CPE301 – SPRING 2019

Design Assignment X

Student Name: Tian Chen

Student #: 2001086575

Student Email: chent5@unlv.nevada.edu

Primary Github address: <https://github.com/Chentian12138/AAAABBBBB>

Directory: DA5

Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

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

We used NRF24L01 chip, atmega328p, and LM34.

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

/\*

\* main.c

\*

\* Created: 4/6/2019 9:49:11 PM

\* Author : chent

\*/

#ifndef *F\_CPU*

#define *F\_CPU* 16000000UL

#endif

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdbool.h>

#include <stdio.h>

#include <string.h>

#ifndef BAUD

#define BAUD 9600

#endif

#include "inctest/STDIO\_UART.h"

//Include nRF24L01+ library

#include "inctest/nrf24l01.h"

#include "inctest/nrf24l01-mnemonics.h"

#include "inctest/spi.h"

void read\_adc(void); //Read temp with the LM35 using ADC

void adc\_init(void); // Initialize ADC

void USART\_init( unsigned int ubrr ); // Initialize USART communication

void USART\_tx\_string(char \*data); //sending string through usart

volatile unsigned int adc\_temp; // raw temp value from adc

volatile unsigned int tempF; // convert to fahrenheit

volatile unsigned int tempC; // convert to celsius

char outs[256]; // out for sending USART commands

volatile char received\_data; // String for receiving USART communication

//Used in IRQ ISR

volatile bool message\_received = false;

volatile bool status = false;

int main(void) {

adc\_init(); // Initialize Analog to Digital Converter on PC4

USART\_init(BAUD\_PRESCALLER); // Initialize the USART

*\_delay\_ms*(500); // Delay to sync all the commuication

nrf24\_init(); // initial the nrf24

nrf24\_start\_listening(); // start listening from the nrf24

status = nrf24\_send\_message(tx\_message);

if (status == true)

{

*printf*("Connection Established\n");

}

while(1)

{

PORTC^=(1<<5); // port pc2 open

read\_adc(); //read from adc

adc\_temp = (adc\_temp\*500)/1023; //calc the temp variable from the raw data

tempC = adc\_temp; // C value

tempF = (tempC\*1.8)+32; // F value

if(message\_received)

{

// Message received, print it

message\_received = false; // Reset to false

*printf*("Received Temperature: %s\n" ,tempF, nrf24\_read\_message());

// Send message as response

*\_delay\_ms*(500);

// Perform ADC execution HERE and store in tx\_message

status = nrf24\_send\_message(tx\_message);

if(status == true)

{

*printf*("Temperature Transmitted\n");

}

}

*\_delay\_ms*(10000);

}

}

void adc\_init(void) //initialize 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)| // ADC2 (PC2 PIN25)

(0<<MUX0); // 010

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);

}

void read\_adc(void) {

unsigned char i =4;

adc\_temp = 0; //initialize adc\_temp variable

while (i--) {

ADCSRA |= (1<<ADSC);

while(ADCSRA & (1<<ADSC));

adc\_temp+= ADC;

*\_delay\_ms*(50);

}

adc\_temp = adc\_temp / 4; // Average a few samples

}

void USART\_init( unsigned int ubrr ) {

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

UCSR0B |= (1 << TXEN0) | (1 << RXEN0)| ( 1 << RXCIE0); // Enable receiver, transmitter & RX interrupt

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

}

// Interrupt on IRQ pin

ISR(INT0\_vect)

{

message\_received = true;

}

void USART\_tx\_string( char \*data ) {

while ((\*data != '\0')) {

while (!(UCSR0A & (1 <<UDRE0)));

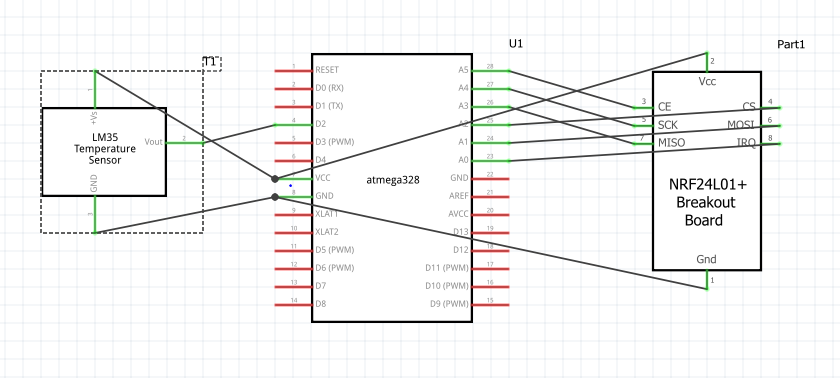
UDR0 = \*data;

data++;

