

**Project Name:** Example\_TRIADC

**Programming Language:** C

**Associated Part Families:** CY8C24x23, CY8C27x43,  
CY8C29x66, CY8C24x94, CY8C21x34

**Software Version:** PSoC® Designer™ 5.2

**Related Hardware :** CY3210 PSoCEval1 Board

## Objective

This project demonstrates the operation of the TriADC User Module.

## Overview

In this project, three analog signals connected to analog input ports are measured using a TriADC user module, converted into digital data and displayed on an LCD display.

## User Module List and Placement

The following table lists the user modules used in this project and the hardware resources occupied by each user module.

User Module	Placement
TriADC	DBB00(Counter1) DBB01(PWM16_LSB) DCB02(PWM16_MSB) DCB03(Counter2) DBB10(Counter3) ASC10(Integrator1) ASD11(Integrator2) ASC12(Integrator3)
LCD	P2[0] to P2[6]
PGAtADC1	ACB00
PGAtADC2	ACB01
PGAtADC3	ACB02

## User Module Parameter Settings

The following tables show the user module parameter settings for each of the user modules used in the project.

TriADC		
Parameter	Value	Comments
ADC Input1	ACB00	Input to Channel1 comes from PGA in ACB00. Input to ACB00 is from P0[1].
ADC Input2	ACB01	Input to Channel2 comes from PGA in ACB01. Input to ACB01 is from P0[0].
ADC Input3	ACB02	Input to Channel3 comes from PGA in ACB02. Input to ACB02 is from P0[3].
Clock Phase1	Norm	Phi1 >Acquire charge Phi2 >Transfer charge
Clock Phase2	Norm	Phi1 >Acquire charge Phi2 >Transfer charge
Clock Phase3	Norm	Phi1 >Acquire charge Phi2 >Transfer charge
Clock	VC1	1.5 MHz column clock is applied to TriADC module.
ADC Resolution	13 Bit	Sampled data is represented using 13 bits.
Calc Time	65	Time for CPU to calculate intermediate integration result is 65 data clock.
Data Format	Signed	12 bits are used to represent magnitude and 1 bit is for sign.

### Notes

- The clock to the digital block of ADCINC should be equal to the column clock of analog block in which the integrator part of ADCINC has been implemented.
- When the input to the ADC is from a CT Block or from direct port pin, set this value to Norm. When the input is from another SC block, set this value to Swapped.
- More details on the ClockPhase and column clock can be found in the article [PSoC® 1 ADCs – The Five Golden Rules](#).

PGAtoADC1		
Parameter	Value	Comments
Gain	1.000	PGA acts as a buffer between TriADC input and P0 [1].
Input	AnalogColumn_InputMUX_0	Output of AnalogColumn_InputMUX_0 is connected to the input of PGA.
Reference	VSS	Reference to the PGA is set to VSS
AnalogBus	Disable	Output of PGA is not connected to analog output buffer.

PGAtoADC2		
Parameter	Value	Comments
Gain	1.000	PGA acts as a buffer between TriADC input and P0[0].
Input	AnalogColumn_InputSelect_1	Output of AnalogColumn_InputMUX_1 is connected to the input of PGA.
Reference	VSS	Reference to the PGA is set to VSS
AnalogBus	Disable	Output of PGA is not connected to Analog output Buffer.

PGAtoADC3		
Parameter	Value	Comments
Gain	1.000	PGA acts as a buffer between ADC input and P0 [3].
Input	AnalogColumn_Input Select_2	Output of AnalogColumn_InputMUX_2 is connected to the input of PGA.
Reference	VSS	Reference to the PGA is set to VSS
AnalogBus	Disable	Output of PGA is not connected to analog output buffer.

LCD		
Parameter	Value	Comments
LCDPort	Port_2	Port 2 is used to send data to LCD
BarGraph	Disable	Bargraph is not used in this code example

## Global Resources

Important Global Resources		
Parameter	Value	Comments
Power Setting (VCC / Sys.Clock Frequency)	5.0 V / 24 MHz	Sys.Clock=24MHz.Clock source accuracy is trimmed for 5 V power setting.
CPU Clock	SysClk/2	Sets the CPU frequency to 12 MHz
VC1	16	Divide 24 MHz system clock by 16 to get a 1.5MHz clock (Column clock for TriADC).
Analog Power	SC On / Ref High	Set the maximum operating power of SC blocks to Ref High.
Ref Mux	(Vdd/2)+/-(Vdd/2)	Ref High = 5 V Ref Low = 0 V AGND = 2.5 V

