CE222867 – MagSense with Feedback LEDs

Objective

This code example demonstrates inductive sensing for buttons and proximity using CY8CKIT-148 PSoC 4700S Inductive Sensing Evaluation Kit.

Overview

PSoC® 4700S Inductive Sensing Evaluation Kit has three coils with a metal overlay for buttons and one coil for proximity sensing. This code example is designed to detect the button press and the proximity of a metallic object using inductive sensing function of MagSense™ Component and control the LEDs on the kit. When any button is pressed, corresponding LED is turned ON and when the metal target (provided with the kit) is brought near the proximity sensor, the LEDs turn ON depending on how close the metal target is to the sensor. Note that multi-touch is not allowed because of the use of common metal overlay over the three buttons.

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C (Arm® GCC 5.4.1 and Arm MDK 5.22)

Associated Parts: All PSoC 4700S parts

Related Hardware: CY8CKIT-148 PSoC 4700S Inductive Sensing Evaluation Kit

Hardware Setup

This example uses the kit's default configuration. Refer to the kit guide to ensure that the kit is configured correctly. This example requires that the kit is placed on a non-metallic platform.

Software Setup

None.

Operation

1. Connect the kit to your PC using the USB Type-A to Type-C cable provided with the kit.
2. Install the MagSense Component from the PSoC Creator using the instructions provided in the User Guide of the kit.
3. Build the project and program the PSoC 4700S device from PSoC Creator by choosing **Debug** > **Program**. For more information on device programming, see PSoC Creator Help.
4. Press any inductive sensing button (one at a time) and observe that the corresponding LED is turned ON.
5. Bring the metal target from the distance of 2 cm towards the proximity coil. Observe LED 7 to LED 4 turns ON as the metal target approaches the sensor.

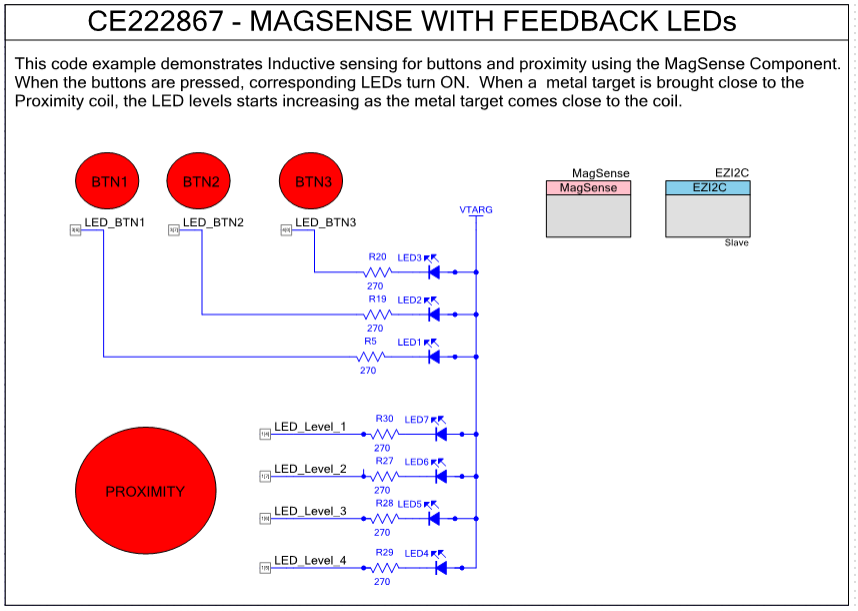
Note: When you press the button or keep the metal target close to the proximity sensor for a longer duration, auto-reset feature of the MagSense Component will automatically reset the button or the proximity sensor along with the LEDs.

Design and Implementation

Figure 1 shows the PSoC Creator schematic for this code example. MagSense Component is configured for three buttons and one proximity sensor. Digital output pins are placed to drive the LEDs.

To monitor the sensor data, MagSense Tuner feature is used which requires the EZI2C Slave Component. To understand the detailed procedure on using the Tuner and selecting values for Thresholds and Baselines, refer to the Inductive Sensing Design Guide.

