2DP4: Microprocessor Systems
Final Project

Instructor: Dr. Doyle

Zeyang Wen – L07 – wenz10 – 400139518

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by **[Zeyang Wen, wenz10, 40013951**

DATA ACQUISITION AND DISPLAY: THE DIGITAL INCLINATION ANGLE MEASUREMENT SYSTEM

I. Table of Contents

***Abstract*—This project is aiming to experience how to obtain the data from a tilt-sensing application and convert electrical signal to analog signal and translate them into PC displace using MATLAB. The digital signal will be show in plot on MATLAB due to the embedded system that was built.**

***Index Terms*—ADC (Analog-to-digital conversion), Bus-clock, E-clock, ESDX, Time Delay,**

II. Introduction & Background

As a component of an engineering project, one of the objectives is acquiring inclination angle measurement data from 0º to 90º. This data will be used as a parameter for numerous custom algorithms, student need to build an ESDX embedded prototype to acquire the angle data, process the data, and transmit the data for display and recording.

Acquiring data system will be widely used in engineering subjects. There is large amount of systems available but most of them are either expensive, limited in capacity or too complex. The experience with this project will provide us a clear and deep understanding of commercial/industrial/medical data acquisition systems operate.

There is one real-life example of ADC, music recoding. People often produce music through computer, therefore, need to create data streams that go into digital music file, like mp4 by using analog-to-digital converters.

III. Design Methodology

To have a general overview of this project, at least, these tasks need to be completed.

First understand the function of accelerometer, quantify he analog signal, transfer the analog signal to ESDX. Then, build an embedded system which convert analog signal to digital signal, and use LEDs as the angle output and another DC voltage output by using MATLAB.

A. *Final Pin Assignment map*

Port AD5 is used as analog signal input pin which is connected to x-axis.

Port AD [0:4] and AD [8:11] as digital signal output.

LED	9	8	7	6	5	4	3	2	1
Port AD	4	11	10	9	8	3	2	1	0

Table 1: LED Map

Turn on the user LED on the DIG 13 to tell people the system is alive.

Port T [0] is used in the interrupt and connected with a button.

Port T [2] is connected to a switch for mode switching.

The voltage source is 3.3V.

Connect the ground to GND pin.

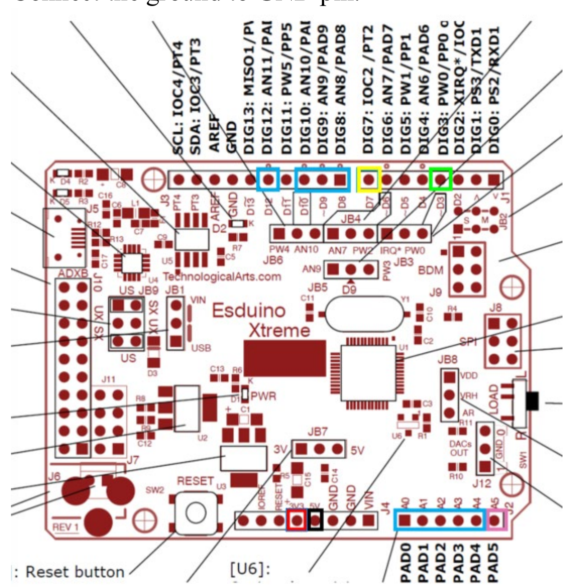


Figure 1: Esduino diagram with ports location.
Blue is output pin to LED. Green is connected to the interrupt button.
Red is 3.3V voltage source. Yellow is connected to switch.
Pink receives analog input signal. Black is ground.

B. *Quantify Signal Properties*

Input voltage = 3.3V;
$V_{RL} = 0\ V; V_{RH} = 3.3\ V;$
ADC resolution =10 bit;
Therefore, the equation of convert voltage and digital code is: $V_{val} = 0\ V + (3.3\ V * val)/(2^{10} - 1);$
Angle =arcsin (x) $\approx x + \frac{x^3}{6} + 3 * \frac{x^5}{40} + 5 * \frac{x^5}{168};$
$x = (val \times 3.3\ V \div 1023 - 1.65\ V)/0.5\text{V};$
Bus Speed= E-clock =8 MHz;
Sample rate=9.091 Hz;

C. *Transducer*

An embedded is built to convert analog signal to digital signal. AN 5 (AD 5) is used to receive the analog signal which produced by ADXL 337, an angle sensing application. By using ATDDR0 register, ESDX can produce digital signal which can be transmit to PC display.

