CE223818 - PSoC Analog Coprocessor Switched Capacitor Filter

Objective

This example demonstrates the use of the Switched Capacitor Filter Component in band-pass and low-pass configurations in PSoC® Analog Coprocessor. Three projects show how the Component filters square wave edges, isolates single frequencies from superimposed signals, and acts as a mixer.

Overview

This example contains three projects that demonstrate use cases for the Switched Capacitor Filter Component. The Low_Pass_Square_Wave example uses the Component as a low-pass filter to remove the high frequency components of a square wave. The Band_Pass_Isolation example demonstrates the Component as a band-pass filter to select a single frequency from two superimposed signals. The FSK_Mixer example uses the Component as a mixer and band-pass filter to shift a frequency shift keyed (FSK) signal to a lower frequency range and isolate the output frequencies.

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C (Arm® GCC 5.4-2016-q2-update)

Associated Parts: PSoC Analog Coprocessor

Related Hardware: CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit

Hardware Setup

This example requires the use of an oscilloscope with two or more channels.

For the Low_Pass_Square_Wave and Band_Pass_Isolation projects:

1. Connect P0[4] (PWM output) to P0[5] (filter input).
2. Connect one oscilloscope channel to P1[2] (filter output).
3. Connect a second oscilloscope channel to P0[5].

For the FSK_Mixer project:

1. Connect P0[4] (PWM output) to P1[3] (filter input).
2. Connect one oscilloscope channel to P1[2] (filter output).
3. Connect another oscilloscope channel to P0[5] (UART output).

Operation

The same instructions apply to each of the three examples.

1. Ensure the device has been configured according to Hardware Setup.
2. Connect the CY8CKIT-048 Kit to a USB port on your PC.
3. Select one of the three projects to program into the device. Right-click the project in the Workplace Explorer and select **Set As Active Project**
4. Build and program the project into the CY8CKIT-048 Kit. For more information on building a project or programming a device, see *PSoC Creator Help*.
5. Observe the output signals on the oscilloscope.

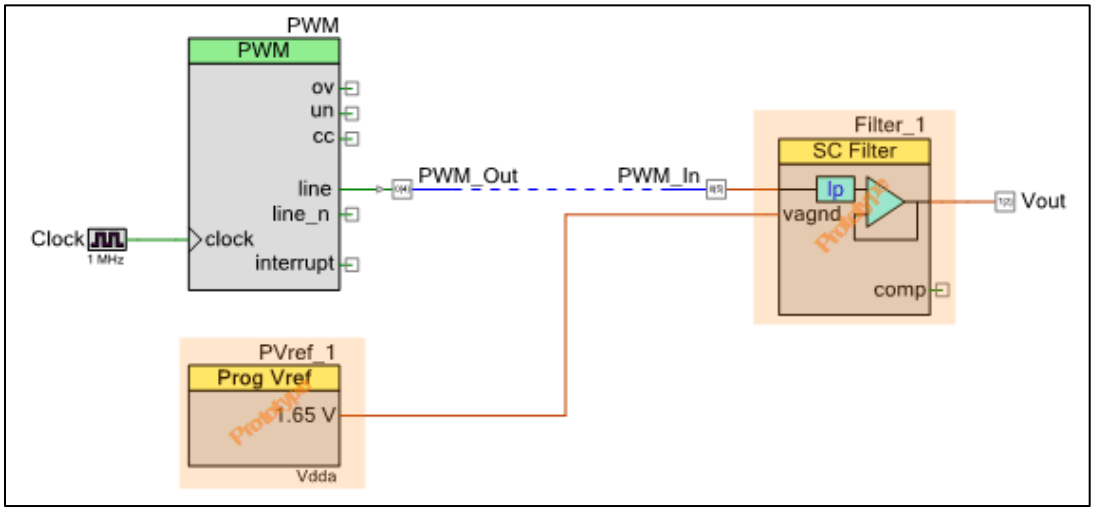
Design and Implementation

This example has three projects. The design and implementation for each project is listed under the heading corresponding to the project.

Low_Pass_Square_Wave

The Low_Pass_Square_Wave design shown in Figure 1 has a Timer Counter Pulse Width Modulator (TCPWM) Component configured as a Pulse Width Modulator (PWM). This component generates a 50% duty cycle square wave with a frequency of 10 kHz as the input signal to the filter Component. The Switched-Cap Filter Component is configured as a low-pass filter with a corner frequency at 10 kHz. A programmable voltage reference (PVref) Component provides the DC bias for the filter at half of the supply voltage. This enables full symmetrical swing of the filtered signal, Vout.

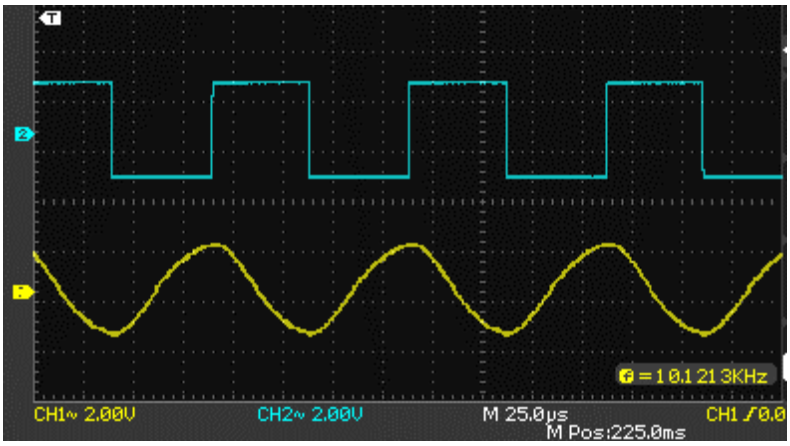
Figure 1. The Low_Pass_Square_Wave Example Schematic



Although the input signal is a square wave at 10 kHz, the output frequency appears as a nearly sinusoidal signal at 10 kHz. The filter Component removes the high-frequency edges of the square wave, resulting in a rounder signal on the output. The output wave form still contains harmonic distortion from the lower-frequency components of the square wave.

A high-Q band-pass filter centered at 10 kHz can reject frequencies outside of the 10-kHz fundamental signal. When a 10-kHz square wave is passed through such a filter, the resulting waveform is largely harmonic-free. To demonstrate this, this project can be reconfigured such that the filter Component is configured as a band-pass filter with a bandwidth of 1 kHz and a center frequency at 10 kHz.

Figure 2. Square Wave Low-pass Filter Input (Blue) and Output (Yellow)



Band_Pass_Isolation

Figure 2 shows the Band_Pass_Isolation design and Figure 3 shows the output. This example superimposes two signals with different frequencies and uses the Switched-Cap Filter Component to isolate the lower of the two frequencies. PWM_1 generates a 10-kHz square wave and PWM_2 generates a 167-kHz square wave. These two signals are XORed in the Smart I/O Component. The output of the Smart I/O Component is a signal that contains both 10 kHz and 167 kHz. The Switched-Cap Filter Component is configured as a bandpass filter centered at 10 kHz with a bandwidth of 1 kHz. The output signal appears as a sinusoidal signal at 10 kHz.

