Measure and Display 0 to 4 V on LCD Using ADC10

Project Name: Measure and Display 0 to 4 V on LCD Using ADC10
Programming Language: C
Associated Part Families: CY8C21x23, CY8C21x34, CY8C21x45, CY8C22x45, CYWUSB6953, CY8C28x45
Software Version: PSoC® Designer™ 5.2
Related Hardware: Customize
Related Document: SSlopeCalc.xls

Objective

This project demonstrates to measure a 0 to 4 V input using ADC10 available in limited analog families of PSoC® 1 (CY8C21x23, CY8C21x34,CY8C21x45, CY8C22x45, CYWUSB6953, CY8C28x45) and display it on an LCD.

Overview

An input voltage signal of 0 to 4 V applied to P0[7] is measured using an ADC10 with 10-bit resolution. The ADC value is converted into a floating-point value that represents the input voltage and is then displayed on the LCD.

User Module List and Placement

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

User Module	Placement
ADC (ADC10)	ACE00, ASE10, DBB00
LCD	Software

User Module Parameter Settings

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

LCD		
Parameter	**Value**	**Comments**
LCD Port	Port_2	Port 2 is used for LCD in CY3210 PSoCEVAL
BarGraph	Disable	Bargraph is disabled as its not required for this example

ADC		
Parameter	**Value**	**Comments**
ADC Input	AnalogColumn_InputMUX_0	Input to ADC comes from Analog Column Input MUX. See note 2 mentioned after this column
Data Clock	VC1	See the note 2 mentioned after this table
Current Setting	Low Current	This decides the current source setting to charge and discharge the input capacitor; low current is 37.5 nA
PWM_High	4VC3 Periods	This decides the measurement period of ADC
PWM_Low	1VC3 Period	This decides the calculation period of ADC

Notes

1. The clock to the ADC is set to VC1; therefore, the column clock for Analog Column 0 should be set to VC1.

2. The input to the ADC comes from Analog Column Input MUX. Select P0.7 as output of Analog Column Input MUX in Chip View of the project. Data clock is used to feed into VC3. The possible values of the Data Clock parameter are limited to VC1 and VC2 or SysClk. The selected clock source should lie between 240 kHz and SysClk/256. The Analog Column Clock for the ADC should be the same as the data clock. This is necessary for the user module to function properly. If SysClk is selected, API software overrides the column clock selection to set it correctly.

3. The setting of Data Clock (source to VC3), current setting, VC3 divider (in Global Resources), PWM_High, and PWM_Low decides the sample rate and input range of the ADC. To assist this selection process, a spreadsheet (SSlopeCalc.xls) is included in project file. For parameter settings in this project and input range of 0 to 4 V, a sample rate of 1.536 K is achieved.

Global Resources

Important Global Resources		
Parameter	**Value**	**Comments**
Power Setting	5.0 V/24 MHz	Circuit must operate at 5 V Vdd and IMO must generate 24 MHz
CPU Clock	SysClk/2	CPU runs at 12 MHz
VC1=SysClk/N	12	24/12 = 2 MHz. This clock is used as the column clock for SC blocks and as a data clock for ADC counter
VC3 Source	VC1	Should be same as Data Clock parameter of ADC user module
VC3 Divider	256	Select according to the required Sample Rate

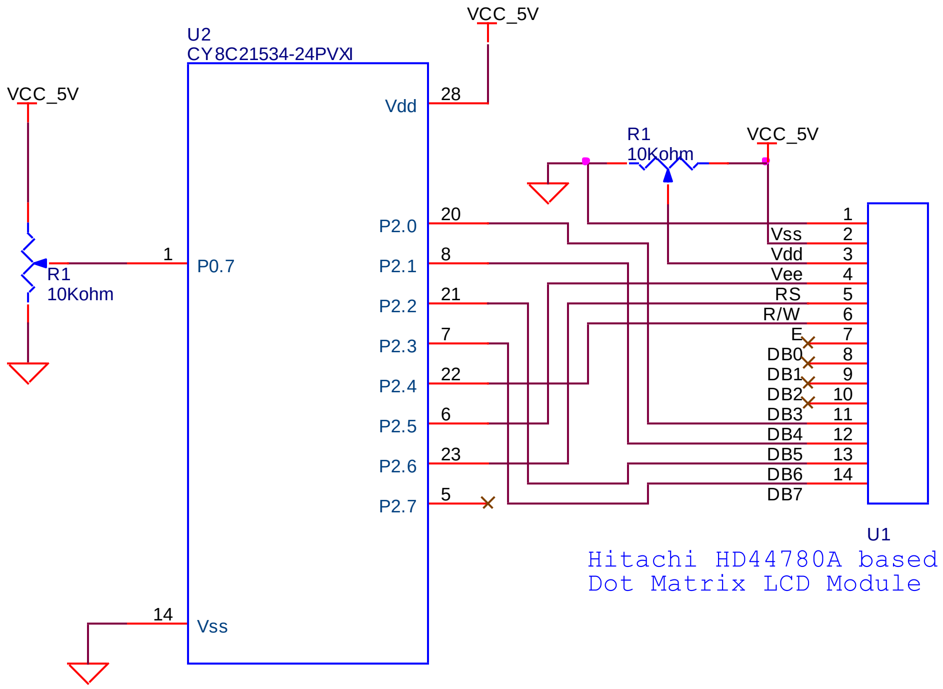
Notes

1. VC1 is selected as Data Clock of ADC user module. Therefore, the clock source provided by VC1 must lie between 240 kHz to SysClk/256.

2. Any other global resources may be left at their default value or configured as required.

Hardware Connections

The schematic diagram for the project is shown here.



The LCD is connected to Port2 of the PSoC® chip CY8C21X34. The analog input is connected to P0[7]. The project can be tested by varying external 0 to 4 V between V_{SS} and P0.7.

Operation

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

Following are the operations performed by firmware:

1. Start LCD
2. Print ‘MEASURED VOLTAGE’ on the LCD at location (0,0)
3. Enable global interrupt
4. Start ADC for FULLRANGE(V_{SS} to V_{DD}) and calibrate. For bandgap reference voltage 1.3 V, start the conversion in continuous sampling mode.
5. Calculate the scale factor to convert the ADC counts to voltage and store it in variable fScaleFactor. The scale factor is calculated as Volts/Count. The input voltage range is 4 V and the number of ADC counts is 1024. Therefore, the scale factor is 4 V/1024.
6. In an infinite loop, the following operations are performed:
   - Wait until ADC data is available
   - Read ADC data into variable iData and clear 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 example
   - Display this ASCII string on LCD at location row 1 column 0 followed by string ‘V’.

To test this project, vary the input voltage given on pin P0.7 and observe the value displayed on the LCD.