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.

CE211305 - Interfacing PSoC® Analog Coprocessor with an Inductive Proximity Sensor

Objective

This code example demonstrates how to implement an analog front end (AFE) for an inductive proximity sensor using the PSoC® Analog Coprocessor.

Overview

This code example demonstrates how to interface the PSoC Analog Coprocessor with an inductive proximity sensor. The code example measures the change in inductance to detect the presence of a metal disk in close proximity to an onboard coil, created using the PCB trace. The brightness of the RGB LED varies based on the proximity distance between the sensor and the metal. The measured sensor data is also sent over I^2C interface to a host PC running the Cypress's Bridge Control Panel (BCP) software.

Requirements

Tool: PSoC Creator™ 3.3 CP3 or later versions

Programming Language: C (ARM® GCC 4.9.3)

Associated Parts: All PSoC Analog Coprocessor parts

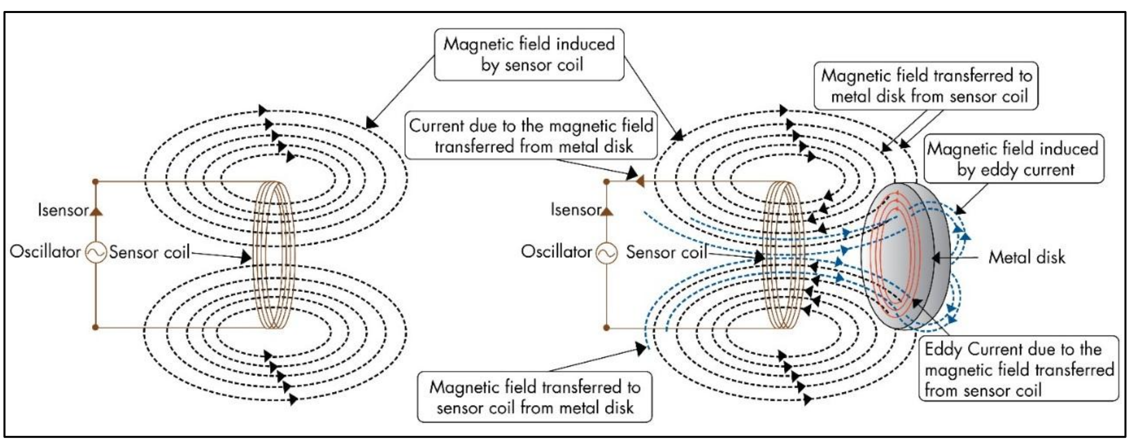
Related Hardware: CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit

Design

Inductive proximity sensing works on the principle of electromagnetic coupling between a sensor coil and the metal disk to be detected. When the metal disk enters the electromagnetic field induced by a sensor coil, some of the electromagnetic energy is transferred into the metal disk as shown in Figure 1. This transferred energy causes a circulating electrical current called eddy current. The eddy current flowing in the metal disk induces a reverse electromagnetic field on the sensor coil, which results in a reduction of effective inductance of the sensor coil.

The sensor coil is placed in parallel with a capacitor to form a tank oscillator circuit. The reduction in the sensor coil inductance causes a shift in the resonant frequency of the tank oscillator circuit. This shift in resonant frequency changes the amplitude of the signal across the sensor coil as shown in Figure 3. The change in the amplitude of the sensor coil signal is measured by the PSoC Analog Coprocessor to detect the presence of the metal disk.

Figure 1. Field Coupling between Sensor and Metal



The inductive proximity sensor implementation on the CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit consists of six stages: a sensor excitation circuit, a tank circuit, a high-pass filter (HPF), a down mixer, a rectifier and an ADC. The main stages of the inductive proximity sensing implementation – PWM of sensor excitation circuit, down mixer, rectifier, and ADC – are implemented in the PSoC Analog Coprocessor. Figure 2 shows the PSoC Creator schematic for inductive proximity sensing.

The inductive proximity sensor on the PSoC Analog Coprocessor Pioneer Kit is an onboard coil implemented using PCB trace; its inductance is 3 µH (L).

The inductive proximity sensor (L) in parallel with a capacitor (C80) forms a tank circuit as shown in Figure 2. The tank circuit is excited using a PWM. This PWM is called Excitation PWM in Figure 2. The resistor (R77) controls the excitation current of the inductive proximity sensor. The resonant frequency (f_r) of the tank circuit is determined by the following equation:

$$f_r = \frac{1}{(2\pi * \sqrt{LC})}$$

The HPF formed by the capacitor (C57) and resistor (R78) prepares the signal for the next stage. It removes the offset voltage and biases the filtered signal to a known reference voltage V_{REF} as shown in Figure 2.

