the CMU register block. Be sure to refer to the reference manuals for both devices to understand the specific differences.

4.2.2 Oscillators

The EFR32xG27 and EFR32xG22 have the same high-frequency crystal oscillator (HFXO). The HFXO specifications are the same between the two devices. The same applies to the high-frequency RC oscillator (HFRCO), low-frequency crystal oscillator (LFXO), precision low-frequency RC oscillator (LFRCO), fast start-up RC oscillator (FSRCO), and ultra-low frequency RC oscillator (ULFRCO).

4.2.3 Operating Frequency

The maximum operating frequency of certain clocks differs between EFR32xG22 and EFR32xG27 as shown in Table 4.3 Clock Branch Frequency Differences on page 7.

Table 4.3. Clock Branch Frequency Differences

Parameter	Symbol	Test Condition	EFR32xG22 Max	EFR32xG27 Max
HCLK and Core frequency	f_{HCLK}	VSCALE2 MODE = WS1	76.8 MHz	80 MHz
EM01 Group A clock frequency	$f_{EM01GRPACLK}$	VSCALE2	76.8 MHz	80 MHz
EM01 Group B clock frequency	$f_{EM01GRPBCLK}$	VSCALE2	76.8 MHz	80 MHz

4.3 Communication Peripherals

The EUART on EFR32xG22 is replaced by the EUSART on EFR32xG27. It retains the same low-frequency asynchronous as the EUART and adds synchronous mode (SPI) functionality comparable to that provided by the USART. The transmit and receive FIFOs have been increased from 4 to 16 entries and are available in both modes .

4.4 EMU

While core functionality is unchanged, the EMU on EFR32xG27 is enhanced with the addition of a coulomb counter and DC-to-DC converter boost mode operation.

4.4.1 Coulomb Counter

The coulomb counter measures the charge delivered by the DC-DC converter and can be used to provide an accurate estimate of the remaining device run-time based on the current battery capacity. Although it is disabled after power-on reset, the coulomb counter remains enabled through all other resets, which can be helpful in systems that make use of EM4 or that have a rechargeable battery. Calibration is required to determine the charge per pulse and is performed by software. The charge per pulse is measured using known on-chip calibration loads, a PRS channel, and the CMU RC oscillator calibration circuitry.

4.4.2 DC-DC Boost Mode

The DC-DC converter on EFR32xG27 supports boost mode operation on some select OPNs. This is an especially valuable feature as it integrates the ability to power a system at 1.8 V from the nominal 1.5 V output such as a single alkaline battery. Through the use of its dedicated BOOST_EN input, the system can enter and exit the boost DC-DC shutdown mode, which is comparable to EM4 under the control of external logic. Refer to the EFR32xG27 Reference Manual for operational details of DC-DC boost mode and to the relevant device datasheet for boost mode electrical specifications.

4.5 Memory

The size of the flash on EFR32xG27 has been increased from 512 kB to 768 kB. The erase page size remains the same between EFR32xG22 and EFR32xG27. The erase cycle endurance and data retention specifications remain unchanged. Program and erase times (including mass erase) may vary slightly between the two devices.

The size of the RAM on EFR32xG27 has been increased from 32 kB to 64 kB. The number of RAM blocks is changed from 2 to 3. The RAM blocks on EFR32xG27 have sizes of 8 kB, 24 kB, and 32 kB. The RAM retention can be controlled by configuring the RAMRETNCTRL field in the SYSCFG_DMEM0RETNCTRL register.

The starting address of flash on EFR32xG27 is changed from 0x0 to 0x08000000. The bootloader starting address is also changed from 0x0 to 0x08000000. The starting address of RAM is the same between the two devices. The User Data and Device Information block reside in the same flash region between two devices.

There are no compatibility issues in memory mapping between EFR32xG22 and EFR32xG27. Changes in the linker file are required to adjust the starting address of flash and also the size of flash and RAM.

4.6 PRS

The PRS peripheral between EFR32xG22 and EFR32xG27 is mostly identical, with the EFR32xG27 PRS adding a few extra signal producers which are related to new or updated features.

Table 4.4. New PRS Producers

Peripheral	SOURCESEL	Signal	SIGSEL
ACMP0	ACMP0 (0x0D)	OUT	0x0
DCDC	DCDC (0x10)	MONO70NSANA	0x0
ETAMPDET	ETAMPDET (0x0C)	TAMPERSRCETAMPDET	0x0
EUSART0	EUSART0L (0x0E)	CS	0x0
		IRDATX	0x1
		RTS	0x2
		RXDATAV	0x3
		TX	0x4
		TXC	0x5
		RXFL	0x6
		TXFL	0x7

Note: Producer EUART0 is replaced by Producer EUSART0.

