

USBFS MIDI PSoC3/5LP Code Example

v1.0

Features

- 2-Inputs/2-Outputs USB MIDI Interface
- Bus-powered
- Suspend mode support

General Description

This example project demonstrates the MIDI interface device operation. The project enumerates as a USB Audio Device with the MIDI feature and does not require additional drivers. The main goal of the USB MIDI interface is to transfer and convert MIDI data between external MIDI devices that use the UART interface, and a PC through the USB bus.

Development Kit Configuration

This example project runs on the CY8CKIT-001 kit from Cypress Semiconductor. A description of the kit, along with more code examples and ordering information is at <http://www.cypress.com/go/cy8ckit-001>.

The project requires configuration settings changes to run on other kits from Cypress Semiconductor. Table 1 is the list of the supported kits. To switch from CY8CKIT-001 to any other kit, change the project's device with the help of Device Selector called from the project's context menu.

Table 1. Development Kits vs Parts

Development Kit	Device
CY8CKIT-001	CY8C3866AXI-040/ CY8C5868AXI_LP035
CY8CKIT-030	CY8C3866AXI-040
CY8CKIT-050	CY8C5868AXI_LP035

The pins assignment for the supported kits is in Table 2.

Table 2. Pins Assignment

Pin Name	Development Kit		
	CY8CKIT-001	CY8CKIT-030	CY8CKIT-050
\USBFS:Dm\	P15[7]	P15[7]	P15[7]
\USBFS:Dp\	P15[6]	P15[6]	P15[6]
LED_InA	P1[4]	P6[2]	P6[2]

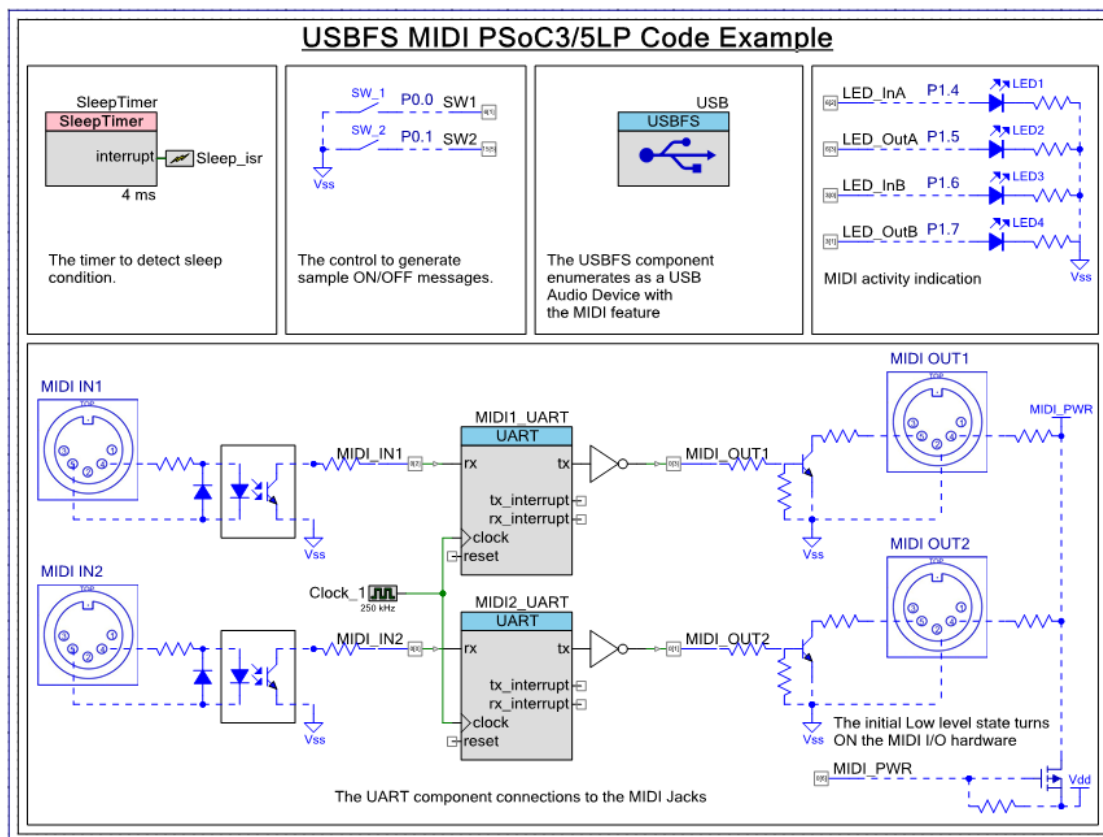
Pin Name	Development Kit		
	CY8CKIT-001	CY8CKIT-030	CY8CKIT-050
LED_InB	P1[6]	P3[0]	P3[0]
LED_OutA	P1[5]	P6[3]	P6[3]
LED_OutB	P1[7]	P3[1]	P3[1]
MIDI_IN1	P5[2]	P0[2]	P0[2]
MIDI_IN2	P5[0]	P0[0]	P0[0]
MIDI_OUT1	P5[3]	P0[3]	P0[3]
MIDI_OUT2	P5[1]	P0[1]	P0[1]
MIDI_PWR	P5[6]	P0[6]	P0[6]
SW1	P0[0]	P6[1]	P6[1]
SW2	P0[1]	P15[5]	P15[5]