Figure 1. Top Design Schematic



To tune the buttons, it is necessary to first measure the inductance and the resistance of the coil alone and then with an aluminum overlay. Similarly, for the proximity sensor, measure the inductance and the resistance of the coil alone and then with the metal target at 2 cm spacing. Table 1 shows the measurement of inductance and resistance of the coil using an LCR meter along with the external capacitor and the resistor on the kit. Based on the measured values, the resonant frequency for buttons are calculated to be 959 kHz and for the proximity sensor, it is calculated to be 793 kHz.

Note: To measure the values of L and R of the coil, make sure to isolate it from other parts of the kit by desoldering the series and parallel components of the coil.

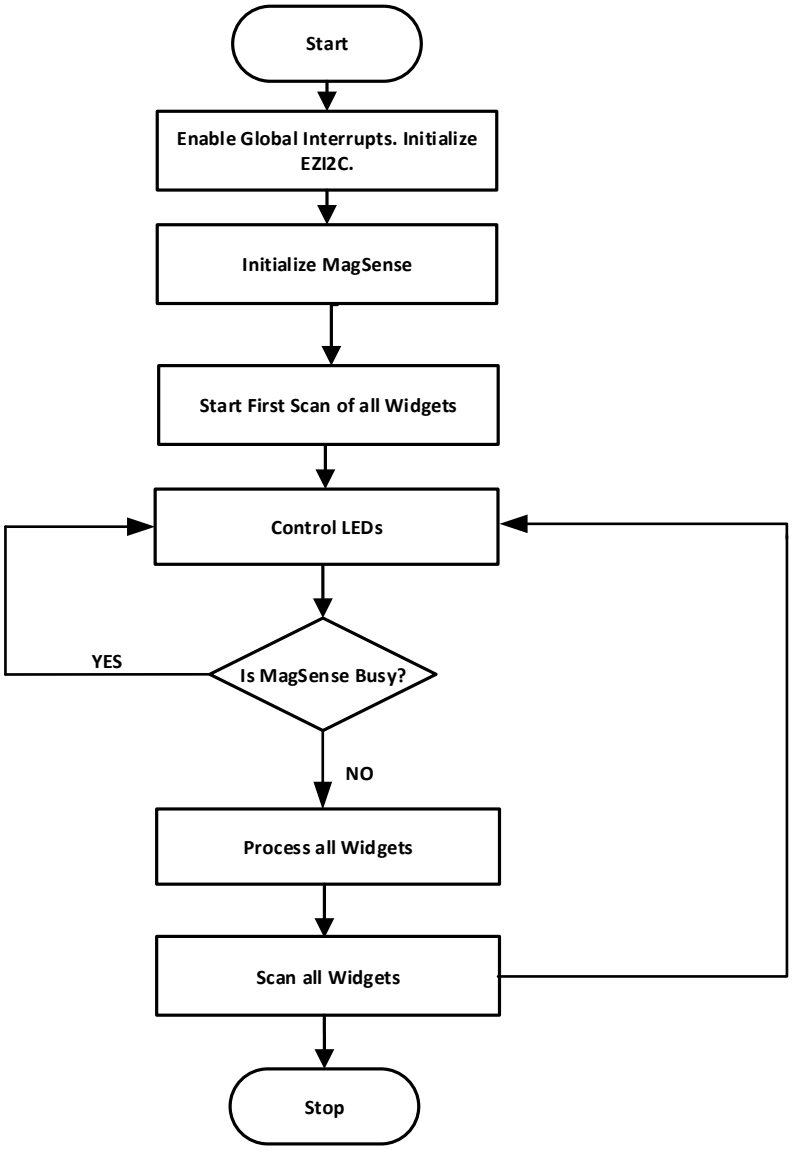
Table 1. Measurement of Lp and Rs using an LCR meter and values for External Components