The signal across resistor R78 is measured by passing it through a down mixer and a rectifier that are implemented using the Universal Analog Block (UAB). Note that custom components are provided with this code example for implementing down mixer and rectifier using the UAB. Refer to the section Implementation of Down Mixer and Rectifier Using PSoC Analog Coprocessor for details. The output of rectifier is connected to an ADC that is operating on synchronous clock as shown in Figure 2. The ADC is set to operate in averaging mode so that ADC value reflects the DC content of rectifier output. The ADC value is compared against the threshold value to detect the presence of the metal disk. The threshold value is selected such that it is less than the peak noise observed on the ADC value to avoid false triggers.

The sensor data is sent to a host PC using I²C. Also the brightness of the LED varies as the metal disk approaches the coil.

Figure 2. Inductive Proximity Sensing Schematic for PSoC Analog Coprocessor

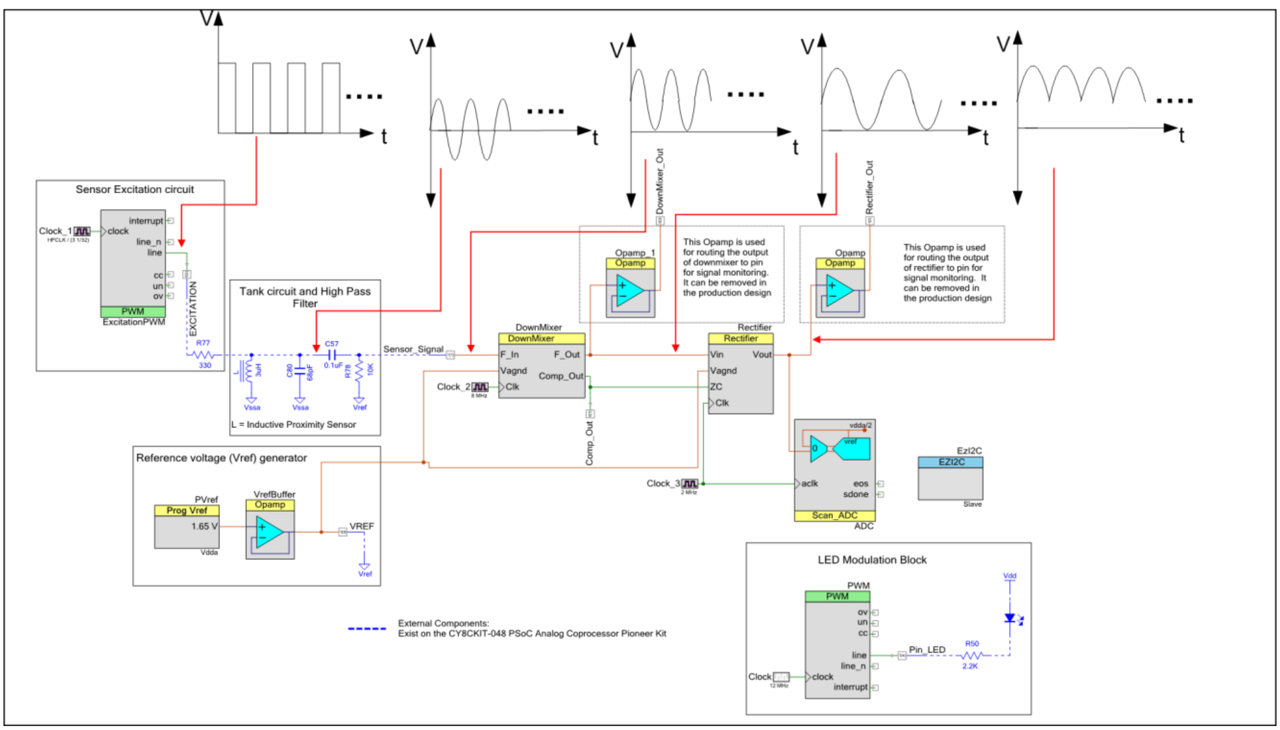


Figure 3 shows the change in the amplitude of sensor signal in the presence of the metal disk.

