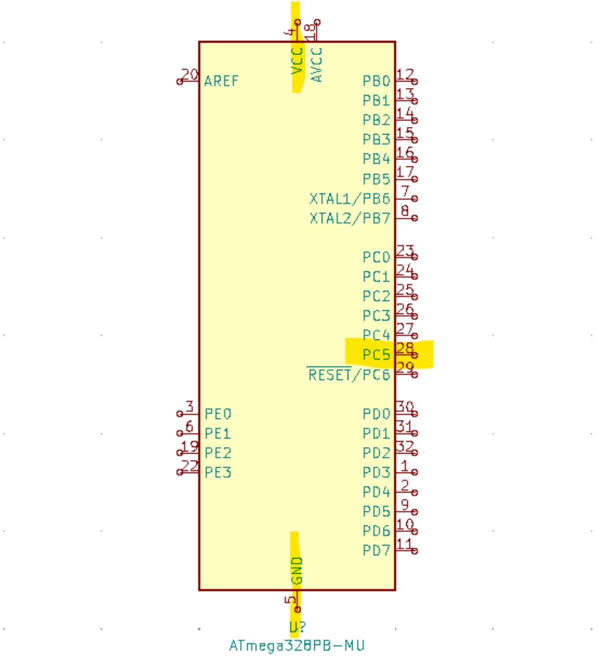
CPE301 – SPRING 2021

Design Assignment 3A

Student Name: Brian Wolak
Student #: 2000509437
Student Email: wolak@unlv.nevada.edu
Primary Github address: https://github.com/brianwolak/submission_da.git
Directory: https://github.com/brianwolak/submission_da/tree/main/DA_3A

Task 1:

The purpose of this design assignment is to create an AVR C code that will monitor an LM35 temperature sensor connected to analog input PC5 and display the temperature in both Fahrenheit and Celsius every 0.25 seconds to the terminal using a timer with interrupt of 0.05s. The 0.05s interrupt will trigger an increment on a counter which will then initialize the read sequence for ADC after 5 completions. This data will then be converted to a string before being output to the terminal. Please note the video link will show both task outputs together in the same video.



ATMEGA328pb Ports Used in Design Assignment 3A

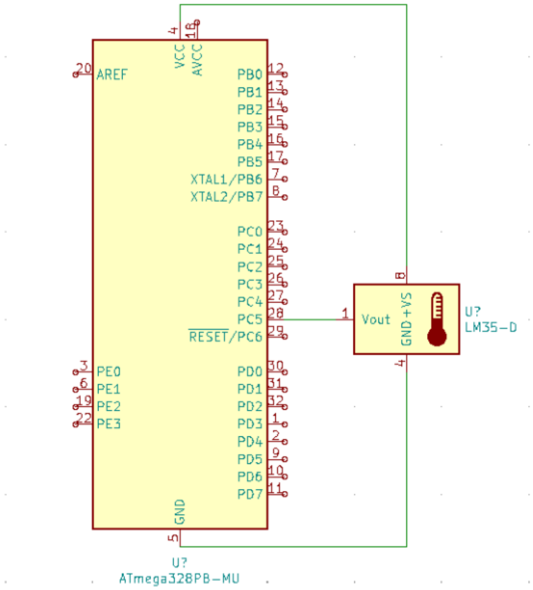
Video Link:

https://youtu.be/ZSA_umiEL_o

C Code:

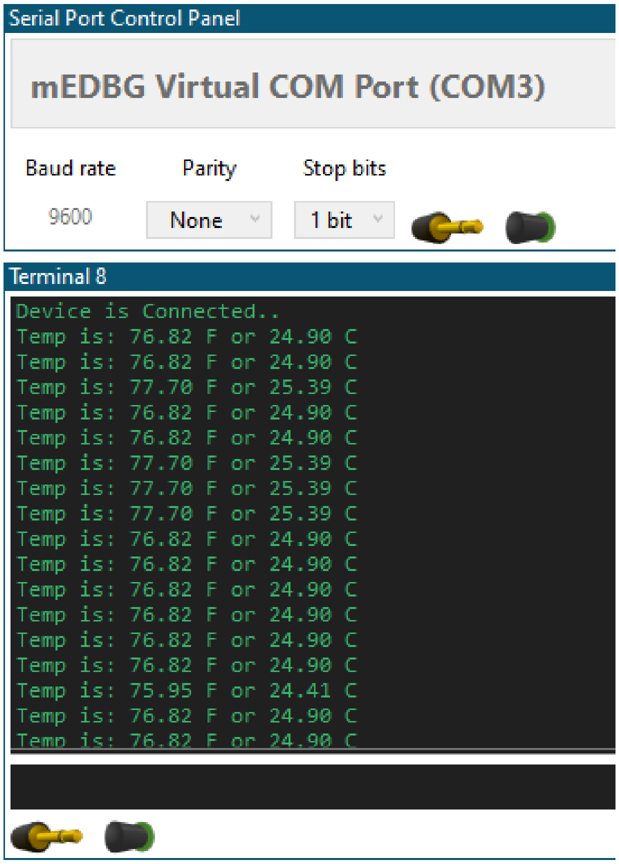
```
#define F_CPU 16000000UL
#define BAUD 9600
#include <avr/io.h>
#include <util/setbaud.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdio.h>

int timer_count;                          //counter looping value
volatile uint8_t tempread;         //temp read value from ADC
char display1[20];                        //fahrenheit string
char display2[20];                        //celsius string

void USART_initialize(void){
       UBRR0H = UBRRH_VALUE;
       UBRR0L = UBRRL_VALUE;
       UCSR0C = _BV(UCSZ01) | _BV(UCSZ00);       //8-bit data transfer
       UCSR0B = _BV(RXEN0) | _BV(TXEN0); //enable RX & TX
}

void interrupt_set(void){
       TCCR1A = 0x00;
       TCCR1B = 0x0D;              //prescale 1024, CTC mode
       OCR1A = 0x030C;             //0.05s counter delay value
       TIMSK1 = 0x02;              //enable OCR1A interrupt flag
       sei();
}

void adc_set(void){
       //reference Vcc, ADC5 input, PINC.5
       ADMUX |= (0<<REFS1) | (1<<REFS0) | (0<<ADLAR) | (0<<MUX3) | (1<<MUX2) | (0<<MUX1)
| (1<<MUX0);
       //enable ADC, 128 prescale
       ADCSRA = (1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (1<<ADPS2) |
(1<<ADPS1) | (1<<ADPS0);
}

void USART_TX_string(char *data){
       while (*data != '\0'){                    //while data DNE 0
              while (!(UCSR0A & (1<<UDRE0)));    //while UNDRE0 DNE 1
              UDR0 = *data;                      //UDR0 gets data value
              data++;                            //next data value
       }
}

void ADC_READ(void){
       ADCSRA |= (1<<ADSC);                      //start transfer
       while((ADCSRA & (1<<ADIF)) == 0);         //wait for ADIF flag
       ADCSRA |= (1 << ADIF);                    //clear ADIF flag
       tempread = ADC;                           //store temp value
}
```

```
void USART_TX_FLOAT(char data){                 //convert for serialplot output
       UDR0 = data;
}

int main(void)
{
       USART_initialize();                      //call USART initialize function
       interrupt_set();                                //call interrupt set function
       adc_set();                                      //call ADC setup function
       float tempc;                                    //celsius temp
       float tempf;                                    //fahrenheit temp
       //confirm connection in terminal
       USART_TX_string("Device is Connected..\r\n");
       while (1)
       {
              if (timer_count == 5){      //enter loop once 0.25s time is reached
                     ADC_READ();                       //call ADC read function
                     tempc = ((tempread * 500.0) / 1024);//tempc from binary to decimal
                     tempf = (tempc * 1.8) + 32;       //convert to farenheit
                     sprintf(display1, "%.2f", tempf); //convert tempf to string
                     _delay_ms(10);                    //10ms delay
                     sprintf(display2, "%.2f", tempc); //convert tempc to string
                     _delay_ms(10);                    //10ms delay

//terminal print statements for both temp C & F
                     USART_TX_string("Temp is: ");
                     USART_TX_string(display1);
                     USART_TX_string(" F or ");
                     USART_TX_string(display2);
                     USART_TX_string(" C");
                     USART_TX_string("\n");
//uncomment one TX_FLOAT line and comment TX_string lines above to output to SERIALPLOT
                     //USART_TX_FLOAT(tempc);
                     //USART_TX_FLOAT(tempf);
                     timer_count = 0;                  //reset 0.05s per step counter
              }
       }
}

ISR(TIMER1_COMPA_vect){
       //increment .05s counter once TIM1 COMPA match is made
       timer_count++;
}
```



Design Assignment 3A Circuit

ATMEL Terminal Output:



Serial Port Control Panel

mEDBG Virtual COM Port (COM3)