	Feature	Lp (Inductance)	Rs (Resistance)	External Parallel Capacitor	External Series Resistor
Without Aluminum Target	BTN1	14.1uH	7 Ohm	C41: 5nF	R32: 330 Ohm
	BTN2	14.1uH	6.78 Ohm	C46: 5nF	R39: 330 Ohm
	BTN3	14.13uH	6.9 Ohm	C51: 5nF	R49: 330 Ohm
	PROX	55.8uH	16.9 Ohm	C42: 2.2nF	R34: 330 Ohm
With Aluminum Target	BTN1	5.5uH	8.9 Ohm	C41: 5nF	R32: 330 Ohm
	BTN2	5.42uH	8.5 Ohm	C46: 5nF	R39: 330 Ohm
	BTN3	5.51uH	8.73 Ohm	C51: 5nF	R49: 330 Ohm
	PROX	18.3uH	20 Ohm	C42: 2.2nF	R34: 330 Ohm

See Figure 3 to Figure 8 for the tuned values of the kit. The tuning procedure is discussed in detail in the Inductive Sensing Design Guide. Once tuning has been completed, sensitivity can be adjusted by reducing the excitation frequency below the resonant frequency by about 5% to 10% depending on the application requirements.

The *main.c* file of the project contains the scanning algorithm described in Figure 2

Figure 2. Flowchart of main.c



Components and Settings

Table 2 lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended.

