CPE301 – SPRING 2019

MIDTERM 1

Student Name: Benjamin Johnson

Student #: 5003284489

Student Email: johnsb18@unlv.nevada.edu

Primary Github address: <https://github.com/johnsb18/ClassRepository>

Directory:

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/Midterm, 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**

Atmega328P

ESP01 Module

Temperature Sensor

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

/\*

\* Midterm\_Project.c

\*

\* Created: 4/5/2019 3:21:18 PM

\* Author : Benjamin Johnson

\* Student #: 5003284489

\* Email: johnsb18@unlv.nevada.edu

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdio.h>

#define BAUDRATE 9600

#define BAUD\_PRESCALER (*F\_CPU* /BAUDRATE/16 - 1)

void USART\_init(unsigned int ubrr);

void ADC\_init(void);

void read\_ADC (void);

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

volatile unsigned int ADC\_temp;

char outs[256];

char instr1[] = "AT";

char instr2[] = "AT+CWMODE=1"; // enables wifi

char instr3[] = "AT+CWLAP"; // list wireless AP in range

char instr4[] = "AT+CWJAP=\"AndroidAP\",\"easymode\""; // join network

char instr5[] = "AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80";

char instr6[] = "AT+CIPSEND=51 \\";

//unsigned char instr7[] = ("GET https://api.thingspeak.com/update?api\_key=W6P5T8ZB7LDLGASQ&field1=%3d \r\n", ADC\_temp);

char instr8[] = "AT+CIPCLOSE";

int main(void)

{

ADC\_init(); // initialize ADC

USART\_init(BAUD\_PRESCALER); // initialize analog to digital

*\_delay\_ms*(125); // some delay before starting

TCNT1 = 49911; // set timer

sei(); // enable interrupts

while (1) {}; // wait for interrupt

}

ISR (TIMER1\_OVF\_vect)

{

USART\_tx\_string(instr1);

*\_delay\_ms*(1000);

USART\_tx\_string(instr2);

*\_delay\_ms*(1000);

USART\_tx\_string(instr3);

*\_delay\_ms*(1000);

USART\_tx\_string(instr4);

*\_delay\_ms*(1000);

USART\_tx\_string(instr5);

*\_delay\_ms*(1000);

USART\_tx\_string(instr6);

*\_delay\_ms*(1000);

//USART\_tx\_string(instr7);

//\_delay\_ms(1000);

USART\_tx\_string(instr8);

*\_delay\_ms*(1000);

read\_ADC(); // read ADC

*snprintf*(outs,sizeof(outs),"GET https://api.thingspeak.com/update?api\_key=W6P5T8ZB7LDLGASQ&field1=%3d\r\n", ADC\_temp); // print float

USART\_tx\_string(outs); // output to terminal

TCNT1 = 49911; // reset timer

}

void USART\_init (unsigned int ubrr)

{

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

UBRR0L = (unsigned char)ubrr;

UCSR0C = (1 << TXEN0); // enable Rx, TX & Rx interrupt

UCSR0B = (3 << UCSZ00); // asynchronous 8 N 1

}

void ADC\_init (void)

{

ADMUX = (0<<REFS1)| // reference selection bits

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

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

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

(0<<MUX1)| // ADC0 (PC2)

(0<<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); // prescaler = 1024

TIMSK1 = (1<<TOIE1); // enable overflow flag

TCNT1 = 49911; // reset timer (65535 - 15624)

}

void read\_ADC (void) // read ADC pins

{

unsigned char i=4;

ADC\_temp = 0;

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\_tx\_string (char \*data)

{

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

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

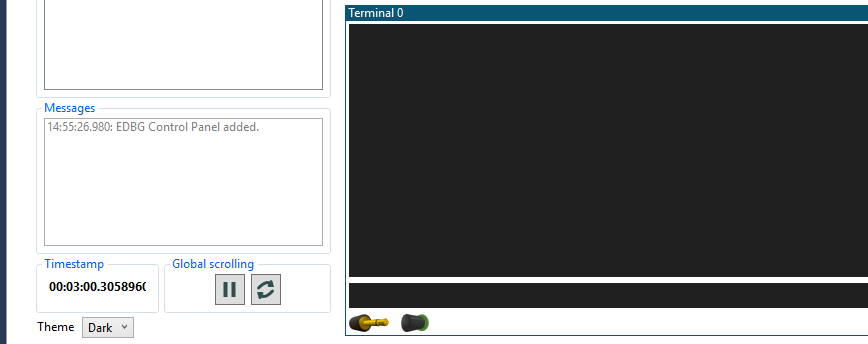
UDR0 = \*data;

data++;

