Amplifier with Dynamic Gain Switching Example Project

1.0

Features

- Opamp as non-inverting amplifier
- ADC is used in single ended mode
- UART display ADC results on HyperTerminal
- LED to indicate when the ADC input is outside the defined voltage window
- Analog Muxes to multiplex three inputs and their corresponding gains
- Debouncer to detect valid switch press

General Description

This example project is also a PSoC Creator starter design for PSoC 4 device. It demonstrates unique and flexible analog routing capability of PSoC 4 to change the Opamp input and feedback network on the fly. In this project, every input have corresponding dedicated feedback networks and hence different gains. The user can change the active channel by pressing a switch.

Development Kit Configuration

The following configuration instructions provide a guideline to test this design. For simplicity, the instructions describe the stepwise process to be followed when testing this design with the PSoC 4 Pioneer Kit (CY8CKIT-042). Please refer to the "Schematic and Pin mapping" section at the end of this document for details.

1. Set jumper J9 to 5.0V position.
2. Connect three input signals to P2[0], P2[1], P2[2].
3. Connect all the external resistors as shown in the top design schematic.
4. Connect P0[5] to pin P12[6] on header J8.
5. Connect USB cable to the PSoC 4 Pioneer Kit DVK and PC with HyperTerminal program.

Project Configuration

This example project consists of ADC SAR Seq, Opamp, AMuxSeq, UART and Debouncer components. The top design schematic is shown in Figure 1. The Opamp is used to amplify the input signal; selection of input channel is done using Input_AMux and selection of feedback network done using Gain_AMux. UART is used to send ADC results to HyperTerminal. Debouncer is used to remove glitches from the input switch. The SAR ADC converts the analog output of the Opamp, into digital values. The ADC also generates an interrupt when its input is outside the defined voltage window (250mV – 750mV). The LED is turned on when ADC generates this interrupt. The UART component sends the ADC output of active channel along with channel number to HyperTerminal.