4.7 Security

EFR32xG27 added peripheral External Tamper Detect (ETAMPDET) that was not present on EFR32xG22. Please refer to the reference manual for a more detailed description of this peripheral.

4.8 Timers

Timer functionality is virtually unchanged when migrating from EFR32xG22 to EFR32xG27. No changes have been made to the features of the Low Energy Timer (LETIMER), Real Time Clock with Capture (RTCC), or Back-Up Real Time Counter (BURTC) nor have the number of instances of these modules (one of each) been changed. The sole change to the timer complement is the expansion of TIMER1 from 16 bits to 32 bits, which makes it functionally identical to TIMER0.

Although its counter, capture, and compare registers now implement all 32 bits, the TIMER_TOP register has a reset value of 0xFFFF. This allows it to retain backward compatibility with the 16-bit implementation of TIMER1 on EFR32xG22, as well as with the existing TIMER[4:2] on both devices.

No changes are required to any software that uses emlib or emdrv to control TIMER1. Code that directly reads 16-bit data from and writes 16-bit data to TIMER1 registers can continue to do so. Note that peripheral register accesses, regardless of the implemented number of register bits, must always be 32-bit reads or writes to avoid undefined behavior.

5. Software Compatibility

A high degree of software compatibility is retained between EFR32xG22 and EFR32xG27. However, in all cases, firmware written originally for EFR32xG22 must be recompiled against the patch release of the relevant SDK that supports EFR32xG27. Even though the software compatibility is highly retained between the two device families, the recompiled firmware cannot be flashed interchangeably between EFR32xG27 and EFR32xG22.

5.1 EMLIB

Because emlib is a peripheral abstraction API, it handles the small differences between devices within the EFR32 famlies, including specific revisions. Certain emlib subcomponents are impacted by the differences between EFR32xG22 and EFR32xG27.

The following sections will provide a summary of certain software differences bewteen EFR32xG22 and EFR32xG27. Please note that this is not necessarily a complete list of the differences, and users are advised to consult to the Gecko SDK and its relevenant documentation for additional details.

5.1.1 `em_cmu.c` and `em_cmu.h`

The APIs in `em_cmu.c` do not have any major changes with regard to EFR32xG27. There are additional `#ifdef` statements to indicate if the API calls are made to serve the EFR32xG27 family. This is done by checking whether the definition `_SILICON_LABS_32B_SERIES_2_CONFIG_7` is defined in the project.

The header file `em_cmu.h` does not have updates that are specific for EFR32xG27.

5.1.2 em_emu.c and em_emu.h

em_emu.c and em_emu.h has several updates to accommodate the newly added DC-DC boost mode in EFR32xG27.

em_emu.h changes

- Added new definition EMU_SERIES2_DCDC_BOOST_PRESENT. This definition is used to distinguish code contents that are used for boost DC-DC mode.
- Added new EMU_DCDCBoostInit_TypeDef which is the initialization structure for boost Mode DC-DC that is used by em_emu.c APIs.

```
/** DCDC Boost regulator initialization structure. */
typedef struct {
  EMU_DcdcBoostTonMaxTimeout_TypeDef    tonMax;              // Ton max timeout control
  bool                                  externalShutdownEn;  // true = disable DCDC boost mode with
                                                             // BOOST_EN=0
  EMU_DcdcBoostDriveSpeed_TypeDef       driveSpeedEM01;      // DCDC drive speed in EM0/1
  EMU_DcdcBoostDriveSpeed_TypeDef       driveSpeedEM23;      // DCDC drive speed in EM2/3
  EMU_DcdcBoostEM01PeakCurrent_TypeDef  peakCurrentEM01;     // EM0/1 peak current setting
  EMU_DcdcBoostEM23PeakCurrent_TypeDef  peakCurrentEM23;     // EM2/3 peak current setting
} EMU_DCDCBoostInit_TypeDef;
```