Figure 3. Voltage Signal Across the Sensor

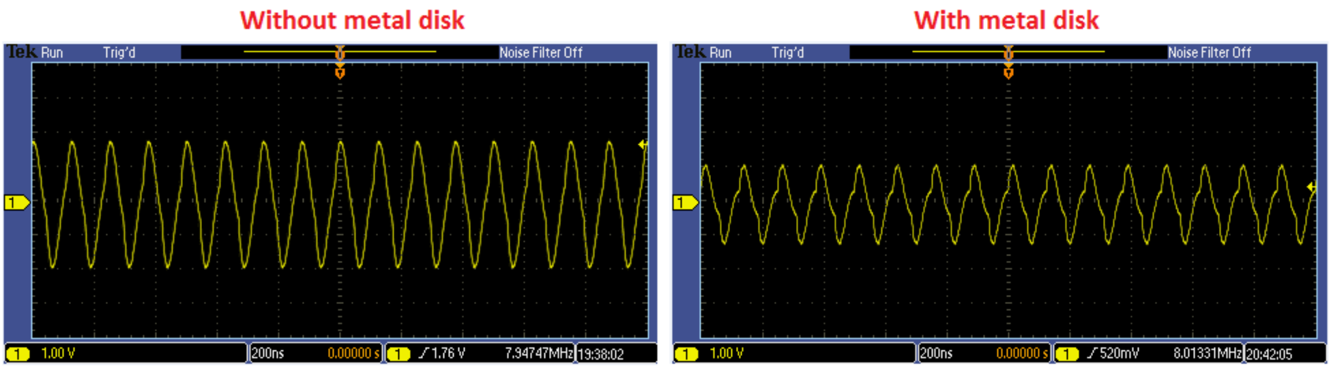
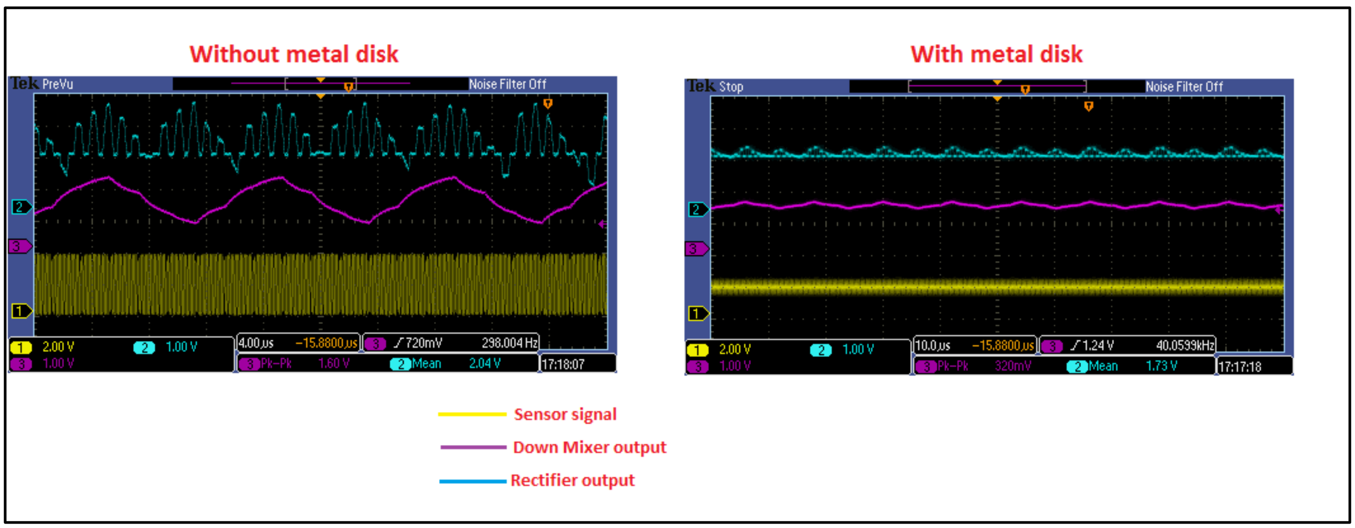


Figure 4 shows the sensor signal, output of down mixer and the output of rectifier with and without the metal disk. In the presence of the metal disk, the amplitude of the sensor signal is reduced. The change in the amplitude of sensor signal reflects on the amplitude of down mixer output and the rectifier output as shown in Figure 4.

Figure 4. Signals with and Without Metal Disk



Note: The output of the rectifier contains return to Vref pulses. This is due to the switched capacitor circuit present in UAB. The output of the rectifier is available in one phase of the switching clock. The envelope of this signal represents the rectified signal of the down mixer output.

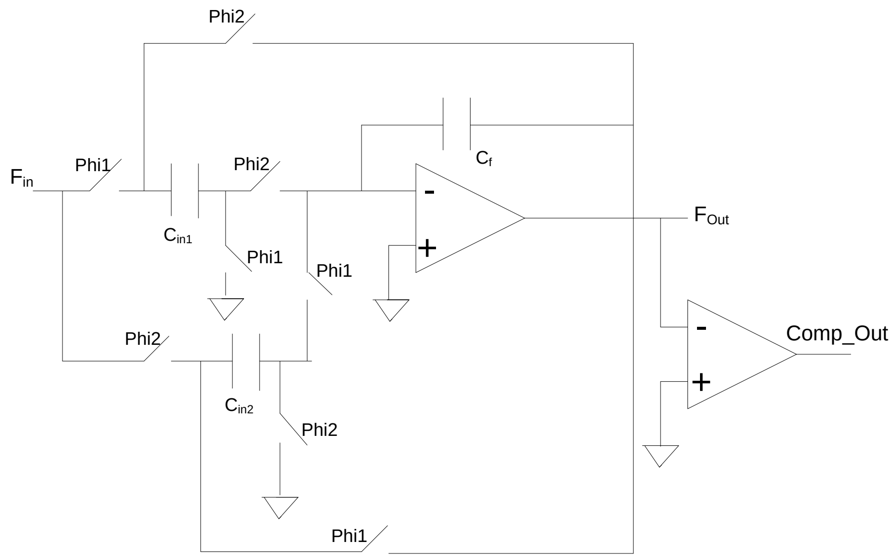
Implementation of Down Mixer and Rectifier Using PSoC Analog Coprocessor

