CPE301 – SPRING 2020

Design Assignment 3B

Student Name: Minsung Cho
Student #: 2001446442
Student Email: chom3@unlv.nevada.edu
Primary Github address: https://github.com/cho-minsung
Directory: https://github.com/cho-minsung/assignment3B

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

DHT11

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

```
#define F_CPU 16000000UL
#define BAUD_RATE 9600

#include <avr/io.h>
#include <util/delay.h>

void usart_init ();
void usart_send(unsigned char ch);

int main(void)
{
       usart_init();

       ADMUX = (0<<REFS1)|  //Reference Selection Bits
                     (1<<REFS0)| //AVcc - external cap at AREF
                     (0<<ADKAR)| //ADC Left Adjust Result
                     (1<<MUX2) | //Analog channel selection bits
                     (0<<MUX1) | //ADC4 (PC4 PIN27)
                     (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 bit
                     (0<<ADPS1)|
                     (1<<ADPS0);

    while (1)
    {
              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)+'0');
              a = a % 10;
              usart_send((a)+'0');
              usart_send('\r');

              _delay_ms(100);
    }
       return 0;
}

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)));
       UDR0 = ch;
}
```

the code from the tutorial video

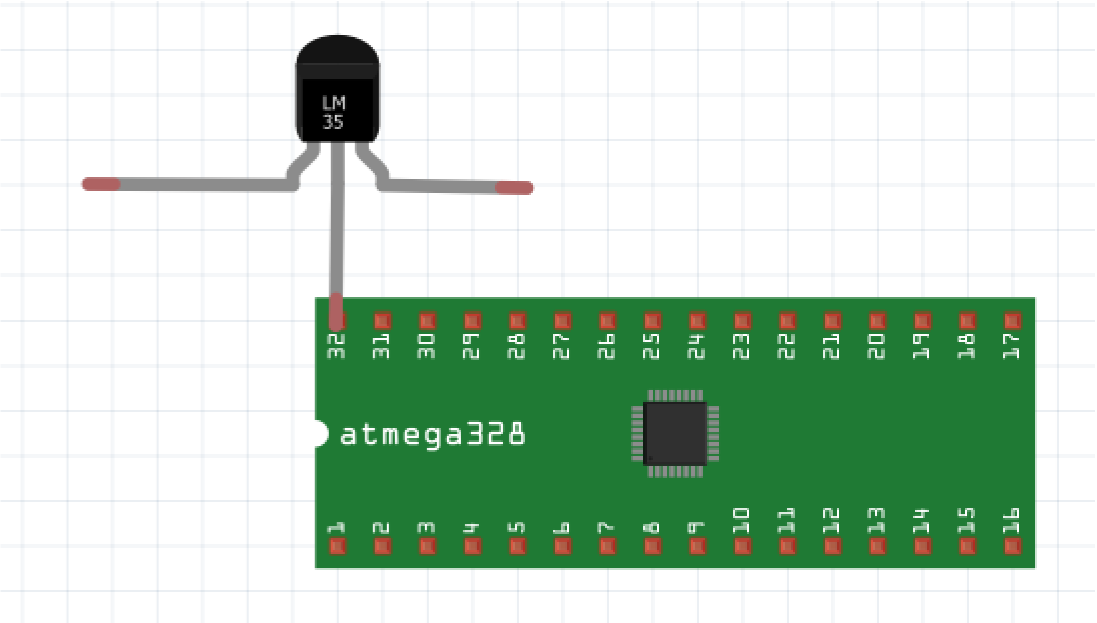
3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

```
#define F_CPU 16000000UL
#define BAUD_RATE 9600

#include <avr/io.h>
#include <avr/interrupt.h>

void usart_init ();
void usart_send(unsigned char ch);
void usart_string(char *data);

int main(void)
{
    usart_init();
    TCCR0A = 0x00; //normal mode set
    TCCR0B = 0x05; //prescaler of 1024
    TIMSK0 = (1<<TOIE0); //timer 0 interrupt enable

    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 (PC4 PIN27)
                 (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 bit
                  (0<<ADPS1)|
                  (1<<ADPS0);

    usart_init();
    sei();

     while (1);
    return 0;
}

ISR(TIMER0_OVF_vect) {
    //60 is roughly 1 second, 30 will be roughly 0.5s.
    for (int i = 0; i <= 30; i++) {
            while (TCNT0 != 255) {}
            TCNT0 = 0;
    }
    adc_read();
    TCNT0 = 0;
    }

void adc_read(void)
{
    ADCSRA |= (1<<ADSC); //START CONVERSION
    while ((ADCSRA & (1<<ADIF))==0); //wait for conversion to finish

    ADCSRA |= (1<<ADIF);

    //Celcius read
    int a = ADCL;
    a = a | (ADCH<<8);
    usart_string("Centigrade: ");
    usart_send((a/100)+'0');
    a = a % 100;
    usart_send((a/10)+'0');
    a = a % 10;
    usart_send((a)+'0');
    usart_send('\r');

    //Farenheit read
    a = ADCL;
    a = a | (ADCH<<8);
    a = (a/1024.0)*5000/10;
    usart_string("Farenheit: ");
    usart_send((a/100)+'0');
    a = a % 100;
    usart_send((a/10)+'0');
    a = a % 10;
    usart_send((a)+'0');
    usart_send('\r');
}
void usart_init(void)
{
    UCSR0B = (1<<TXEN0); //enable interrupts
    UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);
    UBRR0L = F_CPU/16/BAUD_RATE-1;
}
```