- Added new EMU_DCDCBOOSTINIT_DEFAULT. This is the default value for the EMU_DCDCBoostInit_TypeDef structure.

```
/** Default DCDC Boost initialization. */
#define EMU_DCDCBOOSTINIT_DEFAULT                                                         \
  {                                                                                       \
    emuDcdcBoostTonMaxTimeout_1P19us,     /**< Ton max is 1.19us. */                      \
    true,                                 /**< disable DCDC boost mode with BOOST_EN=0 */ \
    emuDcdcBoostDriveSpeed_Default,       /**< Default efficiency in EM0/1. */            \
    emuDcdcBoostDriveSpeed_Default,       /**< Default efficiency in EM2/3. */            \
    emuDcdcBoostEM01PeakCurrent_Load72mA, /**< Default peak current in EM0/1. */          \
    emuDcdcBoostEM23PeakCurrent_Load10mA  /**< Default peak current in EM2/3. */          \
  }
```

- Added new enumerations for boost mode specific type defines. Including:
  - EMU_DcdcBoostDriveSpeed_TypeDef
  - EMU_DcdcBoostEM01PeakCurrent_TypeDef
  - EMU_DcdcBoostToffMaxTimeout_TypeDef
  - EMU_DcdcBoostTonMaxTimeout_TypeDef
  - EMU_DcdcBoostEM23PeakCurrent_TypeDef

  Please refer to the source file em_emu.h for detailed description of each enumeration.

em_emu.c changes

- The void EMU_RamPowerDown(uint32_T start, uint32_t end) API has a new RAM block power down mask for EFR32xG27.

```
#elif defined(_SILICON_LABS_32B_SERIES_2_CONFIG_7)
    mask |= ADDRESS_NOT_IN_BLOCK(start, 0x20006000UL) << 0; // Block 0, 24 kB
    mask |= ADDRESS_NOT_IN_BLOCK(start, 0x20008000UL) << 1; // Block 1, 8 kB
    mask |= ADDRESS_NOT_IN_BLOCK(start, 0x20010000UL) << 2; // Block 2, 32 kB
```

  Consult the source file em_emu.c for complete source code of the EMU_RamPowerDown() API.
- Added new API void EMU_DCDCBoostInit(const EMU_DCDCBoostInit_TypeDef *dcdcBoostInit) for boost mode DC-DC initialization.
- Added new API void EMU_BoostExternalShutdownEnable(bool enable) to enable/disable the boost DC-DC shutdown mode.
- Added new API SL_WEAK void EMU_DCDCUpdatedHook(void). This API is a weakly defined, empty function that would be overwritten by a strong function in RAIL.

5.1.3 em_prs.c and em_prs.h

The emlib library for the PRS peripheral are very similar between EFR32xG22 and EFR32xG27. The em_prs.c has no updates to accommodate changes on the EFR32xG27 PRS peripheral.

em_prs.h added a new PRS signal PRS_DCDC_MONO70NSANA.

```
#if defined(PRS_DCDC_MONO70NSANA)
  prsSignalDCDC_MONO70NSANA = PRS_DCDC_MONO70NSANA, /** DCDC Pulses for Coulomb Counter Calibration Signal. */
```

5.1.4 em_system.c and em_system.h

New device family is added in em_system.h with regard to EFR32xG27.

```
#if defined(_SILICON_LABS_32B_SERIES_2_CONFIG_7)
  /**< EFR32 Mighty Gecko Series 2 Config 7 Value Device Family */
  systemPartFamilyMighty27 = DEVINFO_PART_FAMILY_MG | (27 << _DEVINFO_PART_FAMILYNUM_SHIFT),
  /**< EFR32 Blue Gecko Series 2 Config 7 Value Device Family */
  systemPartFamilyBlue27 = DEVINFO_PART_FAMILY_BG | (27 << _DEVINFO_PART_FAMILYNUM_SHIFT),
#endif
```

em_system.c does not have code updates that are specific for EFR32xG27. However, note that the return value for system-specific APIs such as void SYSTEM_ChipRevisionGet(SYSTEM_ChipRevision_TypeDef *rev) will return a value that is with specific regard to EFR32xG27.

5.1.5 em_timer.c and em_timer.h

Because TIMER0 is already 32 bits wide on all Series 2 EFM32/EFR32 devices, no changes are necessary to the APIs in em_timer.c to handle the 32-bit TIMER1 implementation on EFR32xG27.

5.2 NVM3

The functionality of NVM3 is unchanged on EFR32xG27. Consequently, there is no need to change the number of flash pages allocated to NVM3 in any given application. Due to the flash size increase on EFR32xG27, additional flash pages may be available to be configured for NVM3 usage that was not available on EFR32xG22. Such configuration changes can be done in the linker file.

6. Revision History

Revision 0.2

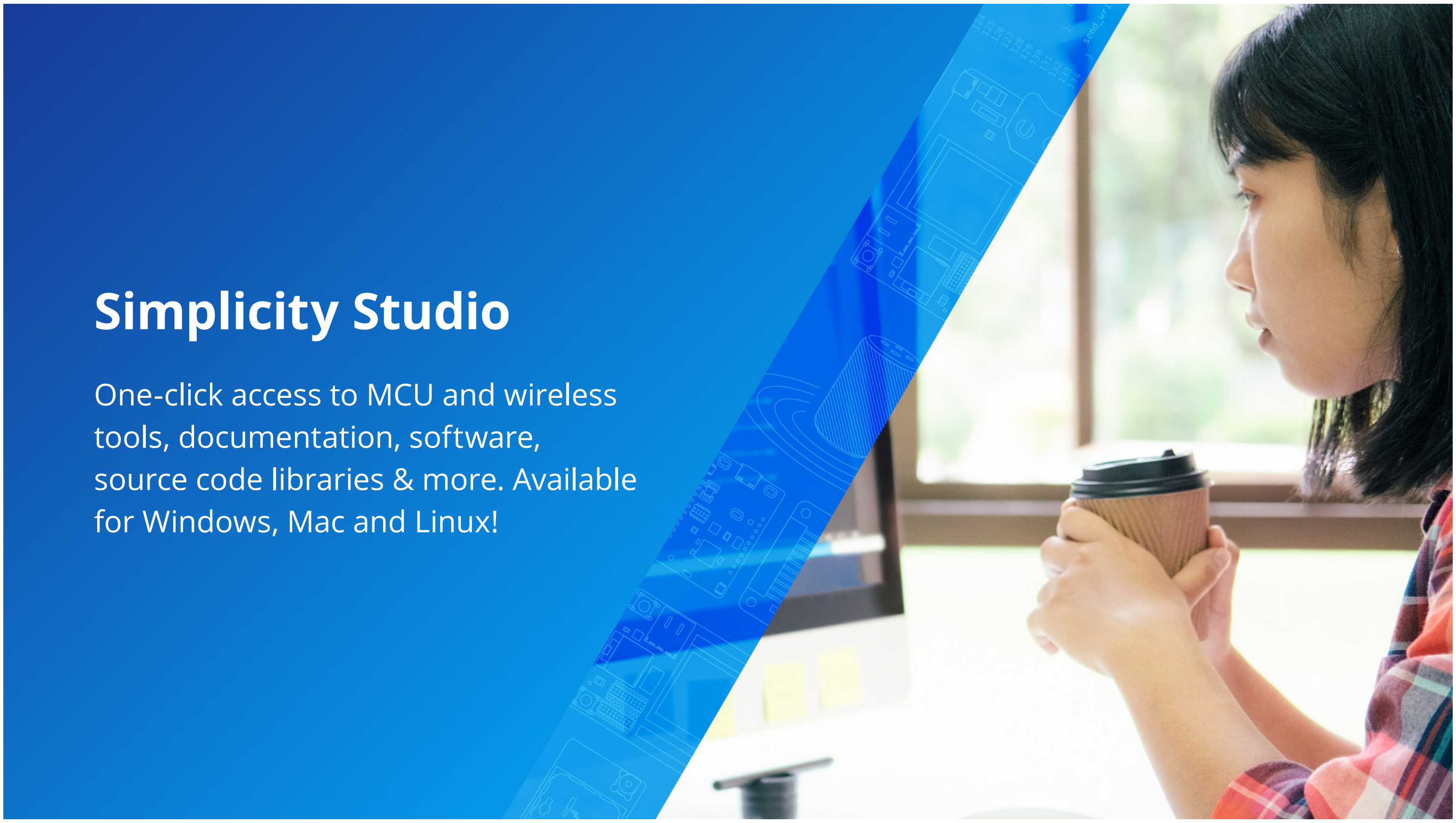
April, 2023

- Table 2.1 Pin Compatibility Overview on page 3 organization improved and updated with a missing OPN.
- Added 4.8 Timers and 5.1.5 `em_timer.c` and `em_timer.h` to cover the upgrade of TIMER1 from 16 bits on EFR32xG22 to 32 bits on EFR32xG27.

Revision 0.1

April, 2022

- 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 & 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 LABS

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

www.silabs.com