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.

Objective

This code example demonstrates how to implement an analog front end (AFE) for a Pyroelectric Infrared (PIR) motion sensor, using the PSoC 4100PS family of devices.

Overview

This code example demonstrates how to measure the voltage signal from a PIR motion sensor and detect the movement of an infrared (IR) emitting object. The measured PIR motion sensor signal and the motion detection status are sent over I^{2}C to a host PC running the Cypress' Bridge Control Panel (BCP) software.

Requirements

Tool: PSoC Creator™ 4.2 or later versions

Programming Language: C (ARM® GCC 5.4)

Associated Parts: PSoC 4100PS

Related Hardware: CY8CKIT-147 PSoC 4100PS Prototyping Kit

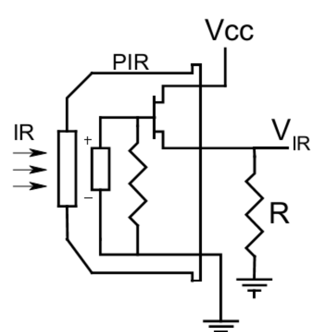
Design

PIR Motion Sensing – Theory of Operation

The PIR motion sensor is based on the pyroelectric effect, where certain materials generate a voltage when exposed to infrared radiation. This radiation is the portion of the electromagnetic spectrum that falls between microwaves and visible light. Infrared radiation has wavelengths longer than visible light but shorter than microwaves. Humans at normal body temperature radiate strongest in the infrared at an approximate wavelength of 10 µm.

The PIR motion sensor uses infrared sensitive materials as the sensing elements. It is packaged with a field effect transistor (FET) in the source follower mode, as Figure 1 shows. The FET is required to buffer the high-impedance output of the sensor element. When the sensor element is exposed to infrared radiation, a voltage is generated across the element.

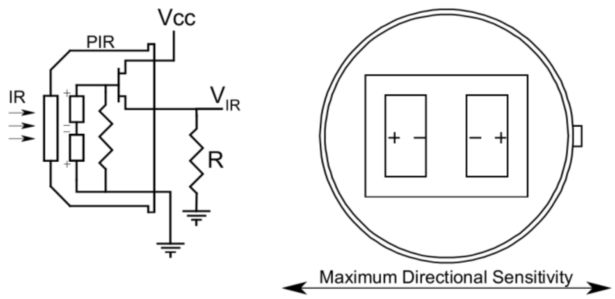
Figure 1. PIR Motion Sensor – Single-Element



Most common PIR motion sensors have two or four sensing elements. These elements are arranged such that the voltage generated by one is subtracted by the other. This arrangement cancels the common signal and generates a voltage only when there is a difference in the incident infrared radiation level on the sensing elements.

Figure 2 shows the dual element PIR motion sensor with the elements connected in series but with an opposite phase, because of which it has maximum sensitivity along a particular axis.

Figure 2. PIR Motion Sensor Dual-Element



The sensor package is designed to have a unique field-of-view for each element. When an IR radiating source moves across the fields of view, the sensor generates a differential signal (see Figure 3). For a 90° field-of-view or more, a Fresnel lens is mounted on the PIR motion sensor. It improves the sensitivity and thus the detection distance.

