Group #12 **Fall ECGR 4101/5101 Lab #06 Report** **10/20/2022**

Naseeruddin Lodge
nlodge@uncc.edu
Srikar Thakkallapally
sthakkal@uncc.edu

Lab Objective:

The objective of this lab is to display ADC value of a potentiometer on a quad digit 7-segment LED using an off-board IC msp430g2452. The potentiometer and the quad digit 7-segment LED were connected to msp430g2452 that was placed on a breadboard. Then uploaded the code to msp430g2452 using TI board. The chip was capable of displaying ADC from the potentiometer to the quad digit 7-segment LED only with Vcc and gnd from the TI board.

Commentary and Conclusion:

The most important thing was to get familiar with how a off board IC works. It took some research and some trials and errors to familiarize how an off-board IC can be programmed and could work just with voltage source if programmed correctly. The next hurdle occurred was oscillations that were produced by the potentiometer and quad digit 7-segment LED. After a lot of modifications, the method that worked the best was filtering out the values using multiple if elseif statements. For example, the oscillations were occurring between 49 to 51, the code figures out the middle number 50 and displays that onto the quad digit 7-segment LED. Multiple such statements were written in order to account for all the cases. Optimal delays were added to assist the quad digit 7-segment LED to display the numbers smoothly. Overall a very interesting lab and helped understand how off board IC can be programmed and gave a better understanding on how embedded systems work.

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 2 - Quad digit 7-segment display connections

Button	Pins
On-board button	P1.3

Table 1 – On-board button connection

Figures:

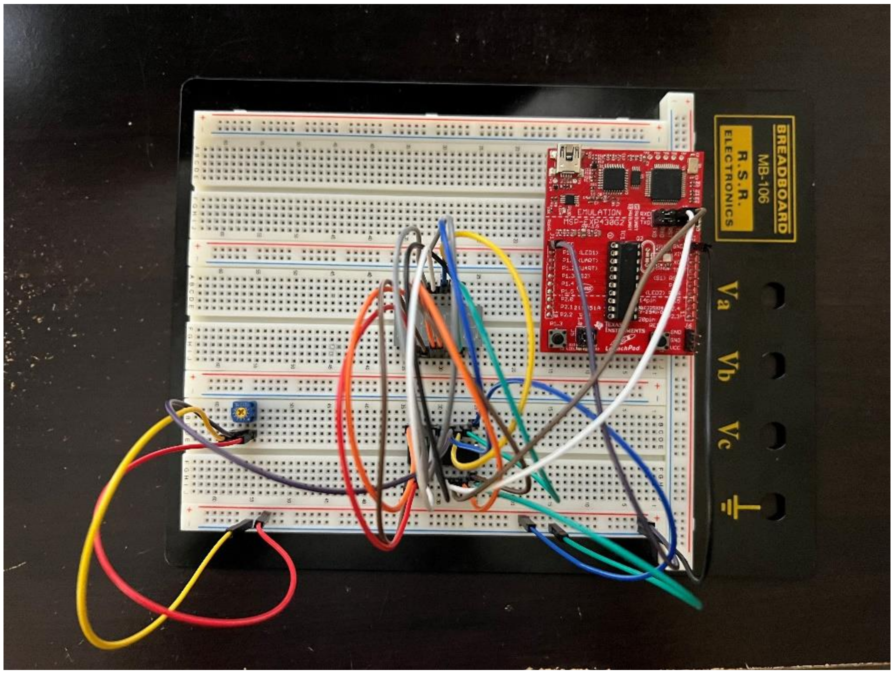

Figure 1 – Circuit Connection

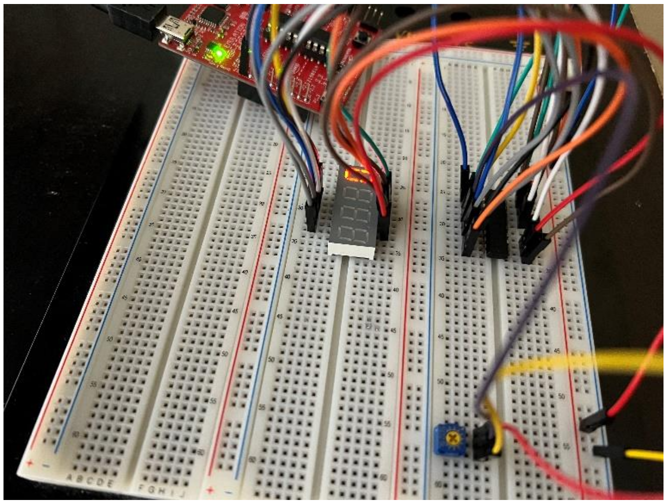

Figure 2 – Quad Digit 7-segment LED displaying ADC Value 0

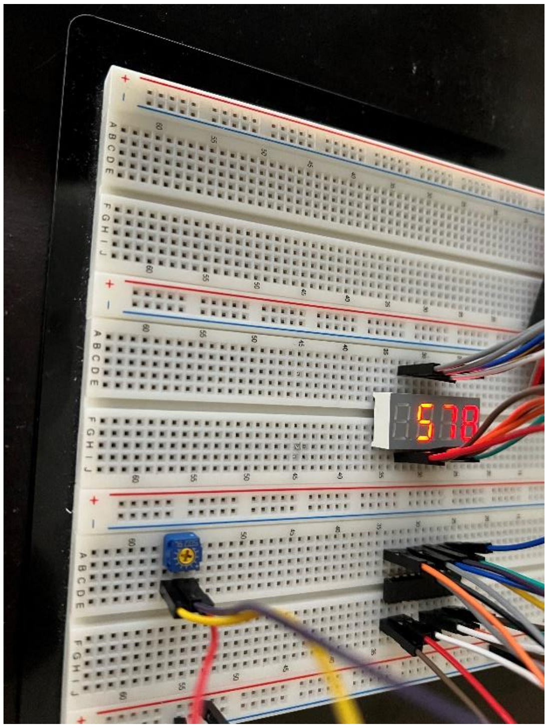

Figure 3 – Quad Digit 7-segment LED displaying ADC Value 578

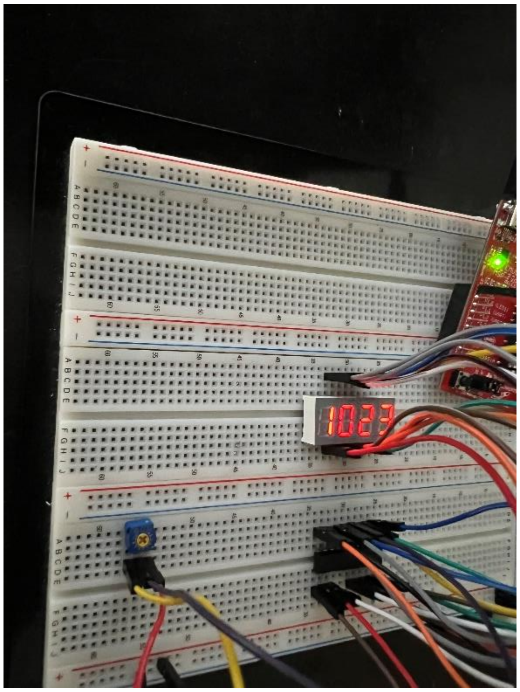

Figure 4 – Quad Digit 7-segment LED displaying ADC Value 1023

Lab Code:

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



//Global Variables
int step1=0, step2=0;



//Prototyped Functions
void PinsSetup();
void ADC_Setup();
void Correct_Osciallations();
void DisplayLED(int n_SingleDigit);
void Control_Dx(int n);



//*****************************************************************
//Main Function
//*****************************************************************
int main(void){

    //Stop watchdog timer
    WDTCTL = WDTPW | WDTHOLD;

    //Function to setup all the Pins required
    PinsSetup();