Note The project control file handles the pins placement automatically according to a chosen PSoC.

Project Configuration

The example project consists of the USBFS, SleepTimer, UART, and pins components. The project schematic is in [Figure 1](#).

Figure 1. Example Project Design Schematic PSoC3/5LP



The important USBFS component configuration Tabs are in the figures below.

Figure 2. USBFS Descriptor Root

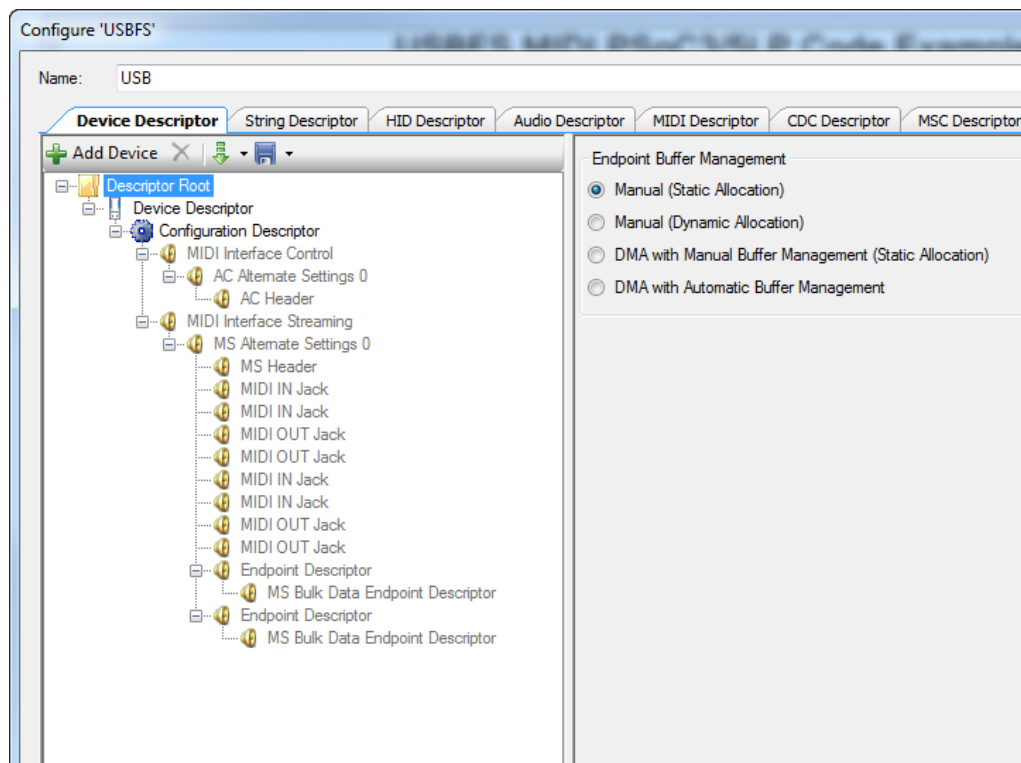