Figure 2. The Band_Pass_Isolation Example Schematic

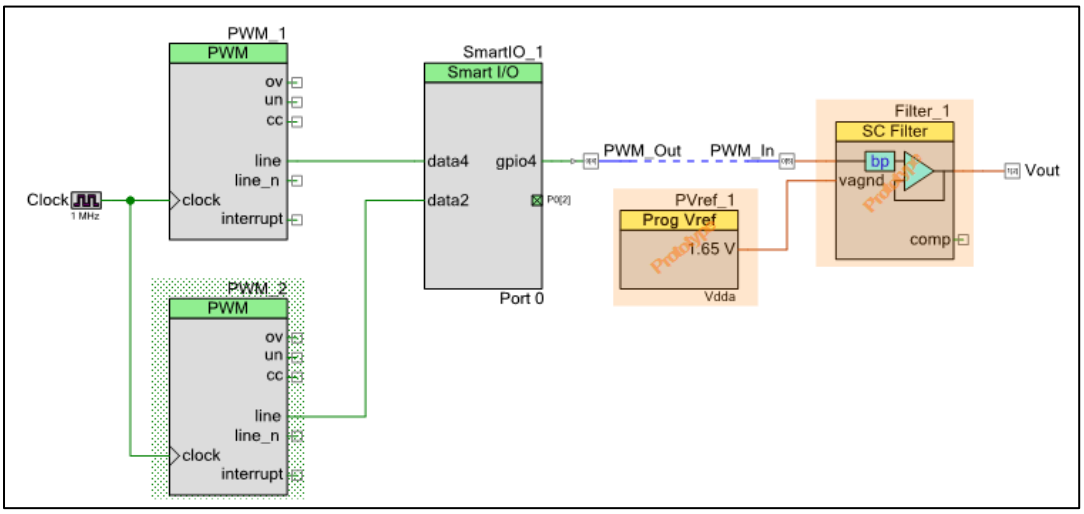
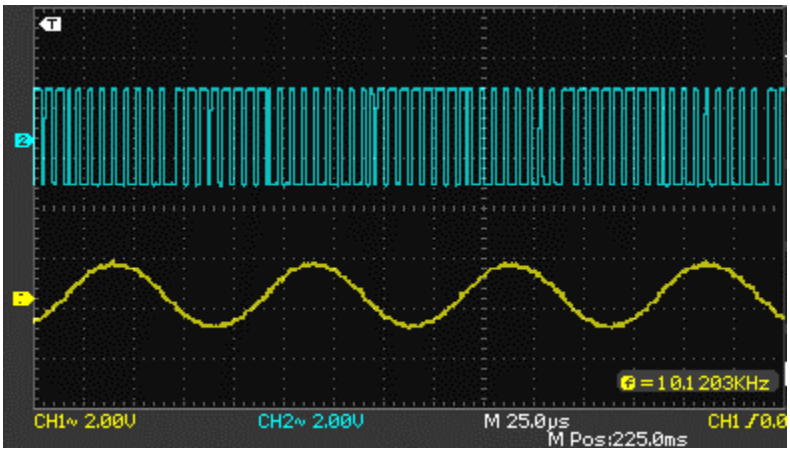


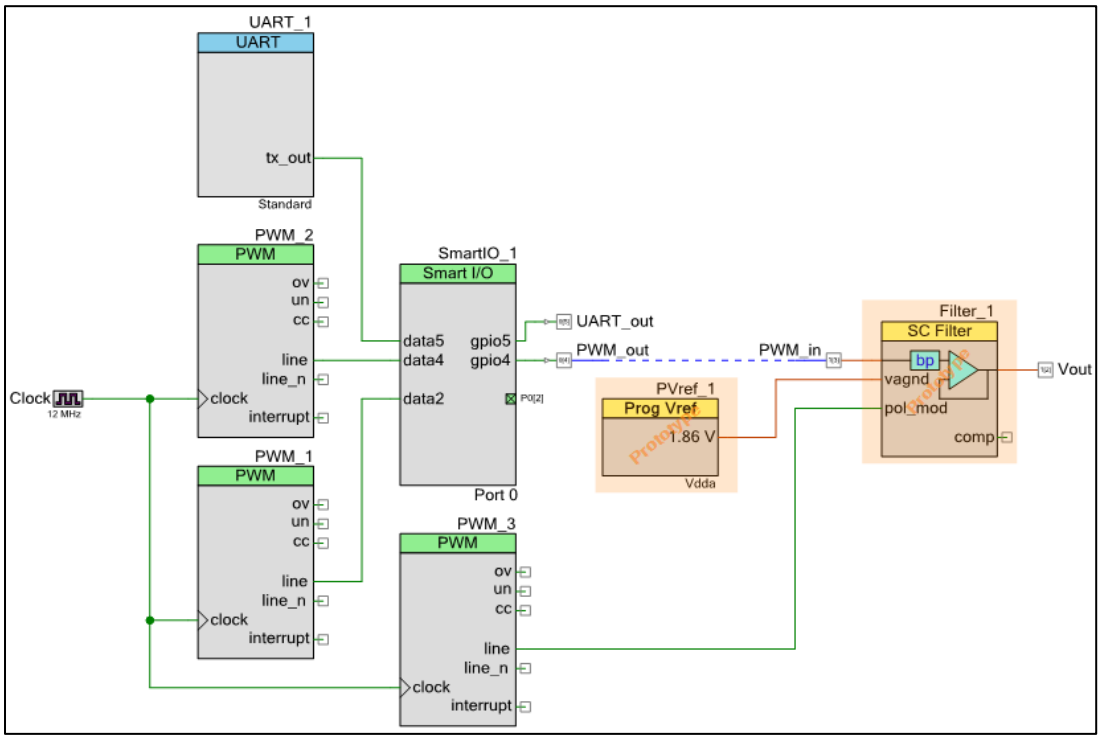
Figure 3. Band_Pass_Isolation Input (Blue) and Output (Yellow)



FSK_Mixer

The FSK_Mixer design shown in Figure 4 generates an FSK signal that switches between two frequencies based on a UART transmission.

Figure 4. FSK_Mixer Example Schematic



PWM_1 and PWM_3 provide 40-kHz and 42.1-kHz signals respectively. The Smart I/O Component is configured to act as a multiplexer with the UART transmission as the select. The filter Component is configured as a band-pass filter with a center frequency at 3 kHz and a bandwidth of 2.1 kHz. The Pol/Mod bit is toggled with a 38-kHz signal from PWM_2, which mixes with the FSK signal. This mixing generates the sum and difference frequencies of the FSK and the mod bit toggle frequency.

$$Sum = f_{fsk} + f_{mod} = (40\ or\ 42.1\ kHz) + 38\ kHz = 78\ or\ 80.1\ kHz \quad \text{Equation 1}$$

$$Difference = f_{fsk} - f_{mod} = (40\ or\ 42.1kHz) - 38\ kHz = 2\ or\ 4.1\ kHz$$

By centering the band-pass filter half-way between the difference frequencies, the sum frequencies from the output signal are removed. By ensuring that the bandwidth of the filter is wide enough, both the 2 kHz and 4.1 kHz signals are preserved. Figure 5 shows the signals as they travel through each stage in the system. Figure 6 shows the output signal.

