Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

CE219490 – PSoC 6 MCU Breathing LED using Smart IO

Objective

This example demonstrates the flexibility of the PSoC® 6 MCU Smart IO Component, by implementing the LED breathing effect exclusively in hardware with no CPU usage beyond initialization.

Overview

This example uses a PWM and PSoC 6 MCU Smart IO Component to implement a breathing LED, where an LED gradually cycles through increasing and decreasing brightness levels. There is no CPU usage except for the initialization of PWM and Smart IO Components. This example also demonstrates how to use Smart IO to route the same signal through multiple I/O pins on the same port. This is demonstrated by inverting the signal using Smart IO and then routing the signal to another pin thus creating two breathing LEDs that are out of phase.

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C (Arm® GCC 5.4)

Associated Parts: PSoC 6 MCU

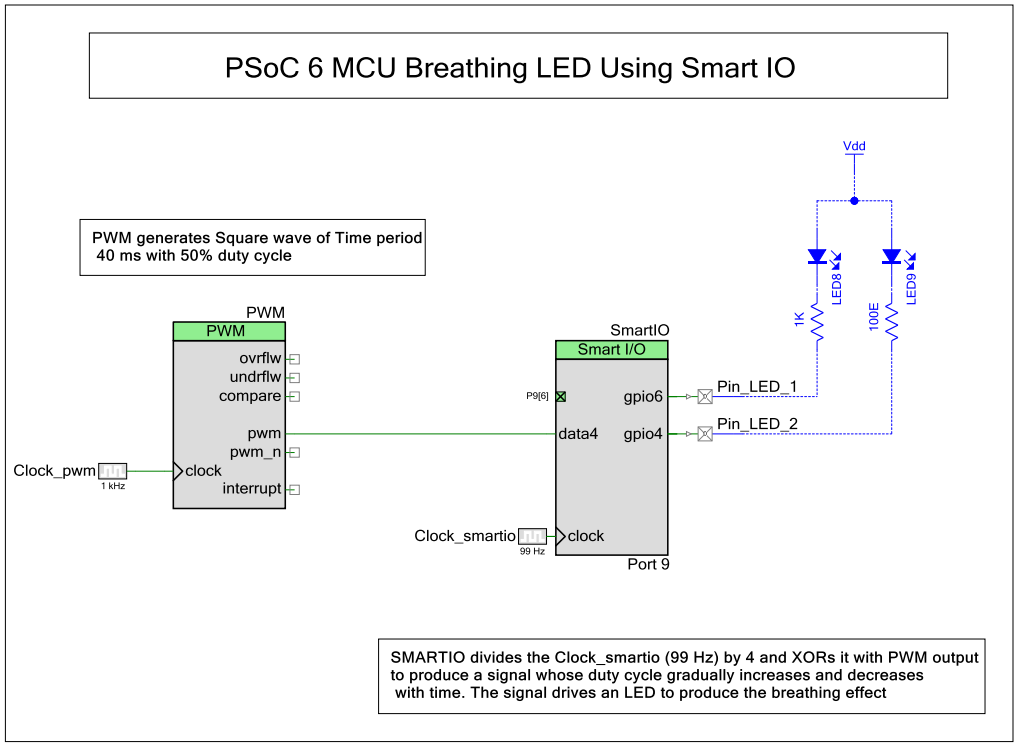
Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Design

This design consists of a PWM Component and a Smart IO Component, both creating square waves of slightly different frequencies. These square waves are routed through an exclusive-OR (XOR) gate within the Smart IO Component, yielding a signal with a gradually changing duty cycle. The rate of change is proportional to the difference between the output square wave frequencies.

The signal is then output to gpio4 and gpio6 on the port. Driving LEDs with this signal results in a “breathing” effect, where the LEDs gradually get brighter and dimmer. Additionally, gpio6 inverts the gpio4 signal and creates a breathing effect that is of opposite polarity to the signal on gpio4.

Figure 1. Breathing LED Project Schematic



The PWM is driven by a 1-kHz clock with a period of 40 counts and a compare value of 20 counts. This gives a 50 percent duty cycle square wave with a 40-ms period. The Smart IO Component is clocked at 99 Hz using a divided clock sourced from PeriClk. This input clock is divided by 4 using the lookup tables (LUTs) of the Smart IO Component to produce a square wave with a 40.4-ms period.

