Beste Öztop 2375624
Deniz Karakay 2443307 12.12.2022

EE447 Introduction to Microprocessors

Preliminary Work-5

Question 1

Program the ATD conversion system on the board, in C language, to convert the analog signal to a 12-bit number between 0x000 and 0xFFF (4095). The input will be taken from PE3. The output value should be stored in an integer variable. Put a screenshot of Keil showing this variable on your report.

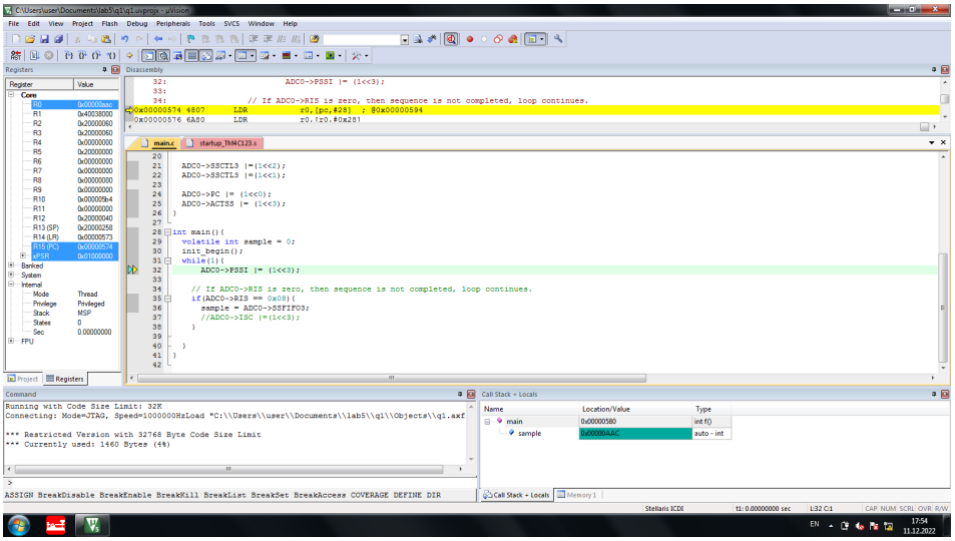


Figure 1: *The potentiometer is set to have 0x00000AAC for "sample" variable*

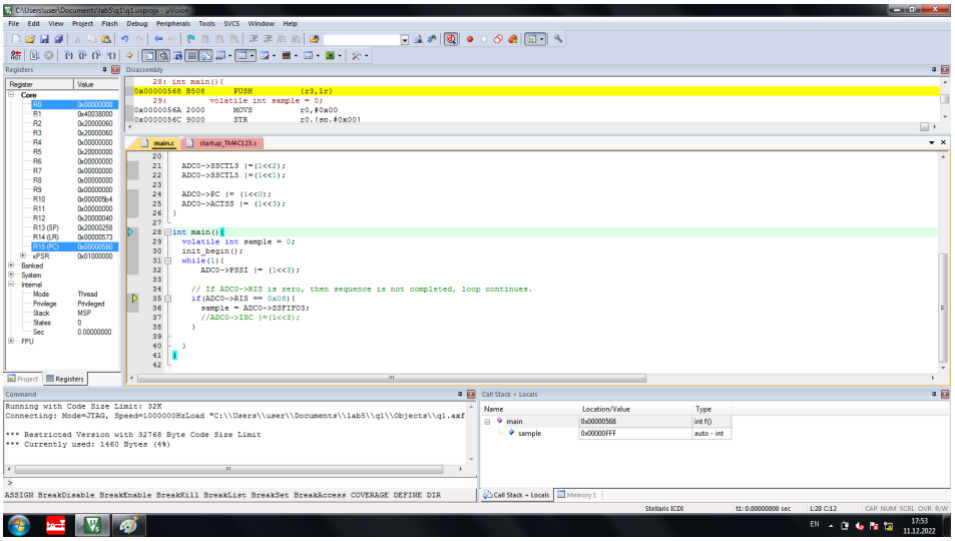


Figure 2: *The potentiometer is set to have 0x00000FFF for "sample" variable*

Beste Öztop 2375624
Deniz Karakay 2443307 12.12.2022

Question 2

Assume that there is a 1.65 Volt DC offset in your input. Subtract the value corresponding to this offset from your reading. Note that new result is a signed number.

