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.

CE224856 – PSoC 6 MCU with BLE Connectivity: Low-Power Beacon with Hibernate (RTOS)

Objective

This code example demonstrates a Bluetooth Low Energy (BLE) beacon that goes into Hibernate mode once the specified active time has elapsed.

Requirements

Tool: PSoC® Creator™ 4.2; Peripheral Driver Library (PDL) 3.0.4

Programming Language: C (Arm® GCC 5.4.1)

Associated Parts: All PSoC 6 MCUs with BLE Connectivity

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

Overview

This code example demonstrates the ability of PSoC 6 MCU with BLE Connectivity (PSoC 6 BLE) to function as a low-power BLE beacon using the Broadcaster role, which transmits Eddystone frames. Eddystone is an open-source BLE beacon profile released by Google. This code example implements the Eddystone Beacon (RTOS) code example, and in addition saves power by entering Hibernate mode after a specified interval of time. The device can wake up from Hibernate mode either from a RTC alarm or from a switch press.

This code example assumes that you are familiar with PSoC 6 BLE and the PSoC Creator Integrated Design Environment (IDE). If you are new to PSoC 6 BLE, see the application note AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity. For a detailed explanation on Eddystone, see CE218139 – PSoC 6 MCU with BLE Connectivity: Eddystone Beacon (RTOS).

This code example uses FreeRTOS. See PSoC 6 101: Lesson 1-4 FreeRTOS training video to learn how to create a PSoC 6 MCU FreeRTOS project with PSoC Creator. Visit the FreeRTOS website for documentation and API references of FreeRTOS.

Hardware Setup

Set the switches and jumpers on the Pioneer Kit as shown in Table 1.

Table 1. Switch and Jumper Selection

Switch/Jumper	Position	Location
SW5	3.3 V	Front
SW6	PSoC 6 BLE	Back
SW7	V_{DDD} / KitProg2	Back
J8	Installed	Back

Software Setup

None.

Operation

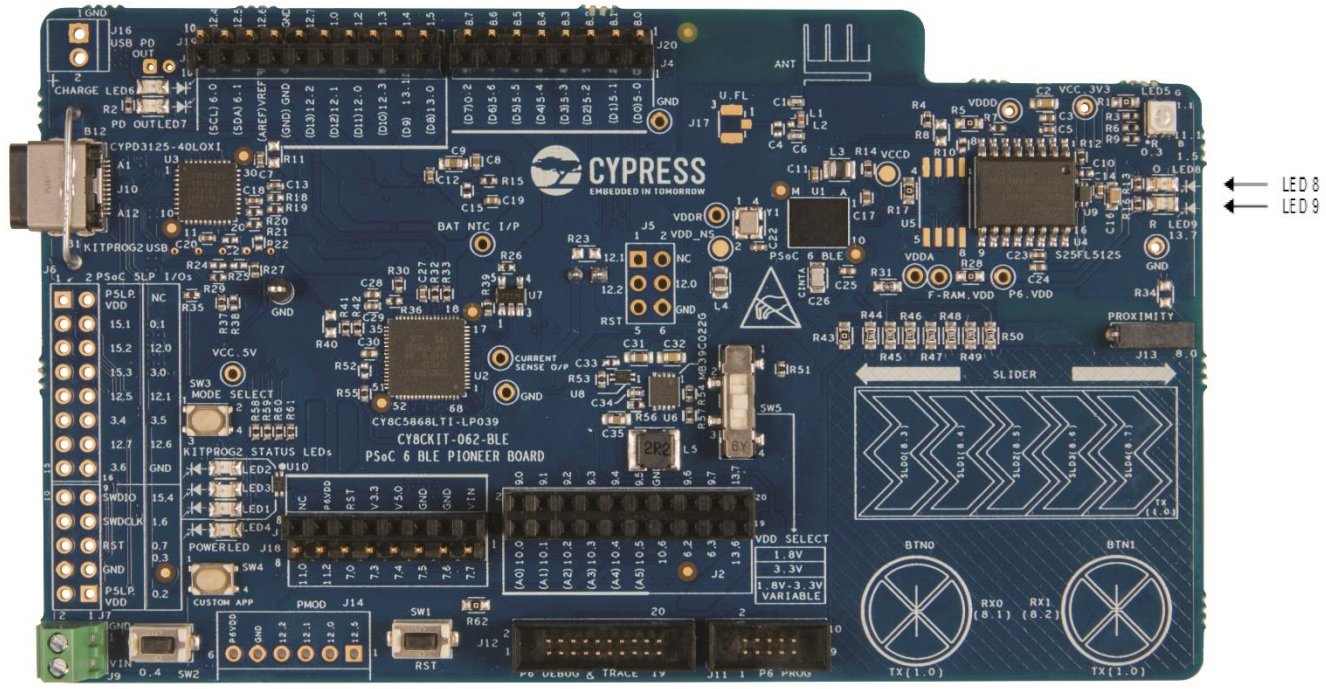
Note: This code example requires an Android device with Android 5.0 or a later version to evaluate.