To generate a square wave signal with a time period close to 40 ms, a 99-Hz clock is divided by 4 using a synchronous sequential circuit, which is realized using the LUTs of the Smart IO Component.

To implement a divide-by-4 sequential circuit, consider the state transition values shown in Table 1:

Table 1. State Transition Table for a Divide-by-4 Sequential Circuit

CLK	Present State		Next State		D0	D1
	Q0	Q1	Q0	Q1		
↑	0	0	1	1	1	1
↑	1	1	0	1	0	1
↑	0	1	1	0	1	0
↑	1	0	0	0	0	0

From this state transition table, you can observe that Q0 is half the frequency of Clock_smartio and Q1 is 1/4th frequency of Clock_smartio. This sequential logic can be implemented using the LUTs of the Smart IO Component.

According to Table 1:

$D0 = \overline{Q0}$ $\quad D1 = Q0\ XNOR\ Q1$

Figure 2. LUT Configuration and Timing Diagram

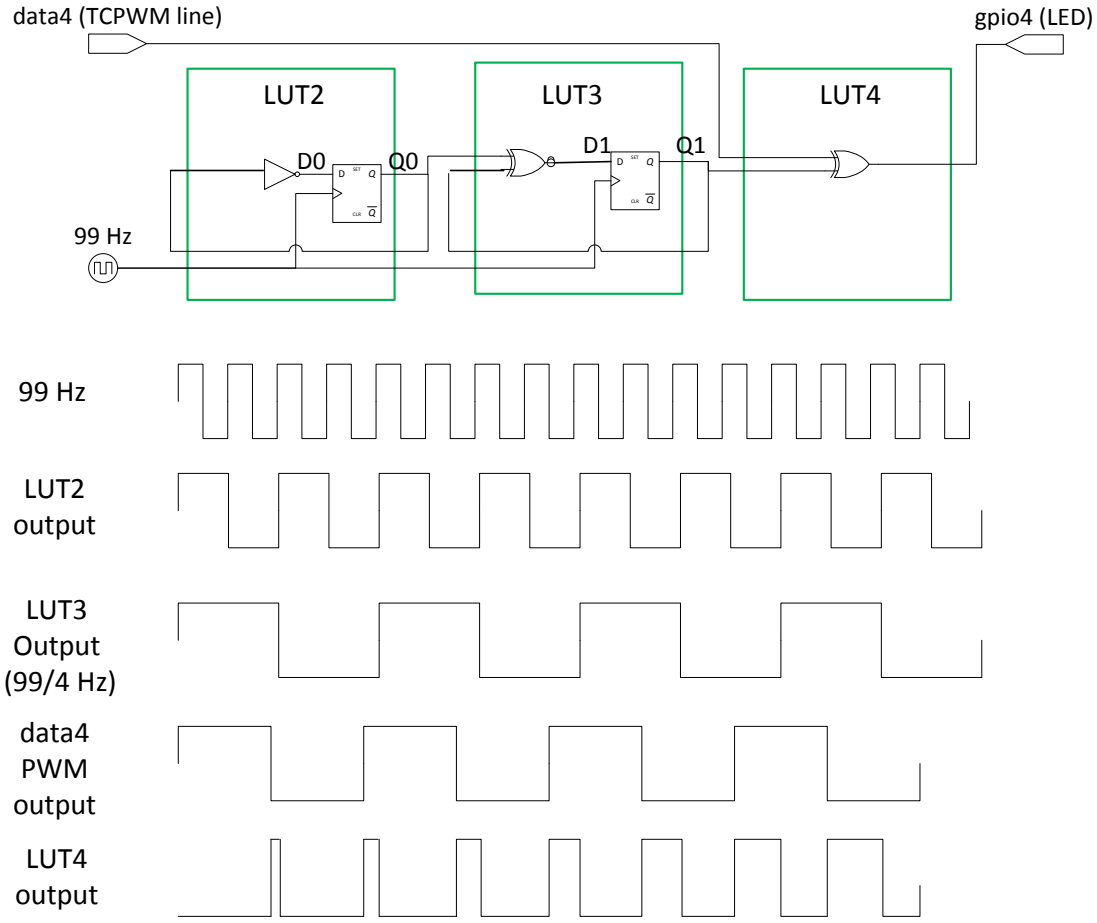


Figure 2 shows the implementation of this logic using LUT 2 and LUT 3. In addition, the divided clock is XORed with the PWM output using LUT 4 to generate a signal with the duty cycle gradually increasing and decreasing over time as shown in Figure 2. The output of LUT 4 is driven to gpio4 output. The LUT 4 output is inverted using LUT 6, and then driven to gpio6 output. This creates a breathing effect that is of opposite polarity to the signal driven on gpio4. To know more about implementation digital functions using the Smart IO component, see the Smart IO Component Datasheet.

The firmware is implemented in *main_cm0p.c* and performs only the component initialization functions:

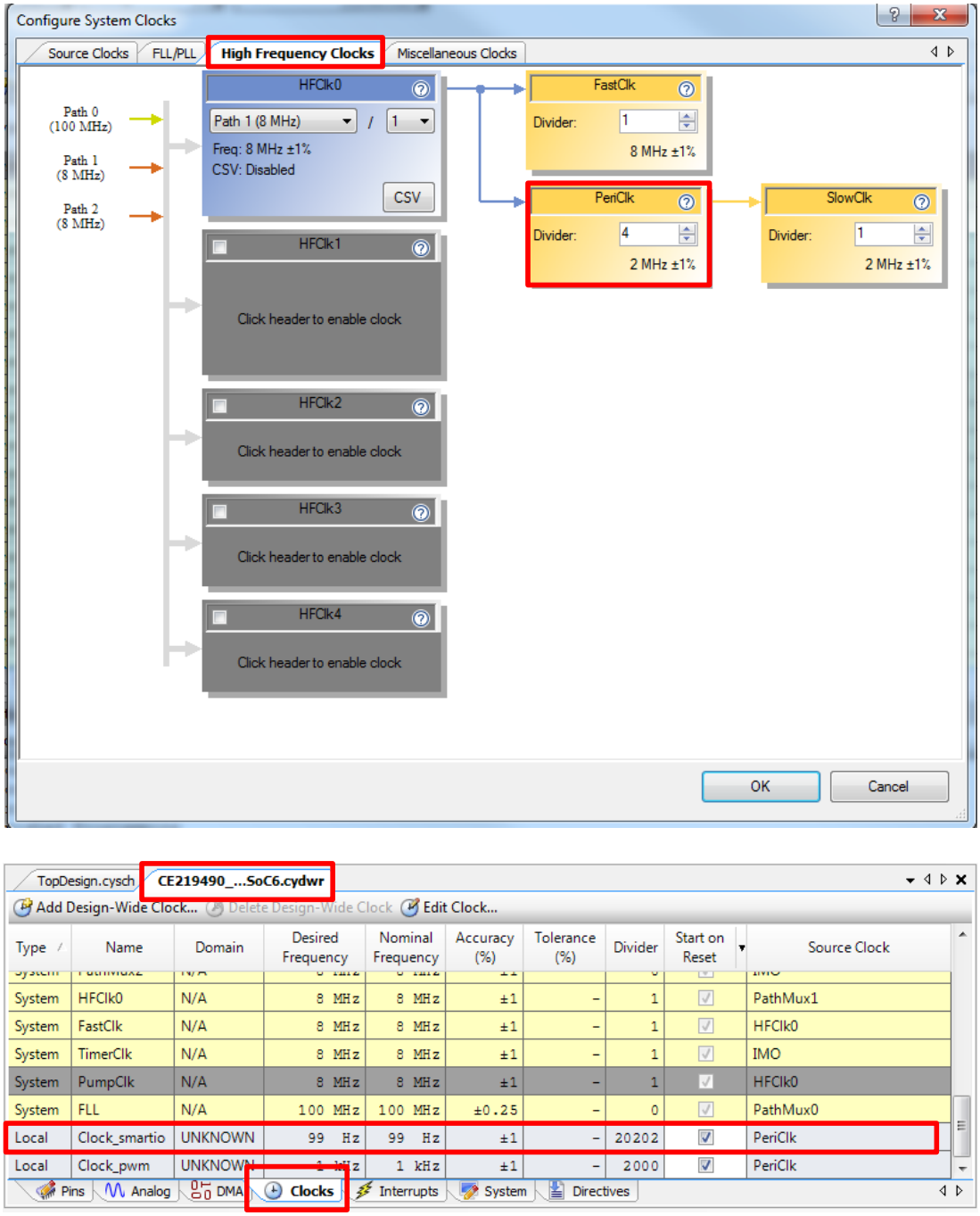
1. Starts the Smart IO Component
2. Starts the PWM Component