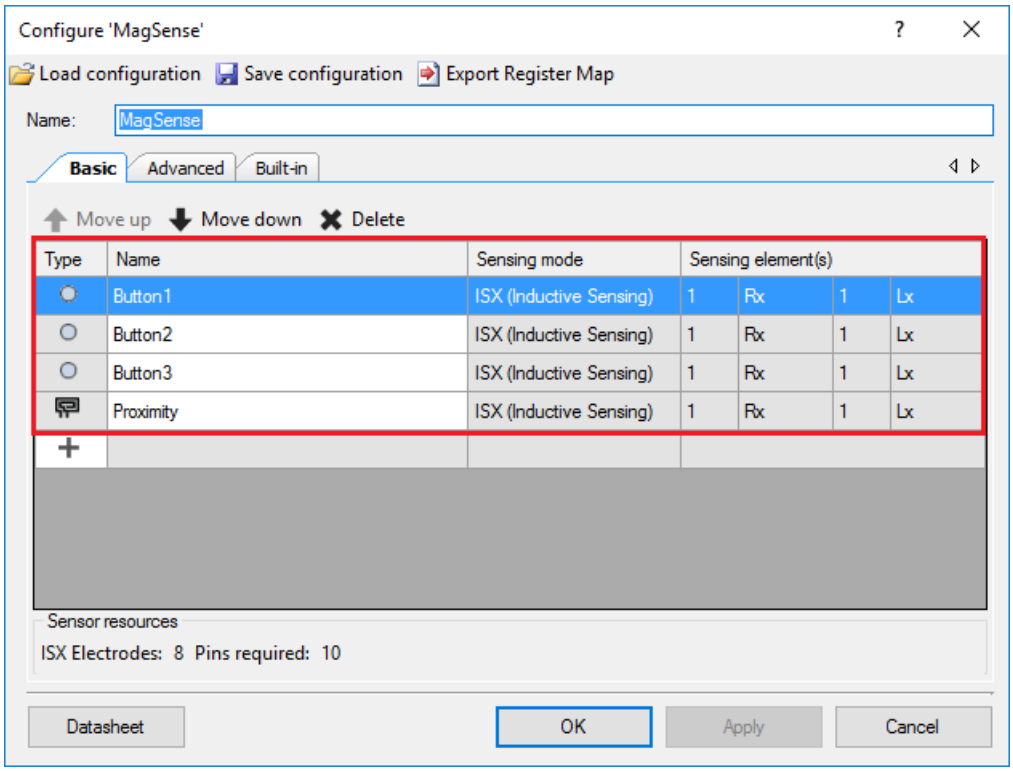
Table 2. PSoC Creator Components

Component	Instance Name	Purpose	Non-default Settings
MagSense (6.0)	MagSense	Inductive Sensor configuration and processing	Figure 3 to Figure 8
PSoC 4 Serial Communication Block (SCB) (4.0)	EZI2C	Communication with external interfaces or with Tuner	Figure 9.
Digital Output Pin (2.20)	LED_BTN1	Provide visual feedback for BTN1 press	Figure 10.
Digital Output Pin (2.20)	LED_BTN2	Provide visual feedback for BTN2 press	
Digital Output Pin (2.20)	LED_BTN3	Provide visual feedback for BTN3 press	
Digital Output Pin (2.20)	LED_Level_1	Provide visual feedback for Proximity	
Digital Output Pin (2.20)	LED_Level_2	Provide visual feedback for Proximity	
Digital Output Pin (2.20)	LED_Level_3	Provide visual feedback for Proximity	
Digital Output Pin (2.20)	LED_Level_4	Provide visual feedback for Proximity	

For information on the hardware resources used by a Component, see the Component datasheet.

Figure 3 shows the configuration for the MagSense Component.

Figure 3. MagSense Component Configuration Basic Tab



Enabling filters are necessary to eliminate noise. Figure 4 shows the settings for filter assignments on the component.

Auto-reset feature is enabled to prevent latch-up of sensors which can happen due to following reasons:

1. Touching the tank circuit
2. Touching the coils
3. Keeping the device on a metallic platform

The auto-reset feature will reset all sensors after about 2 seconds.

Figure 4. MagSense Component Configuration Advance Tab - General Settings

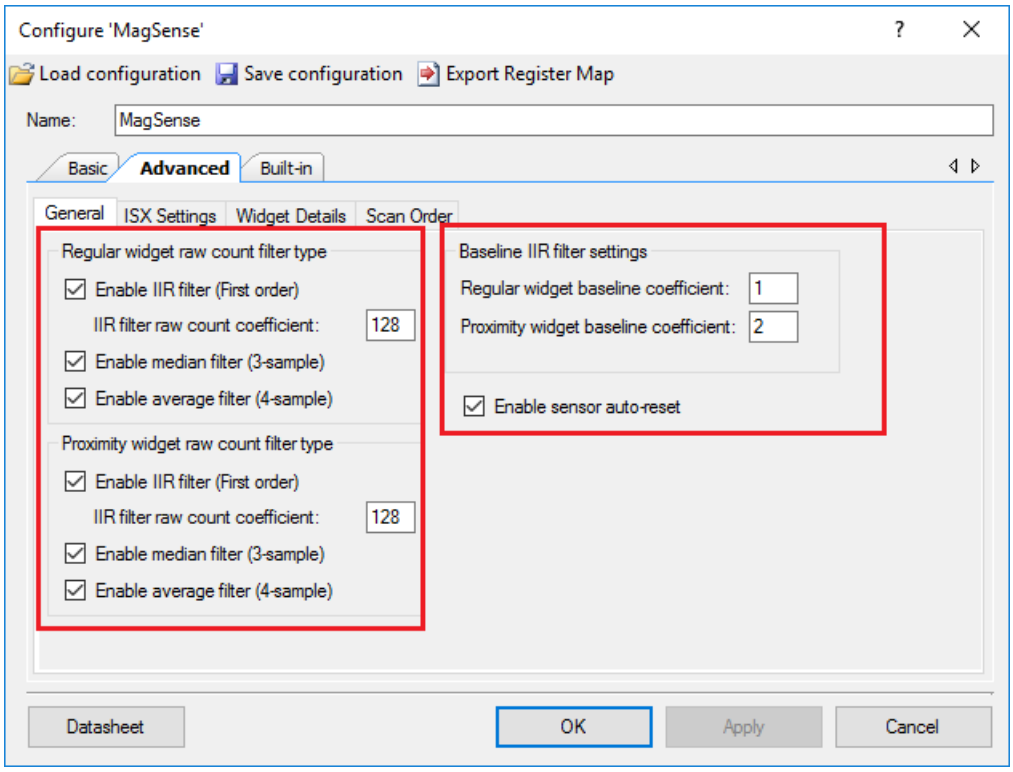


Figure 5 to Figure 8 shows the tuning configuration parameters for the buttons and proximity.

