CPE301 – SPRING 2019

Design Assignment 3B

Student Name: Allis Hierholzer

Student #: 2000160356

Student Email: hierholz@unlv.nevada.edu

Primary Github address: https://github.com/acexhp/submission\_da.git

Directory: Repository/cpe301/DesignAssignment/DA3B

Task:

The goal of the assignment is to modify the above codes to do the following:

1. Write a C AVR program that will monitor the LM34/35 connected to an Analog pin (PC5) to display the temperature in F on the serial terminal every 1 sec. Use a timer with interrupt for the 1 sec delay. Use a FTDI chip for serial to USB conversion.

2. Use the ATMEL Studio Data Visualizer or any Charting program to display the values in time.n.

Submission:

The following are required for successful completion of the design assignment:

a. AVR C code that has been compiled and working.

b. The C code should be well documented with explanation of every instruction.

c. A word document that contains the flow chart of the assembly code along with the snapshots of the schematics, components connected on the breadboard and screen shoots.

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

* Breadboard
* Wires
* USB Cables
* ATMEGA328P XPLAINED MINI
* ATMEL STUDIO 7.0
* FTDI chip

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

/\*

\* DA3B.c

\*/

#define *F\_CPU* 16000000UL

#define BAUD\_RATE 9600

#include <avr/interrupt.h>

#include <avr/io.h>

#include <util/delay.h>

int count\_ovfl=0; //counter for overflow

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)| // ADC5 (PC5 PIN27)

(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);

TIMSK0 |= (1<<TOIE0);

TCNT0 = 0; // setting initial value for counter

sei(); // enable global interrupts

TCCR0B |=(1<<CS02); // setting prescaler to 256

while (1)

{

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

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

ADCSRA |= (1<<ADIF);

if (count\_ovfl==244) // data is updated if overflow is 244 times

{

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');

count\_ovfl=0;

}

}

return 0;

}

ISR(TIMER0\_OVF\_vect)

{

while (!(TIFR0 & 0X01)==0);

count\_ovfl++; //increasing overflow counter

TCNT0=0X00; //resetting counter to zero

TIFR0=0X01; // reset the overflow flag

}

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)

{

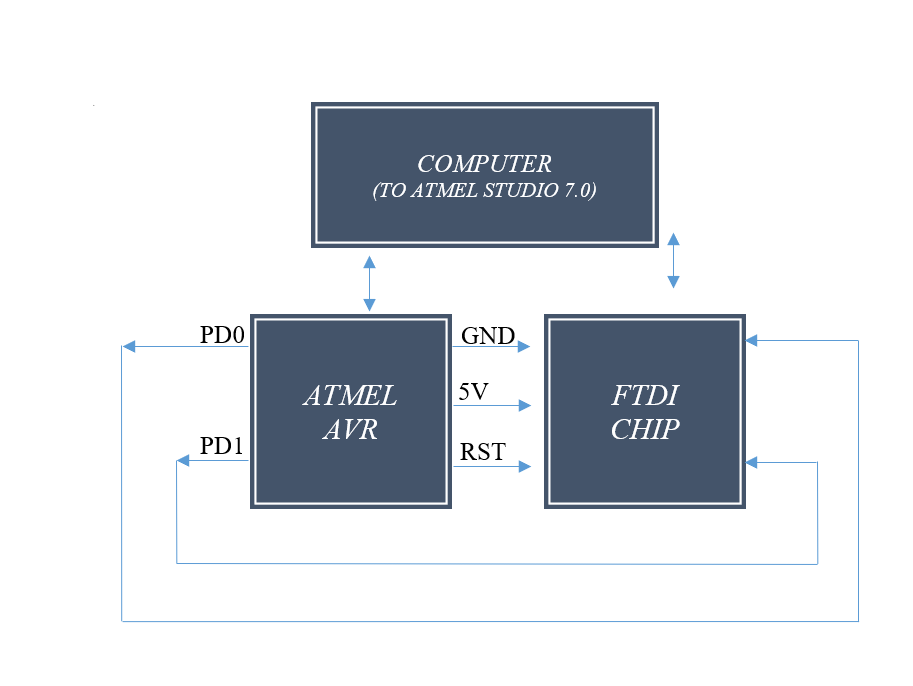
int i = 0;

while(str[i] != 0)