The CM4 core is not used in this example.

Design Considerations

In this example, a clock frequency of 99 Hz is generated. For generating such low-frequency clocks, PeriClk is set to 2 MHz (IMO/4). PSoC Creator automatically sets a clock divider value of 20202 to generate this frequency. You can see this in the **Design Wide Resources** (*cydwr*) tab of PSoC Creator. Figure 3 shows the clock configuration setting used in this example.

Figure 3. Clock Configuration Setting



Hardware Setup

Port 8 and Port 9 are the Smart IO-enabled ports in PSoC 6 MCU. In CY8CKIT-062 BLE, Port 8 of PSoC 6 MCU is dedicated to the CapSense functionality. Therefore, in this kit, only Port 9 can be used for Smart IO-based projects.

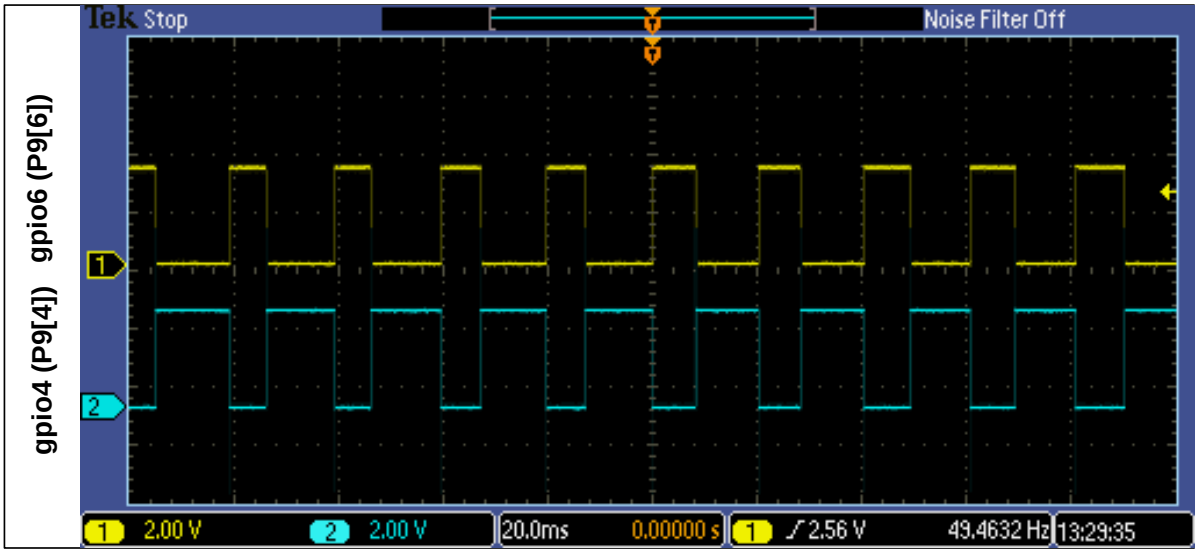
In this example, you need to connect external LEDs to Port 9 because there are no LEDs connected to Port 9 in CY8CKIT-062- BLE. You can use LED8 (**P1 [5])** and LED9 (**P13 [7]**) on the kit. Connect **P9[4]** to **P1[5]** and **P9[6]** to **P13[7]**.

