

Objective

This code example demonstrates how to use the DelSig ADC to take samples of an analog input and send those samples to a PC application over RS-232 via the UART component.

Overview

This code example implements a simple data collection system using the DelSig ADC and the UART component. The ADC continuously samples an analog input. The resulting samples can be sent to a PC over a UART connection a single sample at a time or continuously. Emulated data, which is just an incrementing number, can also be sent over the UART connection to test the communication. The USB UART in the KitProg is used to create an RS-232 connection to a terminal program on a PC. The terminal program is used to send commands to get the ADC sample data and read the resulting responses.

PSoC Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right PSoC device for your design, and quickly and effectively integrate the device into your design. For a comprehensive list of resources, see [KBA86521](#), [How to Design with PSoC 3](#), [PSoC 4](#), and [PSoC 5LP](#). The following is an abbreviated list for PSoC 3 and PSoC 5LP:

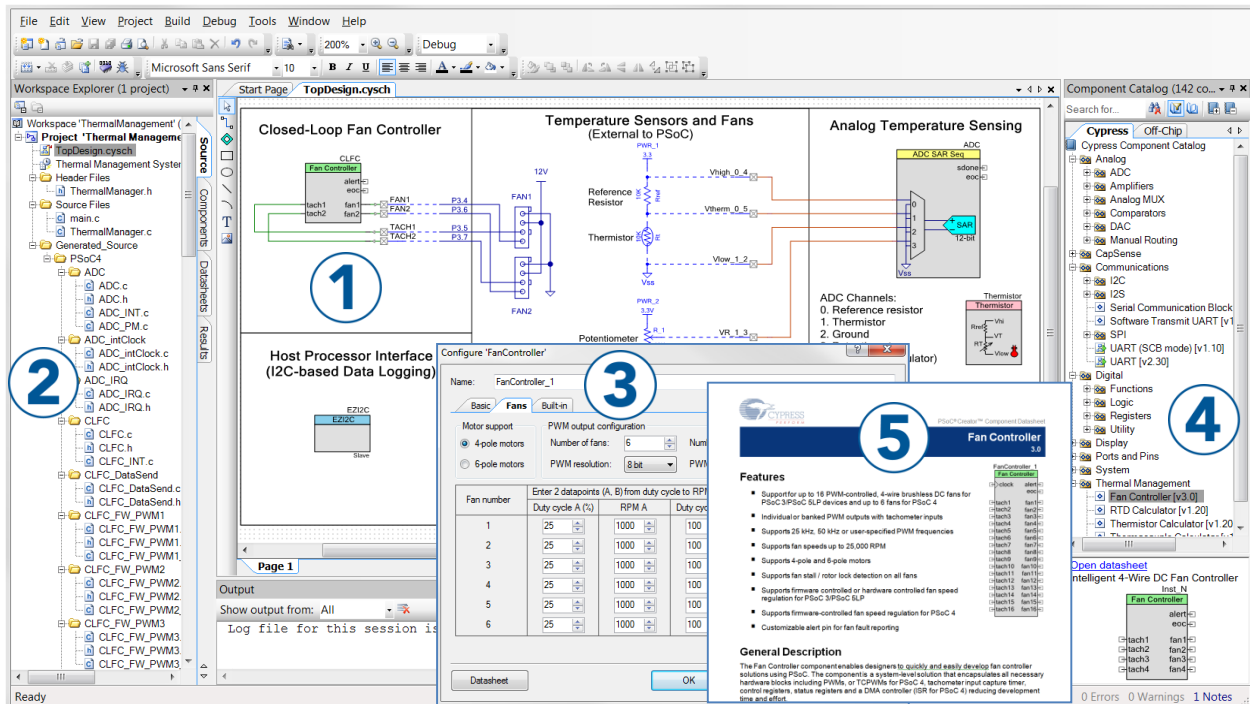
- **Overview: PSoC Portfolio, PSoC Roadmap**
- **Product Selectors: PSoC 1, PSoC 3, PSoC 4, or PSoC 5LP.** In addition, [PSoC Creator](#) includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for the [PSoC 5LP](#) device families
- **CapSense Design Guide:** Learn how to design capacitive touch-sensing applications with the PSoC 5LP family of devices.
- **Application Notes and Code Examples:** Cover a broad range of topics, from basic to advanced level. Many of the application notes include code examples.
- **Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers in each PSoC 5LP device family.
- **Development Kits:**
 - [CY8CKIT-001](#) is a common development platform for all PSoC family devices.
 - [CY8CKIT-050](#) is a development platform targeted at analog intensive designs for PSoC 5LP.
 - [CY8CKIT-030](#) is a development platform targeted at analog intensive designs for PSoC 3.
 - [CY8CKIT-059](#) is a rapid prototyping kit for PSoC 5LP.
- The [MiniProg3](#) device provides an interface for flash programming and debug.