The down mixer and rectifier are implemented in the Universal Analog Block (UAB) using custom components provided with this code example. The down mixer and rectifier occupy one half of the UAB each.

Down Mixer

This component converts the 7.917-MHz sensor signal to a 83-kHz signal by sampling the sensor signal at a rate of 8 MHz. The circuit diagram of the down mixer is provided below.

Figure 5. Down Mixer Circuit Diagram

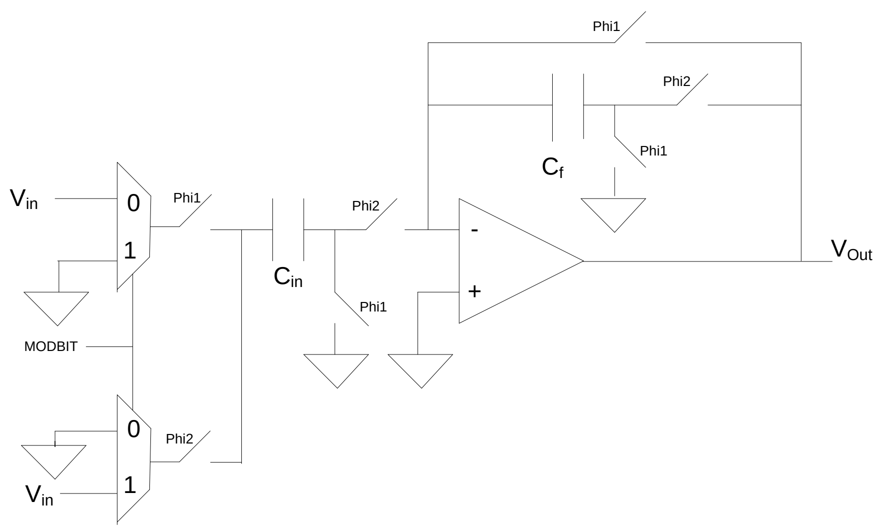


The input signal is sampled on both phases (Phi1 and Phi2) of clock. Of the two input capacitors, when one capacitor is acquiring charge from the input signal, the other capacitor holds the acquired charge. The fractional clock divider used as the source clock for excitation PWM causes harmonic distortion in the sensor signal. This effect of harmonic distortion is reduced by enabling the feedback capacitor C_f as shown in Figure 5. The output of the down mixer is compared against the analog reference voltage by a comparator present in the UAB.

Rectifier

The circuit diagram of the rectifier is shown Figure 6.

Figure 6. Rectifier Circuit Diagram



The input capacitor C_{in} acquires charge from V_{in} in either the Phi1 or Phi2 phase based on the polarity of MODBIT. The charge is transferred from C_{in} to C_f during the Phi2 phase. The Comp_Out of the down mixer is connected to the Zero Crossing (ZC) input of the rectifier as shown in Figure 2. Note that the ZC input is internally connected to MODBIT of the rectifier. As shown in Figure 6, the output of the rectifier is available during the Phi2 phase, and is connected to the analog reference during the Phi1 phase.

Design Considerations

To achieve maximum sensitivity, it is recommended to drive the coil at a PWM frequency that is close to the resonant frequency of the tank circuit. In this example, the resonant frequency of the tank circuit is 11 MHz (L = 3 uH and C = 68 pF) and the PWM frequency is 7.917 MHz.

This design can be extended to support other inductive proximity sensors. The analog front end varies with the characteristics of the inductive proximity sensor.

This code example is designed for the PSoC Analog Coprocessor Pioneer Kit. The design is easily portable to other kits and PCBs.