1. Install a BLE beacon application from Google Play Store that supports Eddystone profile. Locate Beacon, which is the recommended application for this code example, is used to demonstrate the project operation in this section.
2. Plug the CY8CKIT-062-BLE kit board into your computer's USB port.
3. Configure the time-out by editing the parameter `ACTIVE_TIME_SEC` in *hibernate_config.h*.
4. Configure the Hibernate time by editing the parameters `HIBERNATE_TIME_SEC`, `HIBERNATE_TIME_MIN`, and `HIBERNATE_TIME_HR` in *hibernate_config.h*.
5. Build and program the Pioneer Baseboard with the *Low Power Beacon with Hibernate (RTOS) project*. See the Pioneer Kit guide for details on how to program firmware into the device.

   After programming successfully, BLE will start broadcasting URL frames with interleaved Telemetry (TLM) frames. This process will continue until the time-out occurs.

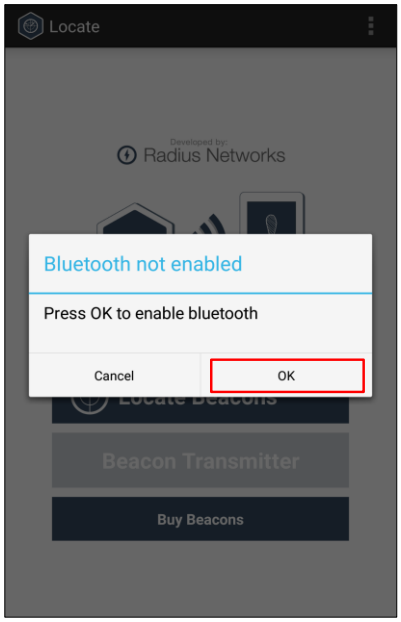
   The red LED (**LED9**) remains ON during the broadcast of URL/UID frames and the orange LED (**LED8**) turns ON during the broadcast of TLM frames. Both the LEDs will remain OFF once time out has occurred. The device can wake up from Hibernate mode either by a RTC alarm or by a switch press.

Figure 1. BLE Broadcasting



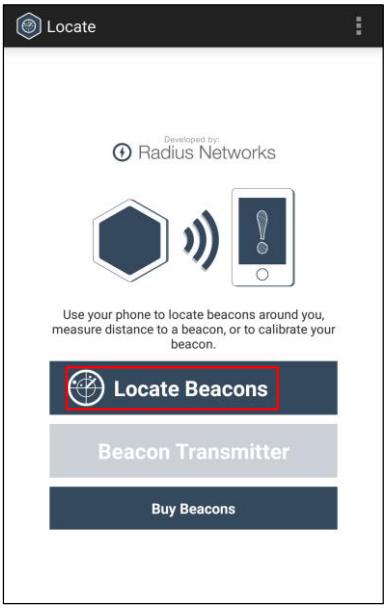
6. Open the Locate Beacon app on the mobile device. If Bluetooth is not enabled on the device, the application will prompt you to enable it.