We have converted the type of "sample" variable from int to float at this point. The main idea is to have signed values with several decimal points.

$$3.3V \text{ corresponds to } 4095 \text{ without the offset}$$
$$Thus, 1.65\ V \text{ offset corresponds to the addition of } \frac{4095}{2} = 2047.5$$
$$This\ value\ is\ subtracted\ from\ the\ reading\ to\ find\ the * actual * value.$$

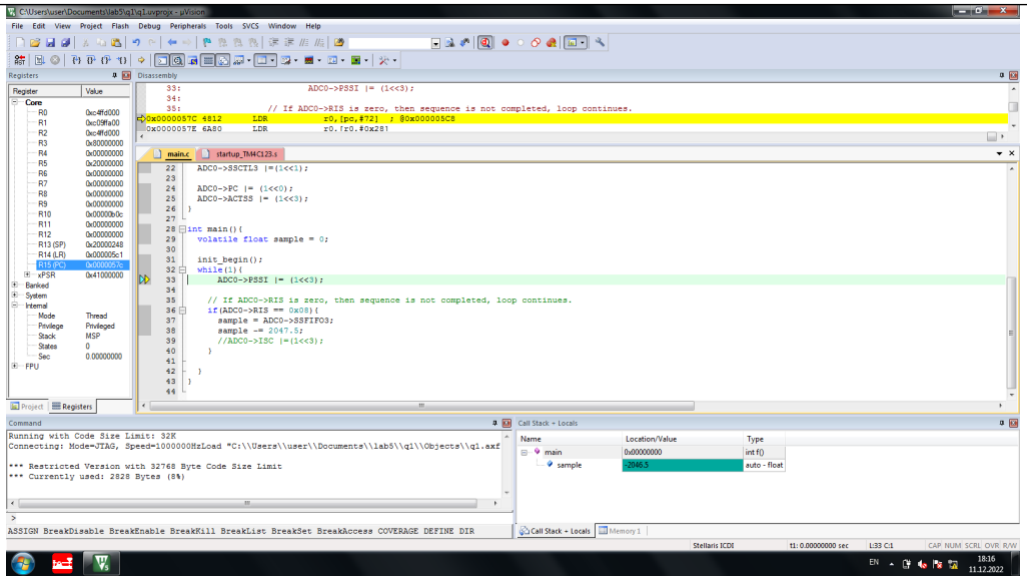


Figure 3: *The potentiometer is set to have -2046.5 for "sample" variable (about -1.65 V, without offset)*