### Notes

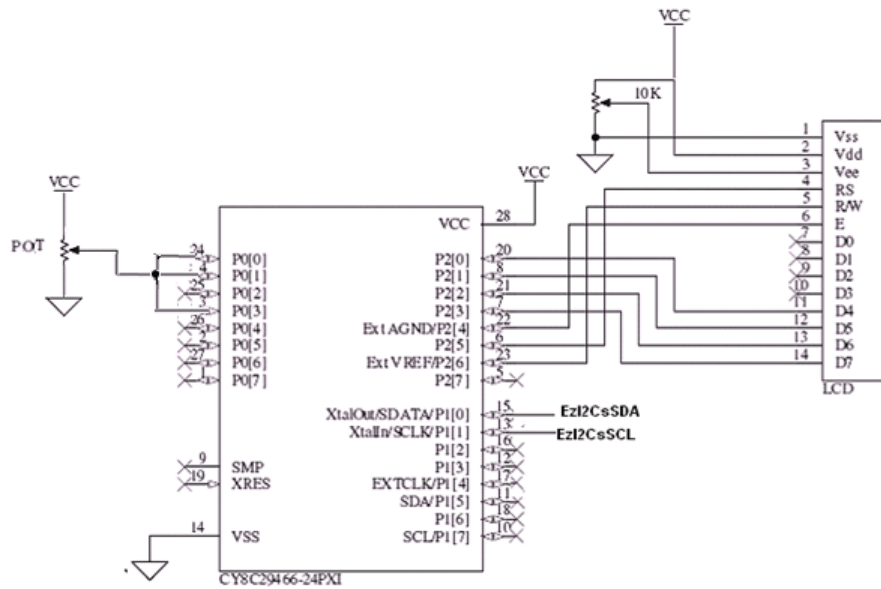
- The analog reference power should be set to the maximum power used by any analog resource. More details on this setting can be found in the article [PSoC 1 ADCs – The Five Golden Rules](#)
- All other global resources are left at their default, as they are not specific to this project.

## Pin Configuration

PinOut			
Pin	Select	Drive	Direction
Port 0_0	Analog Input	High Z Analog	Input
Port 0_1	Analog Input	High Z Analog	Input
Port0_3	Analog Input	High Z Analog	Input
Port2_0-Port2_6	StdCPU	Strong	Output

## Hardware Connections

The schematic of the project is as follows:



The input ports are connected to a potentiometer that generates 0-Vcc. The LCD display is connected to Port2.

The project can be tested using CY3210 PSoC Eval1 board. To test the project the following connections must be made.

- Connect P0[0] of J6 to VR of J5.
- Connect P0[1] of J6 to VR of J5.
- Connect P0[3] of J6 to VR of J5.
- Connect LCD module to J9.
- Connect I2C-USB Bridge to the ISSP header J11.

## Operation

On execution of the program, the hardware settings from the device configuration are loaded into the device and *main.c* is executed. The 24 MHz system clock is divided by 16 (VC1) to generate a 1.5 MHz clock, which is provided as column clock to TriADC user module.

Following operations are performed in *main.c*.

1. Enable global interrupts.
2. Start all PGAs in High power mode.
3. Start LCD module.
4. Start TriADC in high power mode.
5. Run the TriADC in continuous sampling mode.
6. Check if ADC Data is available by calling the TRIADC\_flsDataAvailable() function.
7. If the ADC data is ready, perform the following operations.
  - ❑ Get the integer form of TriADC output from channel 1 using TRIADC\_iGetData1ClearFlag() API and display it on the LCD in Hex form.
  - ❑ Get the integer form of TriADC output from channel 2 using TRIADC\_iGetData2ClearFlag() API and display it on the LCD in Hex form.
  - ❑ Get the integer form of TriADC output from channel 3 using TRIADC\_iGetData3ClearFlag() API and display it on the LCD in Hex form.
8. Go back to Step 6.

## Testing the Project

To test the project using CY3210 PSoCEval1 board, perform the following steps.

- Make the connections as shown in [Hardware Connections](#) section.
- Vary the potentiometer and observe the 3 channel ADC outputs being displayed on the LCD.

PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip" and PSoC Designer are trademarks 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  
<http://www.cypress.com/>

© Cypress Semiconductor Corporation, 2010-2011. 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.