Figure 5. Block Diagram Showing the Signal Flow in the FSK_Mixer Project

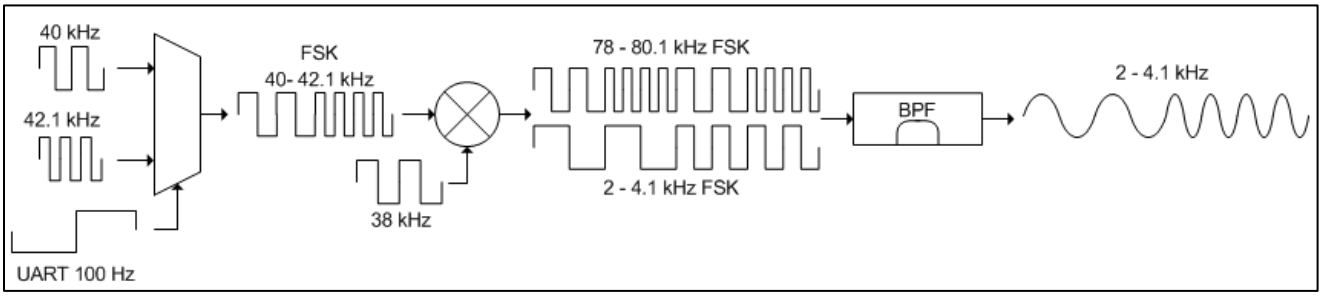
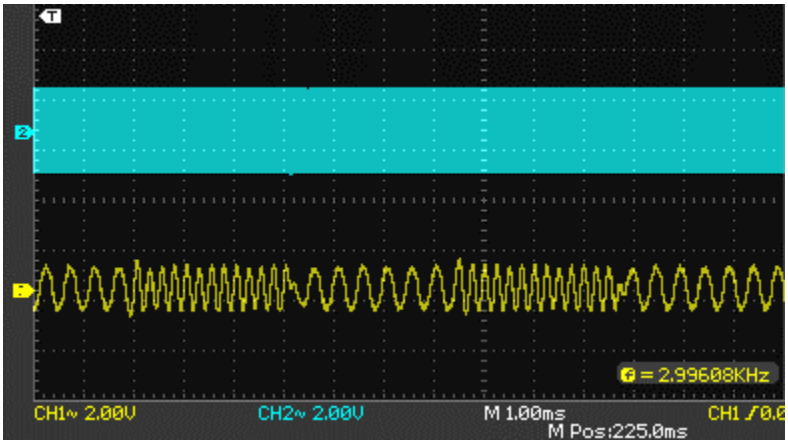


Figure 6. FSK_Mixer Input (Blue) and Output (Yellow)



Components and Settings

Table 1 lists all PSoC Creator Components used in the three examples.

Table 1. PSoC Creator Components

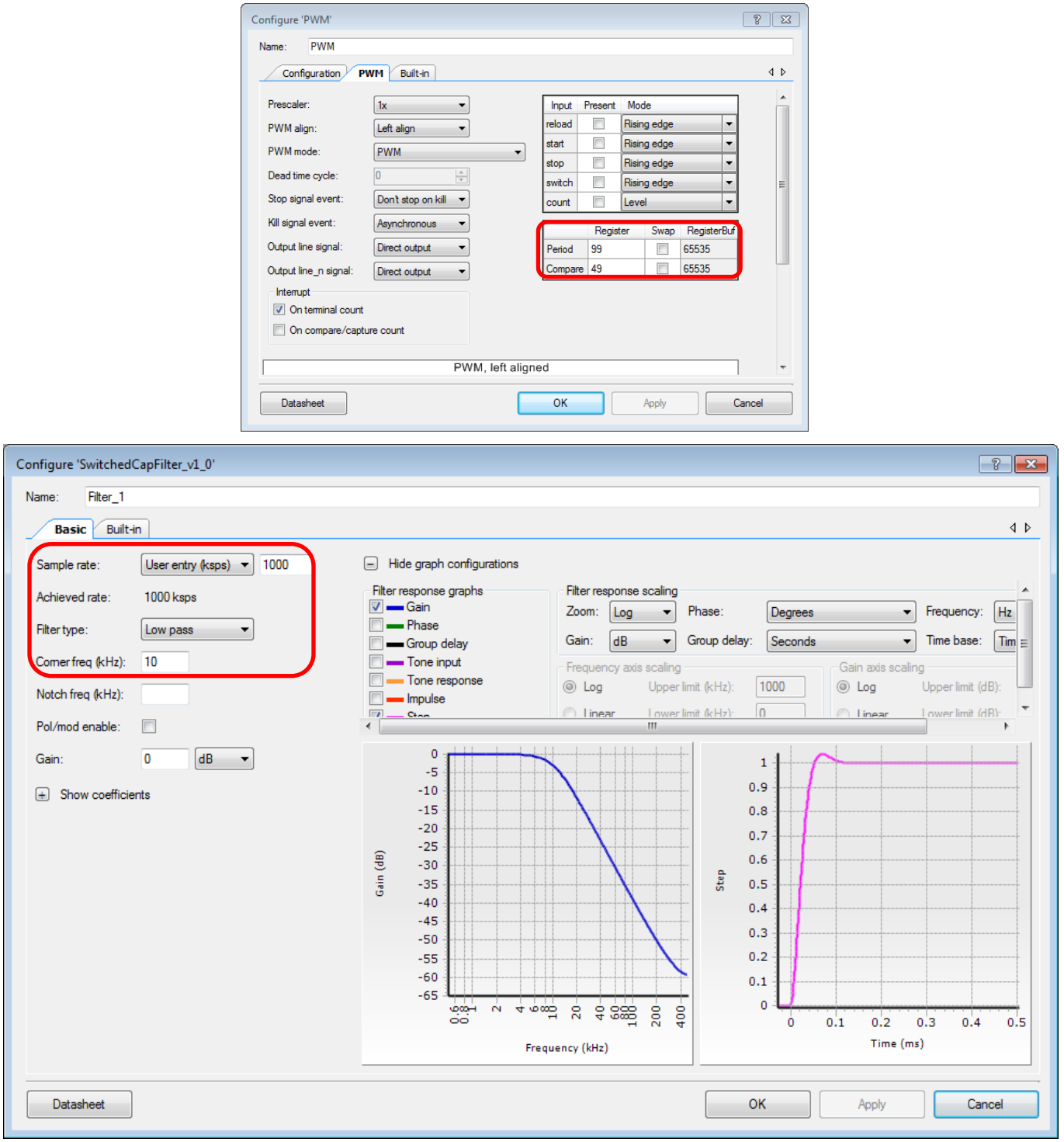
Component	Instance Name	Purpose	Non-Default Settings
Clock	Clock	Drive TCPWM	Frequency: 1 MHz (Low_Pass_Square_Wave and Band_Pass_Isolation examples)
PWM (TCPWM mode)	PWM, PWM_1, PWM_2, PWM_3	Provides signal to be filtered	See Figure 7 for Low_Pass_Square_Wave settings See Figure 8 for Band_Pass_Isolation settings See Figure 12 for FSK_Mixer settings
PVref	PVref_1	Provides an analog reference for the filter	Reference source > V_{DDA} (V): 3.30 Voltage Reference (V): 1.65 (tap 8)
Switched-Cap Filter	Filter_1	Filters analog signals	See Figure 7 for Low_Pass_Square_Wave settings See Figure 11 for Band_Pass_Isolation settings See Figure 13 for FSK_Mixer settings
SmartIO	SmartIO_1	Performs logical operations on PWM signals	See Figure 9 and Figure 10 for Band_Pass_Isolation settings See Figure 14 and Figure 15 for FSK_Mixer settings
UART (SCB mode)	UART_1	Provides multiplexing select data in FSK_Mixer example	UART Basic > Direction: TX only UART Basic > Baud rate (bps) > 300 UART Pins > Show UART terminals