Figure 5. MagSense Component Configuration - Button 1

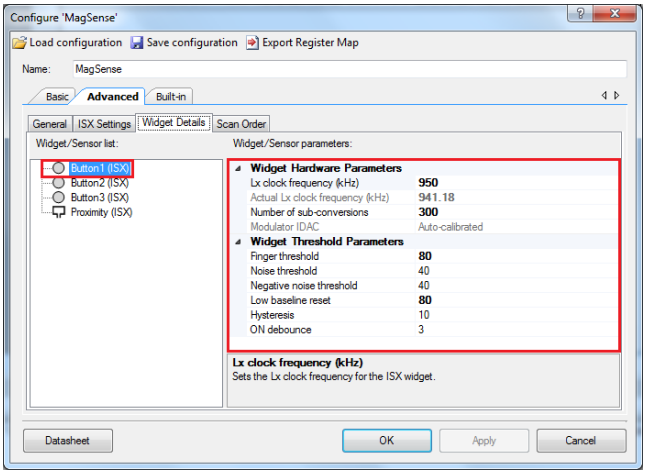


Figure 6. MagSense Component Configuration - Button 2

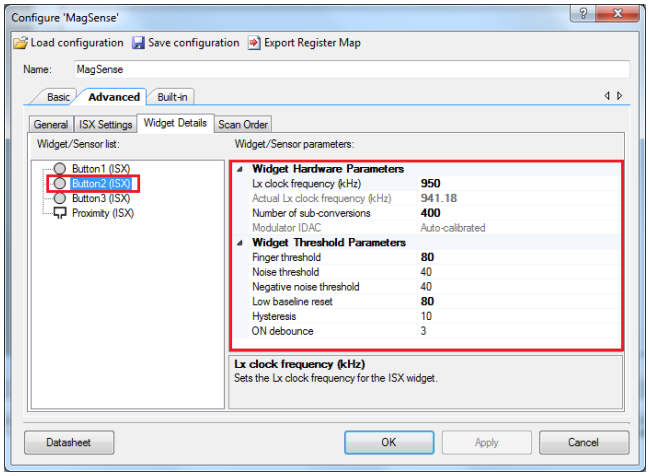


Figure 7. MagSense Component Configuration - Button 3

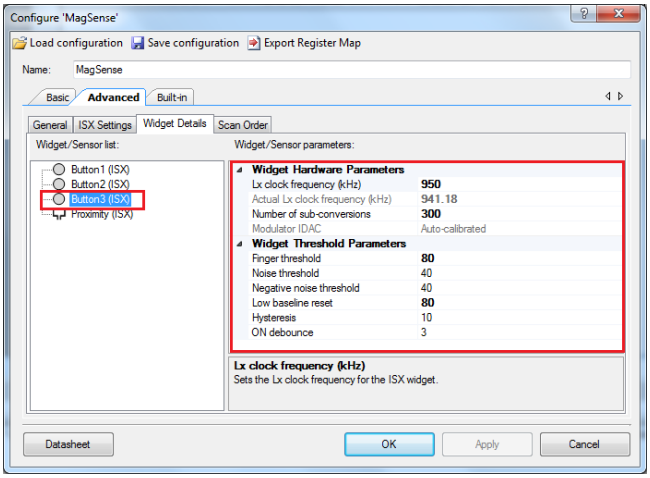


Figure 8. MagSense Component Configuration - Proximity

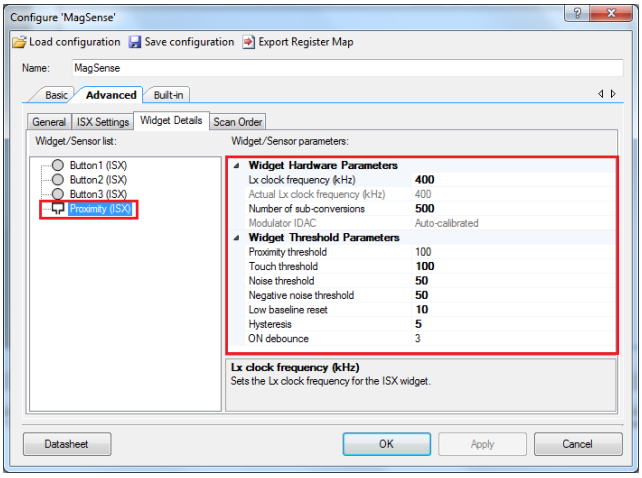


Figure 9 shows the settings for the SCB configured as EZI2C.