PSoC Creator

[PSoC Creator](#) is a free Windows-based Integrated Design Environment (IDE). It enables concurrent hardware and firmware design of systems based on PSoC 3, PSoC 4, and PSoC 5LP. See [Figure 1](#) – with PSoC Creator, you can:

1. Drag and drop Components to build your hardware system design in the main design workspace
2. Codesign your application firmware with the PSoC hardware
3. Configure Components using configuration tools
4. Explore the library of 100+ Components
5. Review Component datasheets

Figure 1. PSoC Creator Features



Requirements

Tool: PSoC Creator 3.1 SP2 or later

Programming Language: C (GCC 4.8.4 or later)

Associated Parts: PSoC 3 and PSoC 5LP parts

Related Hardware: [CY8CKIT-059](#), [CY8CKIT-001](#), [CY8CKIT-050](#), [CY8CKIT-030](#)

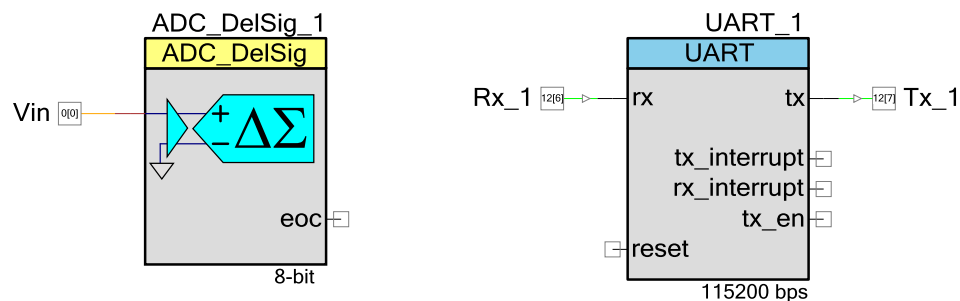
Design

ADC Settings:

Sampling Mode: Continuous
Resolution: 8 Bit
Samples per Second: 10000
Input Range: Vssa to 6.144 V (0.0 to 6*Vref)
Reference: Internal 1.024 V

UART Settings:

Baud Rate: 115200
Data Bits: 8
Parity: None
Stop Bits: 1
Flow Control: None



Test Setup and Procedure:

- 1) Plug CY8CKIT-059 into USB port of PC using the PCB USB connector.
- 2) Program the CE95277 project onto the CY8CKIT-059.
- 3) Connect positive terminal of ADC to a voltage source using P0[0].
- 4) The UART Rx and Tx signals are already connected to the KitProg via the target pins P12[6] and P12[7].
- 5) The KitProg automatically enumerates as a virtual COM port. You can find the virtual COM port number by going to Start -> Control Panel -> Device Manager in a Windows system. Connect to the COM port using a terminal emulator program. Once connected, use the command 'c' to get a single sample, 's' to stream continuous samples, or 'e' to get emulated samples. Use 'x' to cancel streaming of samples.

Figure 2 Code Example Creator Schematic

The code example uses the DelSig component to sample the analog input and the UART component to transmit the sampled data to the PC over RS-232.

All of the firmware for the code example is implemented in main.c. The firmware performs the following functions:

1. Starts the ADC and UART components.
2. Checks for ADC end of conversion. Stores latest result if conversion complete.
3. Checks for UART input.
 - 'C' or 'c' received: transmits the last sample via the UART.
 - 'S' or 's' received: continuously transmits samples as they are completed.
 - 'X' or 'x' received: stops continuously transmitting samples.
 - 'E' or 'e' received: transmits a dummy byte of data.

Hardware Setup

Connect a voltage source to P0[0] on the kit. If P0[0] is not available you can change the Vin pin in the design wide resources file (CE95277 ADC and UART.cywdr, Pins tab). If a voltage source is not available you can test the UART connection using the emulated data by sending 'e' or 'E' over the terminal connection.

This code example is targeted towards the CY8CKIT-059. If using this kit, then no external hardware connections are needed to allow UART communication. If using a different hardware platform you may need to connect the UART Tx (P12[7]) and Rx (P12[6]) to an RS-232 interface.

If you are not using the CY8CKIT-059 then you may also need to target a different PSoC device. To do so, right click the project in the Workspace Explorer and select Device Selector. Select the appropriate PSoC device for your hardware platform.

