CPE301 – SPRING 2019

Design Assignment 3B

Student Name: Mohamad Jundi

Student #: 8000321867

Student Email: jundi@unlv.nevada.edu

Primary Github address: https://github.com/MohamedJundi1994/Submission\_DA.git

Directory: Documents\School\CPE 301\Repository\CPE\_301\DesignAssignments\DA3B

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

USB port => Xplained Mini => 5V (left) => PC5 output (middle) => GND (right) => LM34 (Temp Sensor)

USB port2 => FTDI board => GND => Xplained Mini => Transmitter => PD0 => Receiver => PD1

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**

My Code:

#define *F\_CPU* 16000000UL

#define BAUD\_RATE 9600

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

void usart\_init ();

void usart\_send (unsigned char ch);

int main (void)

{

usart\_init ();

/\*\* Setup and enable ADC \*\*/

ADMUX = (0<<REFS1)| // Reference Selection Bits

(1<<REFS0)| // AVcc - external cap at AREF

(0<<ADLAR)| // ADC Left Adjust Result

(1<<MUX2)| // Analog Channel Selection Bits

(0<<MUX1)| // ADC4 (PC5 PIN28)

(1<<MUX0);

ADCSRA = (1<<ADEN)| // ADC ENable

(0<<ADSC)| // ADC Start Conversion

(0<<ADATE)| // ADC Auto Trigger Enable

(0<<ADIF)| // ADC Interrupt Flag

(0<<ADIE)| // ADC Interrupt Enable

(1<<ADPS2)| // ADC Prescaler Select Bits

(0<<ADPS1)|

(1<<ADPS0);

TCCR1B |= (1 << CS12 | (1 << CS10)) ; // Used to set prescaler of 1024

TIMSK1 = (1 << TOIE1); // TOIE1 will be set high and enable overflow

TCNT1 = 49911; // Set to count from 49911 to count up to 65535 to achieve 1 second

sei(); // Declare global interrupt

while (1)

{

// Infinite loop

}

}

ISR(TIMER1\_OVF\_vect)

{

ADCSRA|=(1<<ADSC); //start conversion

while((ADCSRA&(1<<ADIF))==0);//wait for conversion to finish

ADCSRA |= (1<<ADIF);

int a = ADCL;

a = a | (ADCH<<8);

a = (a/1024.0) \* 5000/10;

usart\_send((a/100)+'0');

a = a % 100;

usart\_send((a/10)+'0');

a = a % 10;

usart\_send((a)+'0');

usart\_send('\r');

TCNT1 = 49911; // Reset TCNT value for loop

}

void usart\_init (void)

{

UCSR0B = (1<<TXEN0);

UCSR0C = (1<< UCSZ01)|(1<<UCSZ00);

UBRR0L = *F\_CPU*/16/BAUD\_RATE-1;

}

void usart\_send (unsigned char ch)

{

while (! (UCSR0A & (1<<UDRE0))); //wait until UDR0 is empty

UDR0 = ch; //transmit ch

}

void usart\_print(char\* str)

{

int i = 0;

while(str[i] != 0)

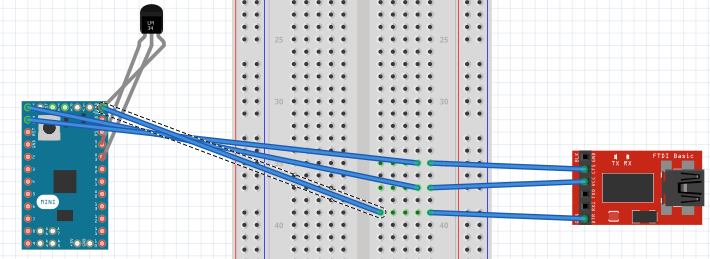
usart\_send(str[i]);

}

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

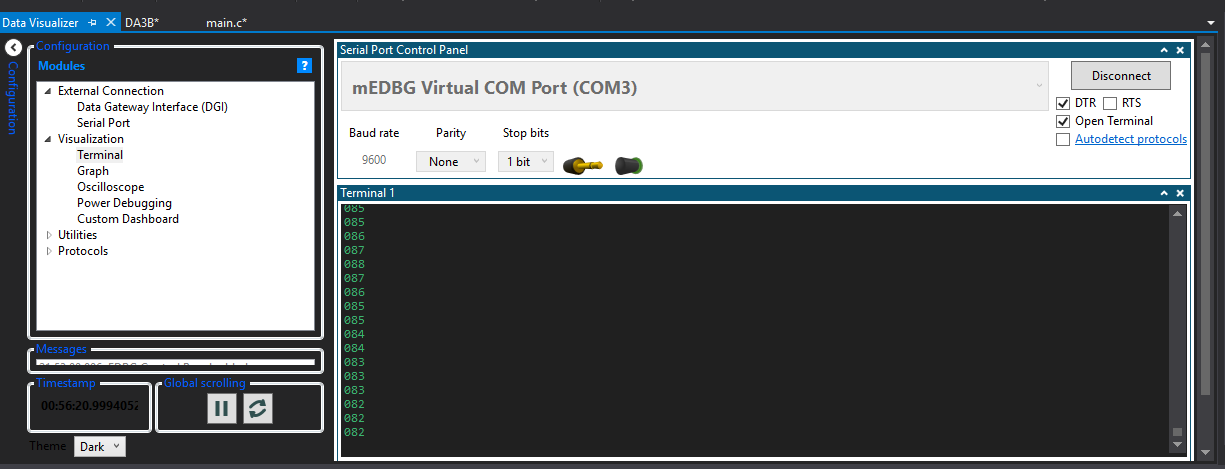
All code is in number 2.