Operation

1. Connect the PSoC 6 BLE Pioneer kit baseboard (CY8CKIT-062 BLE) to your computer's USB port.
2. Build the project and program the PSoC 6 MCU device on the CY8CKIT-062 BLE Kit. For more information on device programming, see PSoC Creator Help.
3. Connect two LEDs to pins **P9[4]** and **P9[6]**. Connect **P9[4]** to **P1[5]** (LED8) and **P9[6]** to **P13[7]** (LED9).

You can observe the breathing effect of opposite polarity on the two LEDs. You can also probe the two pins (**P9[4]** and **P9[6]**) to observe the signals on an oscilloscope as shown in Figure 4.

Figure 4. Breathing LED Signals as Displayed on Oscilloscope



The sections that follow discuss the Components, parameter settings, and resources used to make the example.

Components

Table 2 lists the PSoC Creator Components and hardware resources used in this example

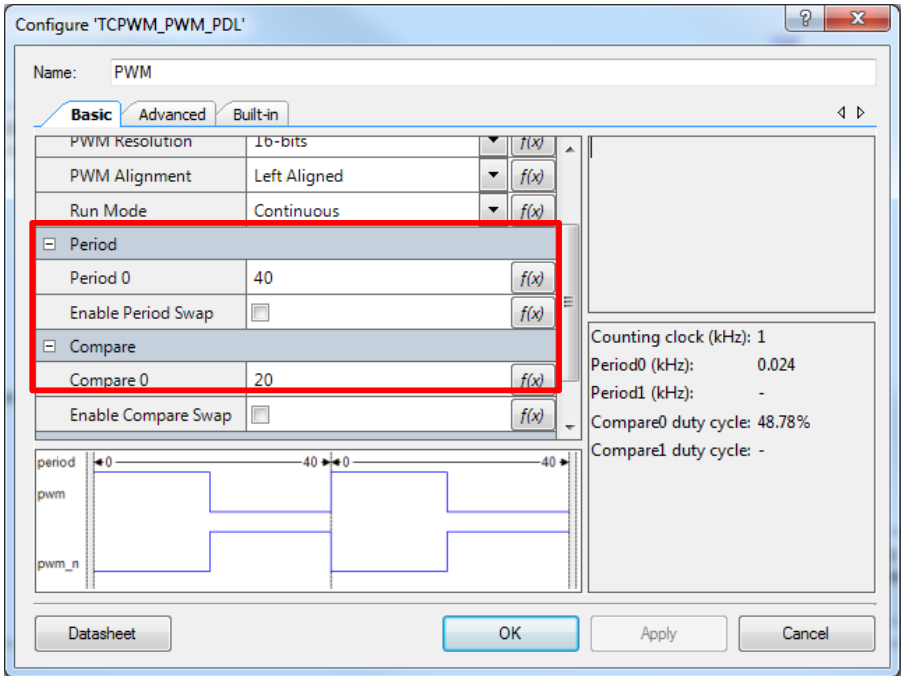
Table 2. List of PSoC Creator Components/PSoC Designer User Modules

Component	Hardware Resources
PWM	1 TCPWM
SmartIO	1 PRGIO
Pin_LED_1, Pin_LED_2	2 GPIOs
Clock_pwm, Clock_smartio	2 Clock dividers