AMPLIFIER WITH DYNAMIC GAIN SWITCHING

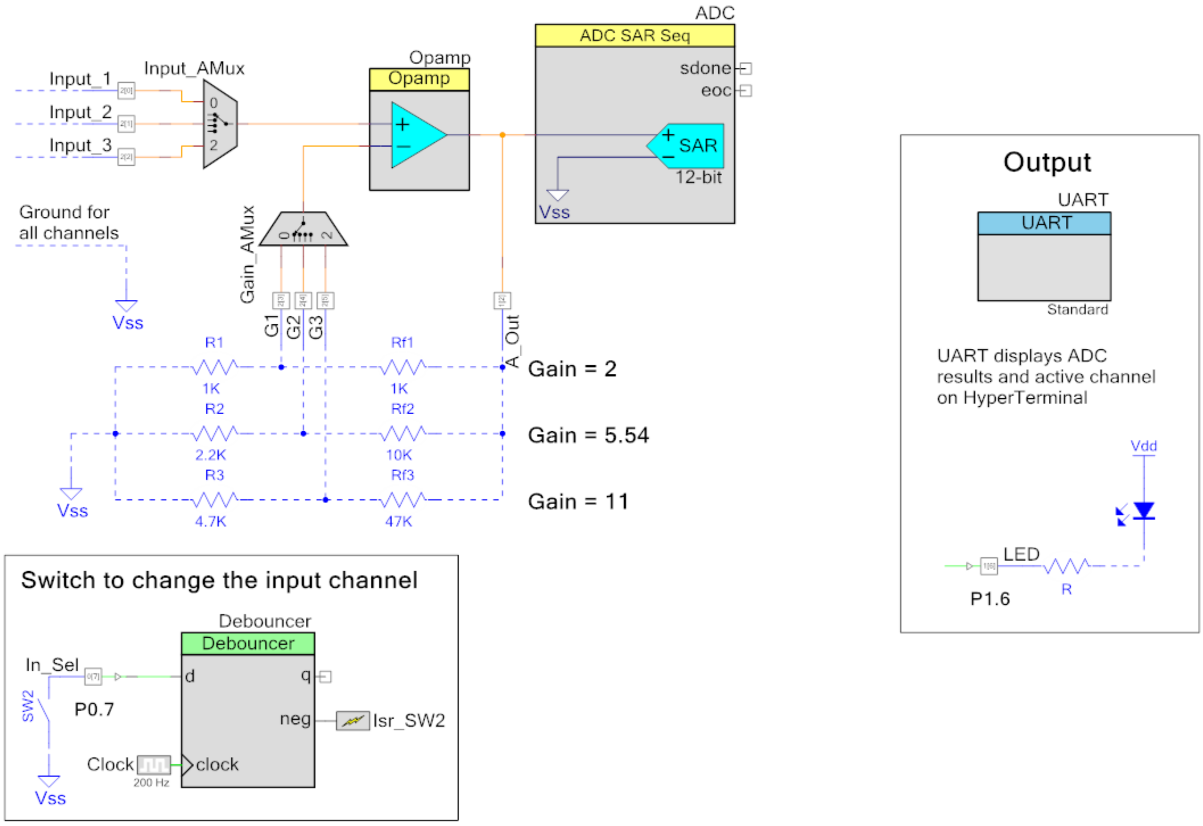

Figure 1. Top Design Schematic

The Opamp is configured in high stability, high power, 10mA output current mode. The ADC is configured in single ended mode. The ADC averages 256 consecutive samples to produce the final result. The ADC component configuration is shown in Figure 2. Switch SW2 and pin In_Sel is used to change the analog mux channels, Input_AMux selects the input channel and Gain_AMux selects the corresponding gain.

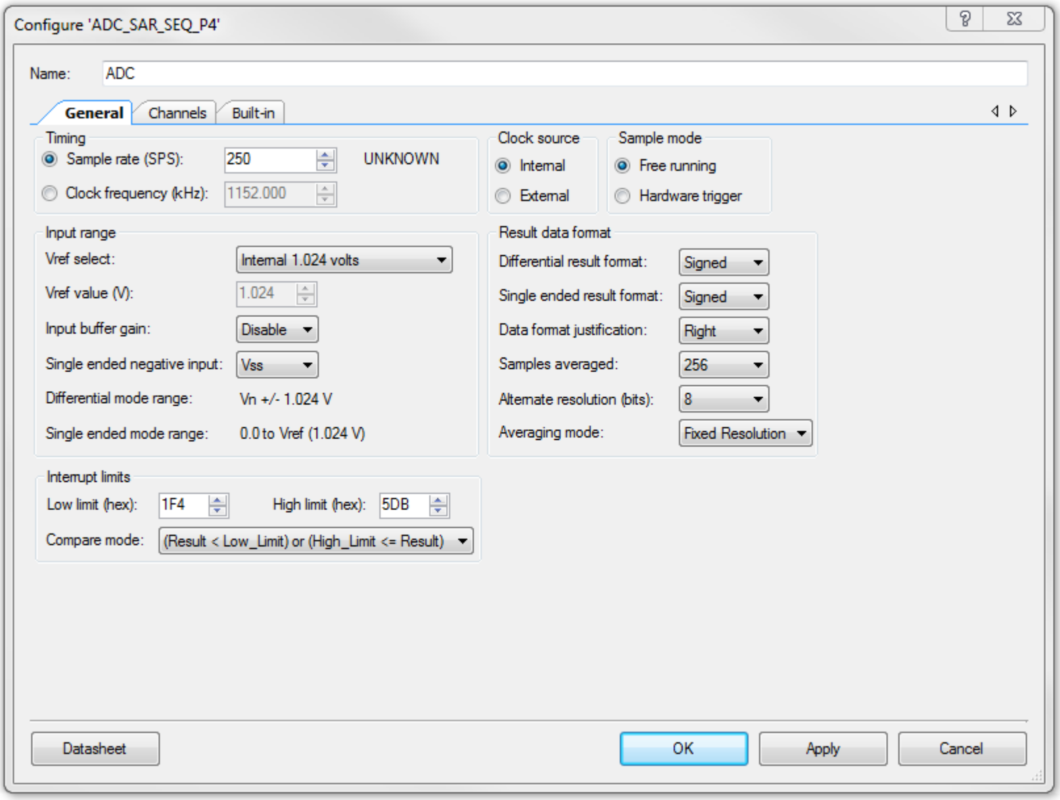

Figure 2 ADC Configuration Window

Project Description

In the main function all components are started, both the analog muxes are initialized to select channel zero and ADC conversion is started. The for loop in the main.c waits for the ADC to finish conversion. When the ADC result is available, it is sent through UART to HyperTerminal. The ADC continuously generates an interrupt when its input is outside the defined voltage window (250mV –750mV). This interrupt is used to control an LED. This LED is turned ON when ADC input is outside the window.