1. **SCHEMATICS**



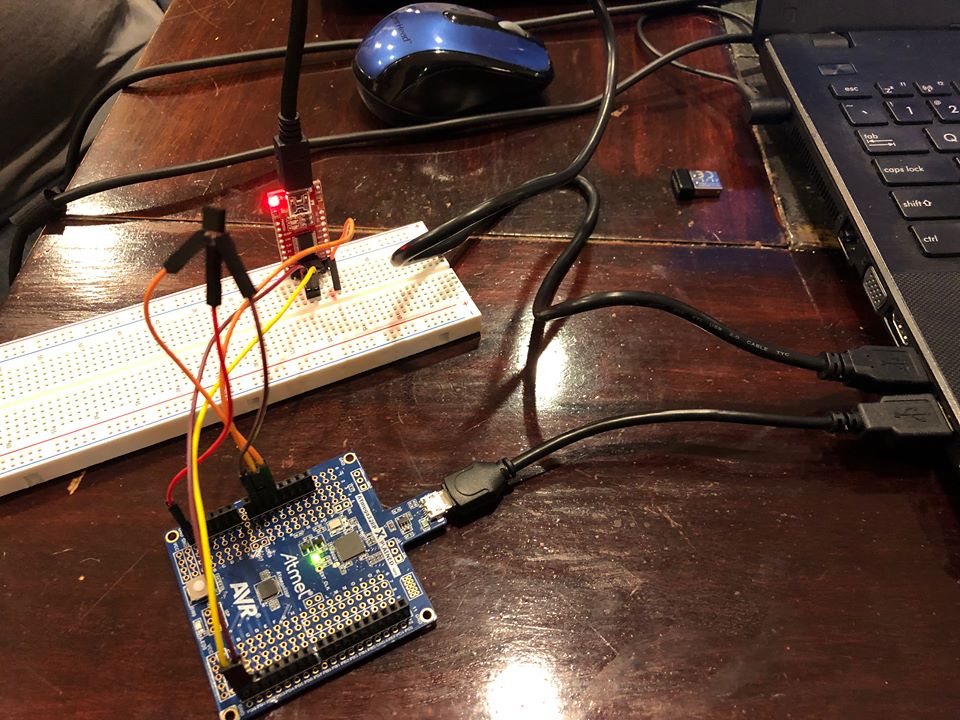
**NOTE:** Used the Arduino board to represent Xplained MINI

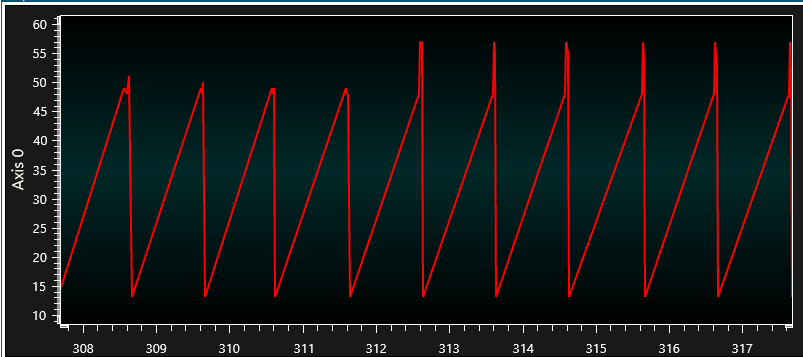
1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



**Next Page**

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

****



1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=xjNccczeVUA&feature=share&fbclid=IwAR1ZMhSVjuDUcfv9DPCW78ZO-lub1gBf8AtTdy2I4tjuH3TIzTOKzElJShE>

1. **GITHUB LINK OF THIS DA**

Link: https://github.com/MohamedJundi1994/Submission\_DA.git

This assignment submission is my own, original work.

MOHAMAD JUNDI