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.

CE223620 – Interfacing PSoC 4 with a Humidity Sensor

Objective

This code example demonstrates how to implement an analog front end (AFE) for a humidity sensor, using PSoC® 4.

Overview

This code example demonstrates how to measure humidity using the capacitance of a humidity sensor. The Cypress CapSense® Component is used to measure the capacitance of the humidity sensor. The CapSense Component provides best-in-class signal-to-noise ratio (SNR) (>5:1) for sensors with a capacitance of up to 200 pF. It also provides high-performance sensing across a variety of environmental factors, such as temperature and humidity.

The measured capacitance and the calculated humidity value are sent over I^2C 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)

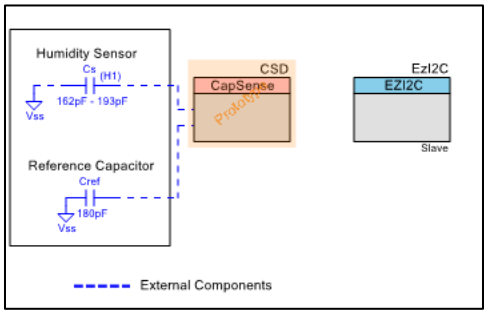
Associated Parts: PSoC 4100PS

Related Hardware: CY8CKIT-147 PSoC 4100PS Prototyping Kit

Design

Figure 1 shows the PSoC Creator schematic for interfacing a humidity sensor with the PSoC 4100PS.

Figure 1. Humidity Sensing Schematic



A humidity sensor (HPP801A031) and a reference capacitor (C_{ref}) of 180 pF are used to measure humidity. C_{ref} is used to calibrate the capacitance measurement. The capacitance of the humidity sensor is calculated using the following equation:

$$C_s = \frac{C_{ref} * (RawCount_{Cs} - RawCount_{Cos})}{RawCount_{Cref} - RawCount_{Cos}} + C_{OS}$$

where, C_s is the capacitance of the humidity sensor

C_{ref} is the capacitance of the reference capacitor

$RawCount_{Cs}$ is the raw count obtained from the CapSense Component when the humidity sensor is being measured

$RawCount_{Cref}$ is the raw count obtained from the CapSense Component when the reference capacitor is being measured

$RawCount_{Cos}$ is the raw count that corresponds to the offset capacitance (trace capacitance with sensor disconnected)

C_{OS} is the offset capacitance (trace capacitance with sensor disconnected from the pin)

Humidity is calculated from the measured capacitance, C_S, using the following equation:

$$\%RH = \frac{C_S - C_{nom}}{Sensitivity} + \%RH_{nom}$$

where, C_S is the capacitance of the humidity sensor

C_{nom} is the nominal capacitance of the humidity sensor

Sensitivity is the sensitivity of the humidity sensor (pF/%RH)

$\%RH_{nom}$ is the nominal humidity value

Note that the C_{nom}, Sensitivity, and $\%RH_{nom}$ are specific to the sensors and are provided by the sensor manufacturer. For the humidity sensor (HPP801A031) used in this code example, the C_{nom}, $\%RH_{nom}$, and Sensitivity are 180 pF, 55%RH, and 0.31 pF/%RH respectively.

The sensor data, such as sensor raw count, measured sensor capacitance, humidity, and reference capacitor raw count, are sent to a host PC using I^2C.

The capacitance of the humidity sensor used in this code example varies from 162 pF to 193 pF for 1% to 100% Relative Humidity (RH). The calibration capacitor is selected closer to the upper range of the capacitance measurement to reduce the gain error in measurement. In this code example, the C_{ref} value is 180 pF.

Design Considerations

This design can be adapted to support other capacitance-based humidity sensors.