Hardware Setup

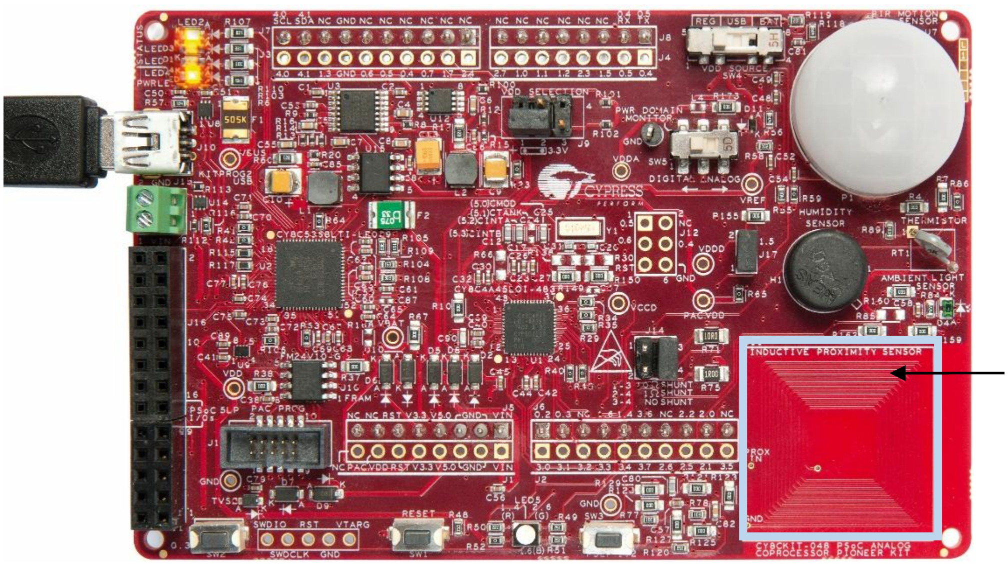
Set the SW4 switch on the PSoC Analog Coprocessor Pioneer Kit to the 'REG' position to select the regulator as the VDD source. Set the J9 jumper to 1-2 for 3.3-V device operation. If you want to use a different power supply source or a different VDD value, select the SW4 and J9 settings according to Table 1.

Table 1. PSoC Analog Coprocessor Pioneer Kit Power Supply Source and VDD Selection

Power Supply Source	V_{DD} (volts)	SW4 (Switch Position)	J9 (Jumper Position)
USB	1.8	REG	open
	3.3	REG	1-2
	5.0	USB	Any position except 2-3
	1.8 - 3.3	REG	4-2
External VIN	1.8	REG	open
	3.3	REG	1-2
	5.0	REG	2-3
Arduino baseboard	1.8	REG	open
	3.3	REG	1-2
Coin Cell	3.0	BAT	NA

Connect the PSoC Analog Coprocessor Pioneer Kit to the computer's USB port using the USB cable provided with the kit, as shown in Figure 7.

Figure 7. Hardware Connection

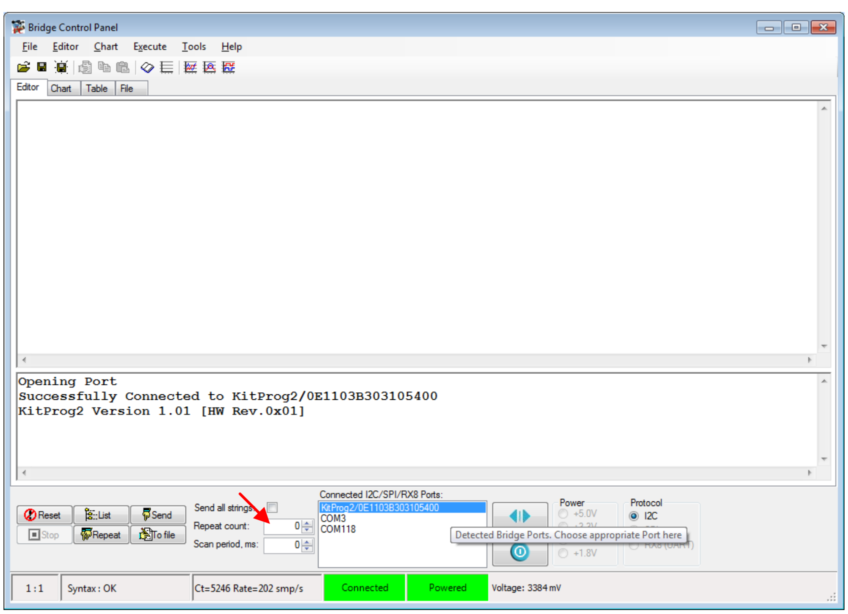


Software Setup

This section describes how to set up the Cypress Bridge Control Panel (BCP) software for viewing sensor data sent over I^2C. BCP is installed automatically as part of the kit software installation. Follow these steps to configure BCP:

1. Open the BCP from **Start** > **All Programs** > **Cypress** > **Bridge Control Panel <version>** > **Bridge Control Panel <version>**.
2. Select **KitProg2/ID** under **Connected I2C/SPI/RX8 Ports** (see Figure 8). Note that the PSoC Analog Coprocessor Pioneer Kit must be connected to the USB port of your computer.

