Measure and Display 0 to 5 V on a LCD Using PSoC® 1

Example Name: Example_Measure_5 V
Programming Language: C
Associated Part Families: CY8C29/27/24/22xxx, CY8C23x33, CY8CLED04/08/16, CY8CNP102, CY8C28x45
Related Hardware: CY3210 PSoCEval1
Software Version: PSoC® Designer™ 5.2

Objective

This example is designed to measure 0 to 5V input using ADCINCVR available in PSoC® 1 and display it on a LCD.

Overview

A 0 to 5 V input voltage applied to P0 [1] is measured using a variable ADC of PSoC 1 ADCINCVR configured for 12 bits of resolution. The ADC value is converted into a floating point value that represents the input voltage and then displayed on the LCD.

User Module List and Placement

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

User Module	Placement
PGA	ACB00
ADC (ADCINCVR)	ASC10
LCD	Port_2

User Module Parameter Settings

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

PGA Module		
Parameter	**Value**	**Comments**
Gain	1.00	The PGA is set as a unity gain buffer. PGA is unity gain is used to make sure that low input impedance of ADC (because ADC is switched capacitor implementation) does not over load input source.
Input	AnalogColumn_InputMUX_0	Input comes from P0 [1] via AnalogColumn_InputMUX_0.
Reference	V_{SS}	Reference to PGA is set to V_{SS}. It can be set to any other value as well as gain of PGA is 1 and reference value does not matter.
AnalogBus	Disable	As the output of the PGA is used internally, the analog bus is disabled.

Note
When measuring unipolar signals referenced to V_{SS}, the reference for the PGA should be set to V_{SS}. When measuring bipolar signals around AGND, the reference should be set to AGND.

LCD Module		
Parameter	**Value**	**Comments**
LCD Port	Port_2	Port 2 is used for LCD.
BarGraph	Disable	BarGraph is not used in this code example and is disabled.

ADC Module		
Parameter	**Value**	**Comments**
Input	ACB00	Input to ADC comes from the PGA in ACB00.
Clock Phase	Norm	See note 2 after this table.
Clock	VC1	This is the clock to the ADC. The sample rate of the ADC is decided by the clock. Make sure that the column clock to the ADC is the same as this setting. See note 1 after this table
ADCResolution	12	ADC is configured for 12 bit resolution.
CalcTime	10	See the note 3 after this table.
DataFormat	Unsigned	Get unsigned value from ADC to keep single ended input from 0 to 5V.

Notes

1. When the clock to the ADC is set to VC1, the column clock for Analog Column 0 should be set to VC1.
2. When the input to the ADC comes from another SC block, the clock phase should be set to Swapped. This is because the output of the SC block is valid during Phase-2 and is zero during Phase-1. If the ADC clock is set to Norm, the ADC samples the output of the SC block during Phase-1 and always reads Zero. If the input to the ADC comes from a continuous signal source such as CT block, Analog Bus, or a direct port pin, the ClockPhase may be set to “Norm” or “Swapped”.
3. The CalcTime is the number of data clocks the ADC holds the result after conversion. The CPU should read the result within this duration. The minimum value for CalcTime is calculated from the data clock, CPU clock, and the number of CPU clocks required to read the result inside the ADC’s ISR. If your application has other interrupts running, there is a chance that the CPU is busy in other ISRs when the ADC has completed the conversion. In this case, add the worst case CPU clocks that may be consumed by other ISRs in the calculation of CalcTime parameter. This ensures that the ADC result is not corrupted by the next conversion in case the CPU cannot read the ADC result at the end of conversion. Refer the user module data sheet for the formula to calculate CalcTime.