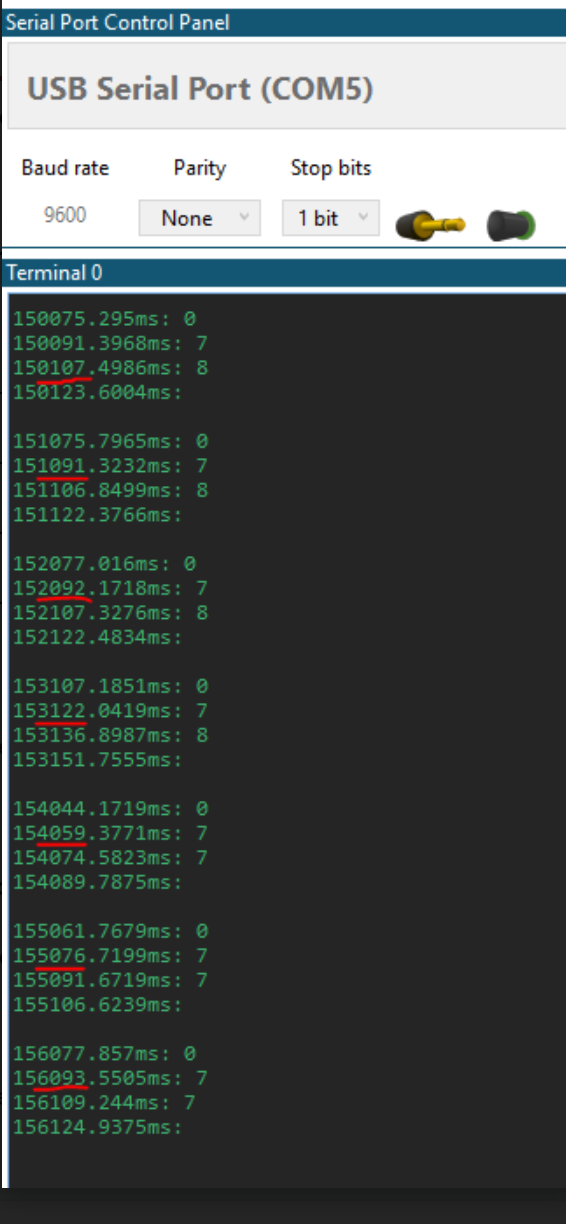
usart\_send(str[i]);

