Group #12 **Fall ECGR 4101/5101 Lab #04 Report** **09/22/2022**
Naseeruddin Lodge
nlodge@uncc.edu
Srikar Thakkallapally
sthakkal@uncc.edu

Lab Objective:

The objective of this lab is to write a code in Code Composer Studio (CCS) that will read the ADC value of through a potentiometer and display it on a quad digit 7-segment display. The potentiometer and ADC assisted in converting the analog value to digital value and then this value is displayed on the quad digit 7-segment display.

Commentary and Conclusion:

The first problem was understanding how a quad 7-digit display works. Then it was very challenging to get 1024 values to display on the quad digit 7-segment display. This was finally achieved by separating a number into single digits and displaying one digit at a time. The most challenging part was to stop the display from oscillating. After multiple attempts, the method that worked best for us was writing a series of if else statements that checked all cases and prevent the display from oscillation. Then finally we had to add a few more if statements to account of 0 and 1023. Overall, it was a very challenging lab but also gave us good practice is ADC and problem solving.

Quad digit 7-segment display	Pins
A	P2.0
B	P2.1
C	P2.2
D	P2.3
E	P2.4
F	P2.5
G	P2.6
DP	P2.7
D1	P1.1
D2	P1.2
D3	P1.4
D4	P1.5

Table 1 - Quad digit 7-segment display connections

Potentiometer	Pins
Vcc	Vcc
Output Pin	P1.3
GND	GND

Table 2 – Potentiometer connections

Figures:

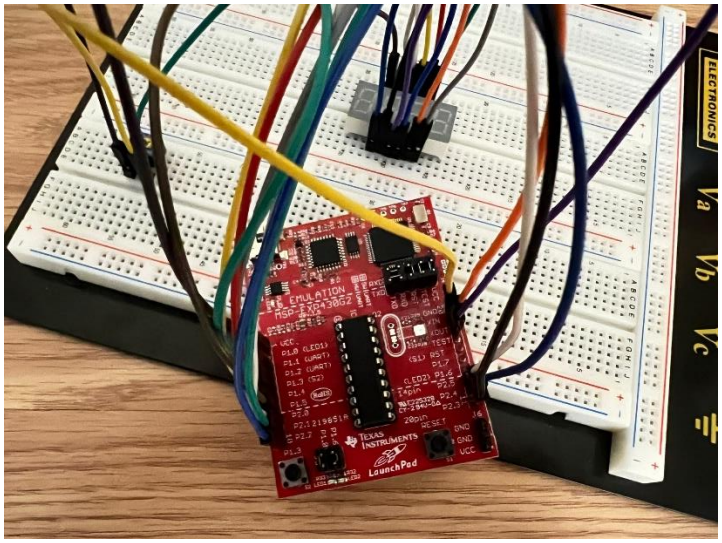

Figure 1 – Circuit Connection

Lab Code:

```
//*******************************************************************
//ECGR 5101
//Lab 04
//Display 4 Digit 7segment LED corresponding to the digital reading
//of a potentiometer
//*******************************************************************
#include <msp430.h>

//*******************************************************************
//Function to Setup ADC
//*******************************************************************
void ADC_Setup(){

    //SREF     -> 000b = VR+ = VCC and VR- = VSS
    //ADC10SHT -> 10b = 16 ADC10CLK cycles
    //ADC10ON  -> ADC10 on
    //INCH     -> Input channel select
```

```
    ADC10CTL0 = SREF_0 + ADC10SHT_2 + ADC10ON;
    ADC10CTL1 = INCH_3;
    ADC10AE0 |= 0x08;
}

//*************************************************************
//Used to display a number.
//Number 0-9 is given as an input and the corresponding
//light turns on using a switch statement
//Case 10 is used to turn off LED and turn off All the LED digits
//*************************************************************
void DisplayLED(int n_SingleDigit){

    switch(n_SingleDigit){

        //P2OUT = 0xhgfedcba
        //0x00111111
        case 0:
            P2OUT = 0xC0;
            break;

        //0x00000110
        case 1:
            P2OUT = 0xF9;
            break;

        //0x01011011
        case 2:
            P2OUT = 0xA4;
            break;

        //0x01001111
        case 3:
            P2OUT = 0xB0;
            break;

        //0x01100110
        case 4:
            P2OUT = 0x99;
            break;

        //0x01101101
        case 5:
            P2OUT = 0x92;
            break;

        //0x01111101
        case 6:
            P2OUT = 0x82;
            break;

        //0x00000111
        case 7:
            P2OUT = 0xF8;
```

```
            break;

        //0x01111111
        case 8:
            P2OUT = 0x80;
            break;

        //0x01110111
        case 9:
            P2OUT = 0x90;
            break;

        case 10:
            P1OUT = 0x00;
            P2OUT = 0xFF;
            break;


    }

}

//*****************************************************************
//This function is responsible for displaying the numbers.
//It gets an input of the reading of the potentiometer.
//Depending on the number of digits required, the digits are turned on.
//Then the Display LED function is used to turn on the numbers, one
//number at a time.
//If the ADC number is 357. 7 is displayed on D1, 5 on D2 and 3 on D3.
//DisplayLED(10) is used to turn off everything
//*****************************************************************
void Control_Dx(int n){

    int SingleDigit;

    //if n = 271, 2 is displayed on D1
    //then D1 is turned off and D2 is turned on
    //and 7 is displayed. Then D3 is turned on
    //and 1 is displayed

    if(n <= 9){

        //D4 - 0x00100000

        P1OUT = 0x02;
        SingleDigit = n;
        DisplayLED(SingleDigit);
        __delay_cycles(1000);

    }

    else if((n>=10) && (n<=99)){

        //D4 - 0x00100000
        //D3 - 0x00010000
```

```
    DisplayLED(10);

    P1OUT = 0x02;
    SingleDigit = n % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(1000);

    DisplayLED(10);

    P1OUT = 0x04;
    SingleDigit = n / 10 % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(1000);

    DisplayLED(10);
}

else if((n>=100) && (n<=999)){

    //D4 - 0x00100000
    //D3 - 0x00010000
    //D2 - 0x00000100

    P1OUT = 0x02;
    SingleDigit = n % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(1000);

    DisplayLED(10);

    P1OUT = 0x04;
    SingleDigit = n / 10 % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(1000);

    DisplayLED(10);

    P1OUT = 0x10;
    SingleDigit = n / 100 % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(1000);

    DisplayLED(10);
}

else if(n >= 1000){

    //D4 - 0x00100000
    //D3 - 0x00010000
    //D2 - 0x00000100
    //D1 - 0x00000010

    P1OUT = 0x02;
    SingleDigit = n % 10;
```

```
        DisplayLED(SingleDigit);
        __delay_cycles(1000);

        DisplayLED(10);

        P1OUT = 0x04;
        SingleDigit = n / 10 % 10;
        DisplayLED(SingleDigit);
        __delay_cycles(1000);

        DisplayLED(10);

        P1OUT = 0x10;
        SingleDigit = n / 100 % 10;
        DisplayLED(SingleDigit);
        __delay_cycles(1000);

        DisplayLED(10);

        P1OUT = 0x20;
        SingleDigit = n / 1000 % 10;
        DisplayLED(SingleDigit);
        __delay_cycles(1000);

        DisplayLED(10);
    }


}

//*******************************************************************
//Main Function
//*******************************************************************
int main(void)
{
    //Stop watchdog timer
    WDTCTL = WDTPW | WDTHOLD;

    int step1=0, step2=0;

    //Turn on GPIO for Xin and Xout
    P2SEL = 0;
    P2SEL2 = 0;

    //P1.1,P1.2,P1.4,P1.5 -> D1,D2,D3,D4
    //Set P1.3 input (potentiometer)
    P1DIR = 0x36;

    //Set P2.0 - P2.5 to output
    //P2.0 - P2.7 -> abcdefgh
    P2DIR = 0xFF;


    //Function to setup ADC
    ADC_Setup();
```

```
//ADC10CTL0 |= ENC + ADC10SC;

//step1 = ADC10MEM;

//Infinite loop to keep the embedded system
//running forever
while(1){

   ADC10CTL0 |= ENC + ADC10SC;

   //Stores the digital value of the potentiometer
   //ADC10MEM is a number between 0 to ((2^10) - 1)
   step1 = ADC10MEM;

   //If n=50 and it oscillates between n=49 and
   //n=51, this code check to see if this kind of
   //oscillation has occured and sets n=50.
   //Also accounts for n=0 and n=1023.
   if(step1 == 0){

       Control_Dx(0);
   }

   else if(step1 == 1023){

       Control_Dx(1023);
   }

   else if(step2 == 0){

       Control_Dx(step1);
   }

   else if((step1 == 1) && (step2 == 1)){

       Control_Dx(step1);
   }

   else if((step2 == (step1+1))){

       Control_Dx(step2);
       step1 = step2;
   }

   else if(step2 == (step1-1)){

       Control_Dx(step2);
       step1 = step2;
   }


   else if((step2 != (step1+1)) || (step2 != (step1-1))){

       Control_Dx(step1);
```

```
        }

        step2 = step1;
    }

}
```