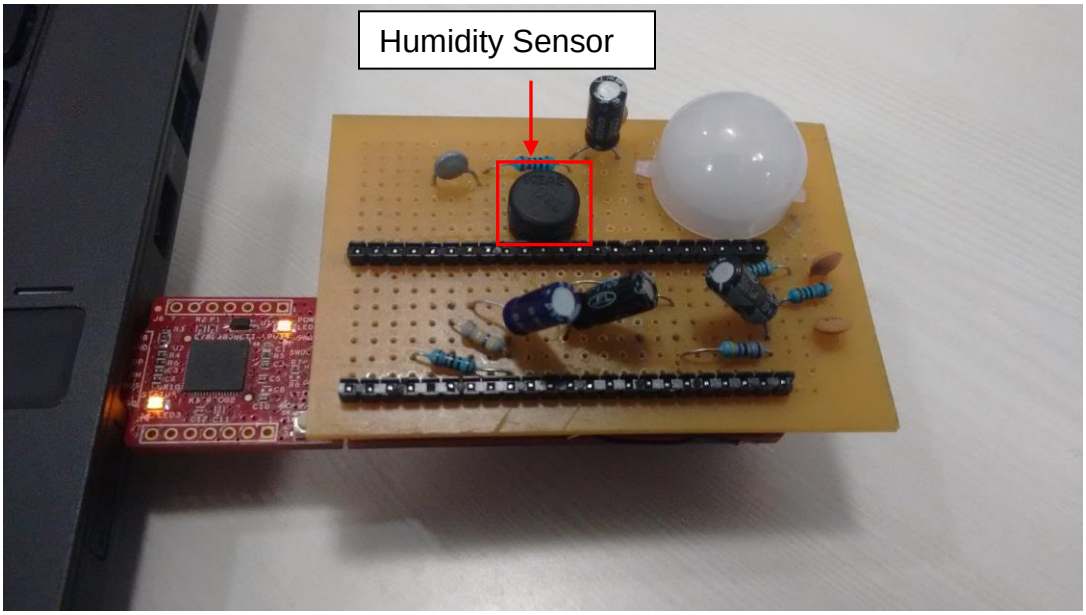
This code example is designed for the PSoC 4100PS Prototyping kit. The design is easily portable to other kits and PCBs, typically by just changing the sensor pin assignments.

Hardware Setup

This code example uses the humidity sensor (HPP801A031) and a reference capacitor (180 pF) mounted on a general-purpose PCB (used as customized shield hardware) to make the connection shown in Figure 1.

Stack up the custom shield and CY8CKIT-147 PSoC 4100PS Prototyping Kit and connect it to your computer's USB port as Figure 2 shows.

Figure 2. Hardware Connection



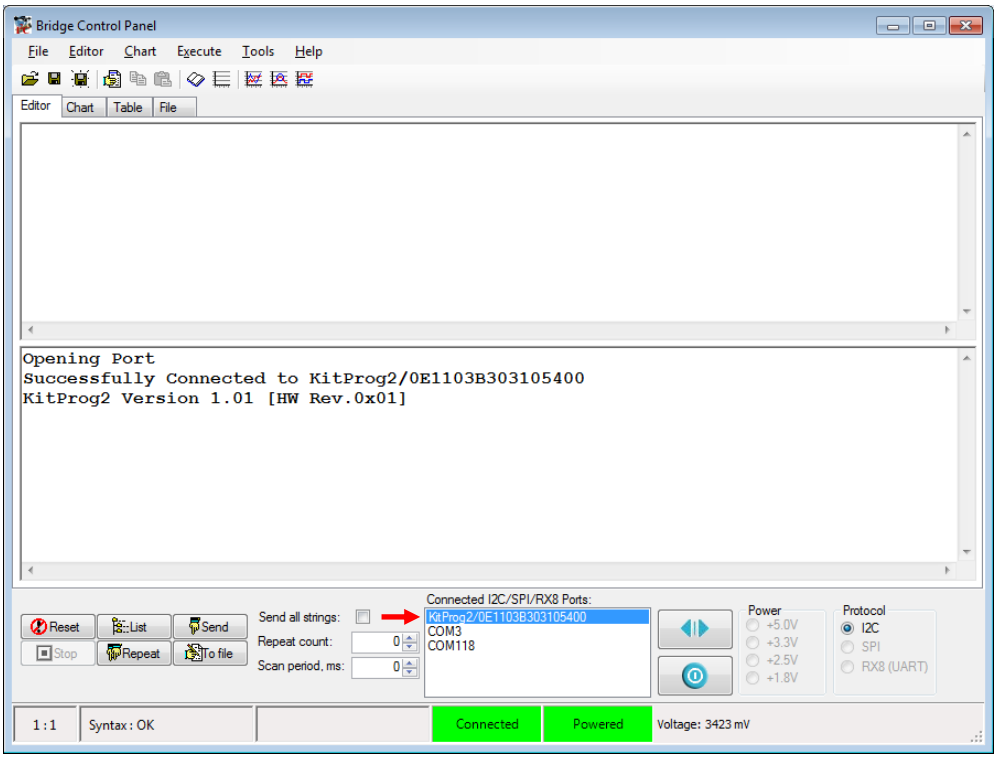
Software Setup

This section describes how to set up the BCP software to view sensor data sent over I²C.

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 3). Note that the PSoC 4100PS Prototyping Kit must be connected to the USB port of your computer.

Figure 3. Bridge Control Panel

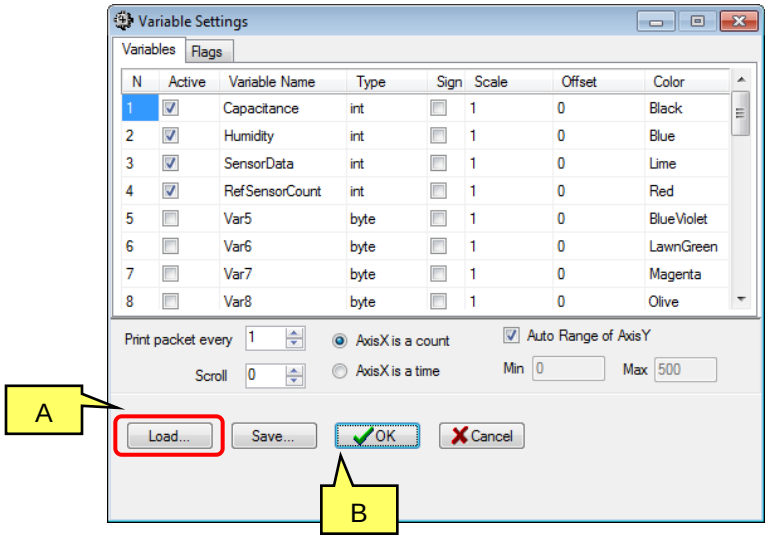


3. Go to **Tools** > **Protocol Configuration**, and navigate to the **I2C** tab, and set the **I2C speed** to '100 kHz'. Click **OK**.

4. Navigate to **Chart** > **Variable Settings**, and do the following;
   A. Load the *CE223620_Humidity_Sensing.ini* file from the following path: `CE223620_Humidity_Sensing\CE223620_Humidity_Sensing.cydsn\BCP Command\`.
   B. Click **OK** as Figure 4 shows.

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

Figure 4. Variable Settings in Bridge Control Panel Software



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

Components

Table 1 lists the PSoC Creator Components used in this example as well as the hardware resources used by each.

Table 1. List of PSoC Creator Components

Component	Instance Name	Version	Hardware Resources
CapSense	CSD	V5.10	CapSense Sigma-Delta (CSD)
EZI2C Slave (SCB mode)	EZI2C	V4.0	Serial Communication Block (SCB)

Parameter Settings

Table 2 lists the non-default settings of all the components used in the design.

Table 2. Components Parameters

Component Instance Name	Settings (Non-Default)
CSD	**Basic tab:** • Add one button and change the Sensing element(s) to 2. • CSD Tuning mode: Manual tuning **Advanced tab > General:** • Enable IIR filter (First order): Enable **Advanced tab > CSD Settings:** • Sense clock source: Direct **Advanced tab > Widget Details:** • Button0(CSD) > Scan resolution: 16 bits • Button0(CSD) > Sense clock frequency (kHz): 187.5
EzI2C	–

Note: EzI2C pins are embedded within the Component.

Design-Wide Resources

Table 3 lists the physical pins used.

Table 3. Pin Names and Locations

Pin Name	Location
EzI2C:SCL	P3[6]
EzI2C:SDA	P3[7]
CSD:Cmod	P5[0]
CSD:Sns [0] (Humidity Sensor)	P1[5]
CSD: Sns [1] (RefSensor)	P1[7]

