CPE301 – SPRING 2019

Design Assignment 3B

Student Name: Kyungseo Yun

Student #: 2001091216

Student Email: yunk93@unlv.nevada.edu

Primary Github address: <https://github.com/biscuit0x/submission_yun.git>

Directory: submission\_yun/DesignAssignments/DA3B/

Submit the following for all Labs:

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

* 1. b. The C code should be well documented with explanation of every instruction.
  2. 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.

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

A. Components

Atmel328p board, LM35 temperature sensor

B. AVR code

#define F\_CPU 16000000UL

#define UBRR\_9600 103// for 16Mhz with .2% error

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include <avr/interrupt.h>

void read\_adc(void);// Function Declarations

void adc\_init(void);

void USART\_init( unsigned intubrr);

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

void TIMER\_init(void);

volatile unsigned int adc\_temp;

char outs[20];

int main(void) {

adc\_init();// Initialize the ADC (Analog / Digital Converter)

USART\_init(UBRR\_9600); // Initialize the USART (RS232 interface)

TIMER\_init();

while(1)

{

}

}

/\* INIT USART (RS-232) \*/

void USART\_init( unsigned intubrr)

{

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

UBRR0L = (unsigned char)intubrr;

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

UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);//asynchronous 8 N 1

}

ISR (TIMER1\_OVF\_vect){

adc\_temp= 0;

TCNT1 = (0xFFFF - ((F\_CPU/1024)\*1) - 60); //set counter for 1 sec

ADCSRA |= (1<<ADSC);

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

ADCSRA |= (1<<ADIF); //clear ADC interrupt flag

adc\_temp = ADCL;

adc\_temp = adc\_temp | (ADCH<<8);

snprintf(outs, sizeof(outs),"%3d\r\n", adc\_temp);// print it

USART\_tx\_string(outs);

}

void TIMER\_init(void) {

TCNT1 = (0xFFFF - ((F\_CPU/1024)\*1) - 60); //set counter for 1 sec

TIMSK1 |= (1 << TOIE0); // Enable Timer1 Interrupt for Timer1

sei(); // Enable Global Interrupt

TCCR1B |= (1<<CS12)|(1<<CS10); // set prescaler = 1024 & start timer

}

void adc\_init(void){/\*\* 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)| // ANalogChannel Selection Bits

(0<<MUX1)| // ADC2

(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 PrescalerSelect Bits

(0<<ADPS1)|(1<<ADPS0); // Select Channel

}

void USART\_sendChar(char ch) {

while (!(UCSR0A & (1<<UDRE0))); // while data reg is not empty: hold

UDR0 = ch; // place character into reg

}

void USART\_sendString(char\* str) {

while ((\*str != '\0')) { // while not the end of the string

while (!(UCSR0A & (1<<UDRE0))); // while data reg is not empty: hold

USART\_sendChar(\*str); //take in character to reg

str++; // next character

}

}

/\* SEND A STRING TO THE RS-232 \*/

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

{

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

{

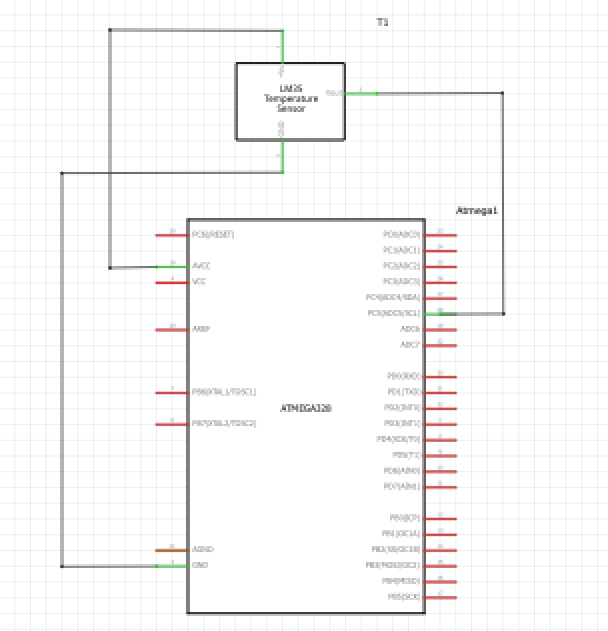
while (!(UCSR0A & (1 <<UDRE0)));