Figure 2. Enabling Bluetooth



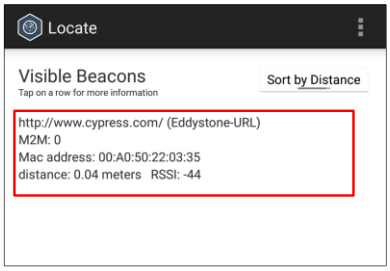
7. After Bluetooth is enabled, select the **Locate Beacons** option as shown in Figure 3.

Figure 3. Locating Eddystone Beacon



The application will search for available beacons and list them. Select the beacon that broadcasts http://www.cypress.com/ in Eddystone-URL mode, as Figure 4 shows.

Figure 4. Selecting the BLE Eddystone Beacon



8. After selecting the beacon, the application will continuously refresh the screen with the URL and TLM frames broadcast by the PSoC 6 MCU, as Figure 5 shows.

Figure 5. Viewing Eddystone Data



9. After the time-out period, the beacon will go into Hibernate mode (LED8 and LED9 will be OFF) and will not be visible in the app.
10. The device wakes up from Hibernate automatically after the RTC alarm time. To wake up the device manually, press the ‘SW2’ switch. The device goes throughs a reset and starts broadcasting again. Repeat from Step 5 to see the beacon.

Design and Implementation

Figure 6 and Figure 7 show the TopDesign schematic of this code example. The BLE Component is configured for Non-discoverable Broadcaster role that transmits Eddystone frames as non-connectable undirected advertisement packets. For more details on the Eddystone configuration, see the BLE Eddystone RTOS code example. A software timer is also configured to create an interrupt that calls a function and puts the system into Hibernate mode.

Two LEDs on the Pioneer Kit are used to indicate the current frame being broadcast. The red LED (**LED9**) remains ON during the broadcast of URL/ Unique Identification (UID) frames and the orange LED (**LED8**) remains ON during the broadcast of TLM frames.

One more software timer is configured to create interrupts at 100-ms time intervals. These time intervals are used to track the uptime (time elapsed since power-on or reset). The uptime data is used in TLM frames.

Switch ‘SW2’ and RTC alarm are configured as Hibernate wakeup sources. The device wakes up from Hibernate automatically after the RTC alarm time. To wake up the device manually press the ‘SW2’ switch.

Table 2. List of PSoC Creator Components

Component	Instance Name	Function
BLE	BLE	The BLE Component is configured as a Non-discoverable Broadcaster role that transmits Eddystone frames as non-connectable undirected advertisement packets.
Digital Output Pin	Pin_LED_Red Pin_LED_Orange	These GPIOs are configured as firmware-controlled digital output pins that control status LEDs.
Digital Input Pin	SW	This GPIO is configured as a digital input pin that wakes up the device from Hibernate mode.

See the PSoC Creator project for more details of PSoC Component configurations and design-wide resource settings.

Figure 6. TopDesign Schematic: BLE and LEDs

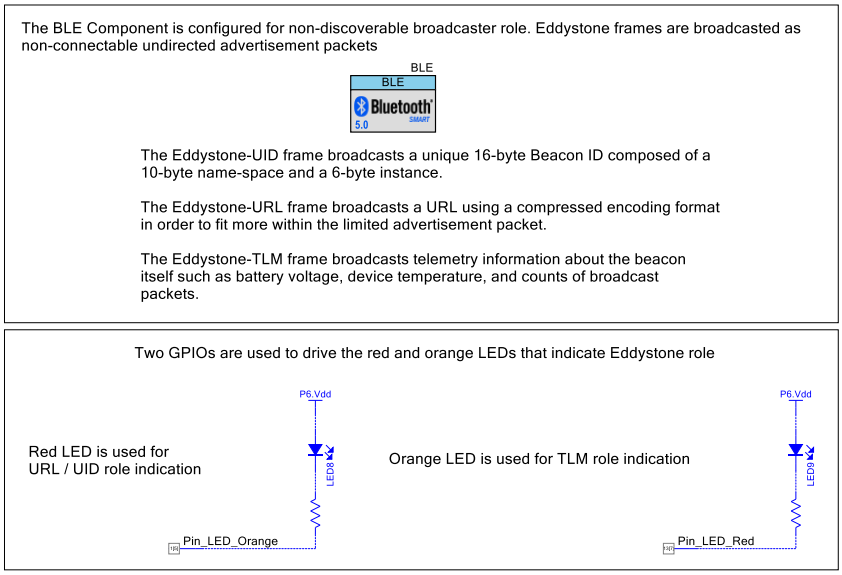
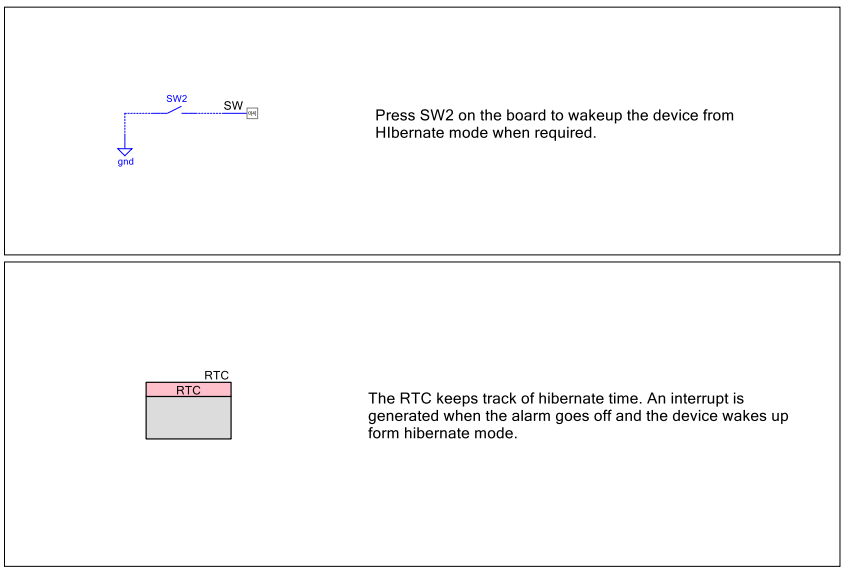


Figure 7 TopDesign Schematic: Hibernate Wakeup



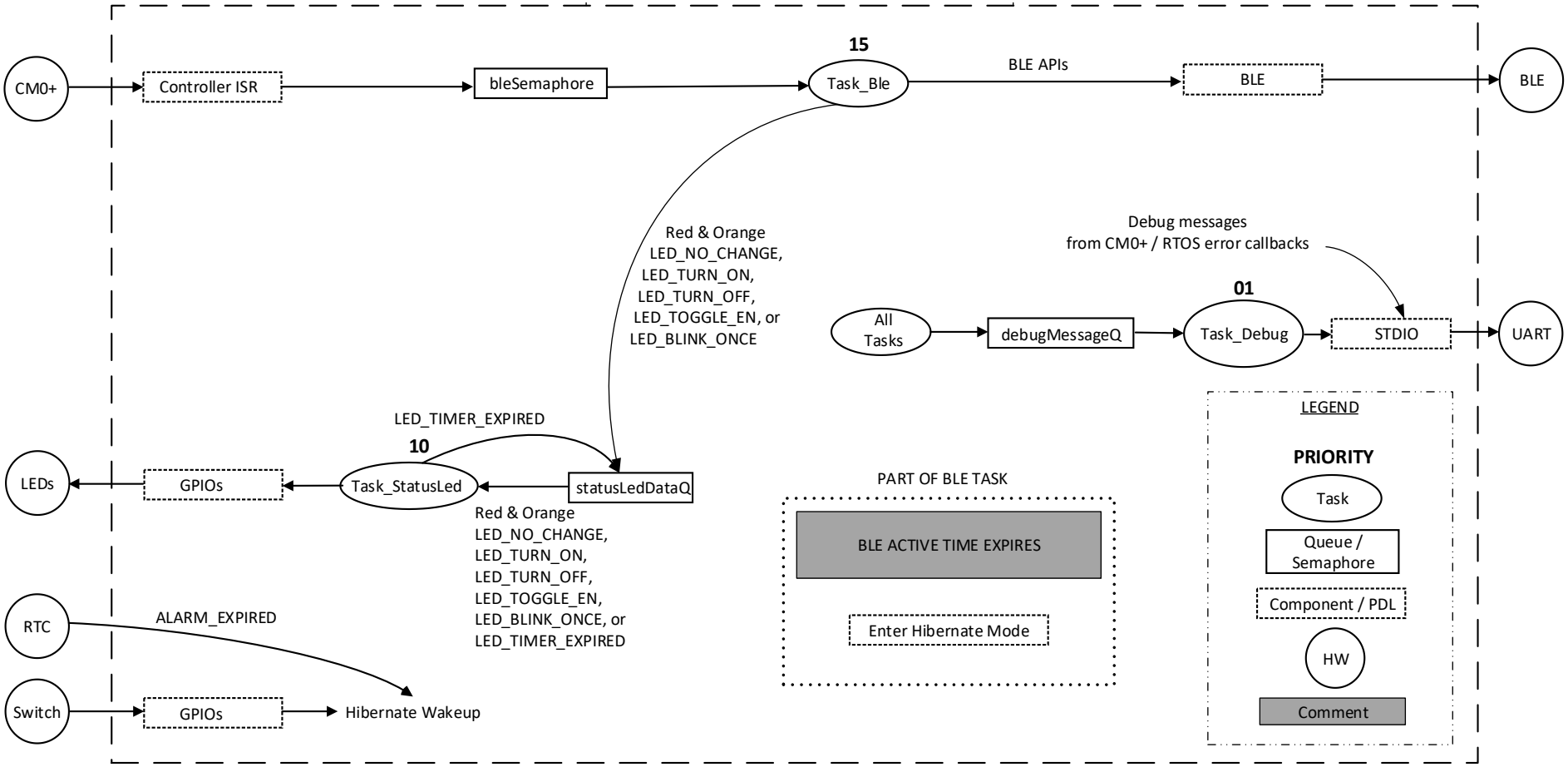
The code example consists of the following files:

- *FreeRTOSConfig.h* contains the FreeRTOS settings and configuration. Non-default settings are explained with in-line comments.
- *main_cm4.c* contains the main function, which is the entry point for execution of the firmware application. The main function sets up user tasks, and then starts the RTOS scheduler.
- *main_cm0p.c* contains functions that start up the BLE controller, start up the CM4, and continuously service BLE stack events.
- *ble_task.c/.h* contain the task and associated functions that handle BLE beacon broadcast.
- *eddystone_config.h* contains the macros that configure Eddystone frame details.
- *status_led_task.c/h* contain the task that controls status LED indications.
- *hibernate_config.h* contains the prototype of Hibernate-related functions and define macros that decide the active time.
- *RTCtimer.c/.h* contains the functions associated with RTC.

See the corresponding header/source files for more details.

Figure 8 shows the RTOS firmware flow of this code example.

Figure 8. RTOS Firmware Flow



Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator project
AN215656 – PSoC 6 MCU: Dual-CPU System Design	Describes the dual-CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-CPU design
AN219434 – Importing PSoC Creator Code into an IDE for a PSoC 6 MCU Project	Describes how to import the code generated by PSoC Creator into your preferred IDE
PSoC Creator Component Datasheets	
Pins	Supports connection of hardware resources to physical pins
Bluetooth Low Energy	Supports BLE connectivity.
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	
Training Videos	
PSoC 6 101: Lesson 1-4 FreeRTOS	

Document History

Document Title: CE224856 – PSoC 6 MCU with BLE Connectivity: Low-Power Beacon with Hibernate (RTOS)

Document Number: 002-24856

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6280078	YEKT	08/22/2018	Initial release

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, 2018. 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.