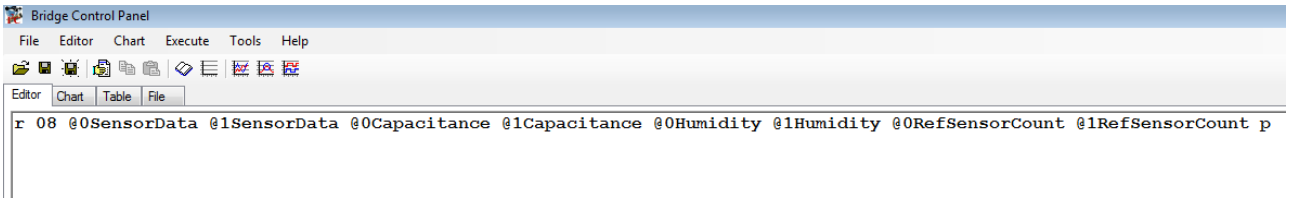
Operation

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 CE223620_Humidity_Sensing**.
3. Connect the PSoC 4100PS Prototyping Kit to your computer's USB port, as described in Hardware Setup.
4. Program the PSoC 4100PS device; select **Debug** > **Program**.
5. Configure the BCP software as described in Software Setup.
6. Select **File** > **Open File**. Open the *CE223620_Humidity_Sensing.iic* file from the following path: *CE223620_Humidity_Sensing\CE223620_Humidity_Sensing.cydsn\BCP Command*

   This file contains the read command to be executed from BCP. The command appears on the panel, as Figure 5 shows.

Figure 5. Read Command in the Bridge Control Panel



7. Click on the read command on the **Editor** tab in BCP and then click the **Repeat** button to read the sensor data continuously. Go to the **Chart** tab and observe the sensor raw count by selecting only the *SensorData* variable.
8. Disconnect the humidity sensor from PSoC. This is required to get the offset count that corresponds to the parasitic capacitance of the trace that connects humidity sensor to PSoC 4100PS.
9. Observe the instantaneous sensor raw count and calculate the average value of sensor raw count using the following equation.

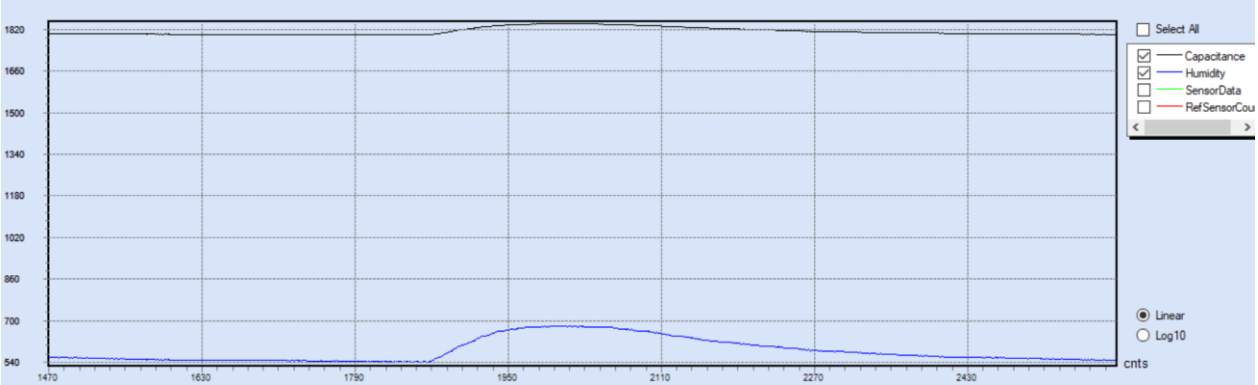
$$\text{Average raw count} = \text{Maximum raw count} + Minimum\ raw\ count/2$$

10. Assign the average raw count calculated in step 9 to the macro `OFFSETCOUNT` in the *main.c* file.
11. Reconnect the humidity sensor to PSoC.
12. Build the project by selecting **Build** > **Build CE223620_Humidity_Sensing**.
13. Program the PSoC 4100PS device by selecting **Debug** > **Program**.

    Note: Disconnect **KitProg2/<serial number>** under **Connected I2C/SPI/RX8 Ports** in BCP to program the device.
14. In the BCP, click the **Repeat** button again and observe the plot of the four values – sensor capacitance, humidity, sensor raw count, and reference capacitor raw count – that are read from the PSoC 4100PS device. See Figure 6.

Note that the sensor capacitance and humidity values are scaled by 10 and displayed on the BCP.

Figure 6. Bridge Control Panel Chart



15. Blow air gently on the humidity sensor and observe that the humidity and sensor capacitance changes on the BCP.

Related Documents

Application Notes		
AN79953	Getting Started with PSoC 4	Describes the PSoC 4100PS.
PSoC Creator Component Datasheets		
CapSense	Supports Capacitive touch sensing applications	
EZI2C Slave	Simplified I2C slave implementation	
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: CE223620 - Interfacing PSoC 4 with a Humidity Sensor

Document Number: 002-23620

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6153495	DIMA	07/10/2018	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

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.