AN1380: EFR32xG24 to EFR32xG27 Compatibility and Migration Guide

This porting guide is for users migrating an existing EFR32BG24 or EFR32MG24 design to an EFR32BG27 or EFR32MG27 design. Migration entails both hardware and software considerations.

This application note covers the MCU compatibility and migration between EFR32xG24 and EFR32xG27. Radio and security considerations are not covered in this application note.

KEY POINTS

- EFR32xG24 and EFR32xG27 maintain a high degree of peripheral and software compatibility.
- Migrating designs from EFR32xG24 to EFR32xG27 will require hardware and software changes depending on the device's feature set.
- Datasheet for each device detail device-specific performance metrics, which may differ between EFR32xG24 and EFR32xG27.

1. Introduction

This application note applies to the following devices:

EFR32xG24 consists of:

- EFR32BG24
- EFR32MG24

EFR32xG27 consists of:

- EFR32BG27
- EFR32MG27

This application note highlights the microcontroller-level differences between EFR32xG24 and EFR32xG27. Certain orderable part numbers (OPNs) of EFR32xG27 are designed to be both pin- and hardware-compatible, as well as largely software-compatible, with EFR32xG24 , thus requiring only minor changes when porting designs. Other OPNs of the EFR32xG27 will require a specific hardware configuration that differs from EFR32xG24.

Four factors must be considered when porting a design from EFR32xG24 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 EFR32xG24 devices in the same package. More information on the footprints, packages, and external hardware compatibility between EFR32xG24 devices and EFR32xG27 devices can be found in 2. Pin Compatibility and 3. External Hardware Migration.

1.2 Peripherals and Software

EFR32xG27 has a different peripheral set compared with EFR32xG24. 4. Peripheral Compatibility covers the peripheral differences between the two devices.

The software features in the SDKs that support EFR32xG24 are available on EFR32xG27; however, existing binaries built for the EFR32xG24 cannot be directly flashed onto the EFR32xG27 or vice versa. 5. Software Compatibility covers these areas in greater detail.

2. Pin Compatibility

Pin compatibility between EFR32xG24 and EFR32xG27 depends on the packages and the device feature set. Table 2.1 Pin Compatibility Overview on page 3 provides an overview of the pin compatibility between EFR32xG24 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 datasheet.

Table 2.1. Pin Compatibility Overview

Package	EFR32xG24OPNs	EFR32xG27 OPNs	Pin Compatible
QFN32 Buck DC-DC	N/A	EFR32BG27C140F768IM32 EFR32MG27C140F768IM32	N/A
QFN32 Boost DC-DC	N/A	EFR32BG27C230F768IM32	N/A
QFN40 Standard/Buck DC-DC	EFR32BG24A020F1024IM40 EFR32BG24A010F1024IM40 EFR32MG24B020F1536IM40 EFR32MG24B010F1536IM40 EFR32MG24A420F1536IM40 EFR32MG24A410F1536IM40 EFR32MG24A020F1536IM40 EFR32MG24A020F1024IM40 EFR32MG24A010F1536IM40 EFR32MG24A010F1024IM40	EFR32BG27C140F768IM40 EFR32MG27C140F768IM40	Yes
QFN40 HFCLKOUT	EFR32MG24A021F1024IM40	N/A	N/A
QFN40 Boost DC-DC	N/A	EFR32BG27C230F768IM40	N/A
QFN48 Standard	EFR32BG24B220F1024IM48 EFR32BG24B210F1024IM48 EFR32BG24A020F1024IM48 EFR32BG24A010F1024IM48 EFR32MG24B220F1536IM48 EFR32MG24B210F1536IM48 EFR32MG24B020F1536IM48 EFR32MG24B020F1024IM48 EFR32MG24B010F1536IM48 EFR32MG24B010F1024IM48 EFR32MG24A420F1536IM48 EFR32MG24A410F1536IM48 EFR32MG24A020F1536IM48 EFR32MG24A020F1024IM48 EFR32MG24A010F1536IM48 EFR32MG24A010F1024IM48	N/A	N/A

Package	EFR32xG24OPNs	EFR32xG27 OPNs	Pin Compatible
QFN48 ADC	EFR32BG24B110F1536IM48 EFR32MG24B310F1536IM48 EFR32MG24B120F1536IM48 EFR32MG24B110F1536IM48 EFR32MG24A110F1024IM48	N/A	N/A
WLCSP39	N/A	EFR32BG27C320F768GJ39	N/A

3. External Hardware Migration

The external hardware components—including crystals, debugger, and power supply components—for EFR32xG27 devices and EFR32xG24 devices are largely compatible, with minor configuration changes needed for certain feature sets on EFR32xG27 devices.

The EFR32xG27 high frequency crystal oscillator (HFXO) has a typical frequency of 38.4 MHz whereas EFR32xG24 has a frequency of 39 MHz. Other specifications regarding the HFXO and other crystals are the same. Furthermore, the debug capabilities and debug connection layout are the same between pin-compatible EFR32xG27 and EFR32xG24 devices. 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 EFR32xG24 design. Therefore, the primary difference in the external hardware component between EFR32xG24 and EFR32xG27 is 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 EFR32xG24 to EFR32xG27.

3.1 Power Supply Compatibility

The power supply compatibility between EFR32xG24 devices and EFR32xG27 devices is dependent on the feature set of the device, with particular regard to the capabilities of the DC-DC converter present on the device.

DC-DC Compatibility

EFR32xG24 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 EFR32xG24. The boost converter, which is only found on specific OPNs of 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 has a different peripheral set than EFR32xG24. Table 4.1 Peripheral Changes on EFR32xG27 on page 7 provides an overview of the peripheral changes on EFR32xG27

Table 4.1. Peripheral Changes on EFR32xG27

Added Peripherals	Removed Peripherals
• Added 1 x Universal Synchronous/Asynchronous Receiver Transmitter (USART1) • Added Pulse-density Modulation (PDM) • Added External Tamper Detect (ETAMPDET) • Added Coulomb Counter • Added Real Time Clock with Capture (RTCC)	• Removed 1 x Analog Comparator (ACMP1) • Removed Voltage Digital to Analog Converter (VDAC) • Removed 1 x Watch Dog Timer (WDOG1) • Removed 1x Enhanced Universal Synchronous/Asynchronous Receiver Transmitter (EUSART1) • Removed Keyscan • Removed Matrix Vector Processor (MVP) • Removed Pulse Counter (PCNT) • Removed Secure Element • Removed System Real-Time Counter (SYSRTC)
Note: For peripherals that are either added or removed, their corresponding bus and peripheral clocks will be added or removed from the Clock Management Unit (CMU) peripheral accordingly. All PRS signals associated with these peripherals will be added or removed from the Peripheral Reflex Signal (PRS) peripheral accordingly as well. 5.1.1 `em_cmu.c` and `em_cmu.h` and 5.1.3 `em_prs.c` and `em_prs.h` will provide information and examples on software adjustment with regard to these changes.	

4.1 Analog Peripherals

There are three main differences in the analog peripherals between EFR32xG24 and EFR32xG27:

1. EFR32xG27 has only one analog comparator peripheral, whereas EFR32xG24 has two analog comparators.
2. The IADC on EFR32xG27 has no high accuracy/high speed mode, whereas EFR32xG24 supports high accuracy/high speed mode on selected OPNs. Table 4.2 IADC High Speed/High Accuracy OPN on page 7 provides a list of EFR32xG24 OPNs that support high speed/high accuracy mode.
3. EFR32xG27 has no Voltage DAC (VDAC), whereas EFR32xG24 has one VDAC.

Table 4.2. IADC High Speed/High Accuracy OPN

OPN	Package Type
EFR32BG24B110F1536IM48-B	QFN48 / ADC
EFR32MG24B310F1536IM48-B	QFN48 / ADC
EFR32MG24B120F1536IM48-B	QFN48 / ADC
EFR32MG24B110F1536IM48-B	QFN48 / ADC
EFR32MG24A110F1024IM48-B	QFN48 / ADC

4.2 Clock Management Unit

Functionally, the CMU is mostly unchanged between EFR32xG24 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 CMU changes regarding clock sources and branches on EFR32xG27 are dependent on the peripherals that are added or removed from EFR32xG24. Peripherals that are added to or removed from EFR32xG27 will have their bus and peripheral clocks added/removed accordingly. Please refer to Table 4.1 Peripheral Changes on EFR32xG27 on page 7 for a comprehensive list of added/removed peripherals between EFR32xG24 and EFR32xG27.

4.2.2 Oscillators

The typical frequency of the high frequency crystal oscillator (HFXO) on EFR32xG27 is 38.4 MHz compared to 39 MHz on EFR32xG24. Other specifications of the HFXO are the same between two devices.

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) have the same specifications between the two devices.

4.2.3 Operating Frequency

The maximum operating frequency of certain clocks differs between EFR32xG24 and EFR32xG27 as shown in Table 4.3 Clock Branch Frequency Differences on page 8.

Table 4.3. Clock Branch Frequency Differences

Parameter	Symbol	Test Condition	EFR32xG24 Max	EFR32xG27 max
HCLK and Core Frequency	f_{HCLK}	VSCALE2 MODE = WS1	78 MHz	80 MHz
EM01 Group A Clock Frequency	$f_{EM01GRPACLK}$	VSCALE2	78 MHz	80 MHz
EM01 Group B/C Clock Frequency[1]	$f_{EM01GRPCCLK}$	VSCALE2	78 MHz	80 MHz
Radio HCLK Frequency	f_{RHCLK}	VSCALE2 or VSCALE1	39 MHz	38.4 MHz

Note:

1. EM01GRPCCLK on EFR32xG24 is renamed to EM01GRPBCLK on EFR32xG27.

4.3 Communication Peripherals

EFR32xG24 has 1 x Universal Synchronous/Asynchronous Receiver/Transmitter peripheral (USART) and 2 x Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter peripheral (EUSART).

Compared with EFR32xG24, EFR32xG27 has 2 x Universal Synchronous/Asynchronous Receiver/Transmitter peripheral (USART) and 1 x Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter peripheral (EUSART). EFR32xG27 also features a digital microphone interface peripheral (PDM) that is not present on EFR32xG24.

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 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 decreased from 1536 kB to 768 kB. The erase page size remains the same between EFR32xG24 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 decreased from 256 kB to 64 kB. The number of RAM blocks is changed from 16 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 and RAM on EFR32xG27 is the same as EFR32xG24. The bootloader starting address is also the same. 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 EFR32xG24 and EFR32xG27. Changes in the linker file are required to adjust the size of flash and RAM.

4.6 Security

The Secure Element (SE) on EFR32xG24 is replaced with the Virtual Secure Element (VSE) on EFR32xG27. EFR32xG27 adds an external tamper detect (ETAMPDET). Refer to the reference manual for a detailed description of this peripheral.

4.7 Counters/Timers

EFR32xG27 has the following changes in the counters/timers peripheral compared with the EFR32xG24

- Replaced System Real Time Clock (SYSRTC) on EFR32xG24 with Real Time Clock Counter (RTCC) on EFR32xG27
- 1 x Watchdog Timer (WDOG) on EFR32xG27 vs 2 x WDOG on EFR32xG24

5. Software Compatibility

A high degree of software compatibility is retained between EFR32xG24 and EFR32xG27. However, in all cases, firmware written originally for EFR32xG24 must be recompiled against the patch release of the relevant Gecko 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 EFR32xG24.

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 EFR32xG24 and EFR32xG27.

The following sections will provide a summary of certain software differences between EFR32xG24 and EFR32xG27. Please note that this is not necessarily a complete list of the differences, and users are advised to consult 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 specific changes for EFR32xG27. There are new define checks added to indicate if the API calls are made to serve 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 specific updates for EFR32xG27. The clock sources of added peripherals will get reused on EFR32xG27. Table 5.1 Clock Sources on page 10 provides an overview of the new clock source enums that are available on EFR32xG27. Please refer to the `em_cmu.h` source file for in-depth information on these clock sources.

Table 5.1. Clock Sources

New Peripherals on EFR32xG27	Clock Source Enum in `em_cmu.h`
Pulse-density Modulation (PDM)	`cmuClock_PDM`
	`cmuClock_PDMREF`
Real Time Clock with Capture (RTCC)	`cmuClock_RTCC`
	`cmuClock_RTCCCLK`
Universal Synchronous/Asynchronous Receiver Transmitter (USART)	`cmuClock_USART1`

5.1.2 em_emu.c and em_emu.h

The em_emu.c and em_emu.h has a few 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 Boost External 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

em_prs.c has no updates to accommodate changes on EFR32xG27 PRS peripheral.

Like em_cmu.h, em_prs.h has definitions for new PRS producer signals that are available in EFR32xG27. Table 5.2 New PRS Producer Signals on page 12 provides a list of the definitions used in em_prs.h for these new signals

Table 5.2. New PRS Producer Signals

New Peripherals on EFR32xG27	PRS Signal Defines in em_prs.h
Coulomb Counter	prsSignalDCDC_MONO70NSANA
External Tamper Detect (ETAMPDET)	prsSignalETAMPDET_TAMPERSRCETAMPDET
Real Time Clock with Capture (RTCC)	prsSignalRTCC_CCV0
	prsSignalRTCC_CCV1
	prsSignalRTCC_CCV2
Secure Element (SE)	prsSignalSE_COREENGATED
	prsSignalSE_STATE0GATED
	prsSignalSE_STATE1GATED
	prsSignalSE_STATE2GATED
Universal Synchronous/Asynchronous Receiver Transmitter (USART1)	prsSignalUSART1_CS
	prsSignalUSART1_IRTX
	prsSignalUSART0_RTS
	prsSignalUSART1_RXDATA
	prsSignalUSART1_TX
	prsSignalUSART1_TXC

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 value that is with specific regard to EFR32xG27.

5.2 Peripheral Software Resources

The software resources for Keyscan and System RTC (SYSRTC) are not within the emlib library. They are located in custom_sdk_loc/platform/peripheral/inc and custom_sdk_loc/platform/peripheral/src.

5.3 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 decrease on EFR32xG27, certain flash pages might not be available to be configured for NVM3 usage that were available on EFR32xG24. Such configuration changes can be done inside the linker file.

6. Revision History

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

SILICON LABS

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

www.silabs.com