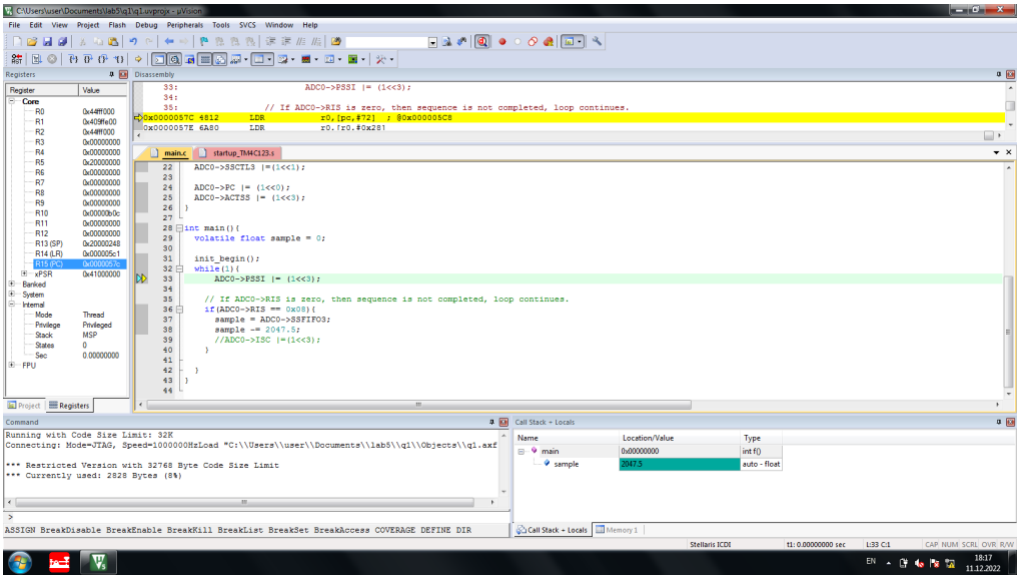


Figure 4: *The potentiometer is set to have 2047.5 for "sample" variable (about 1.65 V, without offset)*

Beste Öztop 2375624
Deniz Karakay 2443307 12.12.2022

Question 3

Convert the resulting value to a char array containing a decimal representation with two decimal places (X.YZ) between -1.65 and 1.65. Put a screenshot on your report.

We have used “sprintf()” function to have the X.YZ representation of a float variable.