Expected Results

The analog input channel should change when switch SW2 is pressed. HyperTerminal displays the active channel and its input voltage. LED will turn on when ADC result is outside the voltage window (250mV – 750mV).

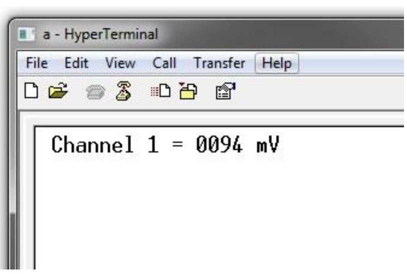

Figure 3. Result

Schematic and Pin Mapping

Schematic

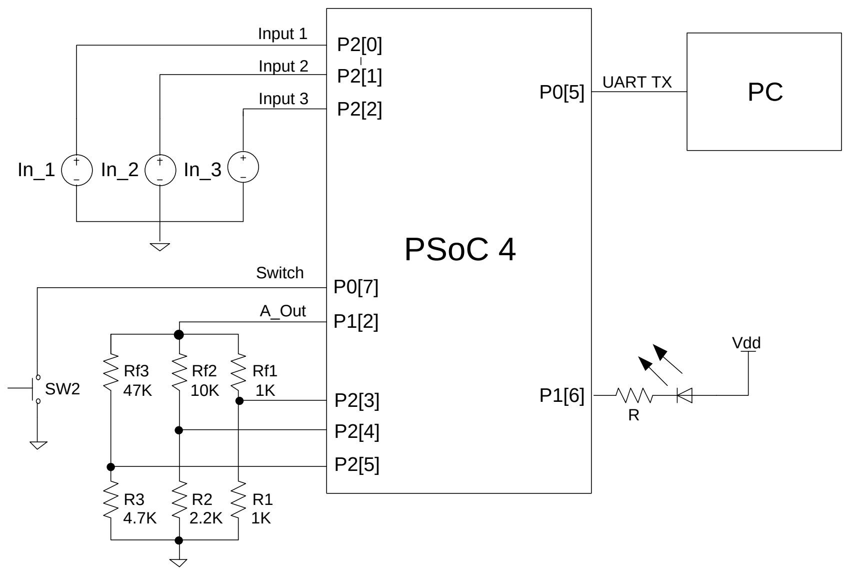

Figure 4. Connection Schematic

Pin Mapping Table:

Refer Table 1 to validate this starter design on any other PSoC 4 development kit:

S.No.	I/O Signals	Project Pin Assignments	Other Possible Connections
1	A_Out	P1[2]	Fixed P1[2] or P1[3]
2	UART	P0[5]	Any GPIO except P1[2]
3	LED	P1[6]	Any GPIO except P1[2]
4	Input Signals	P2[2:0]	Any GPIO except P1[2]
5	Opamp feedback	P2[3], P2[4], P2[5]	Any GPIO except P1[2]
6	Switch	P0[7]	Any GPIO

Note: If LED is active HIGH, then please replace “LOW” with “HIGH” to turn the LED ON and vice-versa. The Section of the code that needs to be changed in “main.c” is mentioned below:

```
/* Turn ON the LED when input is outside the window (250mV - 750mV) */
    LED_Write(LOW);

/* Turn OFF the LED when input is within the defined window (250mV - 750mV) */
    LED_Write(HIGH)
```

Using UART to communicate with PC Host

This example project communicates with a PC host using UART. A HyperTerminal program is required in the PC to communicate with PSoC 4. If you don't have a HyperTerminal program installed, download and install any serial port communication program. Free wares such as HyperTerminal, Bray's Terminal etc. are available on web.

Follow these steps to communicate with PC host.

1. Connect the USB cable between the PC and PSoC 4 Pioneer Kit.
2. Open the device manager program in your PC, find the COM port in which the PSoC 4 is connected, and note the port number.
3. Open the HyperTerminal program and select the COM port in which the PSoC 4 is connected.
4. Configure Baud rate, Parity, Stop bits and Flow control information in the HyperTerminal configuration window. These settings should match the configuration of the PSoC Creator UART component in the project
5. Start communicating with the device as explained in the project description.

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, 2013. 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.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

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 agreemen