D. *Precondition/Amplification/Buffer*

One of the preconditions is the function of the accelerometer, ADXL337, an angle or tilt sensor.
Input voltage = 3.3V;
$V_{RL} = 0\ V; V_{RH} = 3.3\ V;$ And the output voltage will increase with the ascending of angle.

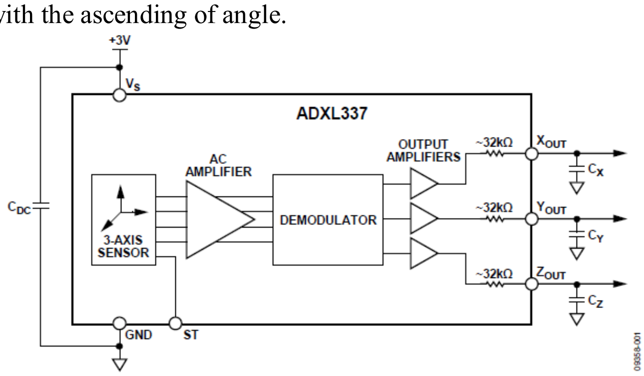


Figure 2: Functional Block Diagram
How the tilter work.

E. *ADC*

The student number is 400139518,

1) *ADC channel is AN5 (PAD 5).*

ATDCTL5 = 0x25 ; //continuous conversion for one channel(channel 5)

Table 14-15. Analog Input Channel Select Coding

SC	CD	CC	CB	CA	Analog Input Channel
	0	1	0	1	AN5

Table 2 : Set analog input channel as AN5

2) *ADC resolution is 10 bits.*

ATDCTL1 = 0x25 ; //set for 10-bit resolution

	7	6	5	4	3	2	1	0
R W	ETRIGSEL	SRES1	SRES0	SMP_DIS	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0
Reset	0	0	1	0	1	1	1	1

Figure 14-4. ATD Control Register 1 (ATDCTL1)

Table 3: ATCDTL1 register

Table 14-4. A/D Resolution Coding

SRES1	SRES0	A/D Resolution
0	0	8-bit data
0	1	10-bit data
1	0	12-bit data
1	1	Reserved

Table 4: ATD Resolution coding map

Table of ADC values:

Analog Voltage (V)	ADC Value
3.3 V	1023
2.475 V	767
1.65 V	511
0.825 V	256
0 V	0

Table 5: Analog Voltage and corresponding ADC value

3) *Bus clock is 8MHz*

Code warrior code:

```
void setCLK8(void){
  CPMUPROT = 0;     //disable clock write protection
  CPMUCLKS = 0x80;    //set PLLSEL=1
  CPMUOSC = 0x80;    // set OSCE=1
  CPMUREFDIV = 0x41;    //set reference frequency to
8/2=4MHZ
  CPMUSYNR = 0x05;      //set VCOCLK frequency to
2*4*(5+1)=48MHZ
  CPMUPOSTDIV = 0x02;   //set pll frequency to
48/(2+1)=16MHZ
  while(CPMUFLG == 0);  //wait for pll to engage
  CPMUPROT = 1;         //enable clock write
protection
}
```

Calculation steps:
Desired bus speed: 8MHz
$f_{PLL} = 16\ MHz.$
$f_{PLL} = \frac{VCO_{clk}}{POSTDIV+1}. POSTDIV = 2.$
$f_{VCO} = 48\ MHz.$
$f_{VCO} = 2 \times f_{REF} \times (SYNDIV + 1).$
(VCOFR = 0 and SYNDIV = 5)
$f_{REF} = 4\ MHz.$
$f_{REF} = f_{osc}/(REFDIV + 1)(OSCE = 1)$
$f_{osc} = 8\ MHz.$
We choose PLLCLK Bus Clock Source.

F. *LED display MODE 0 (BCD) & MODE 1 (Bar)*

Port AD [0:4] and AD [8:11] as digital signal output.

LED	9	8	7	6	5	4	3	2	1
Port AD	4	11	10	9	8	3	2	1	0

```
if(PTT==0xC3){
          PT0AD=angle/10;
          PT1AD=angle%10;
```

Under the condition that switch is off. Port T equals to 0xC3.