Figure 3. PIR Motion Sensor Output Response

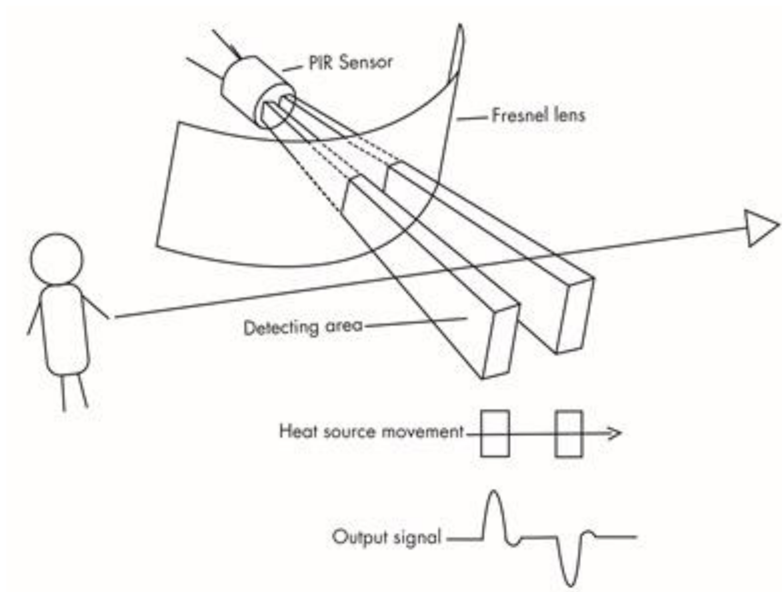
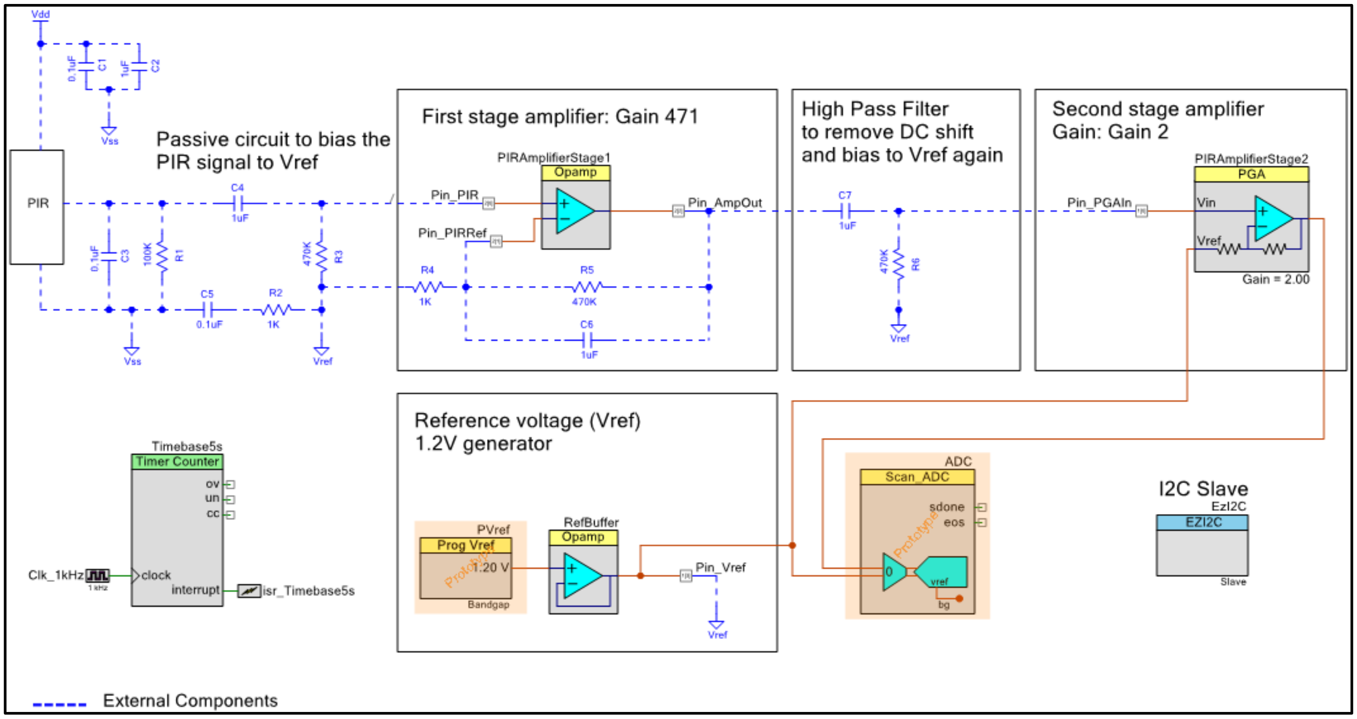


Figure 4 shows the PSoC Creator schematic for interfacing a PIR motion sensor with the PSoC 4100PS.

Figure 4. PIR Motion Sensing Schematic



The PIR motion sensor implementation consists of five stages: a bias circuit for the PIR motion sensor, a first-stage amplifier, a high-pass filter (HPF), a second-stage amplifier, and an ADC. In the main stages of the PIR motion sensor implementation, the two amplifiers are implemented in the PSoC 4.

This code example uses a dual-element PIR motion sensor (ZRE200GE). The voltage signal generated by the sensor is AC-coupled and clamped to the internal reference voltage V_{Ref}. The typical sensor output voltage is in the order of several millivolts and varies depending on the strength of the incident infrared radiation. To detect the motion of a human body at a long distance, a gain of ≈1000 is required. A single-stage amplifier with such a high gain causes the amplifier output to saturate due to the amplification of the input offset voltage. Thus, a two-stage amplifier is best suited for amplifying with high gain.

The total gain is split between two stages. The first stage amplifier gain is set to 471 using internal Opamp and external gain setting resistors. The second-stage amplifier uses a PGA Component with a gain of 2.

A HPF, made using external passive components C7 and R6, is introduced between the first and the second amplifier stages to eliminate the offset voltage.

The PIR motion sensor and gain stages use a 1.2-V bandgap voltage as the reference voltage. The bandgap voltage is independent of supply voltage fluctuations and hence provides a stable voltage reference. This voltage is generated using a programmable reference component, PVref, and is buffered using an Opamp.

The output of the second-stage PGA is connected to the Scanning SAR ADC Component. The Scanning SAR ADC results are compared against threshold values to detect the motion of an IR emitting object.

The PIR raw count, detection thresholds, and detection status are sent to a host PC using I^2C.

Design Considerations

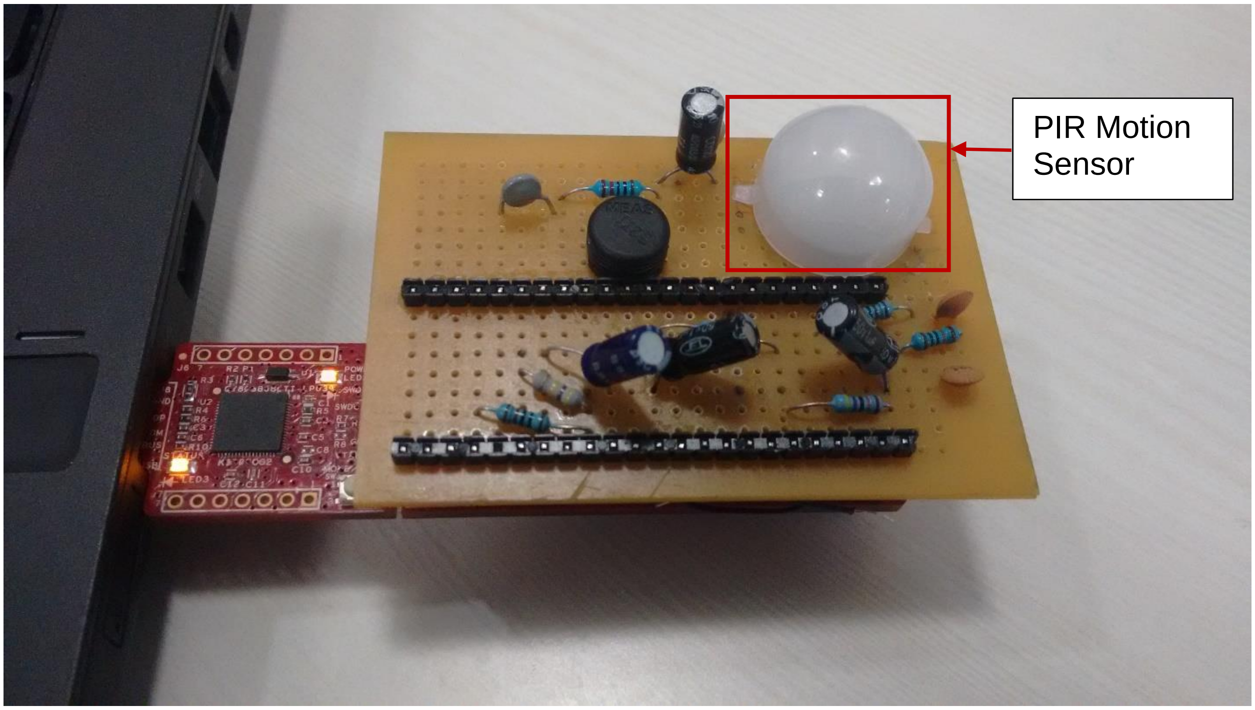
This design can be adapted for other PIR motion sensors. You may need to change amplifier gains depending on the sensor characteristics and the required distance of detection.

This code example is designed for the PSoC 4100PS Prototyping Kit. The analog front end (AFE) shown in Figure 4 is implemented on a general-purpose PCB and is used as a shield board as shown in Figure 5.

Hardware Setup

This code example uses a general-purpose PCB (used as customized shield hardware) to make the connection shown in Figure 4. Stack up the custom shield and CY8CKIT-147 PSoC 4100PS Prototyping Kit and connect it to your computer's USB port as Figure 5 shows.

Figure 5. Hardware Connection



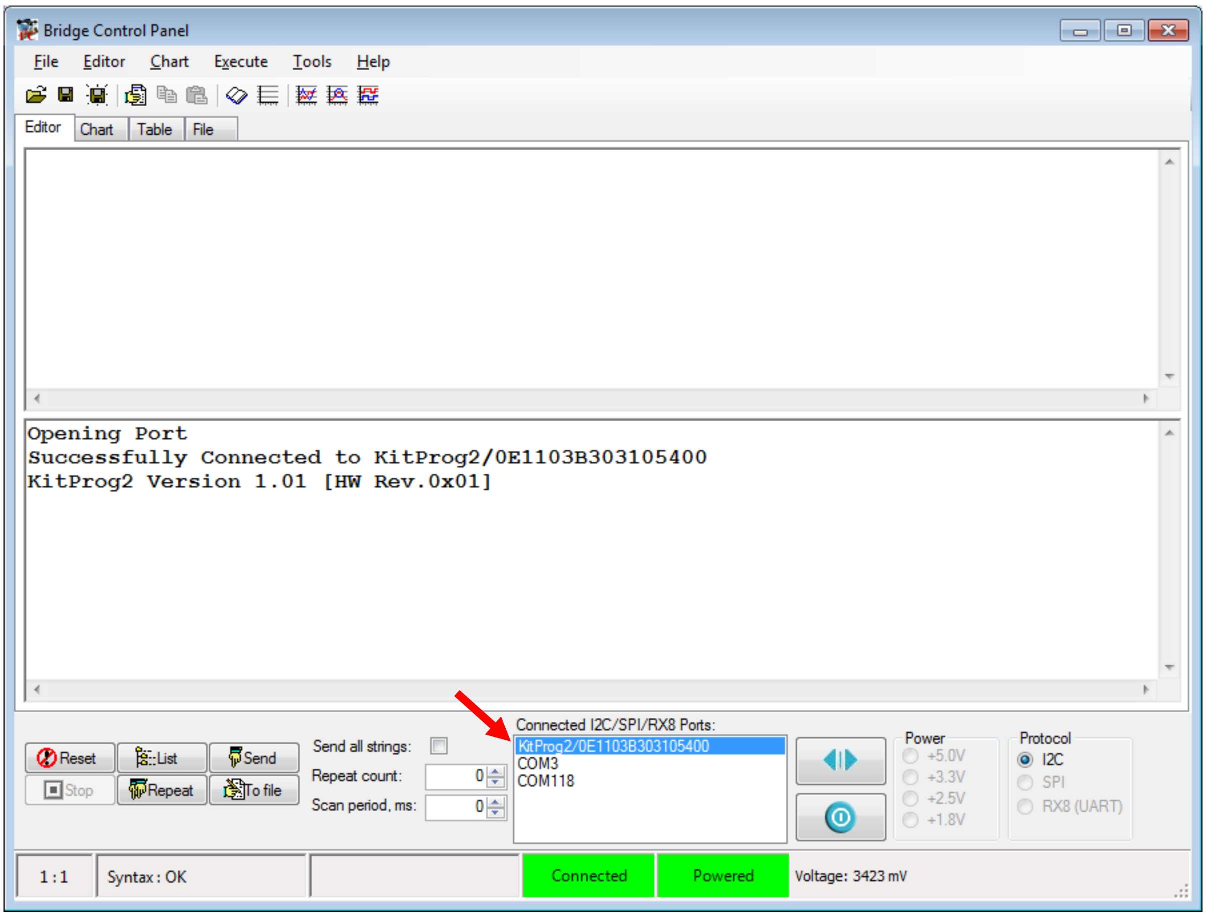
Software Setup

This section describes how to set up the Cypress Bridge Control Panel (BCP) software for viewing sensor data sent over I^2C.