Parameter Settings

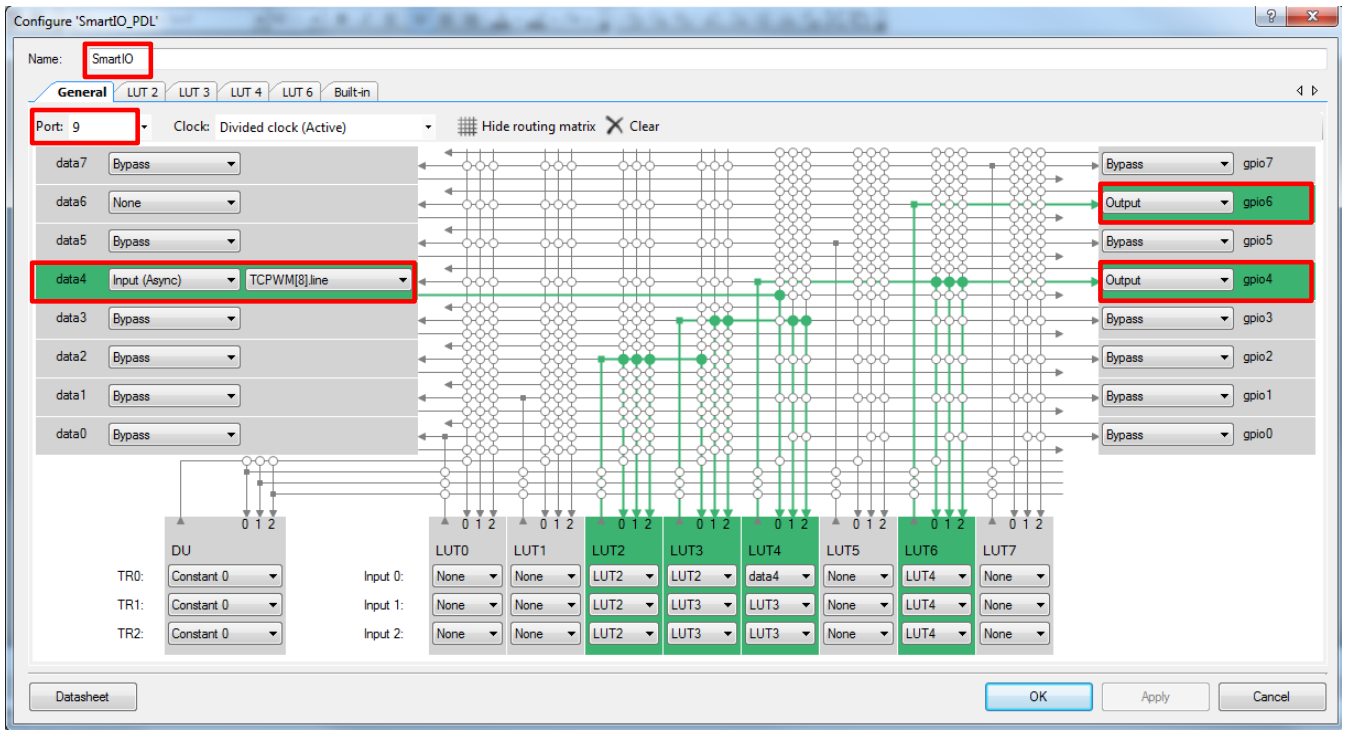
The TCPWM Component is configured as a PWM with a period count of 40 and compare value of 20 counts. The PWM block is driven by a 1-kHz clock source.

