EE 128
Lab 4:Analog to Digital conversion and Display

Section 021
TA: Mohsen Karimi

Tanish Arora
Leya Zeng

1. Abstract

This Lab's main objectives are to getting familiar with :

- A/D conversions in both 9S12 microcontroller and arduino
- 7 segment displays.

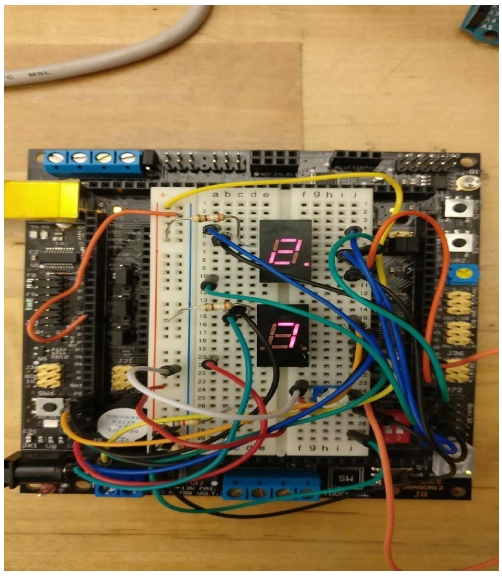
2. Procedure

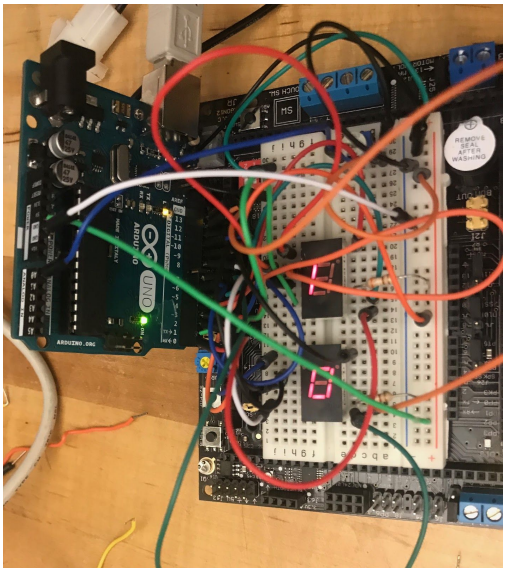
1) Open FreeScale CodeWarrior and start working on the new project. Also, work on the schematic first to make things more clear.
2) Work on the source code to initialize both PORTA and PORTB of 9S12 for outputs, then initialize ADC for an 8 bit resolution mode, wait for the A/D conversion completion and then Read the corresponding A/D result onto both of the 7 segment LED display. First for 8 bit resolution and then a 10 bit resolution.
3) Now, we will work with the Arduino board ADC values. And output them out on the same same set of 7 segment displays.
4) Make sure that the hardware is connected properly i..e, that the LED 7 segment display is connected according to the code to produce different outputs. And the same things goes for the arduino board.