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

Non-default settings for each Component are highlighted in red in Figure 7 to Figure 15.

Low_Pass_Square_Wave Non-Default Settings

Figure 7. PWM and Switched-Cap Filter Settings



Band_Pass_Isolation Non-Default Settings

Figure 8. PWM Settings

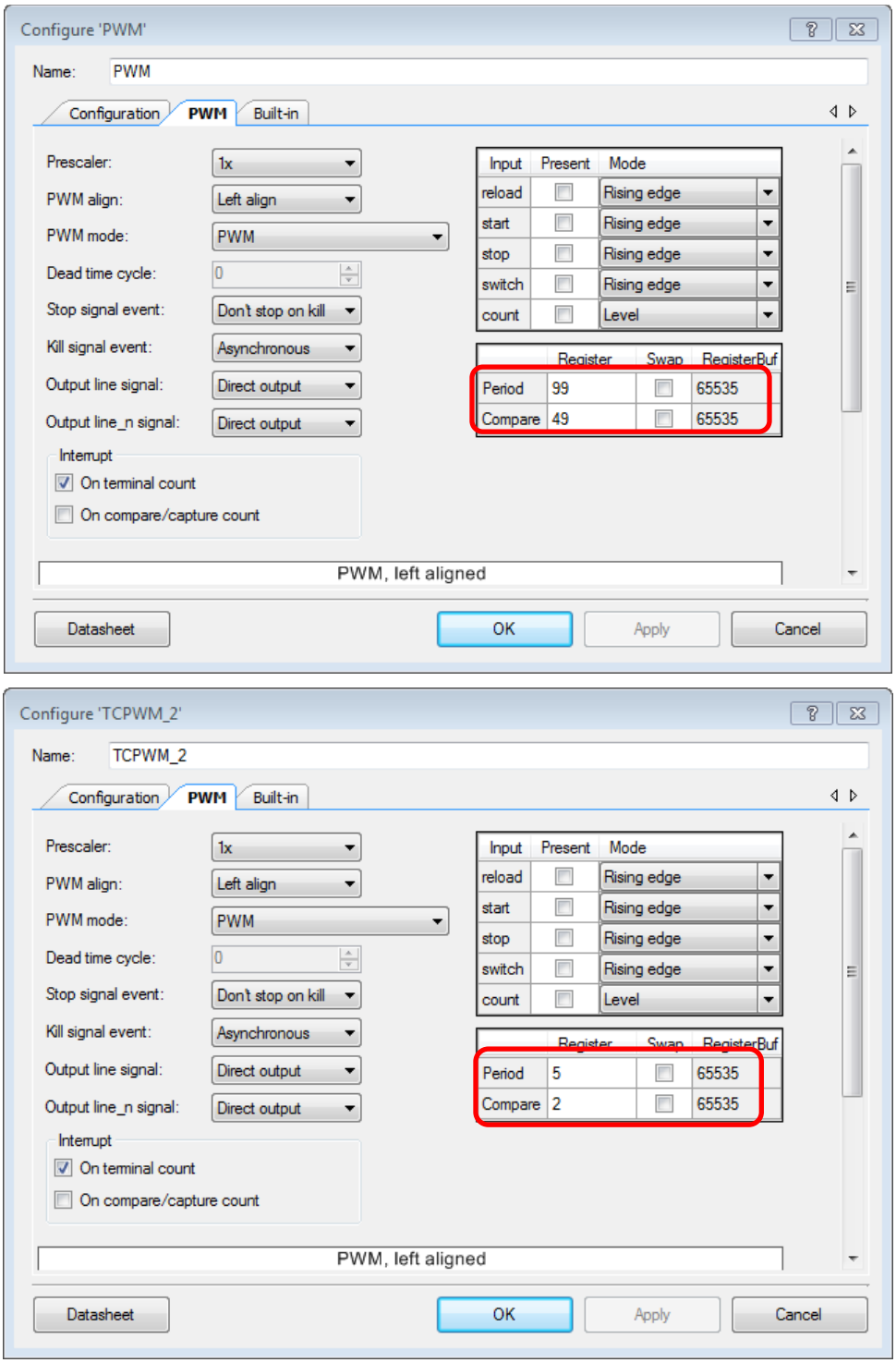


Figure 9. Smart I/O Routing Matrix Configuration

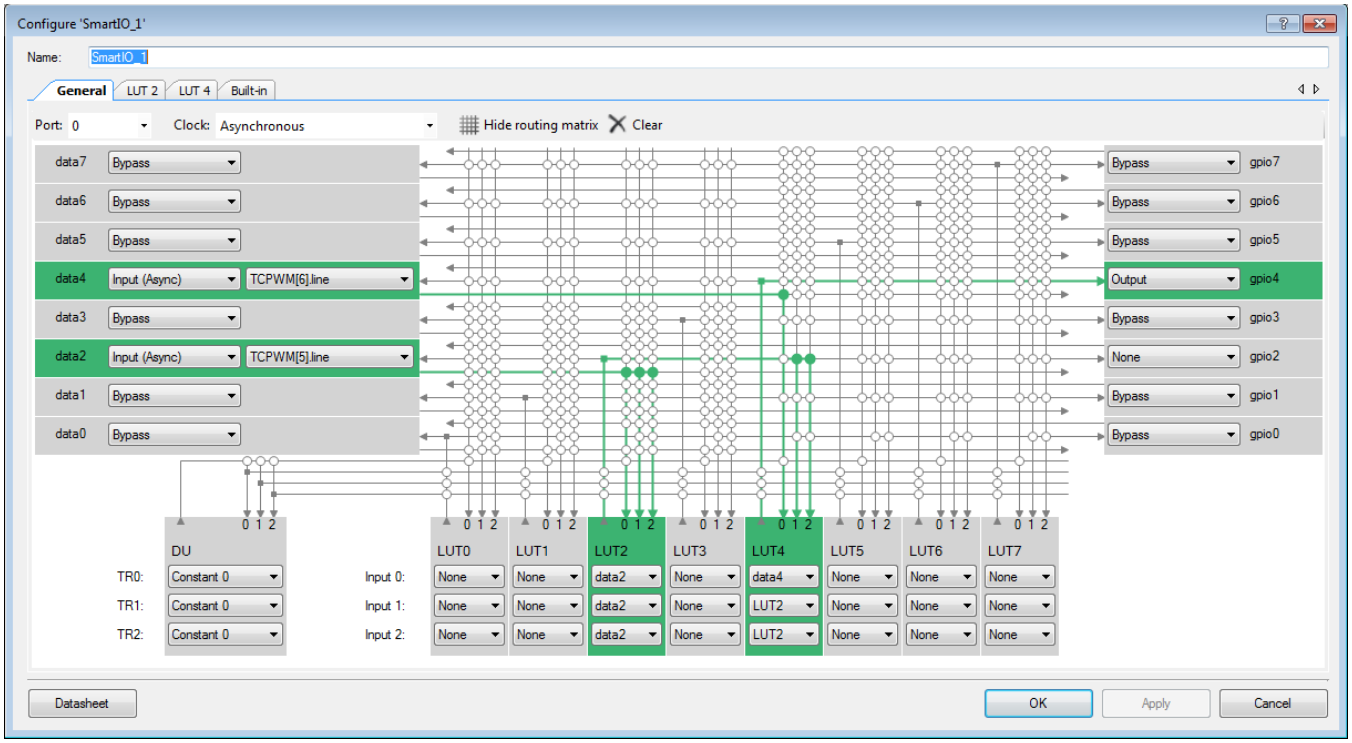


Figure 10. Smart I/O Look Up Table (LUT) Settings

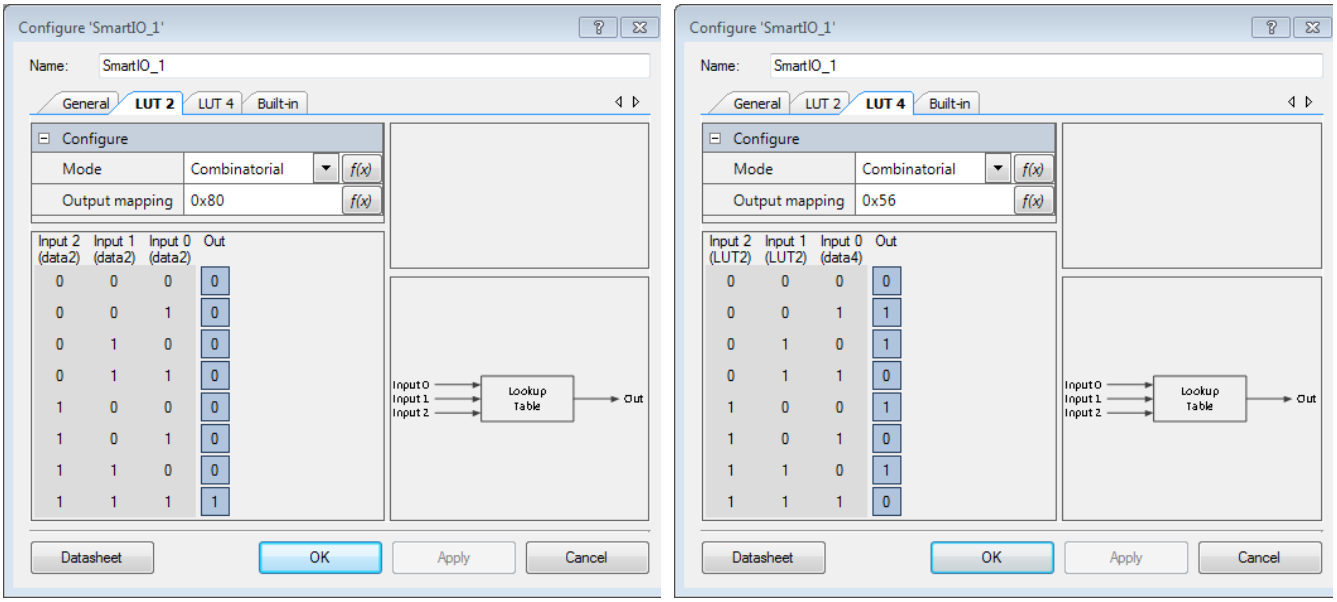
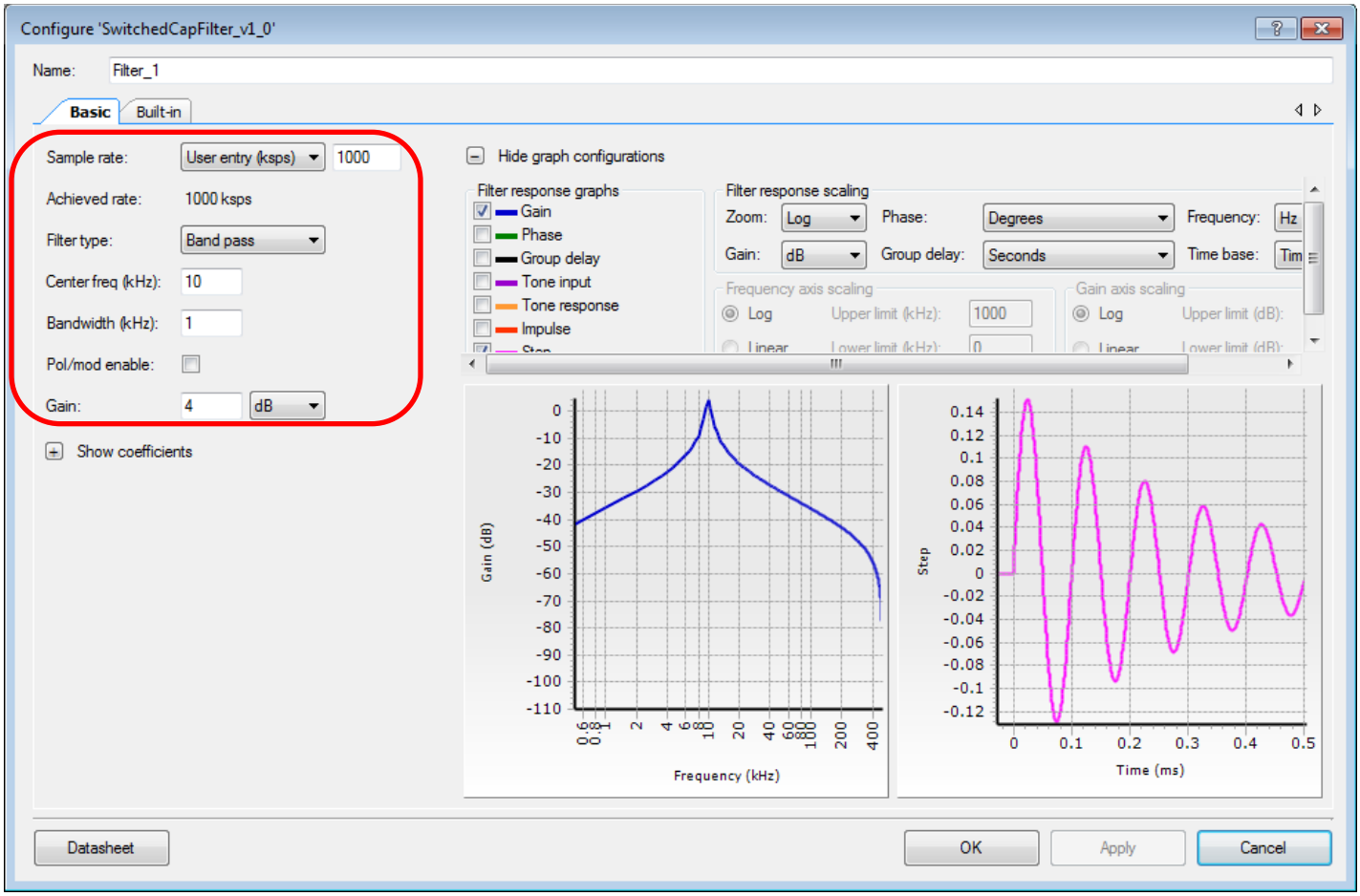
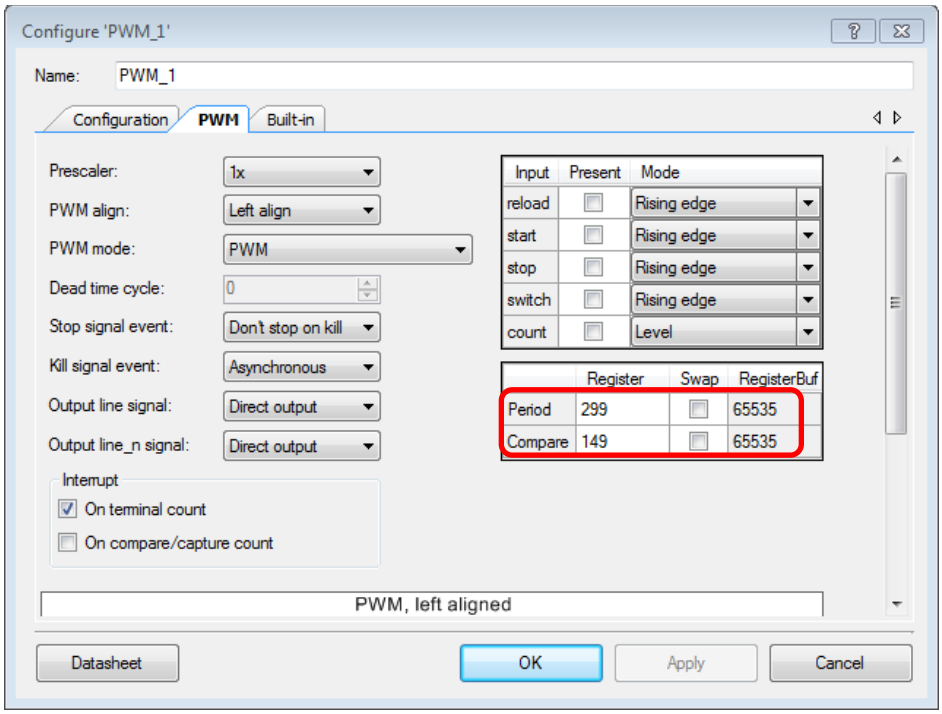