The BCP is installed automatically as part of the kit software installation. Follow these steps to configure the BCP:

1. Open the BCP from: **Start** > **All Programs** > **Cypress** > **Bridge Control Panel <version>** > **Bridge Control Panel <version>.**
2. Select **KitProg2/<serial number>** under **Connected I2C/SPI/RX8 Ports** (see Figure 6). Note that the PSoC 4100PS Prototyping Kit must be connected to the USB port of your computer.

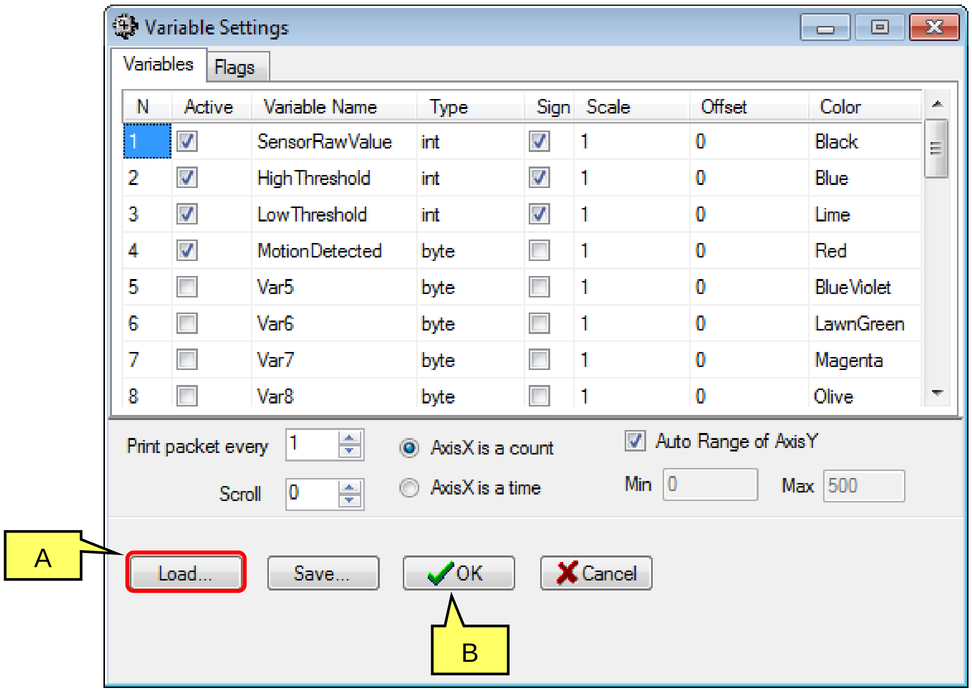
Figure 6. Bridge Control Panel



3. Select **Tools** > **Protocol Configuration**, and navigate to the **I2C** tab, and set the **I2C speed** to ‘100 kHz’. Click **OK**.
4. Select **Chart** > **Variable Settings,** and
   A. Load the *CE223618_PIR_Motion_Sensing.ini* file from the following path: `CE223618_PIR_Motion_Sensing\CE223618_PIR_Motion_Sensing.cydsn\BCP Command`.
   B. Click **OK** as Figure 7 shows.

This file includes the variable names, their data type, and their signs, to represent the data sent over I²C.

Figure 7. Variable Settings in Bridge Control Panel Software



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

Components

Table 2 lists the PSoC Creator Components used in this example and the hardware resources used by each Component.