Global Resources

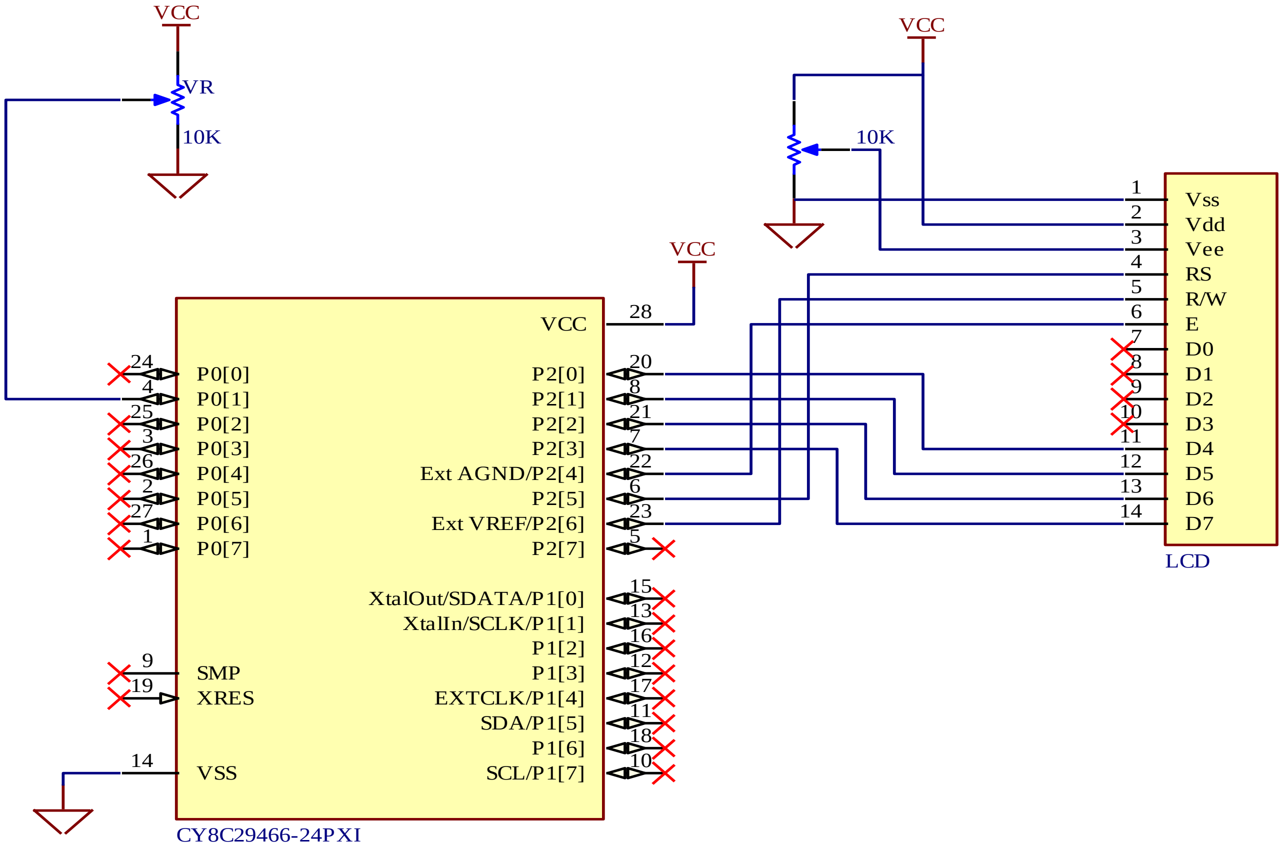
Important Global Resources		
Parameter	**Value**	**Comments**
Power Setting	5.0 V/24 MHz	Circuit must operate at 5V Vdd and IMO must generate 24 MHz.
CPU Clock	SysClk/2	CPU runs at 12 MHz.
VC1=SysClk/N	6	24/6 = 4 Mhz. This clock is used as the column clock for SC blocks and clock to TMR and CNT blocks of ADC.
SysClk Source	Internal24_MHz	Clock source is kept as IMO
Analog Power	SC On/Ref High	SC blocks are kept on and power is set to High.
Ref Mux	$(V_{DD}/2)$ +/- $(V_{DD}/2)$	The analog reference is set to measure from V_{SS} to V_{DD}.

Notes

- The Analog Power parameter should be set to SC On/Ref High. If the power is set to SC Off, all SC blocks are disabled and the ADC does not work.
- The global resources relevant to this project operation are described here. Other global resources are left at their default value or configured as required.

Hardware Connections

The schematic diagram for the code example is shown as follows.



The LCD is connected to Port2 of the PSoC. The analog input is connected to P0[1]. The code example is tested using the CY3210 PSoCEval1 board.

- Connect LCD to J9 on the CY3210 board.
- Connect VR pin of J5 to P0[1] on J6.

Operation

On reset, device configuration is loaded; then the code in *main.c* is executed.

The operations performed by the firmware are as follows:

- PGA is started in HIGHPOWER mode.
- LCD is started.
- On the LCD, at location 0,0 “MEASURED VOLTAGE” is printed.
- Global interrupts are enabled.
- ADC is started in HIGHPOWER mode and the conversion is started in continuous sampling mode.
- The scale factor to convert the ADC counts to voltage is calculated and stored in variable fScaleFactor. The scale factor is calculated as Volts/Count. The input voltage range is 5 V and the number of ADC counts is 4096 (12 bits). Therefore, the scale factor is 5 V/4096.

- In an infinite loop, the following operations are performed:
  - ❑ Wait until ADC data is available.
  - ❑ Read ADC data into variable iData and clear the ADC flag.
  - ❑ Multiply the ADC result by fScaleFactor to get the value of input voltage. In the multiplication, the variable iData is typecast into a float.
  - ❑ Convert this float value in ASCII string using function ftoa. The function returns a pointer to the string that holds the converted ASCII value. To use this function, *stdlib.h* header file is included in code example.
  - ❑ Display this ASCII string on LCD at location row 1 column 0 followed by string “V”.

To test the code example, vary the input voltage on P0[1] and observe the value displayed on the LCD.

Note When varying the input voltage from 0 to 5 V, observe that the display does not vary exactly from 0 V to 5 V. Instead, the display varies from a few tens of millivolts above zero to a few tens of millivolts below 5 V. This is because the output of the PGA is not rail to rail and is in the range of about (V_{SS} + 50 mV) to (V_{DD} – 50 mV). This is an expected behavior.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/

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.