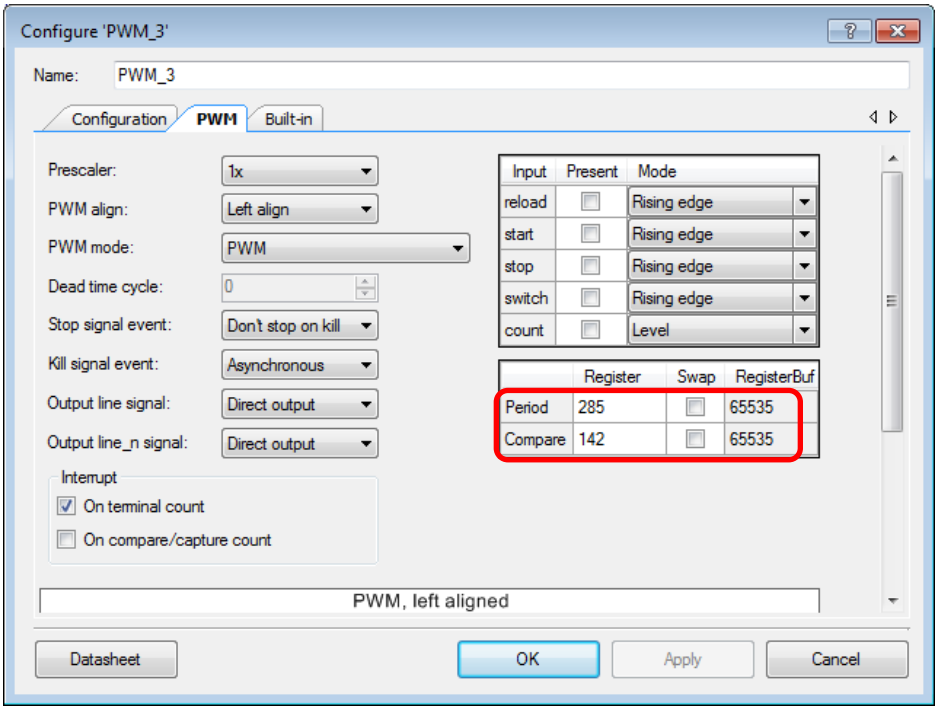
Figure 11. Band_Pass_Isolation Filter Component Settings



FSK_Mixer Non-Default Settings

Figure 12. FSK_Mixer PWM Settings





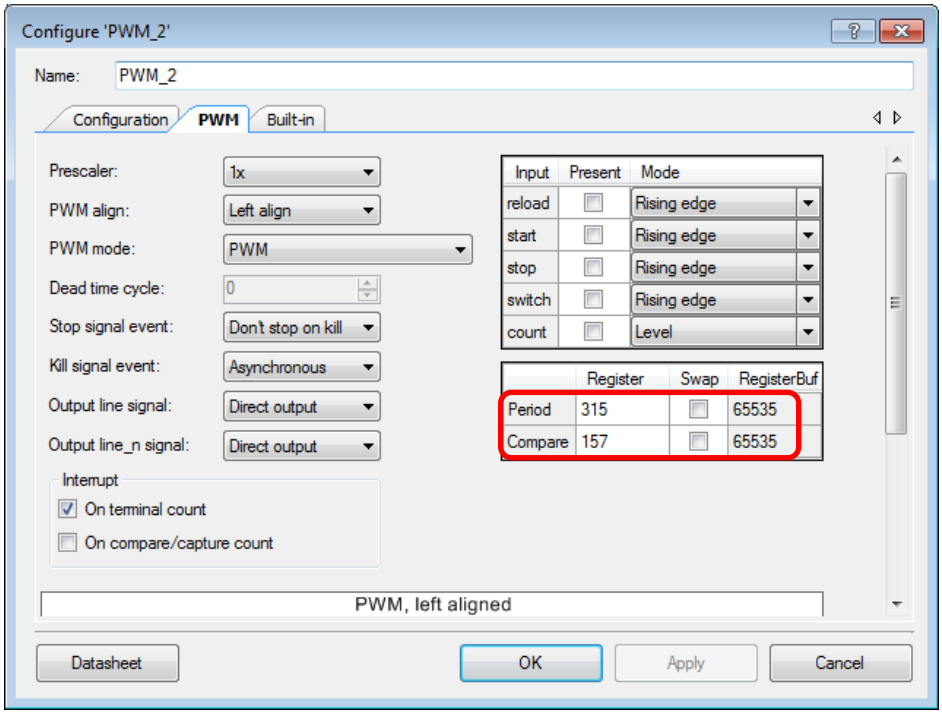

Configure 'PWM_2'
Name:
PWM_2
Configuration
PWM
Built-in
Prescaler:
1x
PWM align:
Left align
PWM mode:
PWM
Dead time cycle:
0
Stop signal event:
Don't stop on kill
Kill signal event:
Asynchronous
Output line signal:
Direct output
Output line_n signal:
Direct output
Interrupt
On terminal count
On compare/capture count
Input
Present
Mode
reload
start
stop
switch
count
Rising edge
Level
Register
Swap
RegisterBuf
Period
315
65535
Compare
157
65535
PWM, left aligned
Datasheet
OK
Apply
Cancel