}

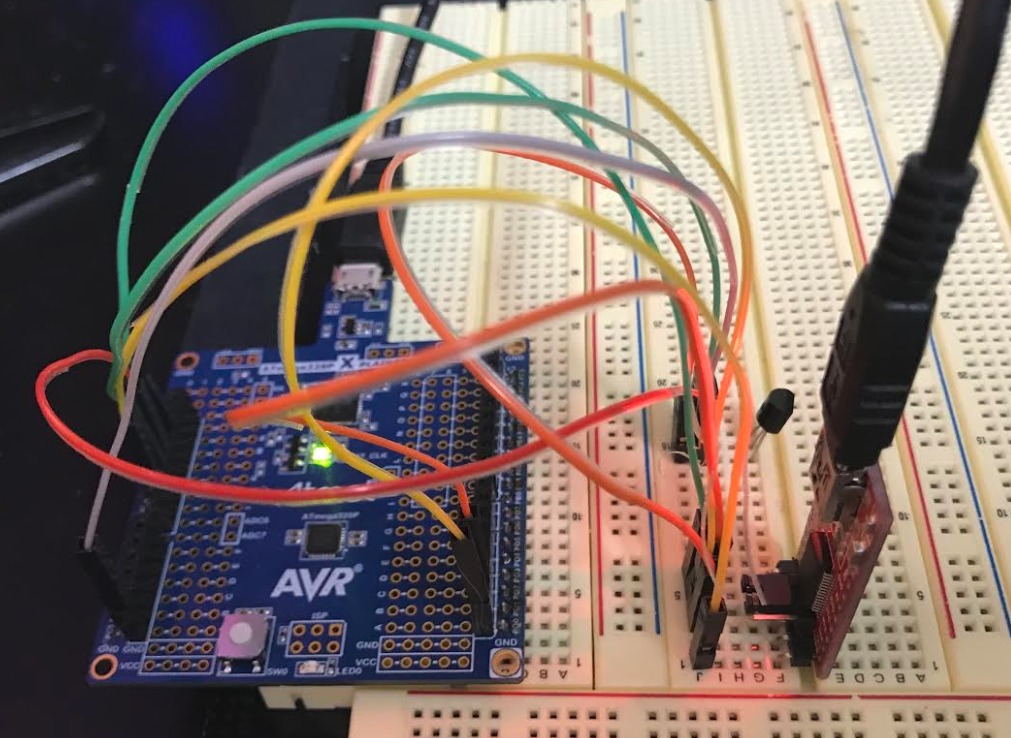
1. **SCHEMATICS**

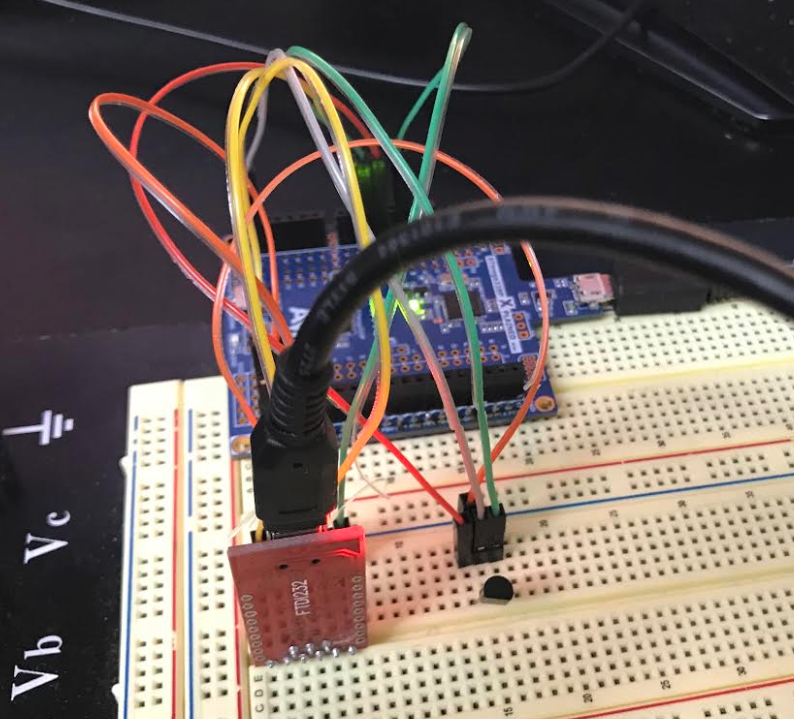
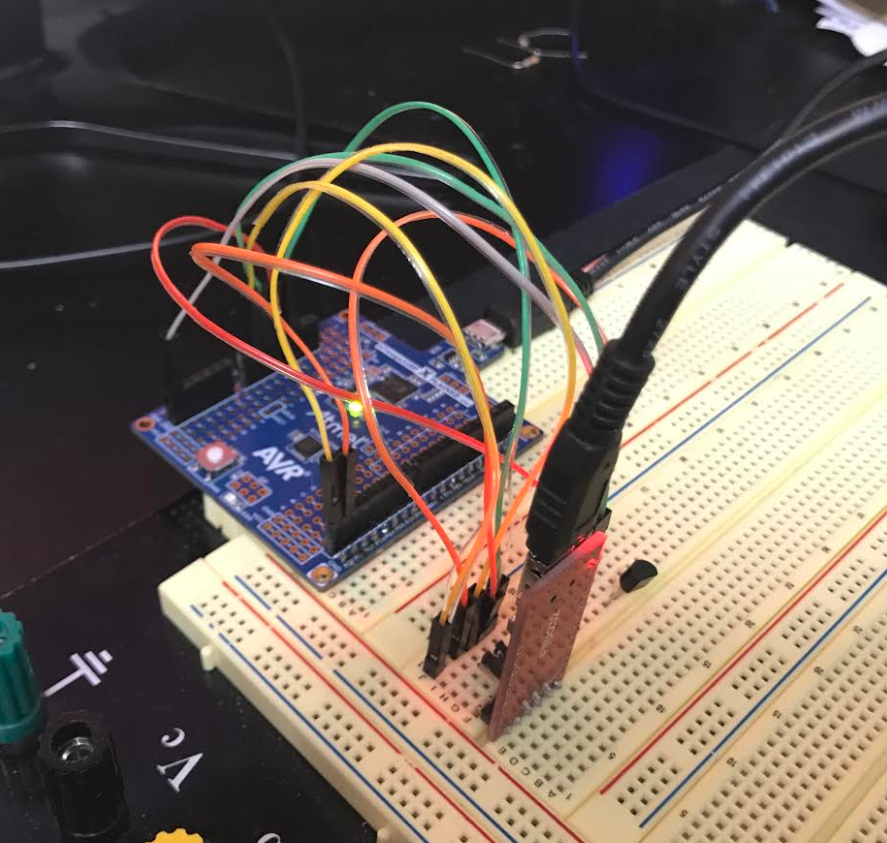


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



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





1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/TrDcCMflFdY>

1. **GITHUB LINK OF THIS DA**

<https://github.com/acexhp/submission_da.git>

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

Allis Hierholzer