Table 2. List of PSoC Creator Components

Component	Instance Name	Version	Hardware Resources
Scanning SAR ADC	ADC	v2.20	SAR ADC
Opamp	RefBuffer	v1.20	Continuous Time Block (CTB)
Opamp	PIRAmplifierStage1	v1.20	Continuous Time Block (CTB)
PGA	PIRAmplifierStage2	v1.10	Continuous Time Block (CTB)
PVref	PVref	v1.0	Programmable Reference Block (PRB)
Clock	Clk_1kHz	v2.20	Clock
Timer Counter (TCPWM Mode)	Timebase5s	v2.10	Timer Counter Pulse Width Modulator (TCPWM)
EZI2C Slave (SCB mode)	EZI2C	v4.0	Serial Communication Block (SCB)
Analog Pin	Pin_Vref, Pin_PGAIn, Pin_AmpOut, Pin_PIR, Pin_PIRRef	v2.20	I/O

Parameter Settings

Table 3 lists the non-default settings of all the components used in the design.

Table 3. Component Parameters

Component Instance Name	Settings (Non-Default)
ADC	Free-run scan rate (SPS): 10000 Number of Channels: 1
RefBuffer	Mode: Follower Output: Output to pin Power/Bandwidth: High
PIRAmplifierStage1	Output: Output to pin
PIRAmplifierStage2	Gain: 2
PVref	-
Clk_1kHz	Frequency: 1 kHz
Timebase5s	Compare/Capture: Compare Period: 4999 Compare: 2500
EZI2C	-
Pin_PIR , Pin_PGAIn, Pin_AmpOut, Pin_PIRRef, Pin_Vref	External terminal: Enabled

Note EZI2C pins are embedded within the Component.

Design-Wide Resources

Table 4 shows the physical pins used.

Table 4. Pin Names and Locations

Pin Name	Location
EZI2C: SCL	P3[6]
EZI2C: SDA	P3[7]
Pin_PIR	P2[0]
Pin_PIRRef	P2[1]
Pin_AmpOut	P2[2]
Pin_PGAIn	P1[0]
Pin_Vref	P1[3]

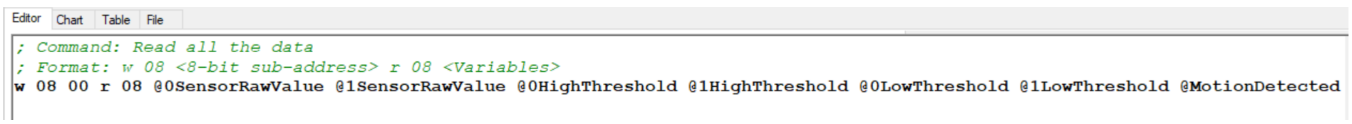
Operation

Follow these steps:

1. Open the project attached with this Code Example in PSoC Creator.
2. Build the project; select the PSoC Creator menu item **Build** > **Build CE223618_PIR_Motion_Sensing**.
3. Connect the PSoC 4100PS Prototyping Kit to your computer's USB port, as described in the section Hardware Setup.
4. Program the PSoC 4100PS device; select **Debug** > **Program**.
5. Configure the BCP software as described in the section Software Setup.
6. Select **File** > **Open File**. Open the *CE223618_PIR_Motion_Sensing.iic* file, from the following path: `CE223618_PIR_Motion_Sensing\CE223618_PIR_Motion_Sensing.cydsn\BCP Command\`

   This file contains the read and write commands to be executed by the BCP. The commands appear on the panel, as Figure 8 shows.