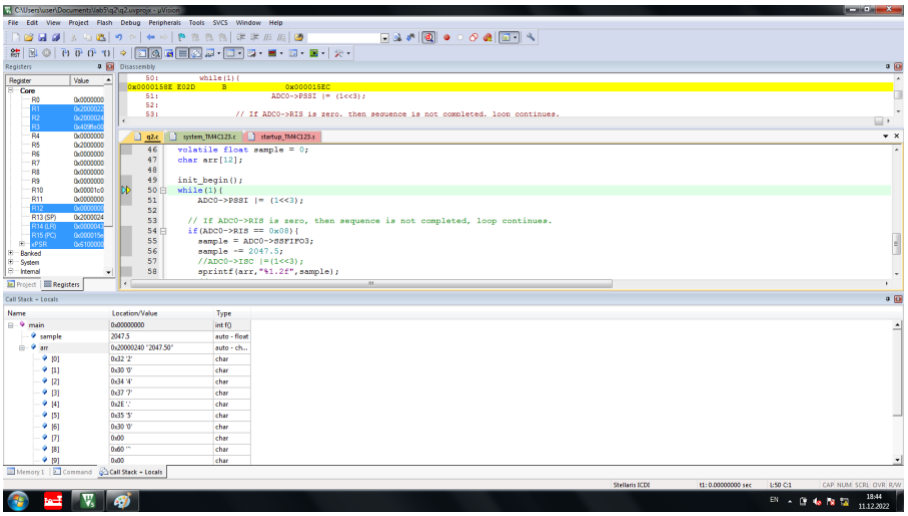


Figure 5: *The potentiometer is set to have 2047.5 for “sample” variable*

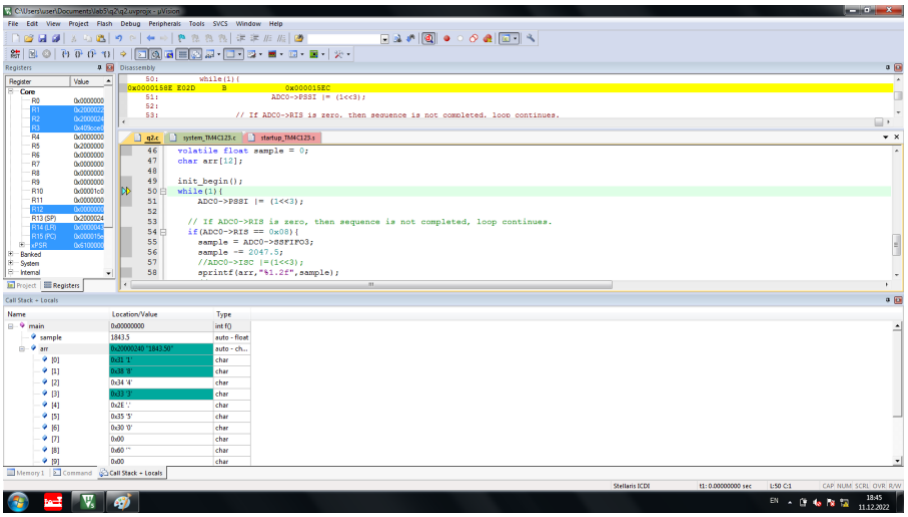


Figure 6: *The potentiometer is set to have 1843.5 for “sample” variable*

Beste Öztop 2375624
Deniz Karakay 2443307 12.12.2022

Question 4

Write a C program that, in an infinite loop, samples the ATD at 1 second intervals, removes the 1.65 Volt DC offset, converts the result to char array and outputs the result on Termite. You may write and use a 1-second delay subroutine. Put a screenshot of Termite on your report.