Figure 9. EZI2C Setting

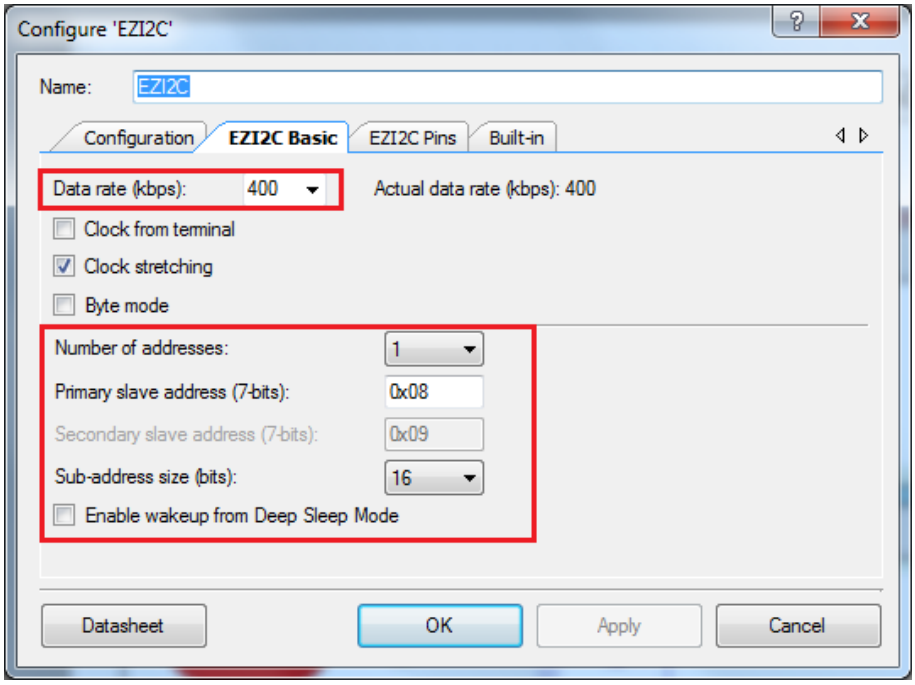


Figure 10 shows the settings for the Digital Output Pin components in PSoC Creator. These pins are used to drive the seven feedback LEDs. The settings are the same for all the Digital output pins in this example.