   Figure 8. Read and Write Command in the Bridge Control Panel

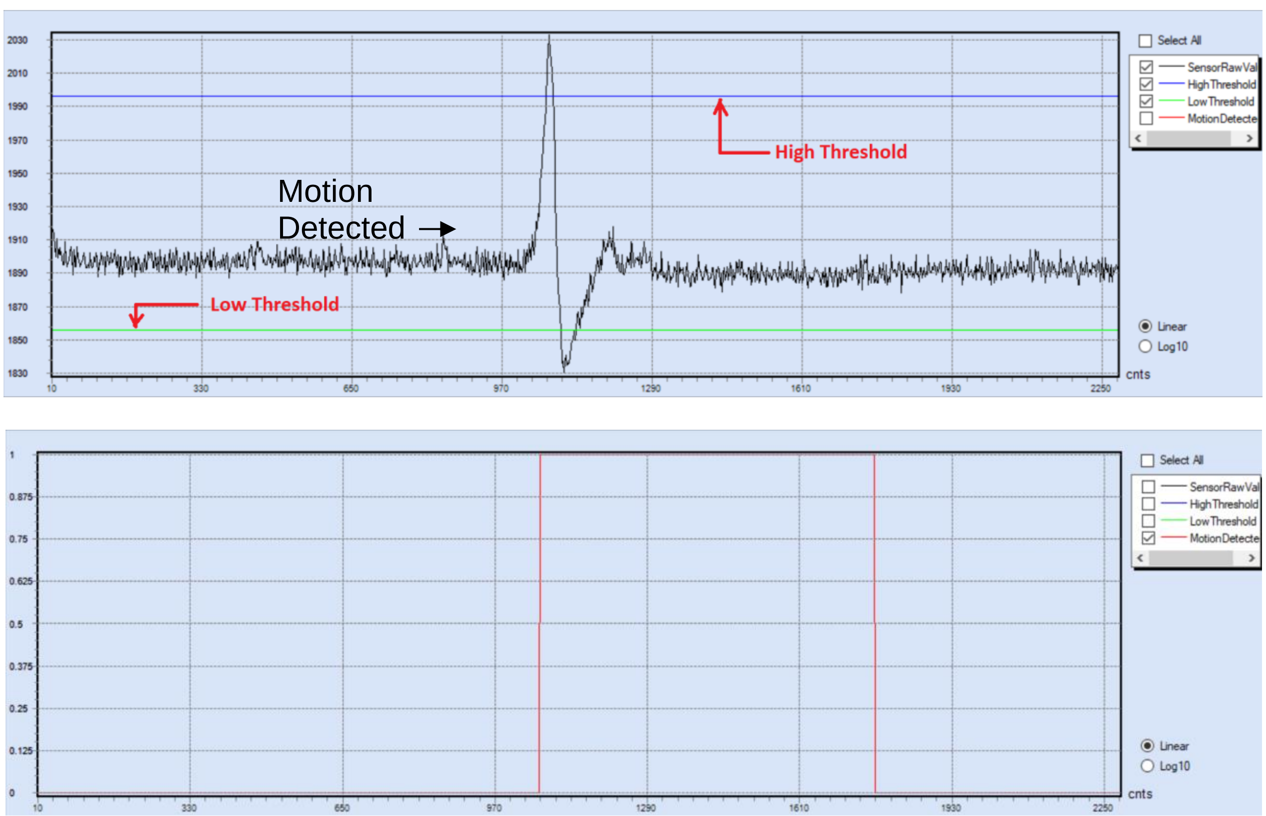


   To execute the command, click on the instruction and then click the **Send** button. See the comments in the BCP for more details on the instruction.

7. To read the sensor data, click on command and then click the **Repeat** button. This reads the data continuously.

   Go to the **Chart** tab and observe the plot of the four values – PIR Motion Sensor raw data, high threshold, low threshold, and motion detection status – that are read from the PSoC 4100PS device. See Figure 9.

   Figure 9. PIR Motion Sensor Data on the Bridge Control Panel Chart



8. Move hand across the PIR Motion Sensor and observe that the 'Motion Detected' flag goes high for about 5s whenever a motion is detected.

Related Documents

Table 5 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 5. Related Documents

Application Notes		
AN79953	Getting Started with PSoC 4	Describes the PSoC 4100PS.
PSoC Creator Component Datasheets		
Scanning SAR ADC	Supports multiple channel hardware scan with single ended and differential input modes	
PGA	Support configurable gain of 2 to 32	
PVref	Generates configurable voltage references using internal bandgap voltage or supply voltage VDDA	
Opamp	Supports voltage follower mode and the Opamp mode with configurable power	
EZI2C Slave	Simplified I2C slave implementation	
Pins	Supports connection of hardware resources to physical pins	
Timer Counter	Supports Timer, Counter, PWM functions	
Device Documentation		
PSoC 4100PS Datasheet		
PSoC 4100PS Architecture Technical Reference Manual		
PSoC 4100PS Register Technical Reference Manual		
Development Kit (DVK) Documentation		
CY8CKIT-147 PSoC 4100PS Prototyping Kit		

Document History

Document Title: CE223618 – Interfacing the PSoC 4 with a PIR Motion Sensor

Document Number: 002-23618

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6153894	DIMA	4/25/2018	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 Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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