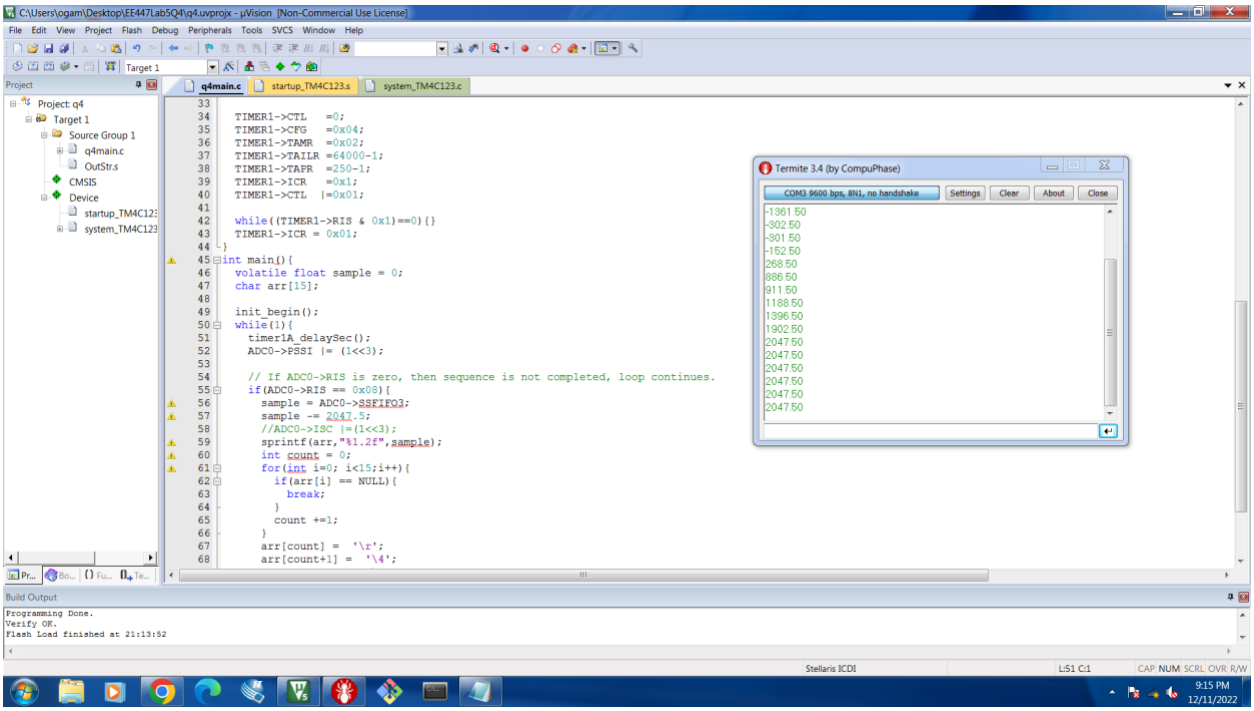


Figure 7: *The readings from the pot is read in every 1 sec, printed to Termite*

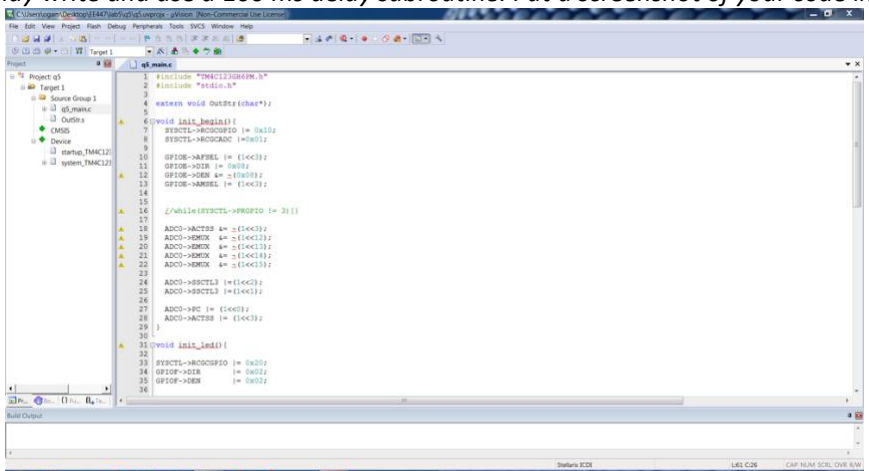
The 1 sec delay subroutine:

```
void timer1A_delaySec(){
SYSCTL->RCGCTIMER |=2;

   TIMER1->CTL    =0;
   TIMER1->CFG    =0x04;
   TIMER1->TAMR   =0x02;
   TIMER1->TAILR =64000-1;
   TIMER1->TAPR   =250-1;
   TIMER1->ICR    =0x1;
   TIMER1->CTL   |=0x01;

   while((TIMER1->RIS & 0x1)==0){}
   TIMER1->ICR = 0x01;
}
```

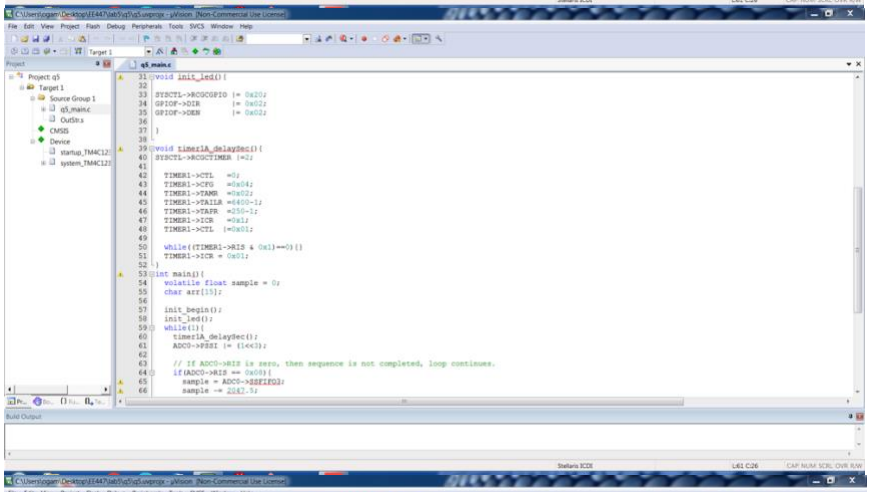
Beste Öztop 2375624
Deniz Karakay 2443307 12.12.2022