3. Experiment System Specification :

***Lab hardware photo*:**

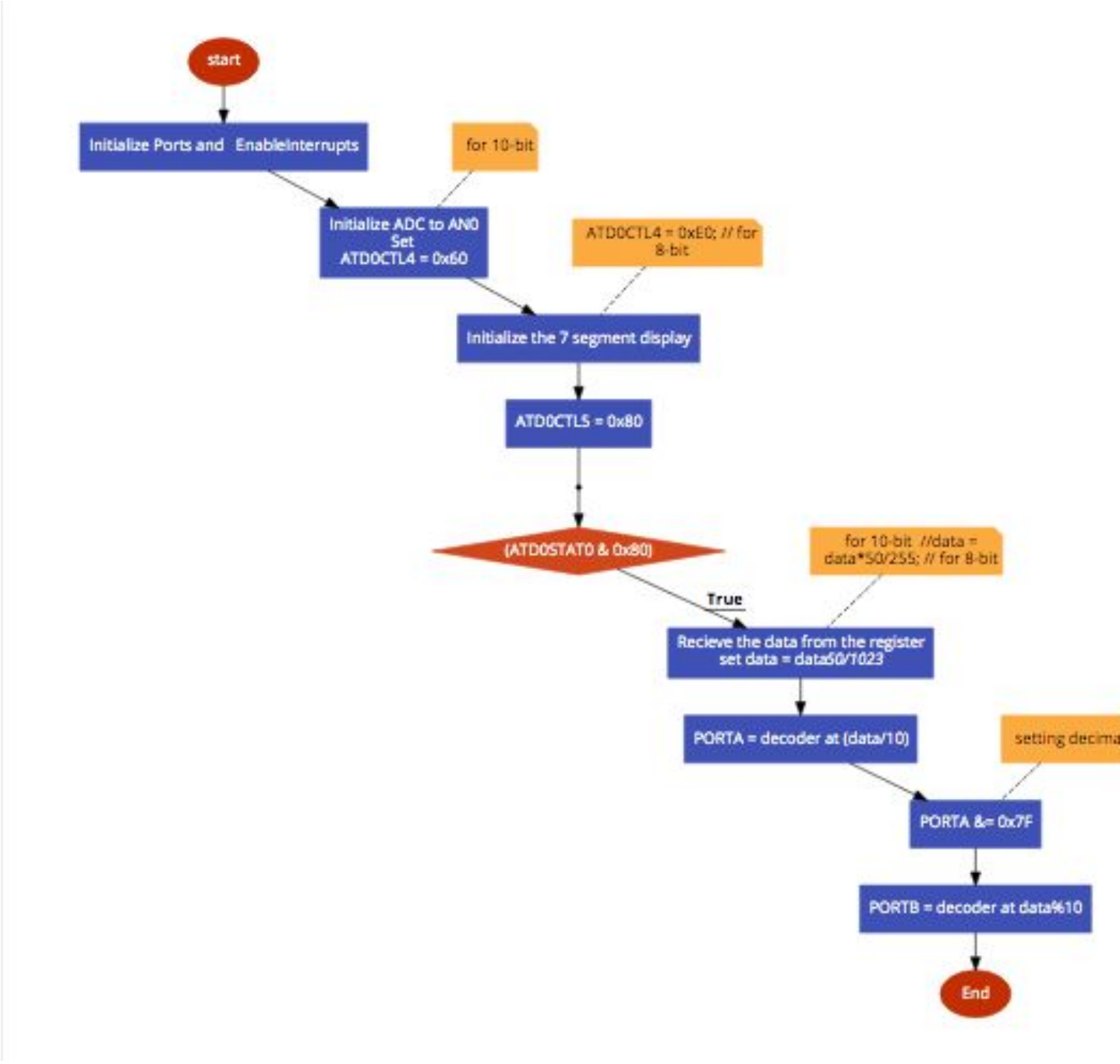