Figure 10. Digital Output Pin Setting

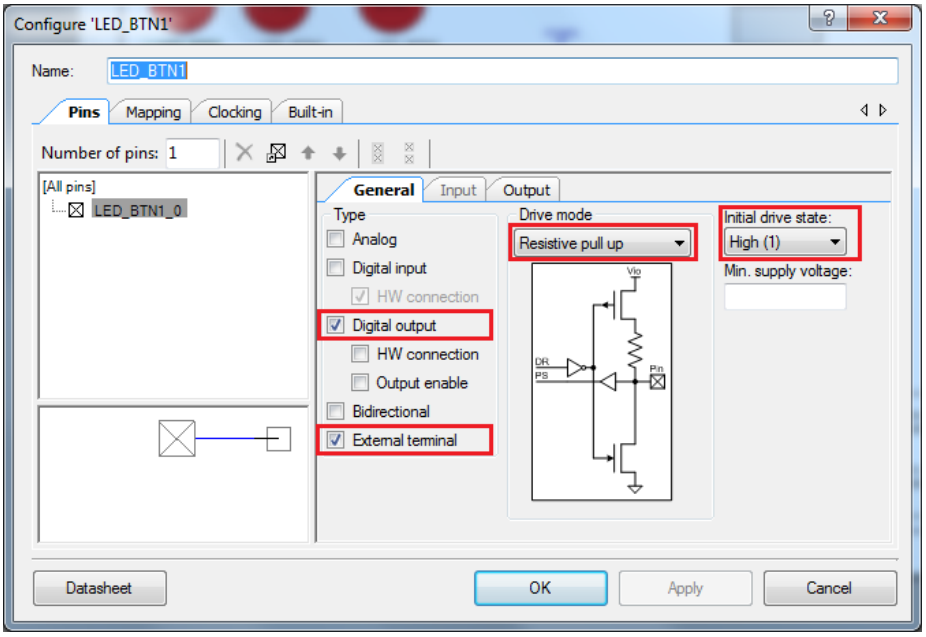


Figure 11 shows the *.cydwr* file (i.e. the pin assignments) of the code example.