UDR0 = \*data;data++;

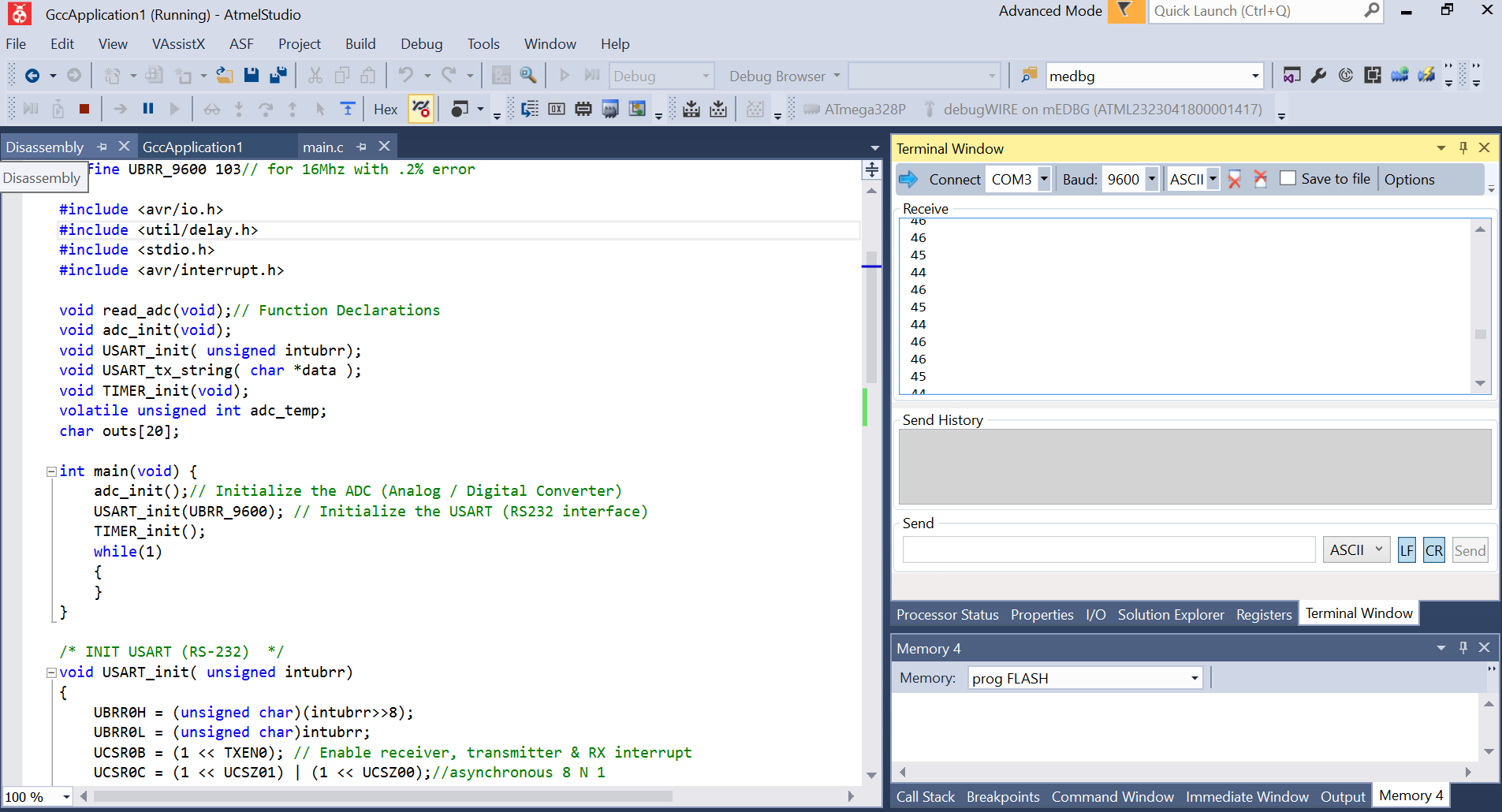
}

}

C. Schematics



D. Screenshot



E. Video / photo

https://youtu.be/Sfb5Uy5cQBs