9S12 ***Arduino***



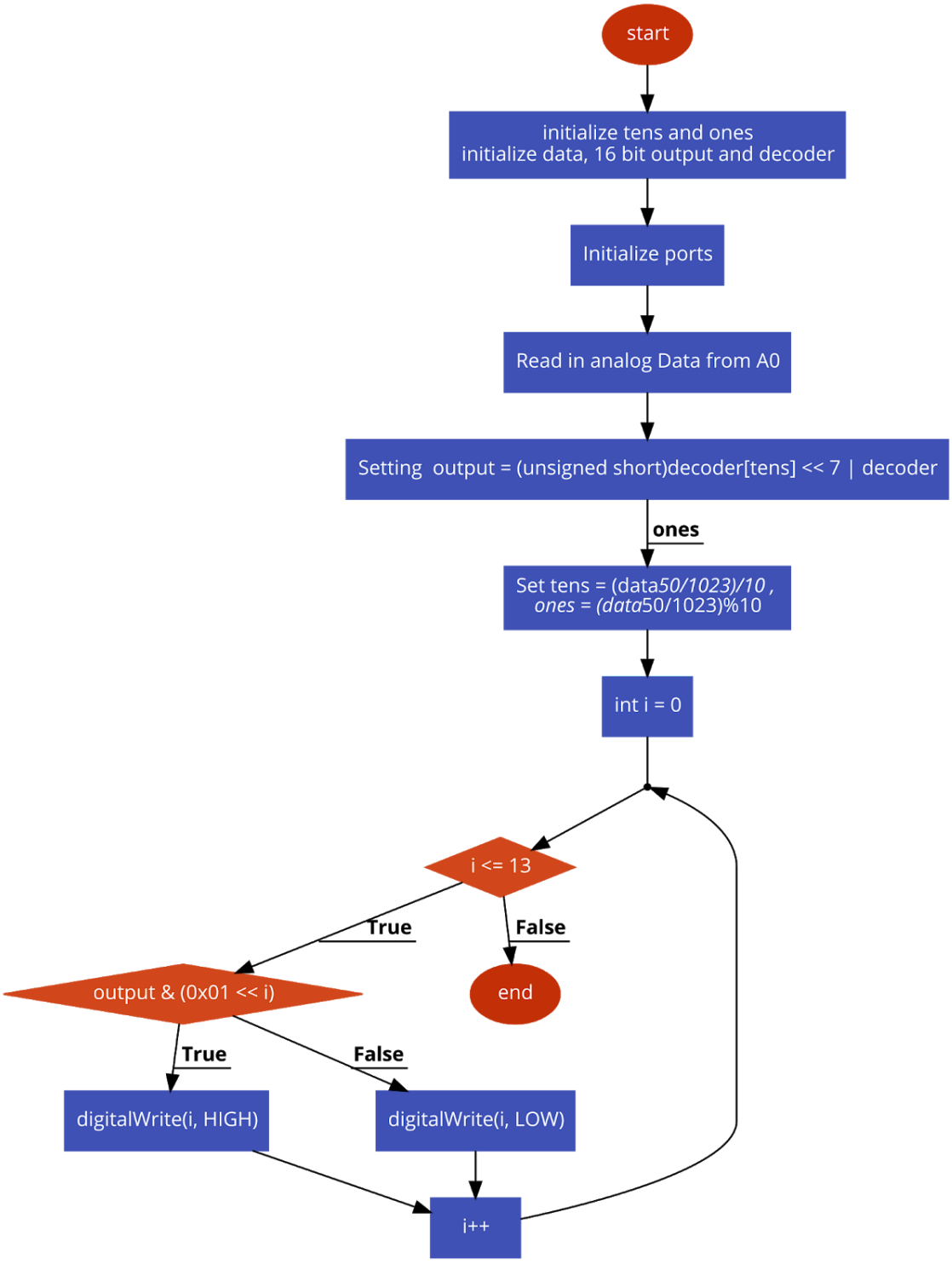


Flowchart;

9S12 microcontroller;