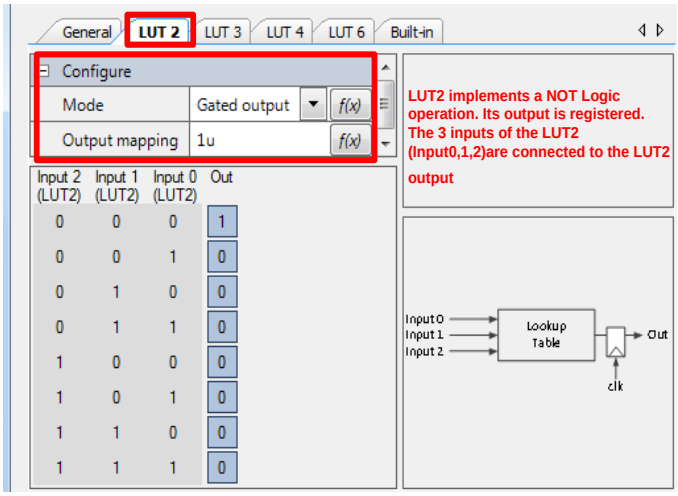
Figure 5. PWM Configuration Settings

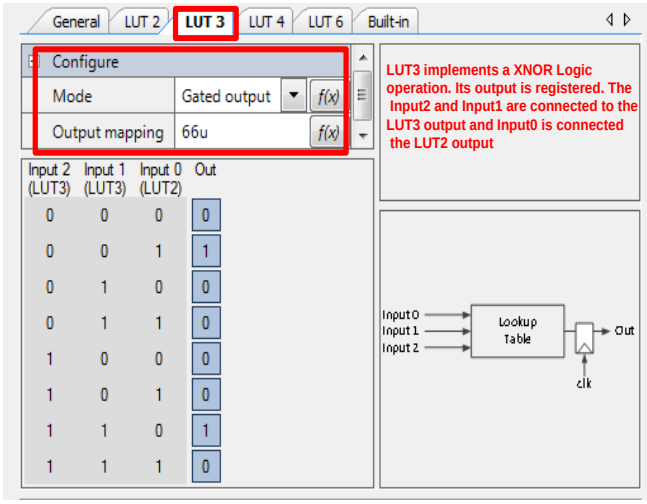


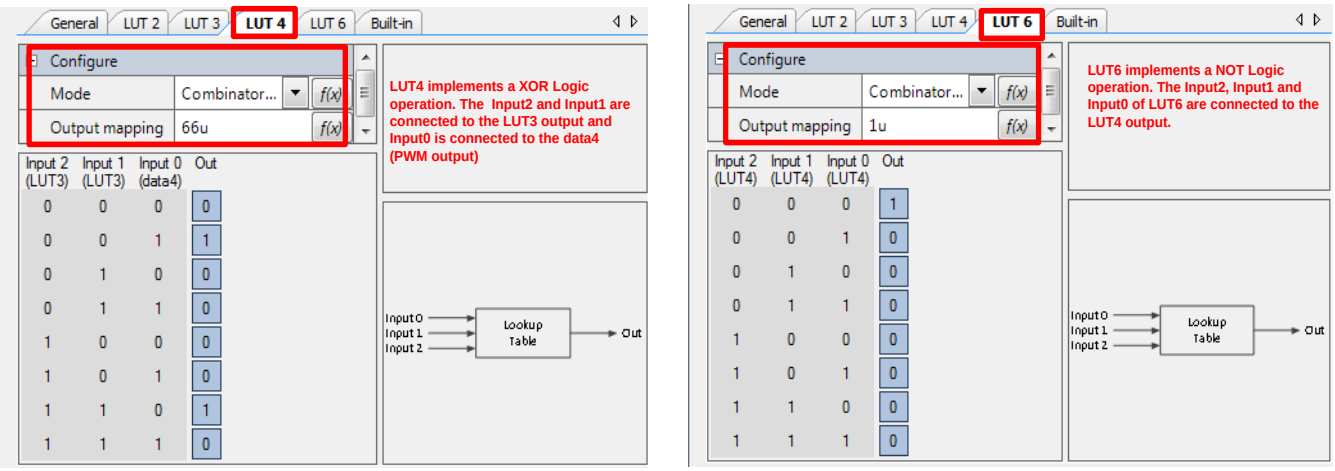
The Smart IO Component is configured to have one input (data4) from the PWM output and two outputs (gpio4 and gpio6). LUT 2 and LUT 3 are configured to divide Clock_smartio by 4 as discussed in the Design section. LUT 4 performs an XOR operation of the divided clock with the PWM output to generate the breathing LED signal. This signal drives the gpio4 output of the Smart IO Component. LUT 6 inverts the breathing LED signal and is brought out to gpio6 output of the Smart IO Component. Figure 6 shows the configuration settings for the Smart IO Component.

Figure 6. Smart IO Configuration









Design-Wide Resources

Figure 7 shows the pin assignments for the CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit.

Figure 7. Device Pin Assignments

	Name	Port	Pin	Lock
	Pin_LED_1	P9[6]	C8	
	Pin_LED_2	P9[4]	C10	

Related Documents

Application Notes	
AN210781 Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes the PSoC 6 MCU with BLE connectivity and how to build your first PSoC Creator project.
PSoC Creator Component Datasheets	
PWM	Supports PWM, Timer/Counter and QuadDec modes
Smart IO	Supports Smart IO peripheral
Pins	Supports connection of hardware resources to physical pins
Clock	Supports clocks dividers for HFCLK
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

Document History

Document Title: CE219490 - PSoC 6 MCU Breathing LED using Smart IO

Document Number: 002-19490

Revision	ECN	Orig. of Change	Submission Date	Description of Change
*A	5842104	VKVK	08/02/2017	Initial public release
*B	6000721	VKVK	12/21/2017	Updated the PSoC project schematic

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.