The mode is 0. The LEDs will represent the angle using binary-coded-decimal format. For example, if the angle is 20 degree. Only LED 6 will be on.

```
if(PTT==0xC3){
          PT0AD=angle/10;
          PT1AD=angle%10;
        }
        if(PTT==0xC7){
         if(angle>=0 && angle<5){
          PT1AD=0x00;
          PT0AD=0x00;
         }
         if(angle>=5 && angle<15){
          PT1AD=0x01;
          PT0AD=0x00;
         }
         if(angle>=15 && angle<25){
          PT1AD=0x03;
          PT0AD=0x00;
         }
         if(angle>=25 && angle<35){
          PT1AD=0x07;
          PT0AD=0x00;
         }
         if(angle>=35 && angle<45){
          PT1AD=0x0F;
          PT0AD=0x00;
         }
         if(angle>=45 && angle<55){
          PT1AD=0x0F;
          PT0AD=0x01;
         }
         if(angle>=55 && angle<65){
          PT1AD=0x0F;
          PT0AD=0x03;
         }
         if(angle>=65 && angle<75){
          PT1AD=0x0F;
          PT0AD=0x07;
         }
         if(angle>=75 && angle<85){
          PT1AD=0x0F;
          PT0AD=0x0F;
         }
         if(angle>=85 && angle<90){
          PT1AD=0x1F;
          PT0AD=0x0F;
         }
        }
```

This part of code is about Mode 1, which the LEDs will represent the angle as a linear bar increasing in 10-degree rounded increments from 0 to 90 degrees. For example, is range of 5 degree to 15 degree, LED 1 will be on.

G. *Data Processing.*

After receiving the digital data form AN 5 with the maximum range of 0~1023. The output from accelerometer is around 630 to 535. Which can be convert to voltage of 2.03 V and 1.726V.

```
value= ((val * 3.3/1023)-1.725)/0.36;
```

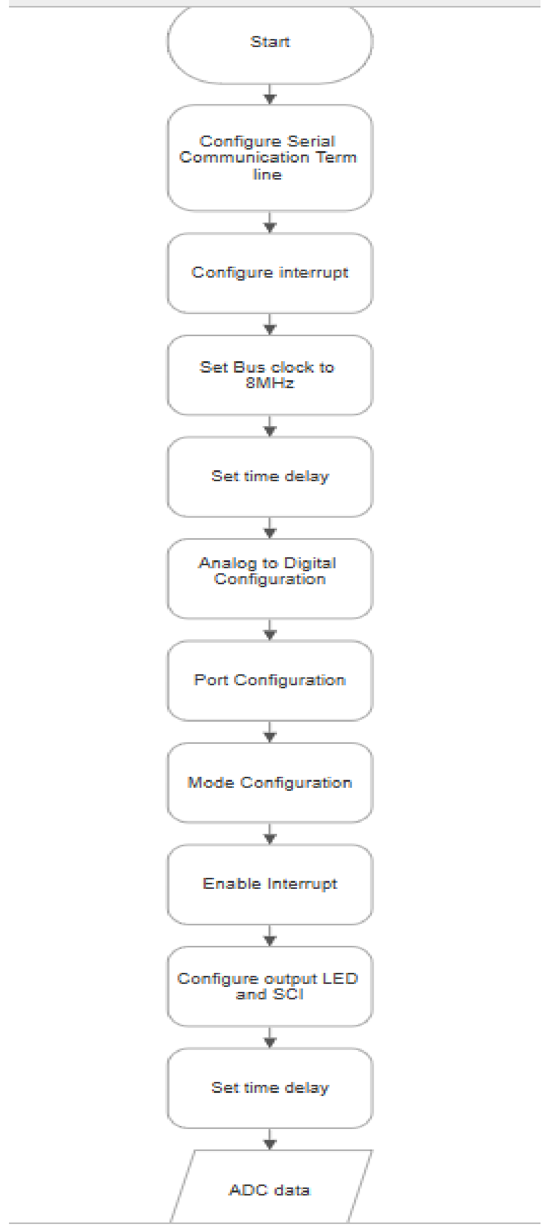
```
angle=(value+(value*value*value)/6+(value*value*valu
e*value*value)*3/40+(value*value*value*value*value*v
alue*value)*15/336)*(18000/314);
```

Value is the parameter of arcsine and angle should be the output around 0 degree to 90 degree.

$$\theta = \sin^{-1}\left(\frac{A_{XOUT}\,[g]}{1\,g}\right)$$

Figure 3 :Angle formula

Data Processing:



Flowchart 1: Data process flowchart

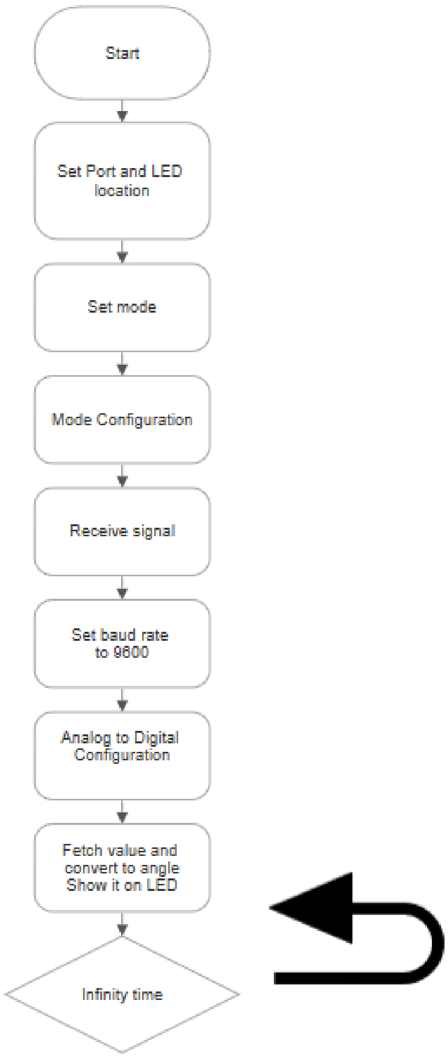
H. *Control/Communicate*

For a correct serial communication, the baud rate parameter in the SCI file need to be change into right value corresponding to 8 MHz bus clock value. I use MATLAB as the PC display that receive the signal from Code Warrior and convert to angle. The port is set to ‘COM 3’ which allow them to transfer signal. MATLAB need a terminator which I choose CR which right at the end of val.

```
SCI_OutUDec(val);
```

```
OutCRLF(); // Receive the digital signal
 s = serial('COM3');
 s.baudrate = 9600;
 s.terminator = 'CR';
```

Figure 4: MATLAB Code



Flowchart 2: Algorithm flowchart

I. Full System Block Diagram

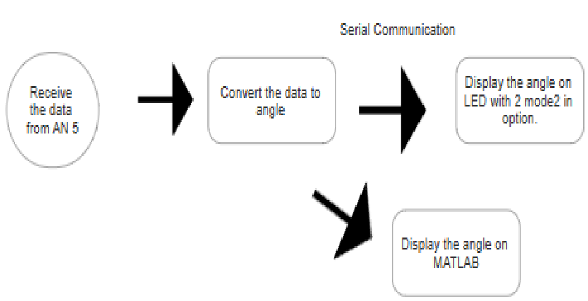


Figure 5: Full System Block Diagram

J. Full System Circuit Schematic

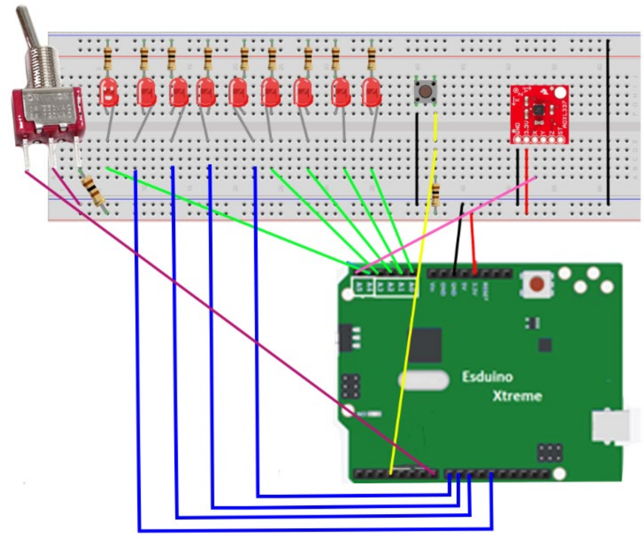


Figure 6 Full System of Circuit Schematic

IV. Result

A. Realterm output result

After set the ports to 3, and baud rate to 9600, digital value from 530 to 630 shows on the screen.

Figure 7: raw value of Realterm output

Figure 8: raw value and corresponding angle

B. MATLAB output result

The unite of x-axis is time and the unite y-axis is angle.

