EE443 - Embedded Systems

Experiment **5** Laboratory Report

Analog Input Output

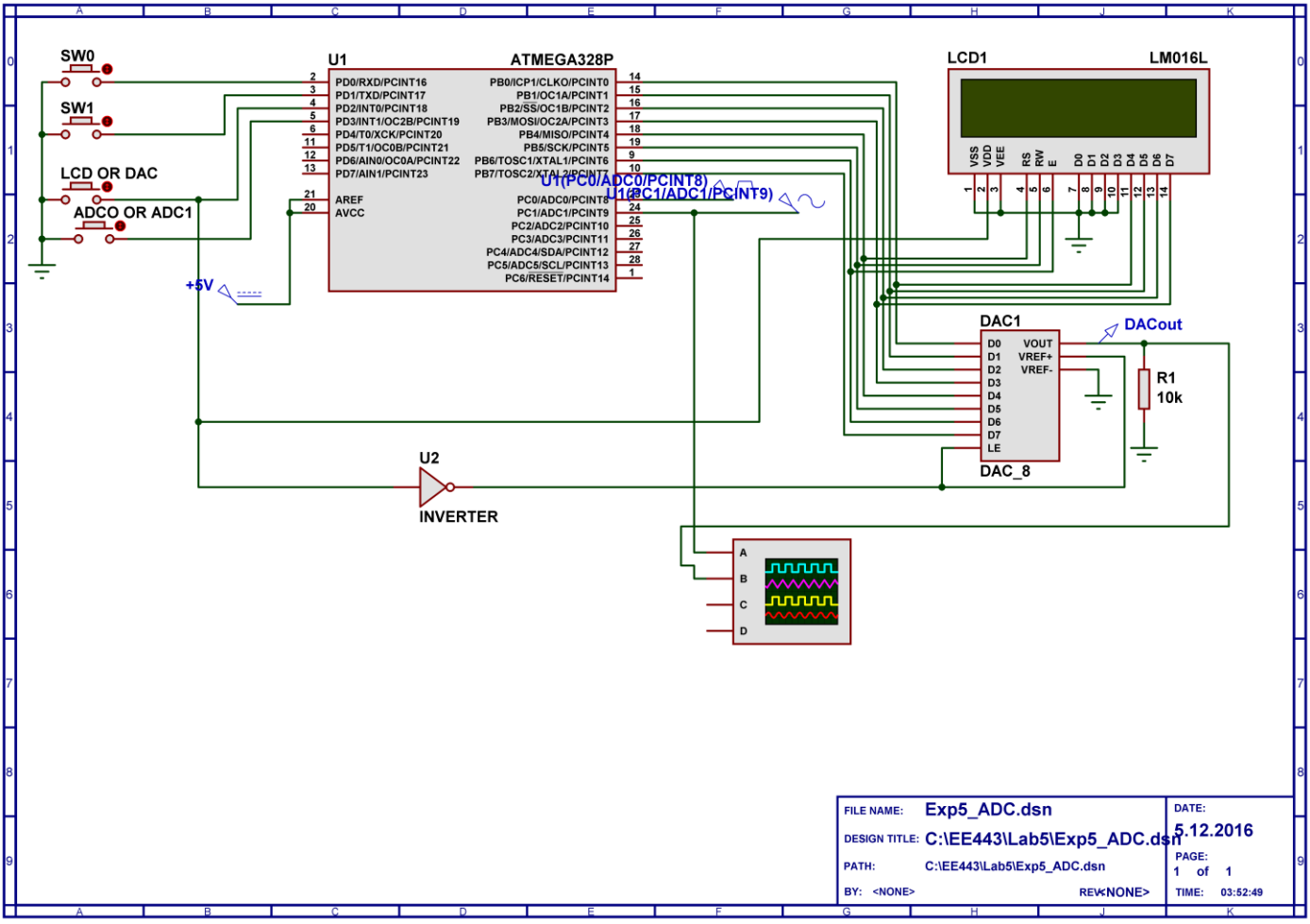
Name: M. Serdar Karaman

Number: 190206038

Date: 05 - December - 2016

Objective

Turning analog signal to digital with ADC and showing result on LCD, reconstruct signal with DAC.