Software Setup

To interface with the UART on the PSoC device a terminal emulator application is needed. Hyperterm and Putty are examples of valid options. Open up a connection to the COM port number that the PC gives to the PSoC device when it enumerates. Make sure the settings in the software COM port match the settings for the UART component, detailed in the Parameter Settings section below.

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 or User Module	Hardware Resources
DelSig ADC	DelSig ADC
Full UART	UDBs (3 Datapath Cells, 58 macrocells, 2 Status Cells, 2 Control Cells, 2 interrupts)
Analog Pin (Vin)	P0[0]
Digital Input Pin (Rx_1)	P12[6]
Digital Output Pin (Tx_1)	P12[7]

Parameter Settings

Component	Non-default Parameter Settings
DelSig ADC	Sampling Mode: Continuous Resolution: 8 Bit Samples per Second: 10000 Input Range: Vssa to 6.144 V (0.0 to 6*Vref) Reference: Internal 1.024 V
UART	Baud Rate: 115200 Data Bits: 8 Parity: None Stop Bits: 1 Flow Control: None
Analog Pin (Vin)	High Impedance Analog
Digital Input Pin (Rx_1)	High Impedance Digital, Hardware Connection
Digital Output Pin (Tx_1)	Strong Drive, Hardware Connection

Design-Wide Resources

Figure 3 shows the pin selections for the code example.

Alias	Name /	Port	Pin	Lock
	Rx_1	P12[6]	20	<input checked="" type="checkbox"/>
	Tx_1	P12[7]	21	<input checked="" type="checkbox"/>
	Vin	P0[0] OpAmp2:vout	48	<input checked="" type="checkbox"/>

Figure 3 Pin Selection

Operation

1. Connect the board to the PC using the USB connector of the KitProg. On the CY8CKIT-059 this is the PCB-USB connector.
2. Program the example project to the board.
3. Connect an external voltage to the PSoC 5LP target device on P0[0]
4. Open a terminal emulator on the PC and connect to the virtual COM port, enumerated as a part of KitProg
5. Type character "C" or "c" in the terminal emulator to receive one sample of ADC data.
6. Type character "S" or "s" in the terminal emulator to receive continuous samples of ADC data.
7. Type character "X" or "x" in the terminal emulator to stop receiving ADC samples.
8. Type character "E" or "e" in the terminal emulator to receive one emulated sample of data. This data will increment by 1 each time an emulated sample is requested.

Upgrade Information

N/A

Related Documents

Table 2 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component datasheets.

Table 2. Related Documents

Application Notes		
AN84783	Accurate Measurement Using PSoC® 3 and PSoC 5LP Delta-Sigma ADCs	Shows how to increase accuracy of DelSig measurements.
AN61102	PSoC® 3 and PSoC 5LP - ADC Data Buffering Using DMA	Shows how to buffer data from the DelSig using DMA.
AN58304	PSoC® 3 and PSoC 5LP – Pin Selection for Analog Designs	Provides an overview of analog routing in PSoC 3/5LP and recommendations for pin selection.
AN58827	PSoC® 3 and PSoC 5LP Internal Analog Routing Considerations	Shows how to choose the best routes for analog sensitive designs.
AN68403	PSoC® 3 and PSoC 5LP Analog Signal Chain Calibration	Shows how to calibrate the analog signal chain in PSoC 3/5LP.
Code Examples		
CE95271	Delta Sigma ADC in Differential Mode with PSoC 3/5LP	
CE95273	Delta Sigma ADC in single-ended mode using DMA and VDAC with PSoC 3/5LP	
CE95299	Delta Sigma ADC using 16 multiplexed single-ended inputs with PSoC 3/5LP	
CE95302	Delta Sigma ADC and I2C Slave with PSoC 3/5LP	
CE95388	UART Receive with PSoC 3/4/5LP	
CE95389	UART Transmit with PSoC 3/4/5LP	
PSoC Creator Component Datasheets		
ADC DelSig	Details the use of the DelSig component.	
UART	Details use of the UART component	
Device Documentation		
PSoC 3 Datasheets	PSoC 3 Technical Reference Manuals	
PSoC 4 Datasheets	PSoC 4 Technical Reference Manuals	
PSoC 5LP Datasheets	PSoC 5LP Technical Reference Manuals	
Development Kit (DVK) Documentation		
PSoC 3 and PSoC 5LP Kits		
PSoC 4 Kits		

Document History

Document Title: Delta Sigma ADC in single-ended mode with PSoC 3/5LP - CE95277

Document Number: 001-95277

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**		KLMZ		New spec

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

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#)

Cypress Developer Community

[Community](#) | [Forums](#) | [Blogs](#) | [Video](#) | [Training](#)

Technical Support

cypress.com/go/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corp. 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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and/or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.