Question 5

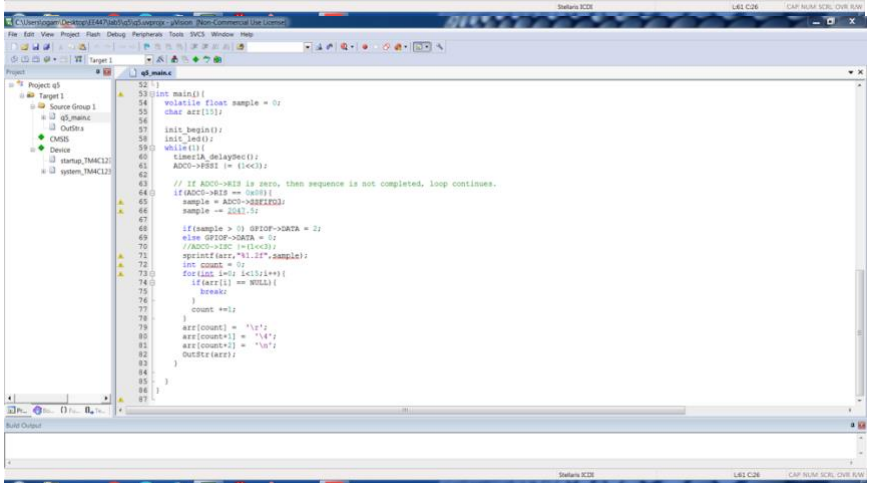
Write another C program that, in an infinite loop, samples the ATD at 0.1 second intervals. If the result is larger than 1.65 Volt, the onboard red LED (PF1) is turned on. If the result is smaller than 1.65 Volt, red LED is turned off. You may write and use a 100 ms delay subroutine. Put a screenshot of your code in your report.



```
#include "TM4C123GH6PM.h"
#include "stdio.h"

extern void OutStr(char*);

void init_begin(){
  SYSCTL->RCGCGPIO |= 0x10;
  SYSCTL->RCGCADC |=0x01;

  GPIOE->AFSEL |= (1<<3);
  GPIOE->DIR |= 0x08;
  GPIOE->DEN &= ~0x08;
  GPIOE->AMSEL |= (1<<3);


  //while(SYSCTL->PRGPIO != 3){}

  ADC0->ACTSS &= ~(1<<3);
  ADC0->EMUX  &= ~(1<<12);
  ADC0->EMUX  &= ~(1<<13);
  ADC0->EMUX  &= ~(1<<14);
  ADC0->EMUX  &= ~(1<<15);

  ADC0->SSCTL3 |=(1<<2);
  ADC0->SSCTL3 |=(1<<1);

  ADC0->PC |= (1<<0);
  ADC0->ACTSS |= (1<<3);
}

void init_led(){

SYSCTL->RCGCGPIO |= 0x20;
GPIOF->DIR       |= 0x02;
GPIOF->DEN       |= 0x02;
```



```
void init_led(){

SYSCTL->RCGCGPIO |= 0x20;
GPIOF->DIR       |= 0x02;
GPIOF->DEN       |= 0x02;

}

void timer1A_delay1Sec(){
SYSCTL->RCGCTIMER |=2;

  TIMER1->CTL   =0;
  TIMER1->CFG   =0x04;
  TIMER1->TAMR  =0x02;
  TIMER1->TAILR =64000-1;
  TIMER1->TAPR  =250-1;
  TIMER1->ICR   =0x1;
  TIMER1->CTL  |=0x01;

  while((TIMER1->RIS & 0x1)==0){}
  TIMER1->ICR = 0x01;
}
int main(){
  volatile float sample = 0;
  char arr[15];

  init_begin();
  init_led();
  while(1){
    timer1A_delay1Sec();
    ADC0->PSSI |= (1<<3);

    // If ADC0->RIS is zero, then sequence is not completed, loop continues.
    if(ADC0->RIS == 0x08){
      sample = ADC0->SSFIFO3;
      sample -= 2047.5;
```



```
}
int main(){
  volatile float sample = 0;
  char arr[15];

  init_begin();
  init_led();
  while(1){
    timer1A_delay1Sec();
    ADC0->PSSI |= (1<<3);

    // If ADC0->RIS is zero, then sequence is not completed, loop continues.
    if(ADC0->RIS == 0x08){
      sample = ADC0->SSFIFO3;
      sample -= 2047.5;

      if(sample > 0) GPIOF->DATA = 2;
      else GPIOF->DATA = 0;
      //ADC0->ISC |=(1<<3);
      sprintf(arr,"%1.2f",sample);
      int count = 0;
      for(int i=0; i<15;i++){
        if(arr[i] == NULL){
          break;
        }
        count +=1;
      }
      arr[count] =  '\r';
      arr[count+1] =  '\4';
      arr[count+2] =  '\n';
      OutStr(arr);
    }

  }
}
```