Experimental Work

Experiment code

```
#include <avr/io.h>
#include <stdio.h>
#include <util/delay.h>
#include "LCDmodule.h"

int main(void){

    PRR   &=~_BV(PRADC); //Turn on ADC peripheral module
    //REFS1 REFS0 ADLAR MUX3 MUX2 MUX1 MUX0
    ADMUX  = 0b00100000; // ADLAR=1, 8bit high
    /*              MUX[3:0] Single Ended Input
                        0000 ADC0
                        0001 ADC1
                        0010 ADC2
                        0011 ADC3
                        0100 ADC4
                        0101 ADC5
                        0110 ADC6
                        0111 ADC7
                        1000 Temperature sensor
                        1001 Reserved
                        1010 Reserved
                        1011 Reserved
                        1100 Reserved
                        1101 Reserved
                        1110 1.1V (VBG)
                        1111 0V (GND)                   */

    ADCSRA = 0x10000110; // ADEN ADSC ADATE ADIF ADIE ADPS2 ADPS1 ADPS0
    /*    ADPS[2:0] Division Factor
                000 2
                001 2
                010 4
                011 8
                100 16
                101 32
                110 64
                111 128                */
    //8M/64=125k <200k(max ADC frequency)


    ADCSRB = 0x00; //ADC B is disabled.

    DDRD  |= 0b11110000; //  pin-0 and pin-1 set as input, others are output
    PORTD |= 0b00001111; //Pull-up resistor for inputs

    DIDR0 |= 0b00000011; //Disabled buffer for ADC0 and ADC1

    DDRB |=0b11111111; // all Port B pins are output


    LCD_Init(); //initialize the LCD display
    LCD_Clear();    //clear the LCD display

    char LCDtext[16];
    unsigned char Atten=1;
    unsigned char PDsave = 0x00;
    unsigned char PDsave_1 = 0x00;


    while(1){
```

```
        if((PIND & 0b00001000) == 0b00000000){// input selection
            ADMUX  = 0b00100001;//ADC1
        }
        else{
            ADMUX  = 0b00100000;//ADC0
        }


        ADCSRA = 0b11000110; // set ADSC, start converting
        while(ADCSRA==0b11000110) {
            ;//wait  until ADC complete
        }


        if((PIND & 0b00000100) == 0b00000000){//SW2=0 DAC active LCD inactive

            PDsave_1=PDsave;
            PDsave = PIND;

            if(((PIND & 0x01) == 0x00) && ((PDsave_1&0x01) == 0x01)){//SW0  pressed
                if(Atten > 1){
                    Atten--;
                }
                else{
                    ;
                }

            }
            else if(((PIND & 0x02) == 0x00) && ((PDsave_1&0x02) == 0x02)){//SW1 pressed
                if(Atten < 6){
                    Atten++;
                }
                else{
                    ;
                }
            }
            else{//Both pressed or not pressed
                ;//Do nothing
            }

            PORTB=(ADCH>>(Atten-1));
        }
        else{
            int ADCout=ADCH;
            PrintByte(LCDtext, "", ADCout);

            LCD_MoveCursor(1, 1);       // Place LCD cursor at column-1 of line-1
            LCD_WriteString(LCDtext);   // Send LCDtext to the LCD module

           _delay_us(1000);
        }

    };

    return 0;
}
```

Answers to the Questions

Question 1: Explain the changes in the DAC output waveform that occur at high attenuation factors. How can you implement an attenuator that does not cause distortion at high attenuation factors by using two DACs?

Answer 1: Low resolution at high attenuation is a foult of wrong refence voltage if attenuation was analog.

We need a voltage controlled resistor to attentuate input voltage and to control this voltage from MCU we need second DAC. Also output of this DAC needs to be connectod to reference voltage.

Question 2: As a home exercise, modify the program to obtain the attenuation steps given below **by using shift and addition operations only**.
Atten = 1=>1/1, **2**=>3/4, **3**=>1/2, **4**=>3/8, **5**=>1/4, **6**=>3/16,
7=>1/8, **8**=>3/32, **9**=>1/16, **10**=>3/64, **11**=>1/32

Answer 2: Only the PORTB calculations were changed. Rest of the code remains same.

```
if(Atten%2 == 0){
      PORTB = ADCH>>(Atten-1);
}
else{
      PORTB = ADCH>>((Atten+3)>>1) + ADCH>>((Atten+3)>>1) + ADCH>>((Atten+3)>>1);
}
```

Question 3: Suggest a method to send data to two external devices by using two more MCU pins. Port-B data should be directed to the target device without causing any unwanted data transfers (i.e. DAC will not receive LCD data, LCD will not receive DAC data).

Answer 3: Solution is alredy implemented on my code.