Figure 11. Pin Assignment

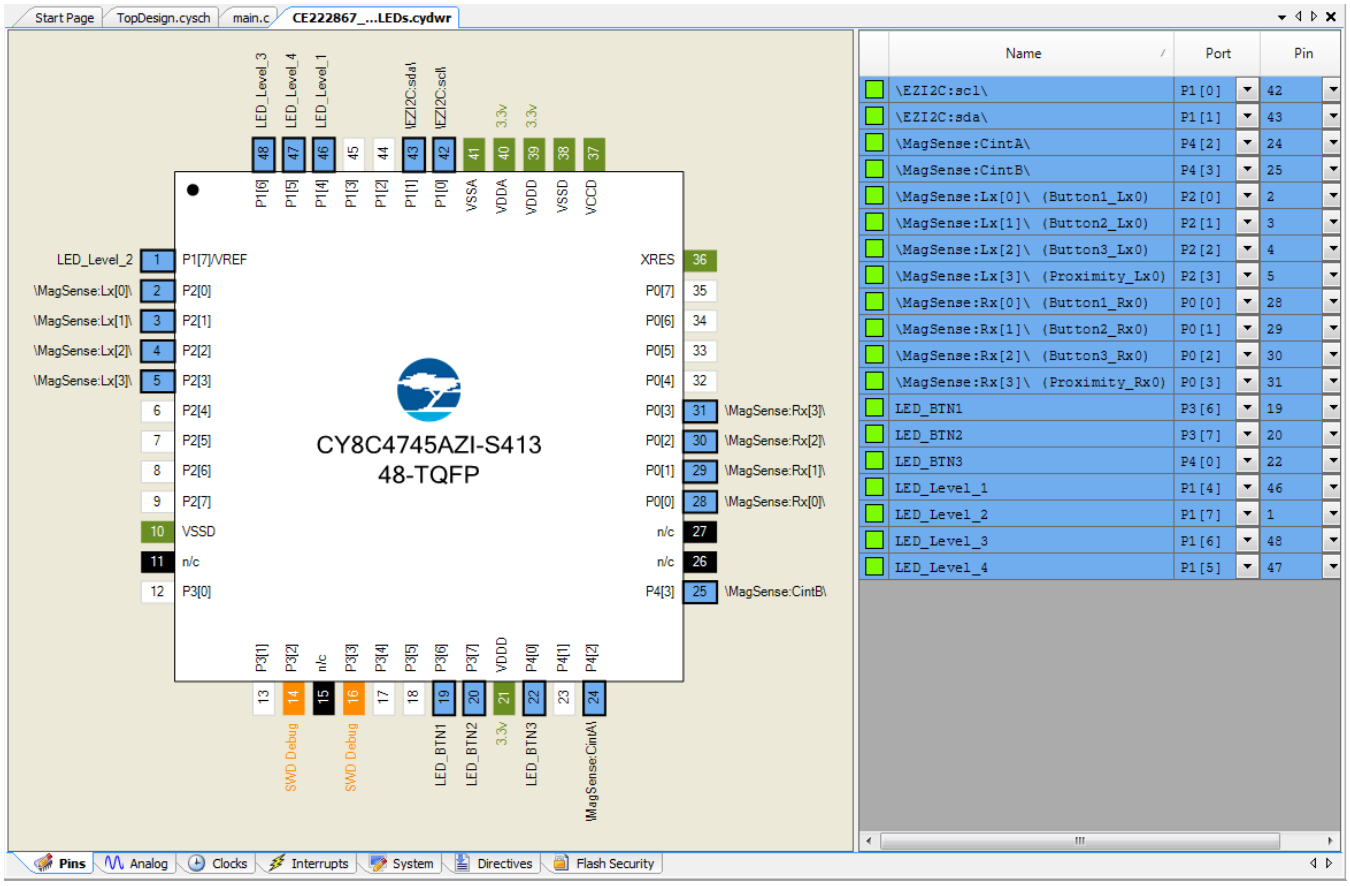


Figure 12 shows the clock settings of the code example.

Figure 12. Clock Settings in PSoC Creator

Type	Name	Domain	Desired Frequency	Nominal Frequency	Accuracy (%)	Tolerance (%)	Divider	Start on Reset	Source Clock
System	ExtClk	N/A	16 MHz	? MHz	±0	–	0	☐	
System	WCO	N/A	32.768 kHz	? MHz	±0.015	–	0	☐	
System	Timer (WDT)	N/A	? MHz	? MHz	±0	–	32768	☐	LFClk
System	ILO	N/A	40 kHz	40 kHz	-50, +100	–	0	☑	
System	LFClk	N/A	? MHz	40 kHz	-50, +100	–	0	☑	ILO
System	Timer_Sel	N/A	40 kHz	40 kHz	-50, +100	–	0	☑	ILO
System	HFClk	N/A	48 MHz	48 MHz	±2	–	1	☑	IMO
System	IMO	N/A	48 MHz	48 MHz	±2	–	0	☑	
System	SysClk	N/A	? MHz	48 MHz	±2	–	1	☑	HFClk
Local	MagSense_ModClk	UNKNOWN	? MHz	188.235 kHz	±2	–	255	☑	HFClk
Local	EZI2C_SCBCLK	UNKNOWN	7.82 MHz	8 MHz	±2	-0, +96.675	6	☑	Auto: HFClk

Reusing This Example

This example is designed for the CY8CKIT-148 PSoC 4700S Inductive Sensing Evaluation Kit. To port this design to a different PSoC 4700S MCU device and/or kit, change the target device using the Device Selector and update the pin assignments in the Design Wide Resources Pins settings as needed. Additional tuning is required for the code to be reused with other boards.

In some cases, a resource used by the code example (for example, an IP block) is not supported on another device apart from 4700S device. In that case, the example will not work. If you build the example targeted at such a device, you will get errors. See the device datasheet for information on what that device supports.

Related Documents

Application Notes	
AN219207 – Inductive Sensing Design Guide	Describes designing methodologies for Inductive Sensing solution with PSoC Devices
Code Examples	
CE209974 - Breathing LED with Smart I/O™	Breathing LED using Smart IO component for various kits
CE209975 - Clock Buffer with Smart I/O™	Uses the Smart IO to implement a clock buffer that can operate in chip Deep Sleep mode
CE209976 - SPI Slave Select Inversion with Smart I/O™	Inverts the polarity of the SCB SPI slave select signal by using the Smart IO component
PSoC Creator Component Datasheets	
MagSense	Supports MagSense component
Smart IO	Supports Smart IO peripheral
TCPWM	Supports PWM, Timer/Counter, and QuadDec modes
Pins	Supports the connection of hardware resources to physical pins
Device Documentation	
PSoC 4 Datasheets	PSoC 4 Technical Reference Manuals
Development Kit Documentation	
CY8CKIT-148 PSoC 4700S Inductive Sensing Evaluation Kit	
Tool Documentation	
PSoC Creator	Look in the downloads tab for Quick Start and User Guides

Document History

Document Title: CE222867 – MagSense with Feedback LEDs

Document Number: 002-22867

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6192165	NMIT	05/15/2018	New code example.
*A	6386292	RJVB	11/16/2018	Updated MagSense component configuration
*B	6455340	RJVB	01/22/2019	Added note in the Overview section about multi-touch

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 | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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

© Cypress Semiconductor Corporation, 2018-2019. 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.