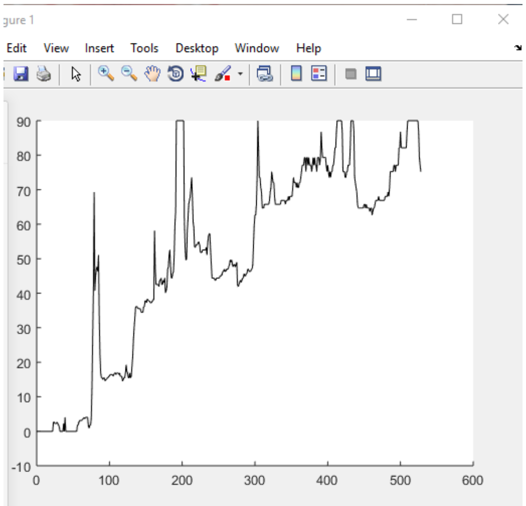
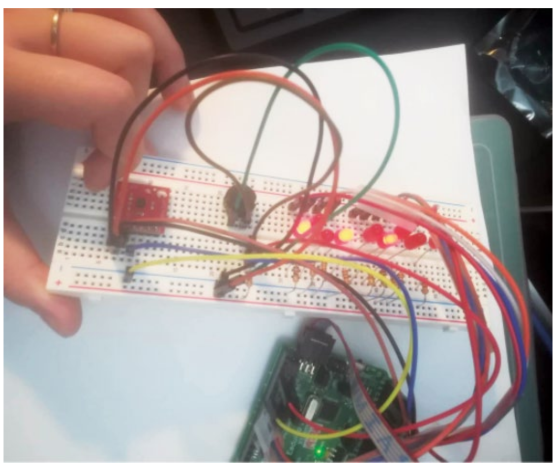
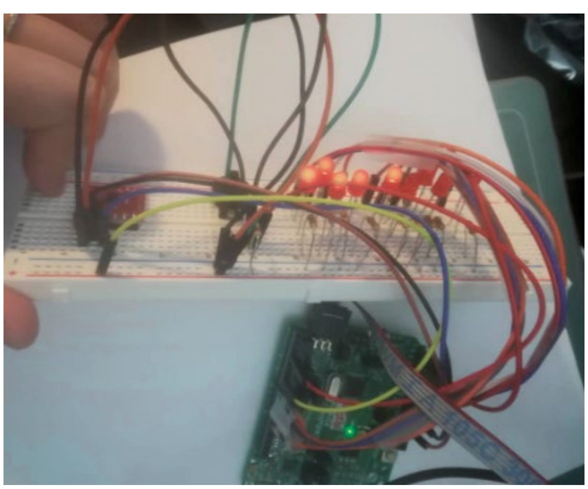


Figure 9: MATLAB output

C. Mode 0 LED Output



D. Mode 1 LED Output



V. Discussion

Address the following questions in your discussion:

(1) Summarize how you were able to overcome the ESDX not having float point capability and not having trigonometric functions.

When I open the new project, I choose floating point format supported, which float is IEEE32, double is IEEE32. Therefore, I can use double and float in 32 bits.

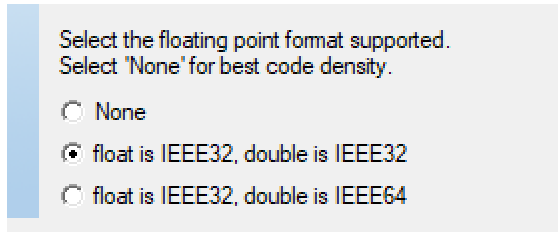


Figure 10: Selecting floating point format

(2) Calculate your maximum quantization error.

$$The\ maximum\ quantization\ error = \frac{3.3}{2^{10}} \approx 0.0032227$$

(3) Based upon your assigned bus speed, what is the maximum standard serial communication rate you can implement. How did you verify?

The bus speed is 8MHz.
Due to the maximum SCI tolerance is $\pm$ 6 %.
The maximum communication rate is 96000.

Baud Rate	Baud Divisor	Percent Error
2400	208	0.16%
4800	104	0.16%
9600	52	0.16%
19200	26	0.16%
38400	13	0.16%
57600	9	3.55%
96000	5	4.17%
115200	4	8.51%

Table 6: Baud Rate and corresponding Divisor with error

(4) Reviewing the entire system, which element is the primary limitation on speed? How did you test this?

The primary limitation is the accelerometer output frequency which limited up to 500Hz. We can measure it by using oscilloscope to test the frequency of AN 5.

(5) Based upon the Nyquist Rate, what is the maximum frequency of the analog signal you can electively reproduce? What happens when your input signal exceeds this frequency?

According to Nyquist Rate, sampling must occur at no less than twice the signal frequency of interest. The maximum frequency of the analog signal that this project can electively reproduce is 1MHz.
$f\_sample > 2 \times f\_signal$
If the input signal is larger than 500Hz, the aliasing will happen.

(6) In general, are analog input signals with sharp transitions (e.g., square, sawtooth, etc.) accurately reproduced digitally? Justify your answer.
The analog input signals can have sharp transition.

VI. Conclusion

With the experience of this project, we have a better understanding of embedded system and analog to digital conversion. Learn how to obtain the data from a angle-sensing application and convert electrical signal to analog signal and translate them into PC displace using MATLAB. This design is useful in real life and future engineering design, and we have a deep understanding of how ADC work.