Baud rate 9600

Parity None

Stop bits 1 bit

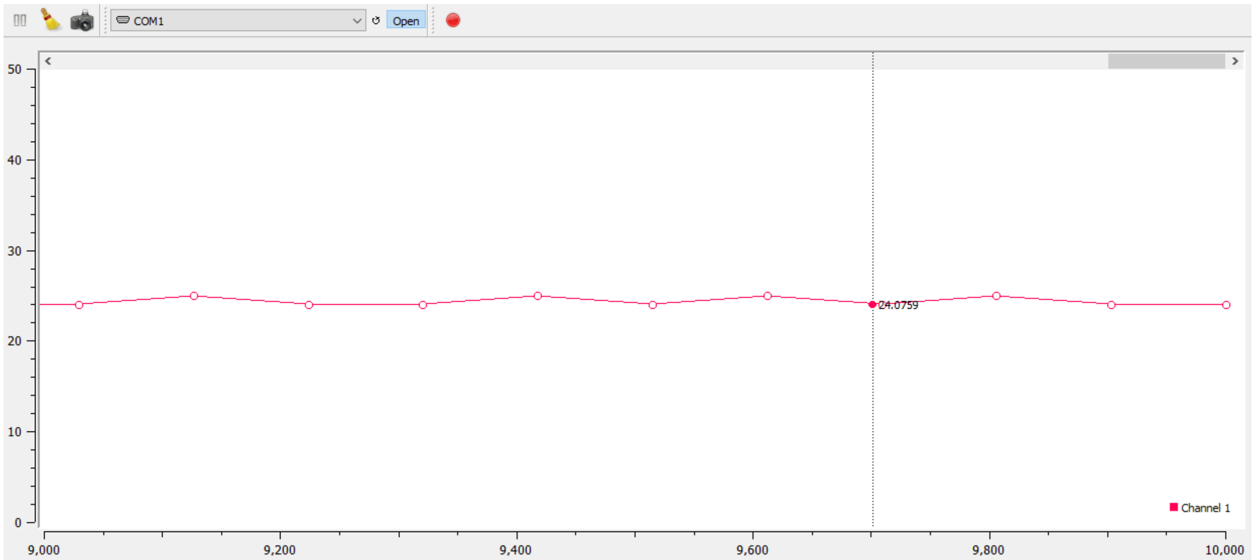
Terminal 8

```
Device is Connected..
Temp is: 76.82 F or 24.90 C
Temp is: 76.82 F or 24.90 C
Temp is: 77.70 F or 25.39 C
Temp is: 76.82 F or 24.90 C
Temp is: 76.82 F or 24.90 C
Temp is: 77.70 F or 25.39 C
Temp is: 77.70 F or 25.39 C
Temp is: 77.70 F or 25.39 C
Temp is: 76.82 F or 24.90 C
Temp is: 76.82 F or 24.90 C
Temp is: 76.82 F or 24.90 C
Temp is: 76.82 F or 24.90 C
Temp is: 76.82 F or 24.90 C
Temp is: 76.82 F or 24.90 C
Temp is: 75.95 F or 24.41 C
Temp is: 76.82 F or 24.90 C
Temp is: 76.82 F or 24.90 C
```

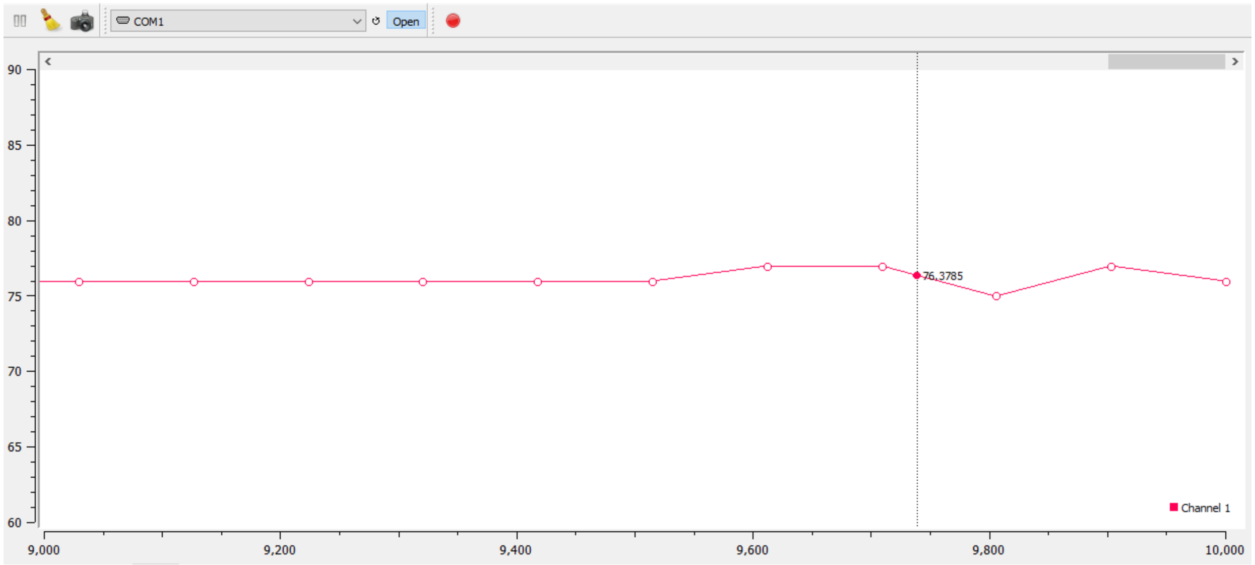
Terminal Window Displaying Connection Confirmation and Temperatures

Task 2:

Task two uses the exact same hardware and code components as task one, but this time we will be reading to a SerialPlot program to show temperature values taken over time. With a couple lines of commenting and uncommenting the same code can be used for both tasks. Please see the video for how to do this as well as the code. Below we can see the SerialPlot program showing Celsius and Fahrenheit temperatures being displayed as expected.



SerialPlot Displaying Celsius Temperature over time



SerialPlot Displaying Fahrenheit Tempertaure over time