}

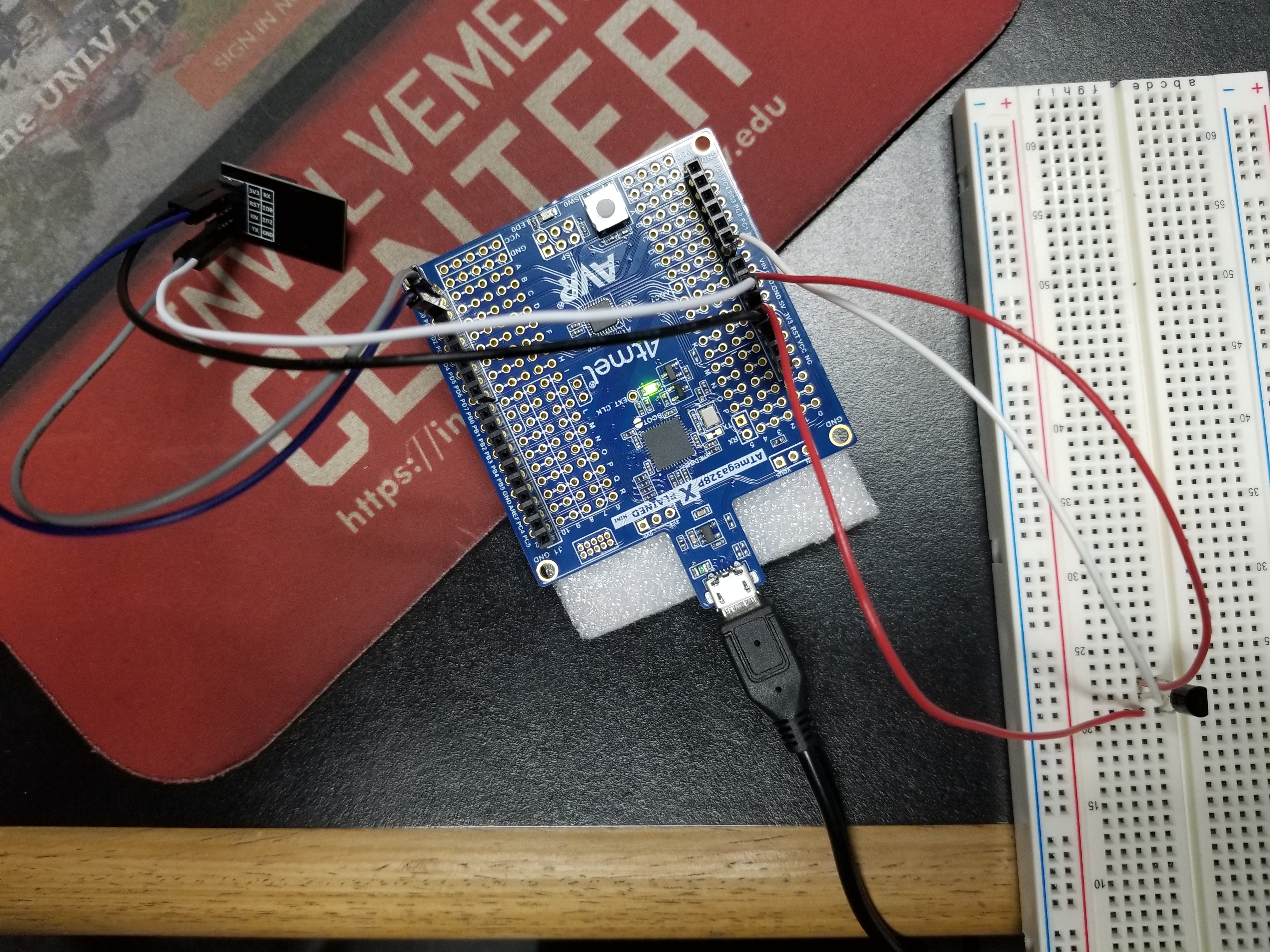
}

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**
2. **SCHEMATICS**
3. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



\*I was unable to get the project to run correctly; I wasn’t getting any output for some reason and could not figure it out.

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



1. **VIDEO LINKS OF EACH DEMO**
2. **GITHUB LINK OF THIS DA**

<https://github.com/johnsb18/ClassRepository/tree/master/Midterms/Midterm%201>

**Student Academic Misconduct Policy**

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

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

Benjamin Johnson