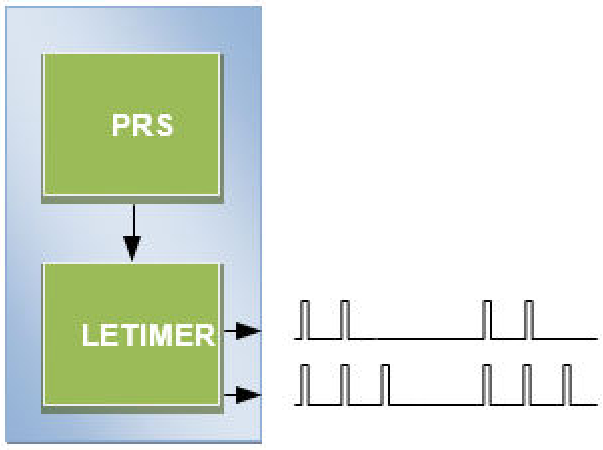
AN0026.1: EFM32 and EFR32 Wireless SOC Series 1 Low Energy Timer

This application note gives an overview of the Low Energy Timer (LETIMER) and demonstrates how to use it on the EFM32 and EFR32 wireless SOC Series 1 devices. For LETIMER information of EFM32 and EZR32 Wireless MCU Series 0 devices, refer to *AN0026.0: EFM32 and EZR32 wireless MCU Series 0 Low Energy Timer*.

This document discusses initializing the LETIMER, a basic setup for operation, and ways to utilize the added LETIMER functionality in more advanced applications.

This application note includes the following:

- This PDF document.
- Source files (zip).
  - Example C-code.
  - Multiple IDE projects.



KEY POINTS

- 16-bit down count timer.
- 2 Compare match registers.
- Compare register 0 can be top timer top value.
- Compare registers can be double buffered.
- Double buffered 8-bit Repeat Register.
- Same clock source as the Real Time Counter.
- LETIMER can be triggered (started) by an RTC event or by software.
- LETIMER can be started, stopped, and/or cleared by PRS.
- 2 output pins can optionally be configured to provide different waveforms on timer underflow:
  - Toggle output pin
  - Apply a positive pulse (pulse width of one LFACLKLETIMER period)
  - PWM
- Interrupt on:
  - Compare matches
  - Timer underflow
  - Repeat done
- Optionally runs during debug
- PRS Output

1. Device Compatibility

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

MCU series 1 consists of the following:

- EFM32 Jade Gecko (EFM32JG1/EFM32JG12)
- EFM32 Pearl Gecko (EFM32PG1/EFM32PG12)
- EFM32 Giant Gecko (EFM32GG11)
- EFM32 Tiny Gecko (EFM32TG11)

Wireless SoC series 1 consists of the followsing:

- EFR32 Blue Gecko (EFR32BG1/EFR32BG12/EFR32BG13/EFR32BG14)
- EFR32 Flex Gecko (EFR32FG1/EFR32FG12/EFR32FG13/EFR32FG14)
- EFR32 Mighty Gecko (EFR32MG1/EFR32MG12/EFR32MG13/EFR32MG14)

2. Introduction

The unique LETIMERTM, the Low Energy Timer, is a 16-bit timer that is available in energy mode EM2 and EM3 in addition to EM1 and EM0. Because of this, it can be used for timing and output generation when most of the device is powered down, allowing simple tasks to be performed while the power consumption of the system is kept at an absolute minimum. The LETIMER runs from the LFACLK which can be clocked by the LFXO, LFRCO, or ULFRCO.

The LETIMER can be used to output a variety of waveforms with minimal software intervention. The waveforms include PWM, pulses with the duration of one LFACLKLETIMER period, and variable frequency waveforms. The LETIMER can also be configured to start counting on compare matches PRS from the RTCC.

An overview of the LETIMER module is shown in Figure 2.1 (p. 3). The LETIMER is a 16-bit down-counter with two compare registers, LETIMERn_COMP0 and LETIMERn_COMP1. The LETIMERn_COMP0 register can optionally act as a top value for the counter. The repeat counter LETIMERn_REP0 allows the timer to count a specified number of times before it stops. Both the LETIMERn_COMP0 and LETIMERn_REP0 registers can be double buffered by the LETIMERn_COMP1 and LETIMERn_REP1 registers to allow continuous operation. The timer can generate a single pin output, or two linked outputs.

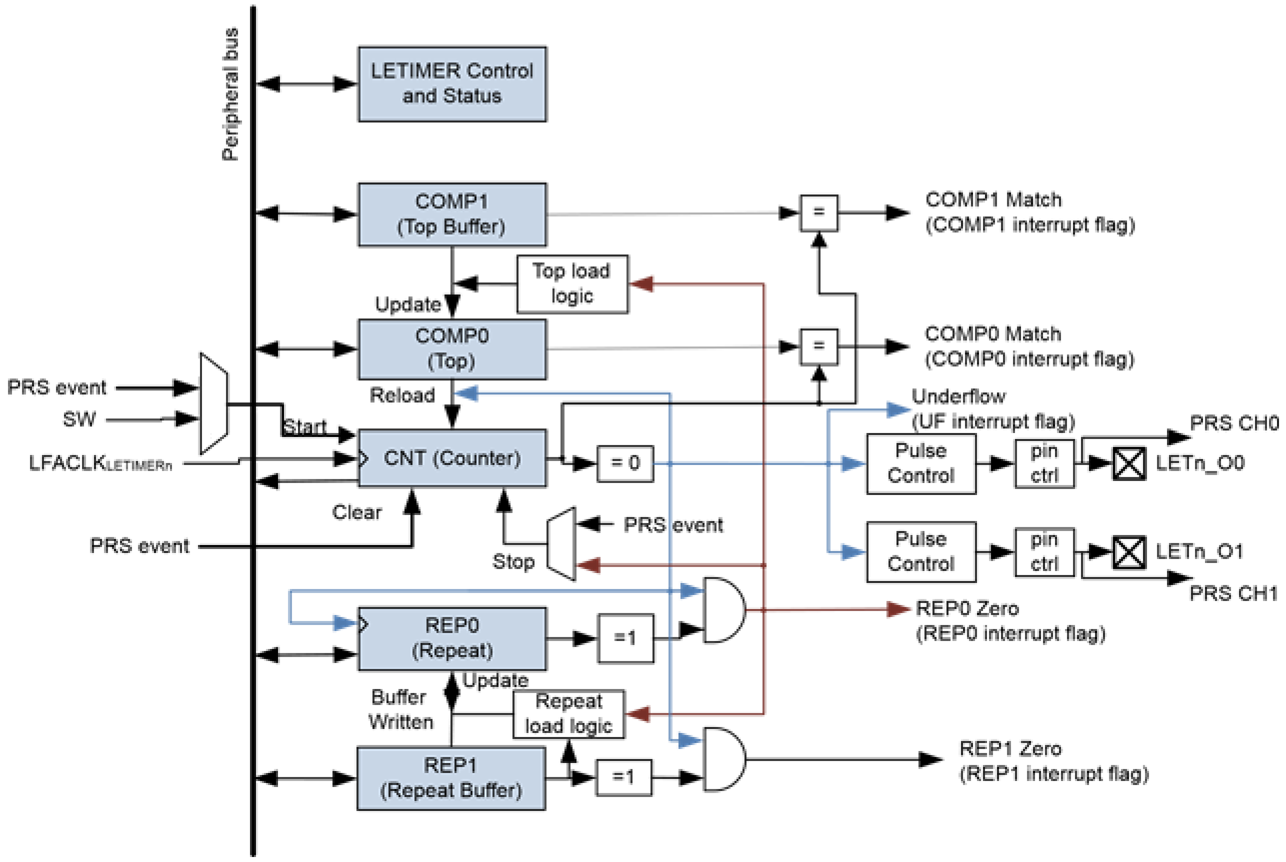


Figure 2.1. LETIMER Overview

3. LETIMER Features

3.1 Compare Register

The LETIMER has two compare match registers, LETIMERn_COMP0 and LETIMERn_COMP1. Each of these compare registers are capable of generating an interrupt when the counter value LETIMERn_CNT becomes equal to their value. When LETIMERn_CNT becomes equal to the value of LETIMERn_COMP0, the interrupt flag COMP0 in LETIMERn_IF is set, and when LETIMERn_CNT becomes equal to the value of LETIMERn_COMP1, the interrupt flag COMP1 in LETIMERn_IF is set.

The compare values can be set using `LETIMER_CompareSet(LETIMER_TypeDef *letimer, unsigned int comp, uint32_t value)` from emlib.

3.2 Top Value

If COMP0TOP in LETIMERn_CTRL is set, the value of LETIMERn_COMP0 acts as the top value of the timer, and LETIMERn_COMP0 is loaded into LETIMERn_CNT on timer underflow. Otherwise the timer wraps around to 0xFFFF. The underflow interrupt flag UF in LETIMERn_IF is set when the timer reaches zero.

If BUFTOP in LETIMERn_CTRL is set, the value of LETIMERn_COMP0 is buffered by LETIMERn_COMP1. In this mode, the value of LETIMERn_COMP1 is loaded into LETIMERn_COMP0 every time LETIMERn_REP0 is about to decrement to 0.

3.3 Repeat Mode

By default, the timer wraps around to the top value or 0xFFFF on each underflow, and continues counting. The repeat counters can be used to get more control of the operation of the timer, including defining the number of times the counter should wrap around. There are four repeat modes available which are detailed in the table below.

Table 3.1. LETIMER Repeat Modes

REPMODE	Mode	Description
00	Free	The timer runs until it is stopped
01	One-shot	The timer runs as long as LETIMERn_REP0 != 0. LETIMERn_REP0 is decremented at each timer underflow.
10	Buffered	The timer runs as long as LETIMERn_REP0 != 0. LETIMERn_REP0 is decremented on each timer underflow. If LETIMERn_REP1 has been written, it is loaded into LETIMERn_REP0 when LETIMERn_REP0 is about to be decremented to 0.
11	Double	The timer runs as long as LETIMERn_REP0 != 0 or LETIMERn_REP1 != 0. Both LETIMERn_REP0 and LETIMERn_REP1 are decremented at each timer underflow.

The interrupt flags REP0 and REP1 in LETIMERn_IF are set whenever LETIMERn_REP0 or LETIMERn_REP1 are decremented to 0 respectively. REP0 is also set when the value of LETIMERn_REP1 is loaded into LETIMERn_REP0 in buffered mode.

The function LETIMER_RepeatSet(LETIMER_TypeDef *letimer, unsigned int rep, uint32_t value) from the emlib can be used to set the values of the repeat registers.

3.3.1 Free Mode

In the free running mode, the LETIMER acts as a regular timer, and the repeat counter is disabled. The LETIMER can be started by writing the START bit in LETIMERn_CMD and runs until it is stopped using the STOP bit in the same register.

3.3.2 One-shot Mode

The one-shot repeat mode is the most basic repeat mode. In this mode, the repeat register LETIMERn_REP0 is decremented every time the timer underflows, and the timer stops when LETIMERn_REP0 goes from 1 to 0. In this mode, the timer counts down LETI-MERn_REP0 times, i.e. the timer underflows LETIMERn_REP0 times. LETIMERn_REP0 can be written while the timer is running to allow the timer to run for longer periods at a time without stopping.

3.3.3 Buffered Mode

The Buffered repeat mode allows buffered timer operation. When started, the timer runs LETIMERn_REP0 number of times. If LETIMERn_REP1 has been written since the last time it was used and it is nonzero, LETIMERn_REP1 is then loaded into LETIMERn_REP0, and counting continues the new number of times. The timer keeps going as long as LETIMERn_REP1 is updated with a nonzero value before LETIMERn_REP0 is finished counting down. If the timer is started when both LETIMERn_CNT and LETIMERn_REP0 are zero but LETIMERn_REP1 is non-zero, LETIMERn_REP1 is loaded into LETIMERn_REP0, and the counter counts the loaded number of times.

3.3.4 Double Mode

The Double repeat mode works much like the one-shot repeat mode with the difference that the LETIMER counts as long as either LETIMERn_REP0 or LETIMERn_REP1 is larger than 0.

3.4 Clock Source

The LETIMER clock source and its prescaler value are defined in the Clock Management Unit (CMU). The LFACLKLETIMERn has a frequency given by the equation below where the exponent LETIMERn is a 4 bit value in the CMU_LFAPRESC0 register.

$$f_{LFACLK\_LETIMERn} = 32.768/2^{LETIMERn}$$

To use this module, the LE interface clock must be enabled in CMU_HFBUSCLKEN0, in addition to the module clock. Clock enabling and prescaling is covered in *AN0004: Clock Management Unit*.

3.5 PRS Trigger

The LETIMER can be configured to start on compare match events PRS signal. RTCC compare match event could generate PRS to start the LETIMER.

3.6 Underflow Output Action

For each of the LETIMER outputs an underflow output action can be set. The configured output action is performed every time the counter underflows while the respective repeat register is nonzero. In PWM mode, the output is similarly only changed on COMP1 match if the repeat register is nonzero. The different output actions are shown in the table below.

Table 3.2. LETIMER Underflow Output Actions

UF0A0/UF0A1	Mode	Description
00	Idle	The output is held at its idle value
01	Toggle	The output is toggled on LETIMERn_CNT underflow if LETIMERn_REPx is nonzero
10	Pulse	The output is held active for one clock cycle on LETIMERn_CNT underflow if LETIMERn_REPx is nonzero. It then returns to its idle value
11	PWM	The output is set idle on LETIMERn_CNT underflow and active on compare match with LETIMERn_COMP1 if LETIMERn_REPx is nonzero.

The LETIMER outputs must be routed to pins using the LETIMERn_ROUTEPEN and LETIMERn_ROUTELOC0 registers. The selected pins must be enabled as output in the GPIO module. Pin configuration is covered in *AN0012: GPIO*.

3.7 Interrupt

There are 5 interrupts available in the LETIMER. One interrupt for when each of the Repeat Counters (REP0 and REP1) reaches zero, one when the LETIMER counter matches the value of each compare register (COMP0 and COMP1), and one when the LETIMER underflows.

These interrupts can be enabled, disabled, and cleared using the following functions from the emlib:

- LETIMER_IntEnable(LETIMER_TypeDef *letimer, uint32_t flags) enables interrupts
- LETIMER_IntDisable(LETIMER_TypeDef *letimer, uint32_t flags) disables interrupts
- LETIMER_IntClear(LETIMER_TypeDef *letimer, uint32_t flags) clears interrupts

3.8 Register Access and Synchronization

There are 2 modes to access the low energy peripheral register, they are uses immediate synchronization and immediate synchronization. For the LETIMER peripheral, the device uses immediate synchronization mode. This doesn't experience a delay from when a value is written to when it takes effect in the peripheral. The values are updated immediately on the peripheral write access. If such a write is done close to an edge on the clock of the peripheral, the write is delayed to after the clock edge. This will introduce wait-states on the peripheral access.

4. Configuration

The LETIMER can be easily and quickly configured using `LETIMER_Init(LETIMER_TypeDef *letimer, const LETIMER_Init_TypeDef *init) function` from emlib. This function allows the configuration of the following parameters:

- Start counting when the initialization is complete
- Counter running during debug
- Use COMP0 register as TOP value
- Load COMP1 to COMP0 when REP reaches 0
- Idle value for output 0
- Idle value for output 1
- Underflow output 0 action
- Underflow output 1 action
- Repeat mode

5. Software Examples

This software example project in this application note are intended for EFM32 Starter Kits (STK) and the EFR32MG radio board. Each project contains three test modes demonstrating the LETIMER features.

The output pins are available in the expansion or breakout headers for different STKs and radio boards in the table below.

Table 5.1. Output Pin Map

STK or radio board	Output 0		Output 1	
	Pin	Position	Pin	Position
EFR32MG1	PA0	J101 P9	PA1	J101 P11
EFR32MG12	PA0	J102 P33	PA1	J102 P34
EFR32MG13	PA0	J101 P9	PA1	J101 P11
EFR32MG14	PA0	J101 P9	PA1	J101 P11
SLSTK3401A	PA0	EXP 12 or J101 P5	PA1	EXP 14 or J101 P7
SLSTK3402A	PA0	EXP 12 or J101 P5	PA1	EXP 14 or J101 P7
SLSTK3301A	PD6	EXP 16 or J102 P13	PD7	EXP 15 or J102 P14
SLSTK3701A	PD6	J102 P26	PD7	J102 P28
SLSTK3301A	PD6	EXP 16 or J102 P13	PD7	EXP 15 or J102 P14

5.1 PWM and pulse Output

In PWM mode the LETIMER is configured to run in free mode with PWM on output 1 and pulses on output 0. The value of COMP0 is used as TOP value for the counter and is loaded after each underflow. Using underflow interrupts, the value of COMP1 is decremented throughout the program execution resulting in a variable PWM duty-cycle.

The PWM frequency and duty-cycle can be obtained using the formulas below.

PWM Frequency Equation

$$f_{PWM} = 32768 / TOP$$

PWM Duty-cycle Equation

$$DS_{PWM} = COMP1 / COMP0 \times 100$$

The purpose of this example is to demonstrate how the LETIMER can be used to output PWM and/or pulses with little CPU intervention while keeping the energy consumption to a minimum.

5.2 RTCC PRS Triggered Counter

The RTCC mode demonstrates how the RTCC PRS can be used to trigger the LETIMER. The LETIMER is configured to start with counting on a PRS signal. RTCC COMP0 match was configured to generate the PRS event. The LETIMER was configured to generate pulses on output 0 and One-shot repeat mode. The figure below illustrates the program flow.

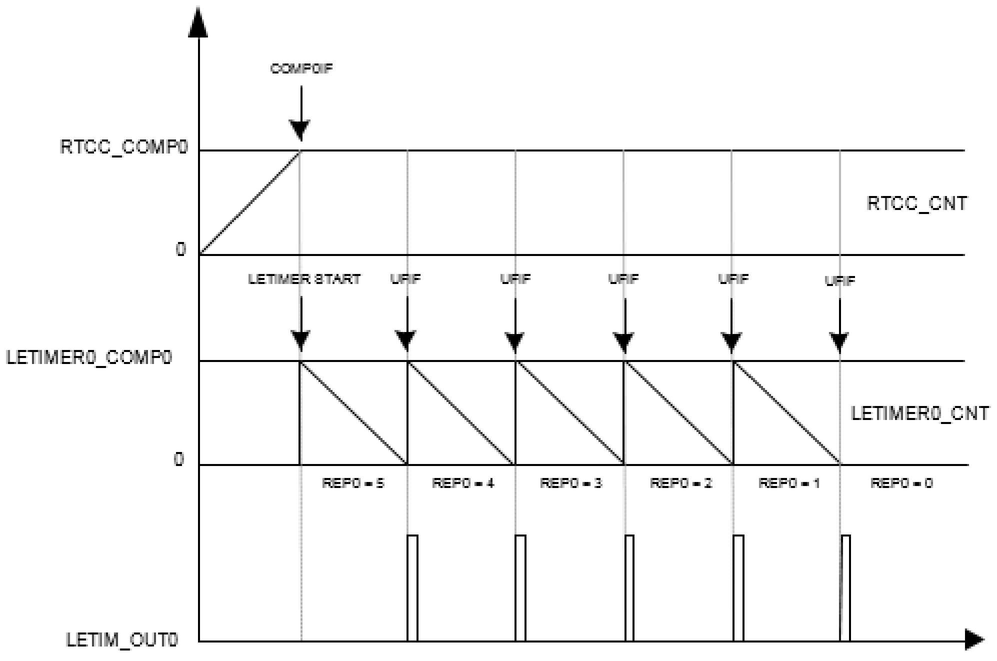


Figure 5.1. RTCC Trigger

The RTCC generates a compare match PRS event after 5 seconds (RTC_COMP0 = 5) of program execution which will trigger the LE-TIMER to start counting. The LETIMER will count down while LETIMERn_REP0 != 0 generates a pulse on each underflow. For this project, LETIMER_REP0 has the value of 5 so there will be 5 pulses.

Note: The RTCC continues counting after the compare match. If it wraps around the top value and generates a new compare match the LETIMER will not be triggered because LETIMERn_REP0 = 0.

5.3 GPIO PRS Start, Stop counter

The GPIO mode demonstrates how the GPIO can be used to start and stop the LETIMER through PRS function. The LETIMER is con-figured to start counting when push button 0 is pushed, with pulses on output and free mode, in the meantime the LETIMER is config-ured to stop counting when push button 1 is pushed.

6. Revision History

Revision 1.08

February, 2018

- Split AN0026 into AN0026.0 and AN0026.1 for MCU/Wireless Series 0 and MCU/Wireless Series 1 respectively.
- Added support for EFM32 Series 1 and EFR32 Series 1 devices.
- Added more feature projects.
- Re-organized the example code structure.

Revision 1.07

May, 2014

- Updated example code to CMSIS 3.20.5
- Changed to Silicon Labs license on code examples
- Added project files for Simplicity IDE
- Removed makefiles for Sourcery CodeBench Lite

Revision 1.06

September, 2013

- New cover layout.

Revision 1.05

May, 2013

- Added software projects for ARM-GCC and Atollic TrueStudio.

Revision 1.04

November, 2012

- Adapted software projects to new kit-driver and bsp structure.
- Added projects for Tiny and Giant Gecko STKs.

Revision 1.03

August, 2012

- Added projects for Tiny and Giant Gecko STKs.

Revision 1.02

April, 2012

- Adapted software projects to new peripheral library naming and CMSIS_V3.

Revision 1.01

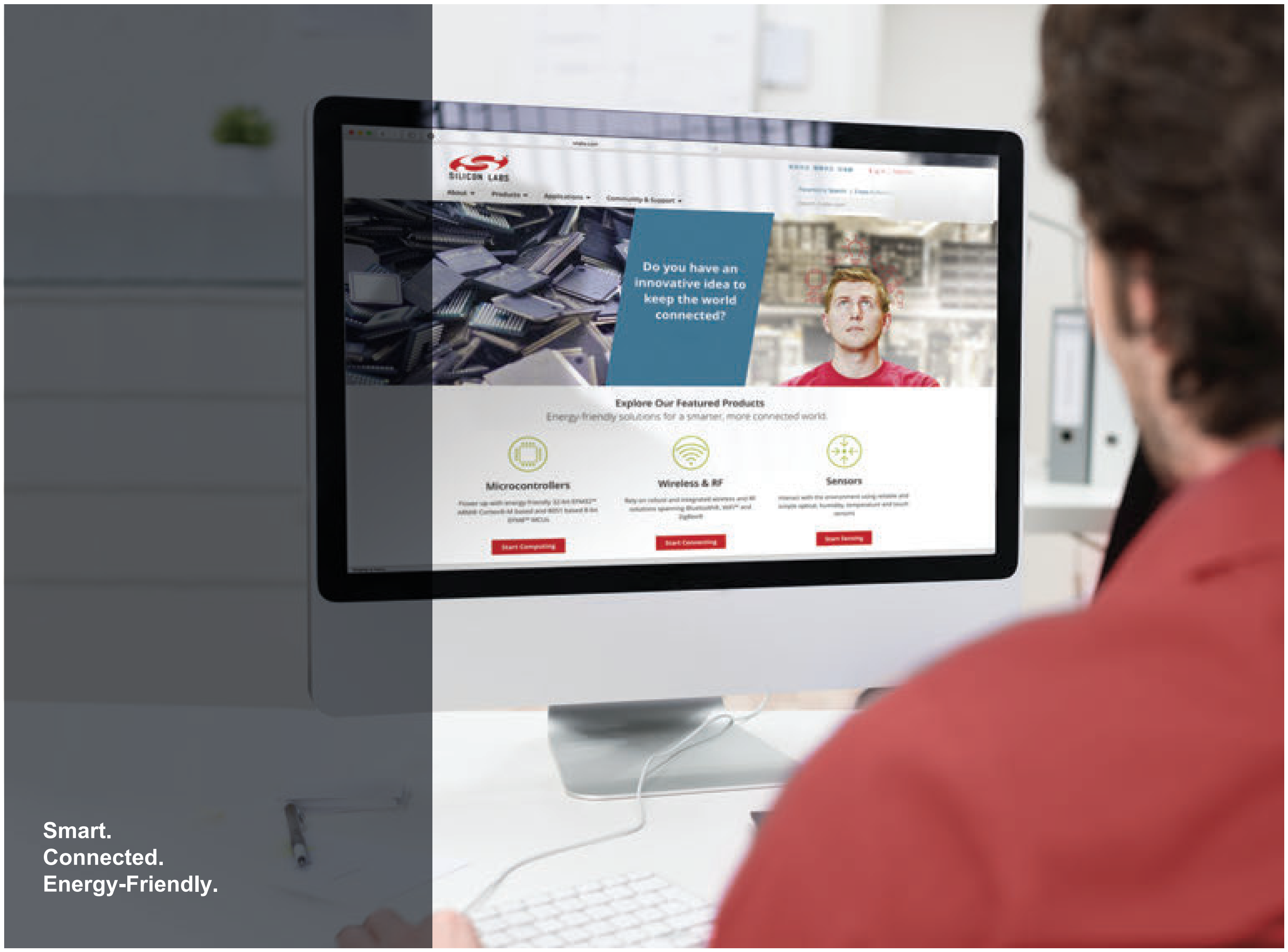
October, 2011

- Updated IDE project paths with new kits directory.

Revision 1.00

December, 2010

- Initial revision.



Products
www.silabs.com/products

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