Figure 13. Switched-Cap Filter Settings

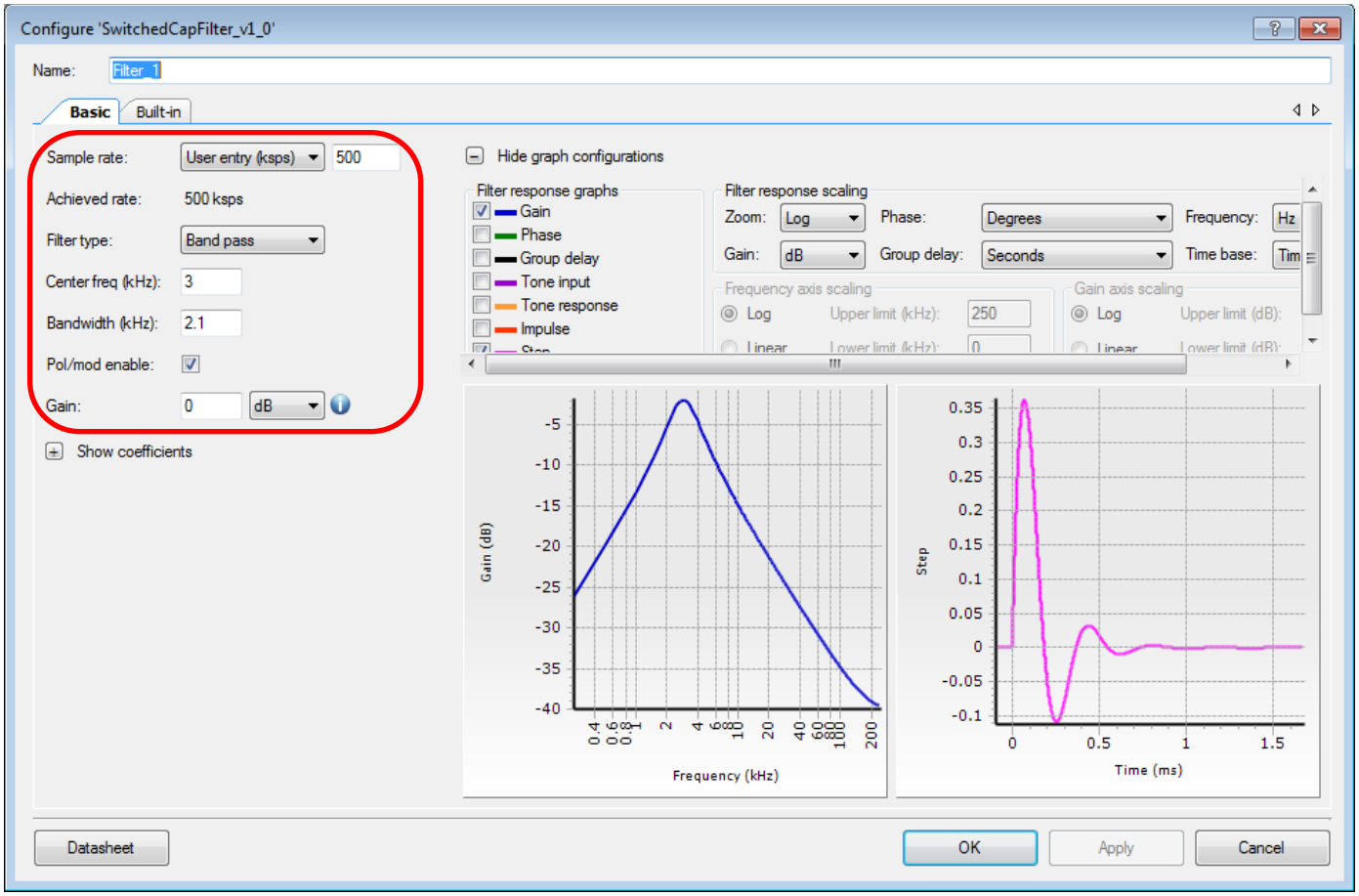


Figure 14. FSK_Mixer Smart I/O Routing Matrix

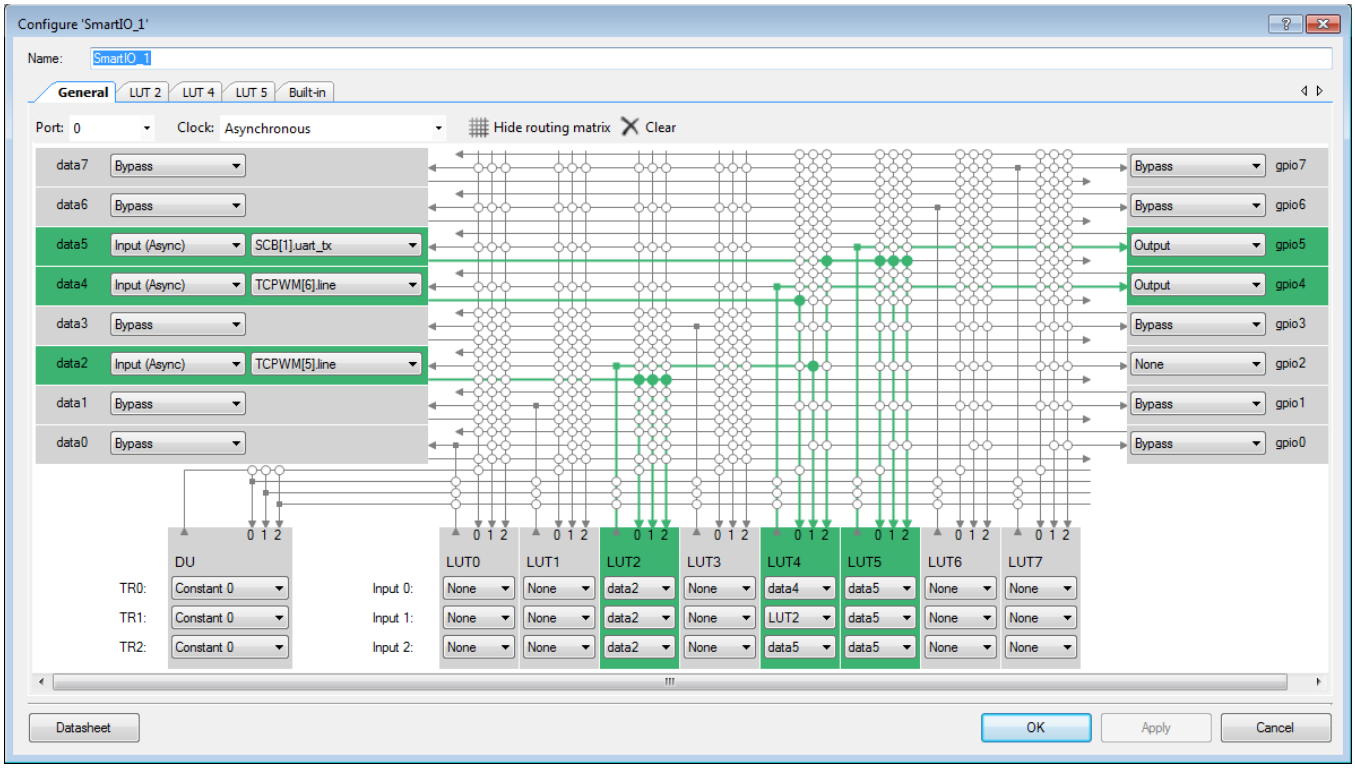
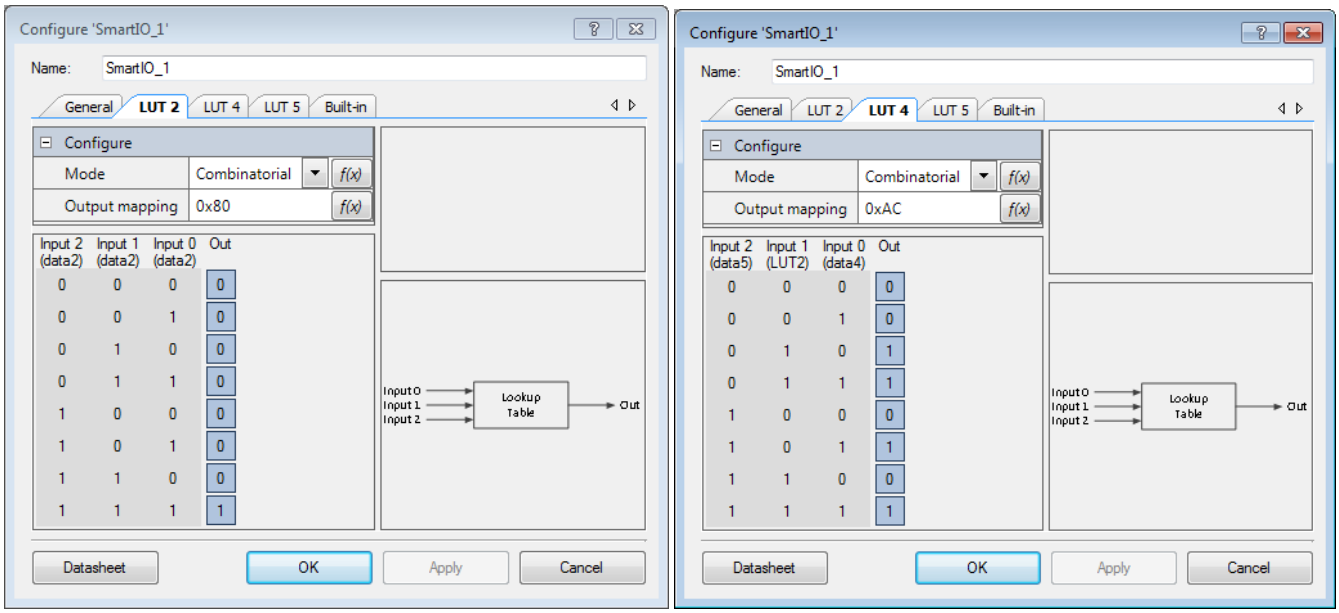
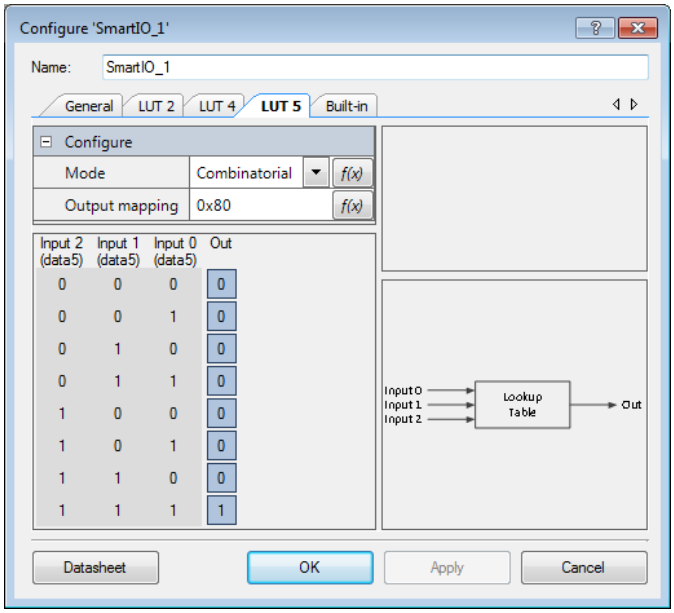


Figure 15. FSK_Mixer LUT Settings





Reusing This Example

This example is designed for the CY8CKIT-048 pioneer kit. The designs require a PSoC Analog Coprocessor device to work. Each project in this example uses a hardware focused approach. This leaves the CPU free to perform any task that you need to implement.

Related Documents

Application Notes	
AN211293 – Getting Started with PSoC Analog Coprocessor	Describes PSoC Analog Coprocessor and shows how to create a simple design in PSoC Creator
PSoC Creator Component Datasheets	
Serial Communication Block (SCB)	Supports UART, SPI, and I^2C communication
Timer Counter Pulse Width Modulator (TCPWM)	Supports PWM generation
Smart I/O	Supports programming logic on GPIO signals pins
Clock	Supports local clock creation and connection to system clocks
PVref	Provides local voltage reference
Switched-Cap Filter	Supports analog signal filtering
Device Documentation	
PSoC Analog Coprocessor: CY8C4AXX Family Datasheet	PSoC Analog Coprocessor Architecture Technical Reference Manual (TRM)
Development Kit (DVK) Documentation	
CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit	

Document History

Document Title: CE223818 - PSoC Analog Coprocessor Switched Capacitor Filter

Document Number: 002-23818

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6186421	BFMC	11/26/2018	New Code Example

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

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.