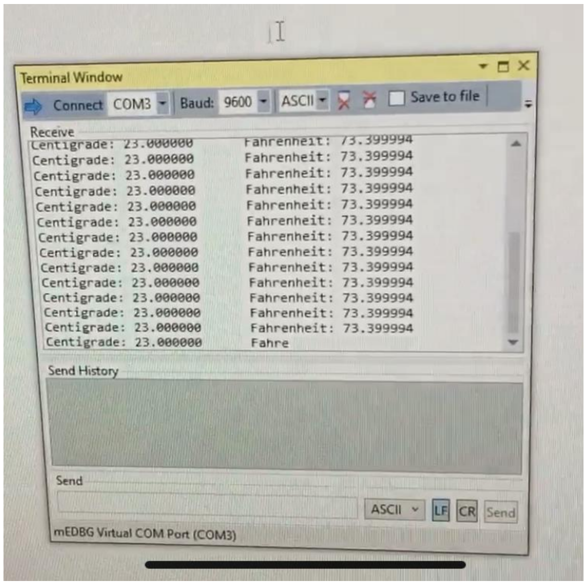
```
void usart_send(unsigned char ch)
{
    while (!(UCSR0A & (1<<UDRE0)));
    UDR0 = ch;
}

void usart_string(char *data) {
    while((*data != '\0'))      {
            while(!(UCSR0A & (1<<UDRE0)));
            UDR0 = *data;
            data++;
    }
}
```

4. SCHEMATICS



The 32 is actually PC0 and the rest of the legs are grounded and connected to the power.

5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)



Connect COM3 Baud: 9600 ASCII □ Save to file

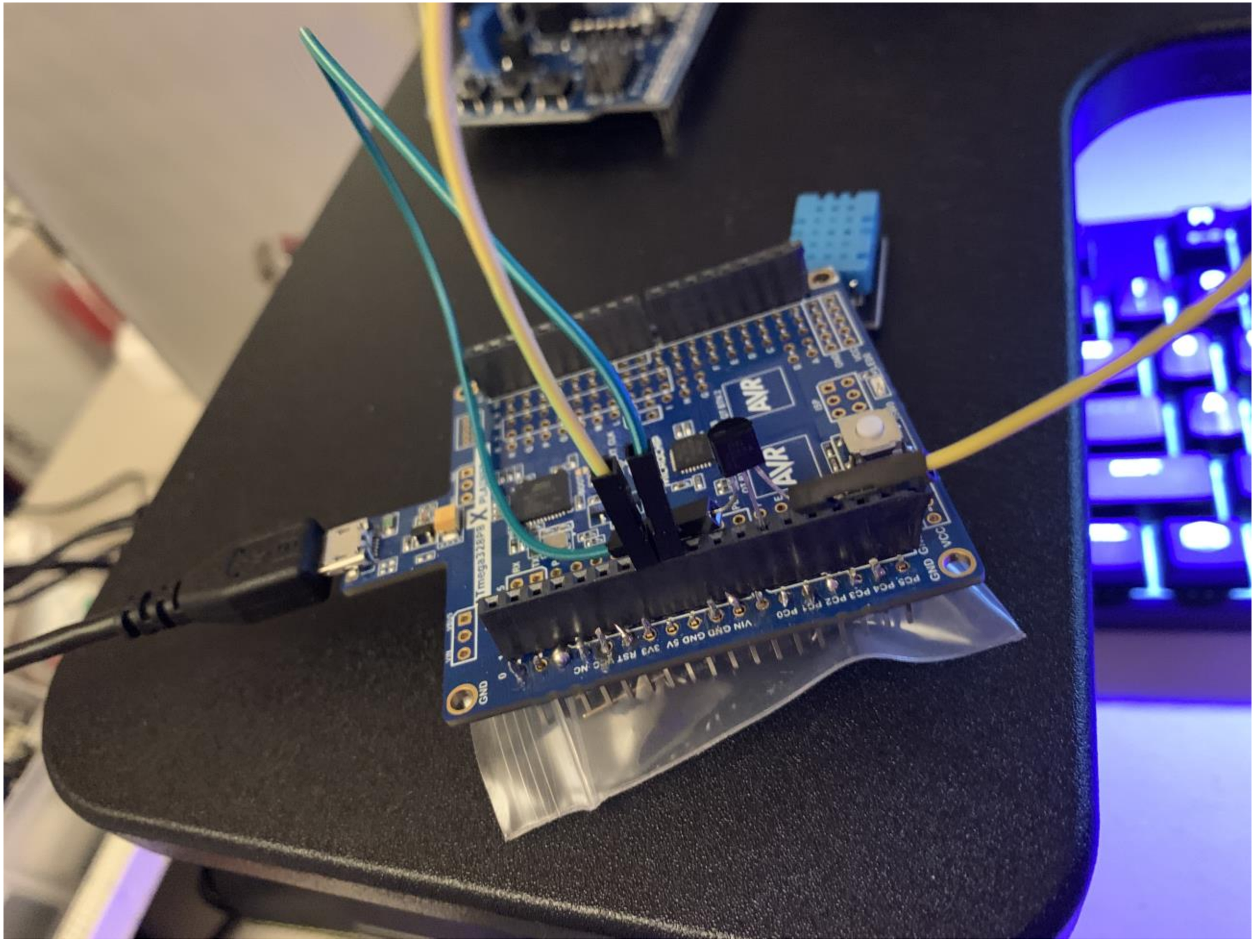
Receive

```
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahrenheit: 73.399994
Centigrade: 23.000000        Fahre
```

Send: ASCII

mEDBG Virtual COM Port (COM3)

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

https://youtu.be/UqAnStqKjmE

8. GITHUB LINK OF THIS DA

https://github.com/cho-minsung/assignment3B

Student Academic Misconduct Policy

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

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