    //Function to setup ADC
    ADC_Setup();

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

        //Gets ready for ADC Conversion
        ADC10CTL0 |= ENC + ADC10SC;

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

        //Function used to fix oscillations
        Correct_Osciallations();
    }

}



//****************************************************************
//Name    : PinsSetup()
//Input   : void
//Returns : void
//
//Function to setup all the Ports and Pins used
//****************************************************************
void PinsSetup(){

    //P1REN |= 0x08;

    //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;
}



//****************************************************************
//Name    : ADC_Setup()
//Input   : void
//Returns : void
//
//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;
}



//****************************************************************
//Name    : correct_osciallations(int step)
```

```
//Input   : void
//Returns : void
//
//This function is responsible for fixing oscillations.
//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.
//*****************************************************************
void Correct_Osciallations(){


    //These series of if else if loops check to see if a number is
    //if a number is oscillating between for example 4,5 and 6. These
    //if else if loops fix that oscillation and set the ADC value to 5.
    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+2))){

        Control_Dx(step2);
        step1 = step2;
    }

    else if(step2 == (step1-2)){
        Control_Dx(step2);
        step1 = step2;
    }

    else if((step2 == (step1+3))){

        Control_Dx(step2);
        step1 = step2;
    }

    else if(step2 == (step1-3)){

        Control_Dx(step2);
        step1 = step2;
    }

    else if((step2 == (step1+4))){

        Control_Dx(step2);
        step1 = step2;
    }

    else if(step2 == (step1-4)){

        Control_Dx(step2);
        step1 = step2;
    }


    else if((step2 != (step1+1)) || (step2 != (step1-1)) ||
            (step2 != (step1+2)) || (step2 != (step1-2)) ||
            (step2 != (step1+3)) || (step2 != (step1-3)) ||
            (step2 != (step1+4)) || (step2 != (step1-4))){

        Control_Dx(step1);
    }


    step2 = step1;

}



//*******************************************************************
//Name    : DisplayLED(char n_SingleDigit)
//Input   : char
//Returns : void
//
//Used to display a number.
//char 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
    //0
    case 0:
        P2OUT = 0xC0;
        break;

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

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

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

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

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

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

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

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

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

        //Turns off D1, D2, D3, D4
        //and all the 7 segments
        case 10:
            P1OUT = 0x00;
            P2OUT = 0xFF;
            break;


    }

}



//*****************************************************************
//Name    : Control_Dx(int n)
//Input   : int
//Returns : void
//
//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(5000);

    }
```

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

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

    DisplayLED(10);

    P1OUT = 0x02;
    SingleDigit = n % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(5000);

    DisplayLED(10);

    P1OUT = 0x04;
    SingleDigit = n / 10 % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(5000);

    DisplayLED(10);
}

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

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

    P1OUT = 0x02;
    SingleDigit = n % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(5000);

    DisplayLED(10);

    P1OUT = 0x04;
    SingleDigit = n / 10 % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(5000);

    DisplayLED(10);

    P1OUT = 0x10;
    SingleDigit = n / 100 % 10;
    DisplayLED(SingleDigit);
    __delay_cycles(5000);

    DisplayLED(10);
}

else if(n >= 1000){

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

```
        //D2 - 0x00000100
        //D1 - 0x00000010

        P1OUT = 0x02;
        SingleDigit = n % 10;
        DisplayLED(SingleDigit);
        __delay_cycles(5000);

        DisplayLED(10);

        P1OUT = 0x04;
        SingleDigit = n / 10 % 10;
        DisplayLED(SingleDigit);
        __delay_cycles(5000);

        DisplayLED(10);

        P1OUT = 0x10;
        SingleDigit = n / 100 % 10;
        DisplayLED(SingleDigit);
        __delay_cycles(5000);

        DisplayLED(10);

        P1OUT = 0x20;
        SingleDigit = n / 1000 % 10;
        DisplayLED(SingleDigit);
        __delay_cycles(5000);

        DisplayLED(10);
    }


}
```