Figure 3. USBFS Device Descriptor

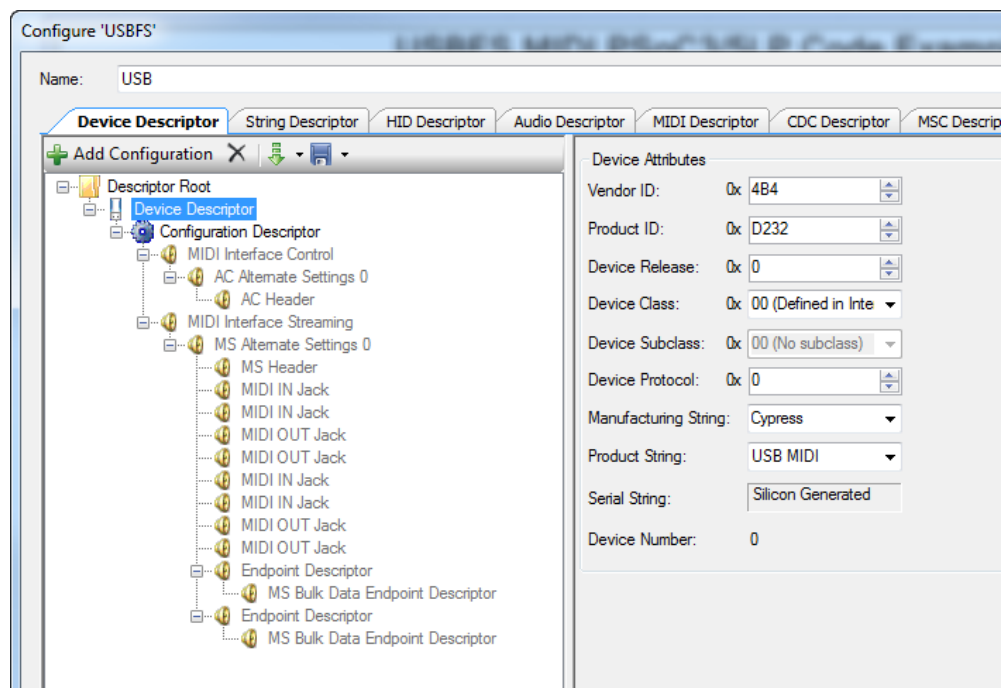


Figure 4. USBFS Configuration Descriptor

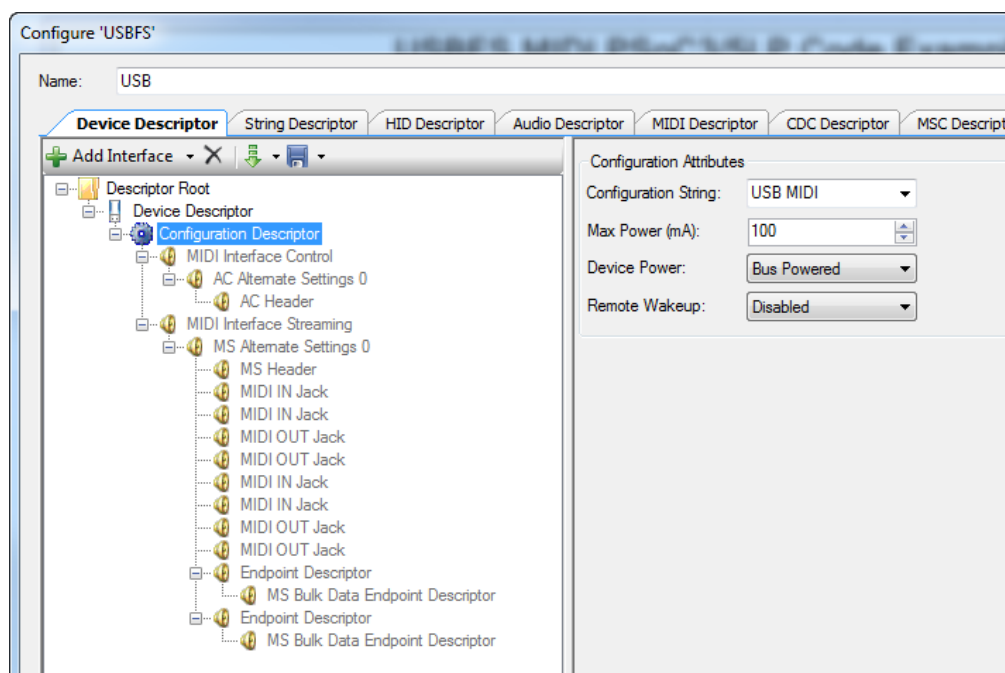


Figure 5. USBFS AC Alternate Settings Descriptor

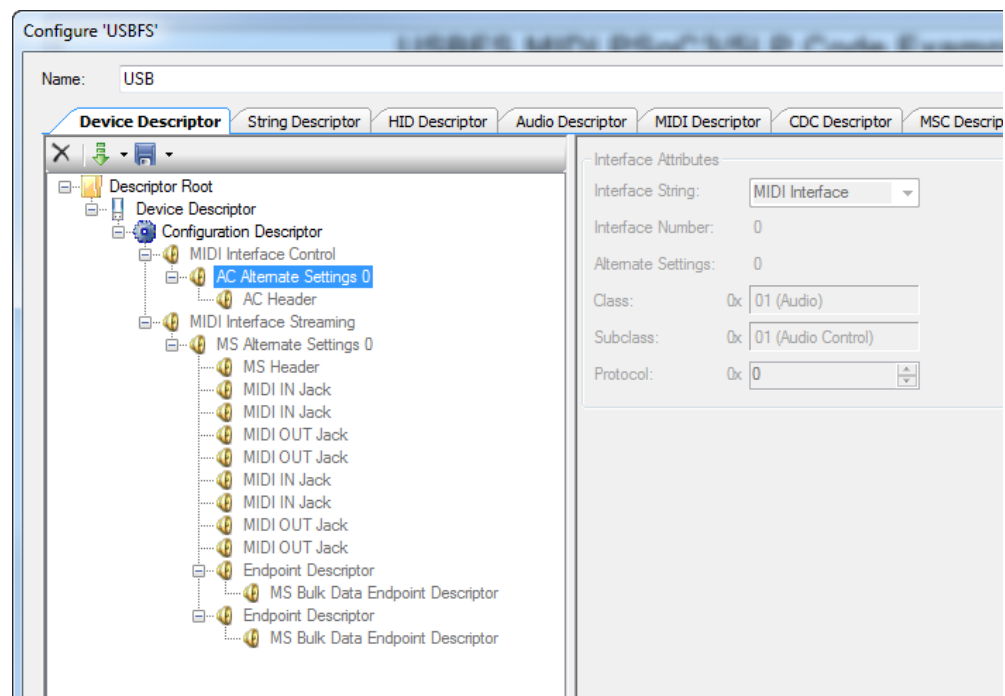


Figure 6. USBFS MS Alternate Settings Descriptor

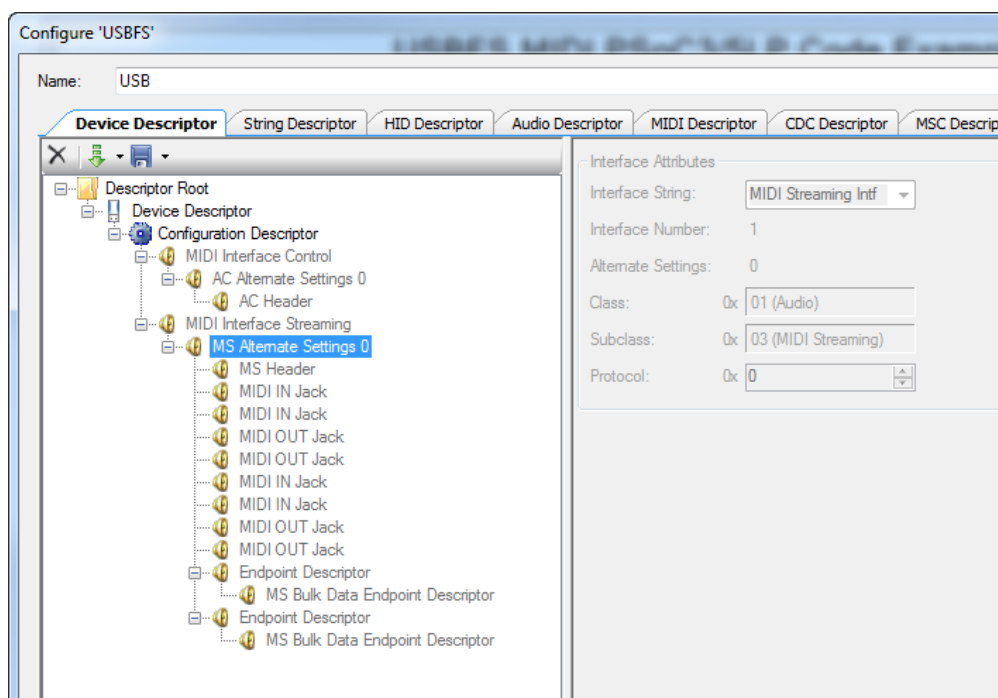


Figure 7. USBFS EP1 Descriptor

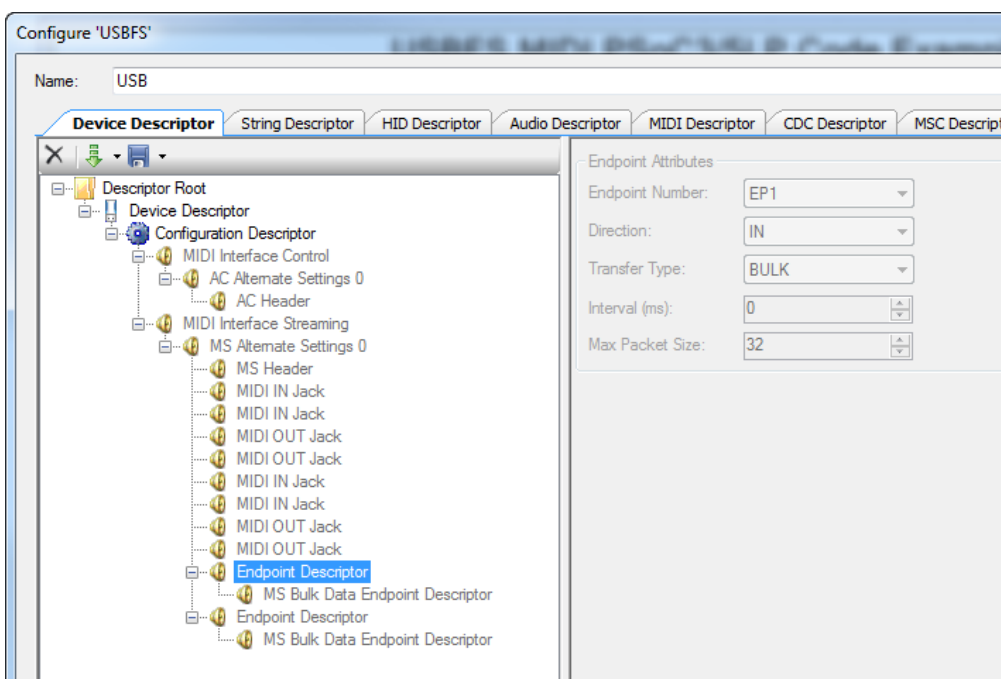
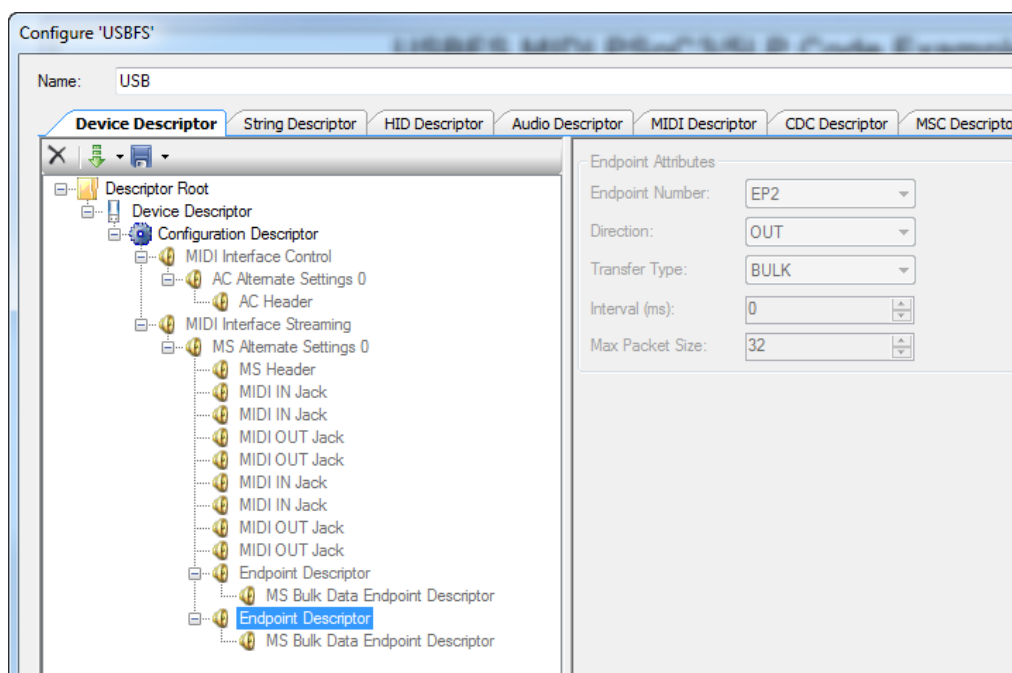


Figure 8. USBFS EP2 Descriptor



Project Description

The main firmware routine configures the USBFS component for operation and starts it. The code waits for the USBFS device enumeration and loops MIDI streams MIDI_IN1 to MIDI_OUT1 and MIDI_IN2 to MIDI_OUT2. The LED indicates data transfers for MIDI_IN/OUT_1/2. The code generates simple notes according to the state of the SW1 and SW2 buttons.

Example Project Execution Flow

To execute the USBFS component code example you need the following equipment:

- USB 3/5LP Kit (CY8CKIT-001/030/050)
- CY8CKIT-044 MIDI I/O EBK extension board
- MIDI Software [MIDI-OX](#) (on Windows), [GarageBand](#) (on Mac OS).

Follow the procedure below:

1. Make sure that PSoC 3/PSoC 5 LP DVK CY8CKIT-001 is in the default configuration.
2. Connect the CY8CKIT-044 MIDI I/O EBK extension board to CY8CKIT-001 PORT A.
3. Connect pins LED_InA, LED_OutA, LED_InB, LED_OutB to the LED1, LED2, LED3, LED4 respectively (refer to the table of pins). These LEDs indicate the MIDI input and output activity.

4. Connect pins SW1, SW2 to the SW1 and SW2 respectively (refer to the table of pins).
5. Set the JP8 jumper to the VBUS position.
6. Connect the computer USB cable to the CY8CKIT-001.
7. Build the project and program the hex file into the target device.
8. Open the Device Manager and observe that the USB Audio Device is working properly.
9. Connect the MIDI output port of your MIDI device (keyboard) to one of the MIDI input ports (MIDI_IN1/MIDI_IN2).
10. Connect one of the MIDI output ports (MIDI_OUT1/MIDI_OUT2) to the MIDI input port of your MIDI device (synthesizer, sound module, etc.)
11. Configure your MIDI capable music software to use the USB Audio Device for incoming and outgoing MIDI messages. Some applications also require selecting a specific MIDI input and output port for each track used within the software
12. (Check your software's documentation for instructions). Software examples: [MIDI-OX](#) (on Windows), [GarageBand](#) (on Mac OS)
13. Press SW1/SW2 to generate sample note ON/OFF messages.

Expected Results

The device continuously transfers data between the USB and both UARTs and indicates the activity on the LEDs. When the computer is in the sleep mode, the device also goes into the sleep mode and reduces the power consumption. The wake-up source is configured to PICU.

Note Some applications may not properly support the PC's Stand-By mode (MIDI-OX). In this case, restart the software after the PC's wake-up to continue operation.

© Cypress Semiconductor Corporation, 2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