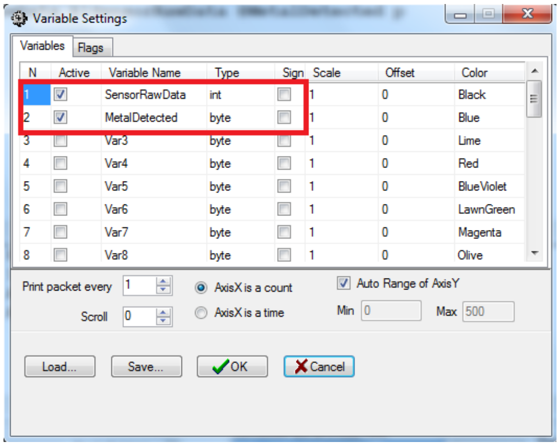
Figure 8. I^2C Devices List



3. Select **Tools** > **Protocol Configuration**, navigate to the **I2C** tab, and set the **I2C speed** to ‘100 kHz’. Click **OK**.
4. Select **Chart** > **Variable Settings** and **Load** the *CE211305_Inductive_Proximity_Sensing.ini* file from the following path: `<Install_Directory>\CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit\<version>\Firmware\PSoC Analog Coprocessor\BCP Command\`. Click **OK**. See Figure 9.

   This file includes the variable names, their data type, and the sign to represent the data sent over I^2C.

Figure 9. Variable Setting



BCP is now ready for reading and displaying the sensor data. Refer to the Operation section for the testing procedure.

Components

Table 2 lists the PSoC Creator Components used in this example, the hardware resources used by each Component and non-default parameter settings for each Component.

Table 2. PSoC Creator Components

Component	Name	Version	Hardware Resources	Non-Default Parameter Settings
Scanning SAR ADC	ADC	v1.10	SAR ADC	**Config tab:** Free-run scan rate (SPS): 1000 Input Range: Vref select: Vdda/2 Vneg for S/E: Vref Result Data Format: Samples averaged: 64 Number of channels: 1 Input mode: Single ended Avg: Enable **Common tab:** Show analog clock (aclk) terminal: Enable
Opamp	VrefBuffer	v1.20	CTB	Mode: Follower Output: Output to pin
Prog Vref	PVref	v1.0	PRB	Reference source: Vdda (V) Voltage reference (V): 1.65 (tap 8)
Opamp	Opamp_1	v1.20	CTB	Mode: Follower Output: Output to pin
Opamp	Opamp	v1.20	CTB	Mode: Follower Output: Output to pin

Component	Name	Version	Hardware Resources	Non-Default Parameter Settings
PWM	ExcitationPWM	v2.10	TCPWM	Period : 1 Compare : 1
EzI2C	EzI2C	v3.10	SCB	-
PWM	PWM	v2.10	TCPWM	Period : 1200 Compare : 1200
DownMixer (Custom Component)	DownMixer	v0.0	UAB	InputCapacitor:5
Rectifier (Custom Component)	Rectifier	v0.0	UAB	InputCapacitor: 63 FeedbackCapacitor: 31

Design-Wide Resources

Table 3 lists the physical pins used.

Table 3. Pin Names and Locations

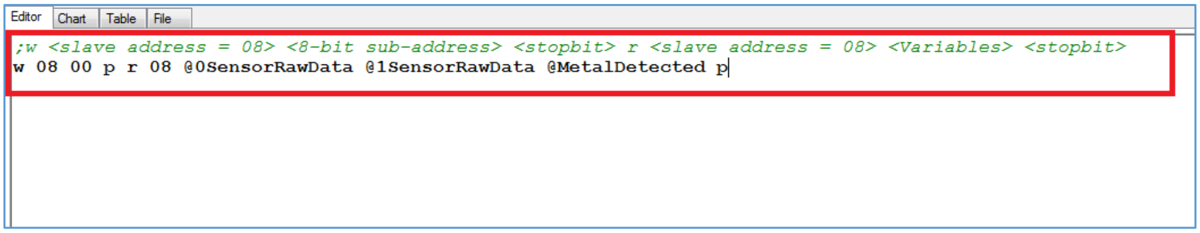
Pin Name	Location
Sensor_Signal	P1[1]
Rectifier_Out	P1[2]
VREF	P1[3]
EXCITATION	P0[2]
EzI2C:SCL	P4[0]
EzI2C:SDA	P4[1]
PIN_LED	P1[4]
DownMixer_Out	P2[2]
Comp_Out	P0[7]

Note: EzI2C pins are embedded within the Component.

Operation

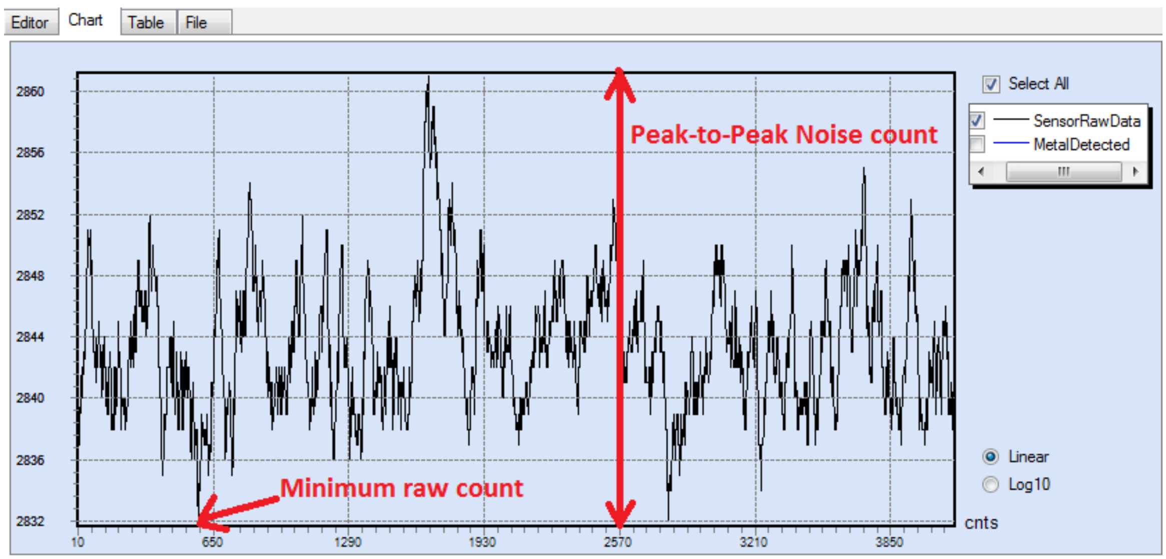
1. Select the *CE211305_Inductive_Proximity_Sensing.cywrk* file in the PSoC Creator Start page under **Examples and Kits** > **Kits** > **CY8CKIT-048**. Select a location to save the code example.
2. Build the project (**Build** > **Build CE211305_Inductive_Proximity_Sensing**).
3. Connect the PSoC Analog Coprocessor Pioneer Kit to your computer's USB port, as described in the section Hardware Setup.
4. Program the PSoC Analog Coprocessor device; select **Debug** > **Program**.
5. Configure the BCP software as described in the section Software Setup.
6. Select **File** > **Open File**. Open the *CE211305_Inductive_Proximity_Sensing.iic* file from the following path:
```
<Install_Directory>\CY8CKIT-048 PSoC Analog Coprocessor Pioneer
Kit\<version>\Firmware\PSoC Analog Coprocessor\BCP Command\
```
   This file contains the read command to be executed by the BCP. The command appears on the panel as Figure 10 shows.

Figure 10. Read Command in the Bridge Control Panel



7. Click the command on the BCP **Editor** tab and then click the **Repeat** button to read the sensor data.
8. Go to the **Chart** tab and observe the plot of sensor raw data and metal detection status.
9. Select only the SensorRawData variable.
10. Take the *minimum raw count* from the SensorRawData as shown in Figure 11.

Figure 11. Noise Count Chart



11. Calculate *Threshold = Minimum raw count – Peak to Peak Noise count.*
12. Assign the threshold calculated in step 11 to the macro `THRESHOLD` in *main.c*.
13. Build the project (**Build > Build CE211305_Inductive_Proximity_Sensing**).
14. Program the PSoC Analog Coprocessor device (**Debug** > **Program**).

    Note: Disconnect the **KitProg2/<serial number>** under **Connected I2C/SPI/RX8 Ports** in BCP in order to program the device.
15. Bring the metal disk (provided with the PSoC Analog Coprocessor Pioneer Kit) close to the onboard sensor such that the common area between the surface of the metal disk and the coil is maximum as shown in Figure 12 and observe the change in sensor data as shown in Figure 13.

Figure 12. Metal Disk at Proximity Distance to Coil

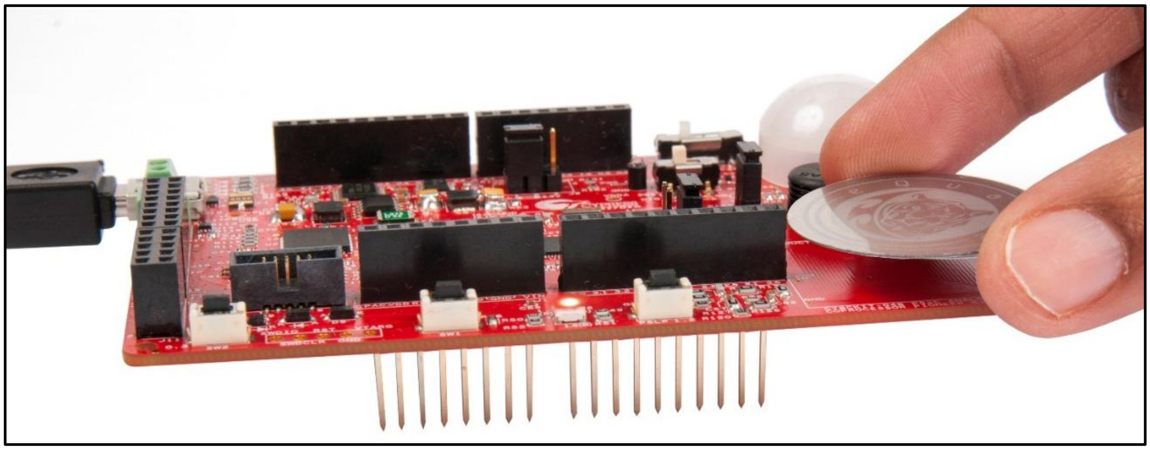
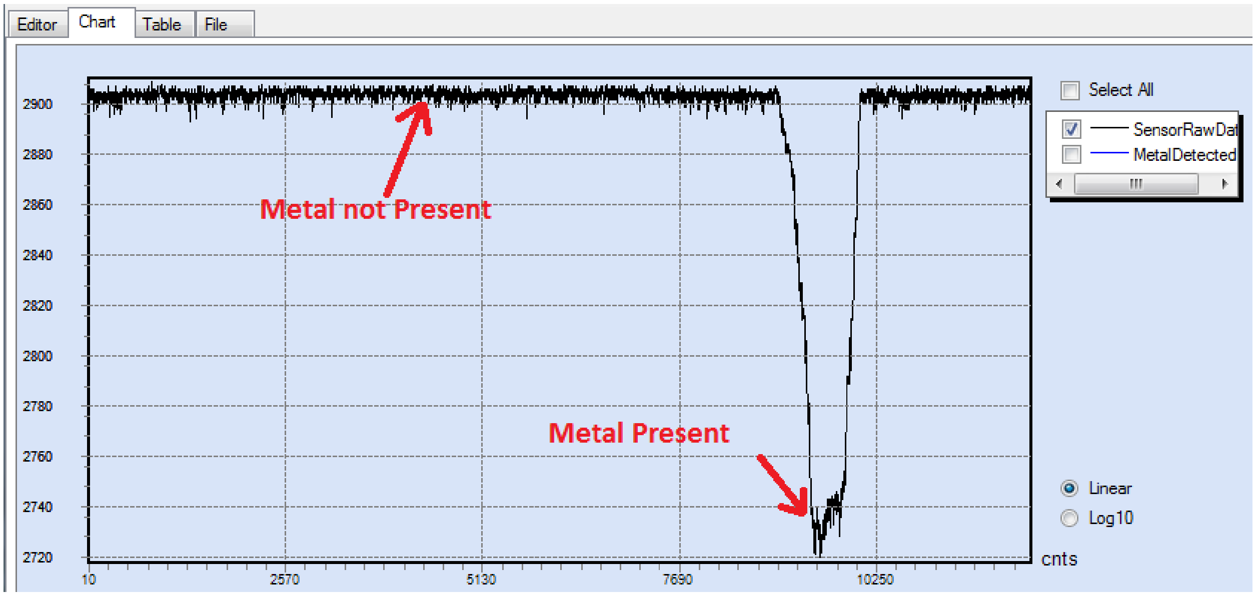


Figure 13. Output on Bridge Control Panel



16. The intensity of the red LED increases as the metal disk approaches the coil.

Related Documents

Table 4 lists the relevant application notes, code examples, Component datasheets, device documentation, and DVK documentation.

Table 4. Related Documents

Application Notes		
AN211293	Getting Started with PSoC Analog Coprocessor	Describes the PSoC Analog Coprocessor
PSoC Creator Component Datasheets		
PGA	Support configurable gain of 2 to 32	
Scanning SAR ADC	Supports multiple channel hardware scans with single-ended and differential input modes	
PVref	Generates configurable voltage references using internal bandgap voltage or supply voltage VDDA	
Opamp	Supports voltage follower mode and Opamp mode with configurable power	
EZI2C Slave	Simplified I^2C slave implementation	
Pins	Supports connection of hardware resources to physical pins	
PWM	Supports Timer, Counter, PWM functions	
Device Documentation		
PSoC Analog Coprocessor Datasheets		
PSoC Analog Coprocessor Architecture Technical Reference Manual		
PSoC Analog Coprocessor Register Technical Reference Manual		
DVK Documentation		
CY8CKIT-048 PSoC Analog Coprocessor Pioneer Kit		

Document History

Document Title: CE211305 – Interfacing PSoC® Analog Coprocessor with an Inductive Proximity Sensor

Document Number: 002-11305

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5302416	DIMA	06/11/2016	New code example.
*A	5739878	AESATP12	05/26/2017	Updated logo and copyright.

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

Cypress Developer Community

Forums | WICED IOT 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