}

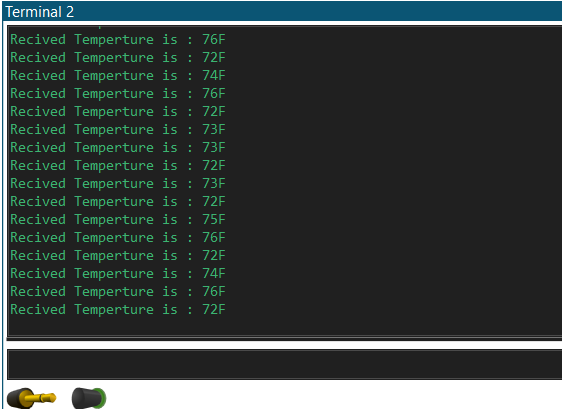
}

1. **SCHEMATICS**

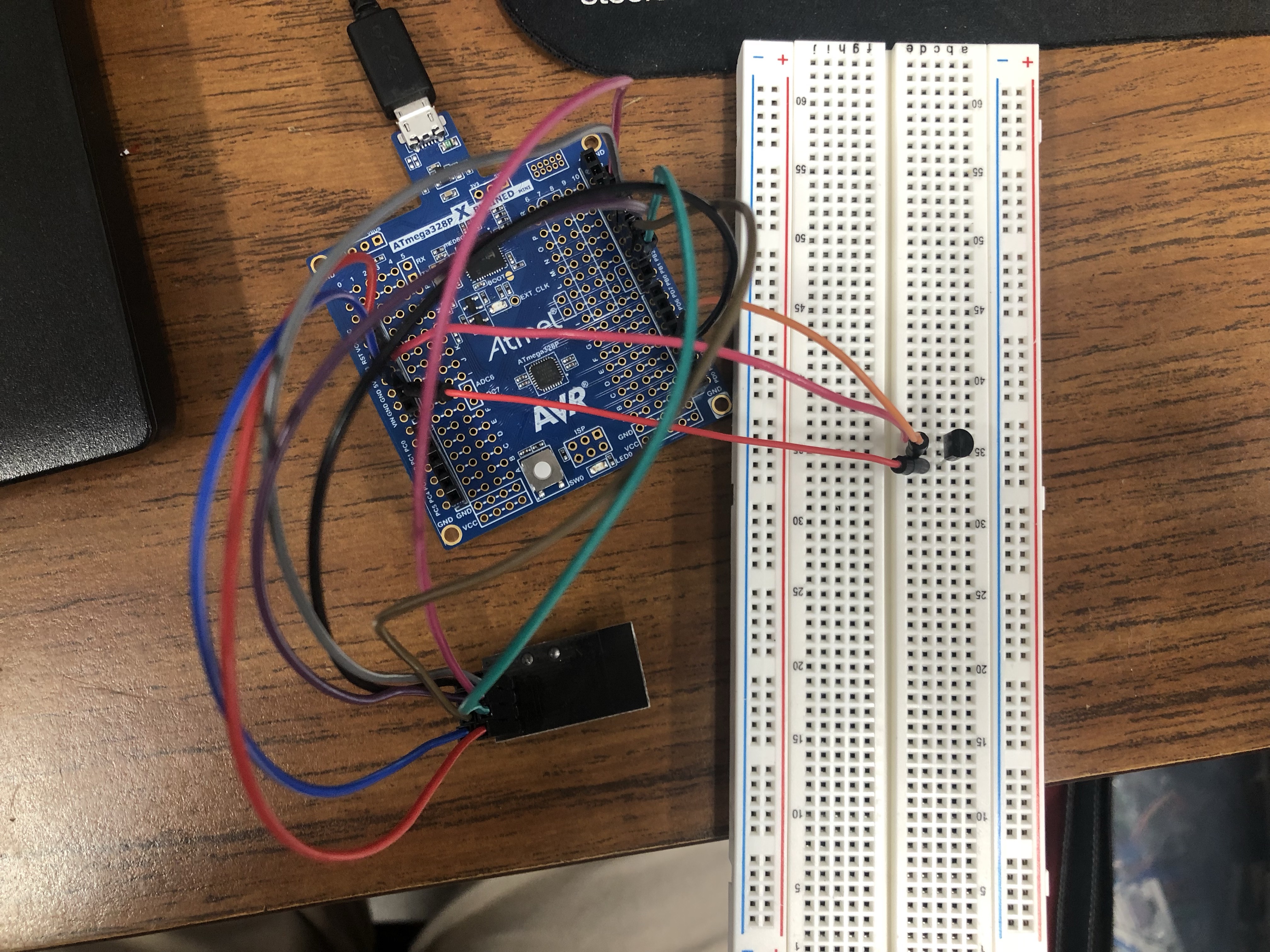


Use fritzing.org

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



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



1. **GITHUB LINK OF THIS DA**

<https://github.com/Chentian12138/AAAABBBBB>

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT