**EE443 - Embedded Systems**

Experiment 5 Laboratory Report

**Analog Input Output**

**Name: Erdem ŞİMŞEK**

**Number: 200206011**

**Date: 30-11-2016**

**Objective**

The main purpose of this experiment is that learning how to use ADC and observing the ADC performance by using DAC.

**Experimental Work**

ATmega328P is used in this experiment at 8 MHz clock speed. Also, we simulated the our code by using Proteus. We built the circuit as specified in prelab. ADC is used for converting analog signal to digital signal and we observe the ADC performance by using DAC on oscilloscope.

I included the necessary library which are “avr/io.h” and “LCDmodule.h”. LCDmodule library is to control the LCD. I copied the LCD library files to the same workspace with project and then, i added the files by using command ‘Add files recursively’. Also, i set the compiler and linker path setting for successful compiling.

All port-B pins are configured as an output and all port-C pins are configured as an input. Port-B is used for driving the LCD and DAC. ADC used the PORTC1 pin as analog input and ADC uses AREF as reference voltage and division factor is 64.

If SW0 is pressed and Atten value is greater than 1, Atten value will be decremented. Also, SW1 is pressed and Atten value is less than 6, Atten value will be incremented. This Atten value represents the attenuation factor.

**#include <avr/io.h>**

**#include <LCDmodule.h>**

**void PrintByte(char \*,char \*,char);**

**int main(void)**

**{**

**char LCDtext[16] = "Atten";**

**unsigned char Abyte = 1;**

**LCD\_Init();**

**DDRB = 0xFF;**

**DDRC = 0x00;**

**ADMUX = 0b01100000; //** AREF is selected as reference. Left adjust is selected

**ADCSRA = 0b10000110;** // ADC is enabled. Division factor is 64

**ADCSRB = 0b00000000;** // ASCSRB is in default value

**while(1){**

**ADCSRA = 0b11000110;** // ADC is started

**while(ADCSRA & (1<<ADSC)){** // Wait the ADSC pins becomes 0

**if((PIND & \_BV(PIND0)) == 0x00 ){** // SW0 is pressed

**if(Abyte > 1)**

**Abyte--;**

**while(!(PIND & \_BV(PIND0)));**

**}**

**else{**

**if( (PIND & \_BV(PIND1)) == 0x00 ){** // SW1 is pressed

**if(Abyte < 6)**

**Abyte++;**

**while(!(PIND & \_BV(PIND1)));**

**}**

**}**

**}**

**PORTB = ADCH >> (Abyte-1);** // Shift ADC conversion result due to Abyte

**ADCSRA = 0b11000110;** // Restart the ADC

**}**

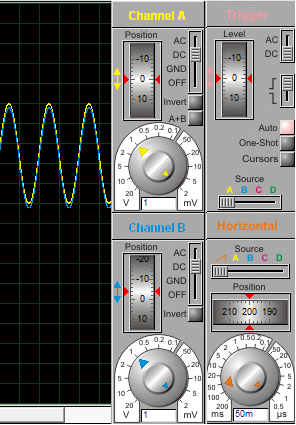
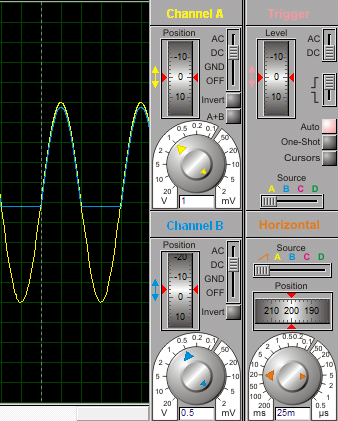
**return 0;**

**}**

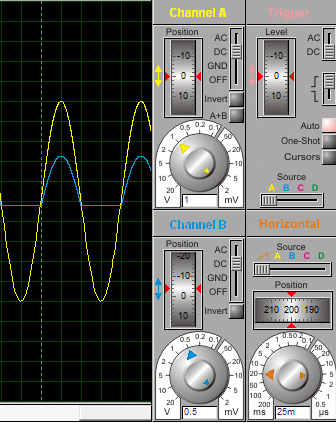
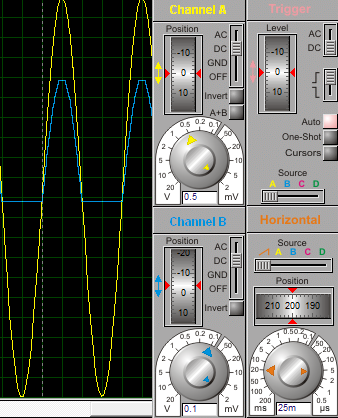
ADMUX register keeps the reference voltage, data adjust and input channel selection information. In this experiment, i used the AREF as reference voltage and data is adjusted to left. Also, the PINC1 is used for input channel. ADCSRA register keeps control register of ADC and prescaler information. The prescaler is 64 for that experiment. For starting the ADC conversion, the ADEN and ADSC registers are set as 1.

**Results and Conclusion**

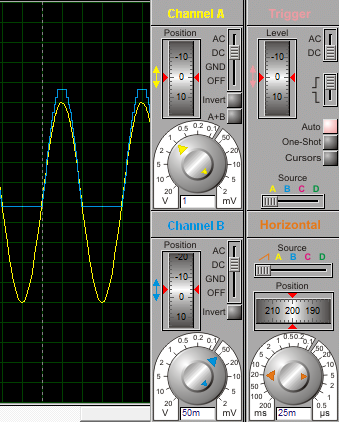
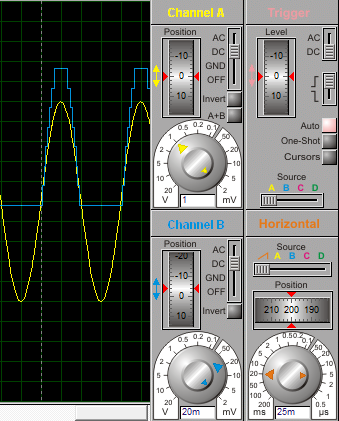
As seen from the figures, when the attenuation factor increases, it becomes harder to recover the signal. Because, the step size interval is getting decrease due to increasing Atten value. It seems like step function.

** **

***Figure 1 : When Atten is equal to 1 Figure 2 : When Atten is equal to 2***

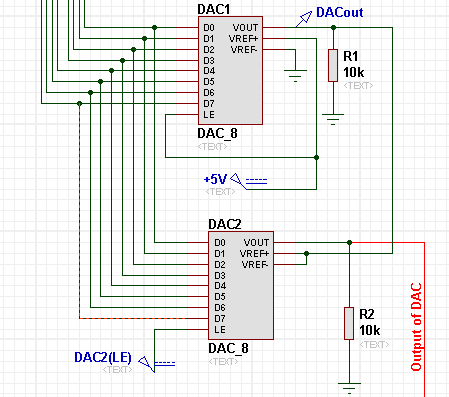
** **

***Figure 3 : When Atten is equal to 3 Figure 4 : When Atten is equal to 4***

** **

***Figure 5 : When Atten is equal to 5 Figure 6 : When Atten is equal to 6***

**Question 1 :** By using two DAC, the problem can be minimized. DAC1 output are used as reference voltage for DAC2. Thus, the distortion problem can be used.



**Question 2 :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Atten** | **Shift Operation** | **Atten** | **Shift Operation** |
| 1 | PORTB = ADCH >> 0 | 7 | PORTB = (ADCH >> 3) |
| 2 | PORTB = (ADCH >> 1) + (ADCH >>2) | 8 | PORTB = (ADCH >> 4) + (ADCH >> 5) |
| 3 | PORTB = (ADCH >> 1) | 9 | PORTB = (ADCH >> 4) |
| 4 | PORTB = (ADCH >> 2) + (ADCH >> 3) | 10 | PORTB = (ADCH >> 5) + (ADCH >> 6) |
| 5 | PORTB = (ADCH >> 2) | 11 | PORTB = (ADCH >> 5) |
| 6 | PORTB = (ADCH >> 3) + (ADCH >>4 ) |  |  |

**Question 3 :** There is an enable pin for activating the LCD and DAC. These pins are E pin for LCD and LE pin for DAC. For proper communication without causing any interfere. Two pins on port C are can be used for activation operation. PINC2 register is connected to the enable pin of LCD which is E pin and PINC3 register is connected to the enable pin of DAC which is LE pin.

If want to drive LCD, we need to set PINC2 pin as high and PINC3 pin as low. Also, for driving the DAC, you have to set PINC3 pin as high and PINC2 pin as low. Thus, LCD will be enable during LCD operation and DAC will be enable during DAC.