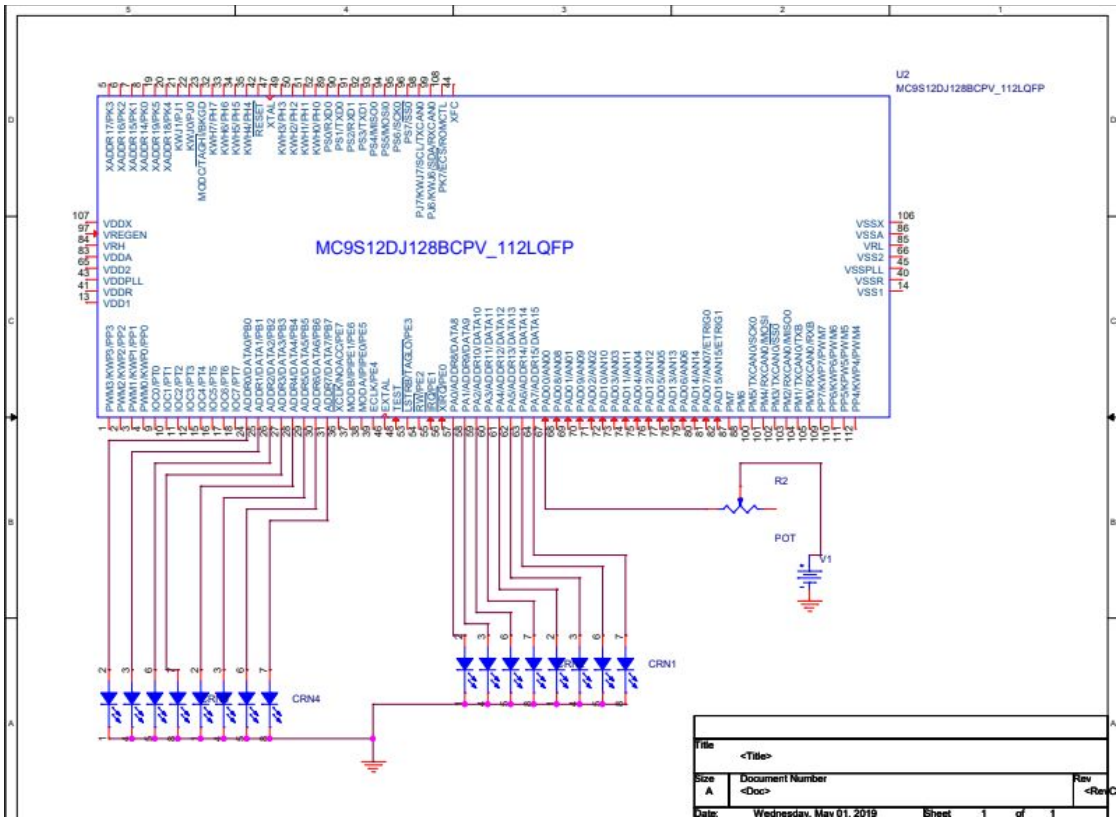


<u>Arduino board ;</u>

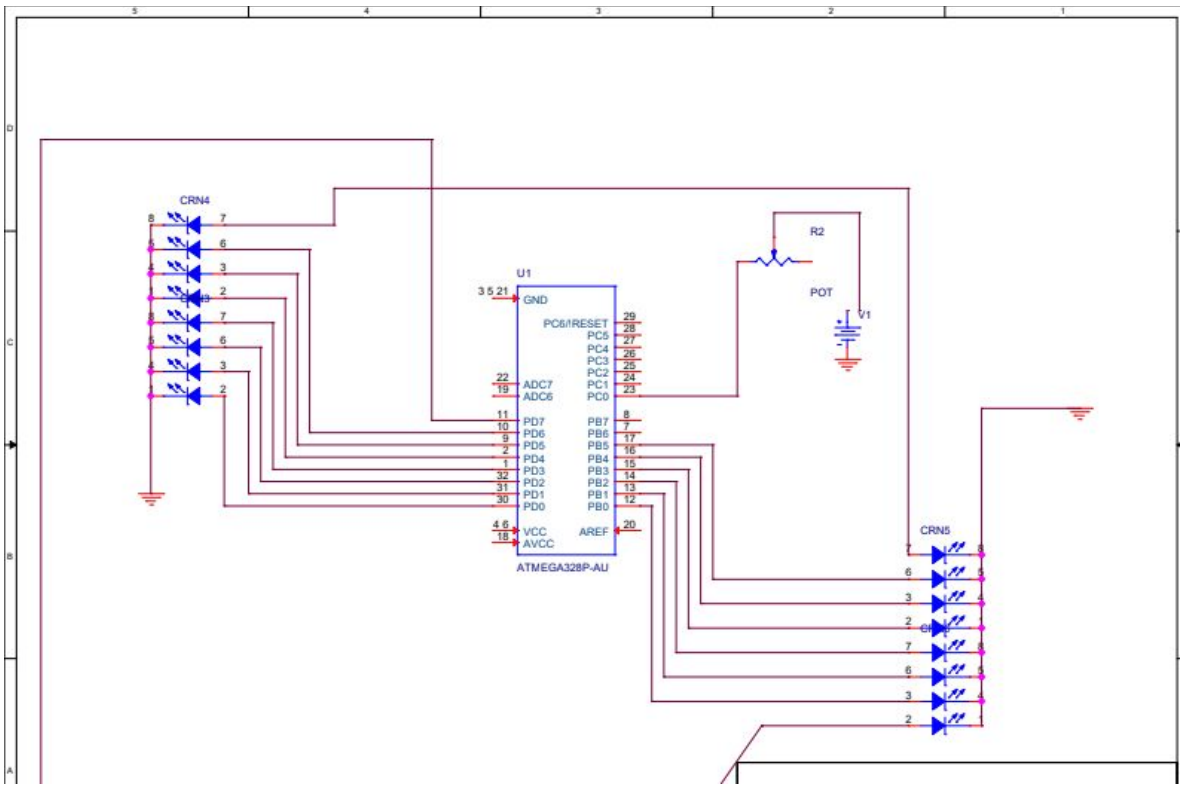

start
initialize tens and ones
initialize data, 16 bit output and decoder
Initialize ports
Read in analog Data from A0
Setting output = (unsigned short)decoder[tens] << 7 | decoder
ones
Set tens = (data50/1023)/10 ,
ones = (data50/1023)%10
int i = 0
i <= 13
True
False
end
output & (0x01 << i)
True
False
digitalWrite(i, HIGH)
digitalWrite(i, LOW)
i++

4. Hardware Design :

9S12 Controller;



Arduino Board:



5. Software Design:

- ***Dragon Board Code (Codewarrior) Part1***

```
#include <hidef.h> /* common defines and macros */
#include <mc9s12dg256.h>
#include <stdio.h>

unsigned int data;
unsigned long i;
unsigned char decimal = ~0x80 ;

void init(){
  DDRA = 0xff;
  DDRB = 0xff;
  PORTA = 0x00;
  PORTB = 0x00;
  EnableInterrupts;

  ATD0CTL2 = 0xc0;
  for (i=0;i<150000;i++);//wait 20us
  ATD0CTL3 = 0x08;
  ATD0CTL4 = 0x60;//for 10-bit
  //ATD0CTL4 = 0xE0; // for 8-bit
}

void main(void) {
  unsigned char decoder[10] =
{~0x3F,~0x06,~0x5B,~0x4F,~0x66,~0x6D,~0x7D,~0x07,~0x7F,~0x6F};

  init();
  while (1) {
    ATD0CTL5 = 0x80;
    while (!(ATD0STAT0 & 0x80));
    data = ATD0DR0;
    data = data*50/1023;//for 10-bit
    //data = data*50/255; // for 8-bit

    PORTA = decoder[data/10];
    PORTA &= 0x7F;
    PORTB = decoder[data%10];
    for(i = 0; i < 300000; i++);
  }
}
```

- ***High Level Description (Part 1) Dragon Board***
    1. Set port DDRA and DDRB to be the output port for the 7 Segment LED output ports.
    2. Turn on and set ADC ports and bits.

3. ATD0CTL4 is set for resolution, when set to 0xE0 measures 8 bit and 0x60 measures 10 bits.
4. Set decoder for 2 common Anode 7 segment LED
5. Within the loop, on timer read from ATD0DR0 and assign it to data, as a temp variable.
6. Separate data into the Tenth number and the Oneth number by divide data by 10 and mod data by 10.
7. Load each data value into decoder for proper display on 7 segment LED.

- ***Arduino Board Code (Arduino) Part2***

```
int tens = 0;
int ones = 0;
unsigned int data;
unsigned short output;

unsigned char decoder[] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};

void setup(){
    DDRB = 0xff;
    DDRD = 0xff;
    DDRC = 0x00;
}

void loop() {
    data = analogRead(A0);

    output = (unsigned short)decoder[tens] << 7 | decoder[ones];

    tens = (data*50/1023)/10;
    ones = (data*50/1023)%10;

    for (int i = 0; i <= 13; i++) {
        if (output & (0x01 << i)) {
            digitalWrite(i, HIGH);
        } else digitalWrite(i, LOW);
    }
}
```

- ***High Level Description (Part 2) Arduino***
    1. Create Tenth and Oneth number to be placeholder, data to read in value, output to be set output value.
    2. Define 7 Segment LED hex decoder.
    3. Setup DDRB and DDRD to be output ports and DDRC to be input port.
    4. Read in values from A0 and assign the value to data.
    5. Set Tenth and Oneth numbers with divide and mod.

6. Set output values, both Tenth and Oneth numbers to output.
7. Since there are 13 pins for output, use loop to check each bit of output for high and/or low, and set that bit to high and/or low as output (digitalWrite()) to the 7 segment LED.

6. Problems Encountered:

There were no major problems with this lab. The lab went pretty smooth, The minor problems which we did face was the fact that since we were not able to demo the correct functionality of Lab 3 on time, so we had to meet outside of lab hours to work on the lab in order to finish it on time. Another problem that we faced was getting the data from the ADC port in order to show it up on both 7 segment display. But we were able to fix that mainly as it required some binary to floating point conversions.

8. Questions:

1) **What is the resolution (minimum distinguishable input voltage) of a 10-bit ADC with V_{RH} = 8*V* and V_{RL} = 2*V*?.**
   Ans) 5.9mV
2) **Suppose that there is a 10-bit A/D converter with V_{RL} = 2*V* and V_{RH} = 8*V*. Find the corresponding voltage values for the A/D conversion results of 5, 110, 250 and 800.**
   Ans) 5-> **0.0293 V**
   110-> **0.6452 V**
   250 -> **1.466 V**
   800-> **4.692 V**
3) **Assume that we have a 12-bit Successive Approximation ADC, driven by a 4 KHz clock. What is the minimum conversion time of one sample with this ADC? Express your answer in seconds.**
   Ans) 0.003 secs

9. Conclusion:

This Lab was overall really interesting as we got to work with 9S12 more and understand about the A/D converter and also work with 7-Segment displays. This lab was not just limited to working with the 9S12 controller but also the arduino uno board , where we got to work with Arduino programming. For this Both Leya and Tanish worked together on software and hardware designs