CPE301 – SPRING 2019

Design Assignment 3A

Student Name: yeeunLee

Student #:2001614222

Student Email: leey81@nevada.unlv.com

Primary Github address: https://github.com/yeeun219/submission\_da.git

Directory: cpe301\DesignAssignments\DA3B

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

while (1) //Repeat continuously, display a temperature,

void adc\_init(void) /\* INIT ADC \*/

void read\_adc(void) /\* READ ADC PINS \*/

void USART\_init( unsigned int ubrr ) /\* INIT USART (RS-232) \*/

void USART\_tx\_string( char \*data ) /\* SEND A STRING TO THE RS-232 \*/

ISR(TIMER0\_COMPA\_vect) // the interrupt service routine

1. **DEVELOPED CODE OF**

/\*

\* GccApplication6.c

\*

\* Created: 2019-04-01 am6:51:27

\* Author : llje2

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include <avr/interrupt.h>

#define BAUDRATE 9600

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

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

void adc\_init(void);

void USART\_init( unsigned int ubrr );

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

volatile unsigned int adc\_temp;

char outs[20];

int extraTime=0; //to count 1second interrupt

int main(void)

{

DDRB = 0x01; // setting the LED as an output

TCCR0A = (1 << WGM01); // Set CTC Bit

OCR0A = 195; // number of ticks we need for our specific time

TIMSK0 = (1 << OCIE0A);

TCCR0B = (1 << CS02) | (1 << CS00); // use 1024 prescaler

sei(); // setting the interrupt

adc\_init(); // Initialize the ADC

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

USART\_tx\_string("Connected!\r\n"); // we're alive!

*\_delay\_ms*(125); // wait a bit

while(1)

{

read\_adc();

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

USART\_tx\_string(outs);

*\_delay\_ms*(1000); // wait a bit

}

}

/\* INIT ADC \*/

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)| // Analog Channel Selection Bits

(0<<MUX1)| // ADC5 (PC5 PIN28)

(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);

}

/\* READ ADC PINS \*/

void read\_adc(void)

{

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

}

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

void USART\_init( unsigned int ubrr )

{

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

UBRR0L = (unsigned char)ubrr;

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

UCSR0C = (3 << UCSZ00); //asynchronous 8 N 1

}

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

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

{

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

{

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

UDR0 = \*data;

data++;

}

}

ISR(TIMER0\_COMPA\_vect) // the interrupt service routine(see list of vectors for all interrupt sources!!!!

{

extraTime++;

if(extraTime > 100)

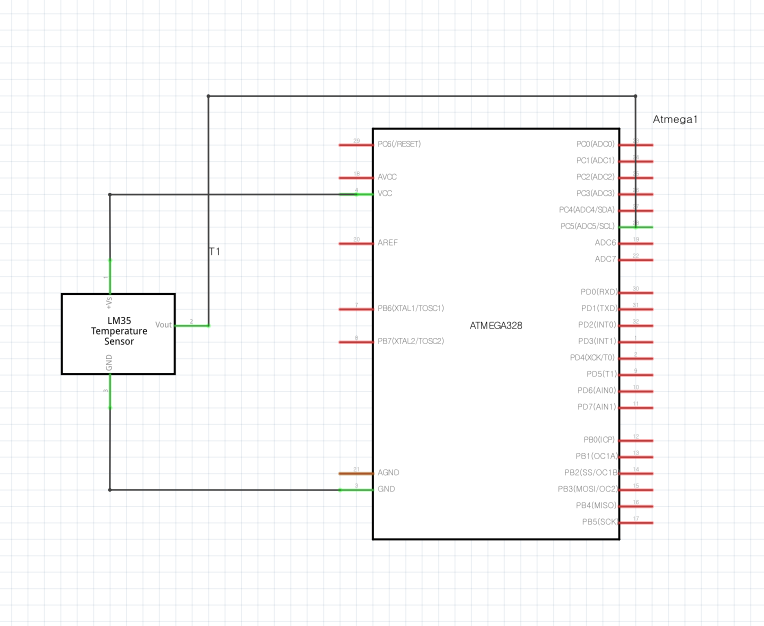
{

PORTB ^= (1<<PORTB0); // toggle LED

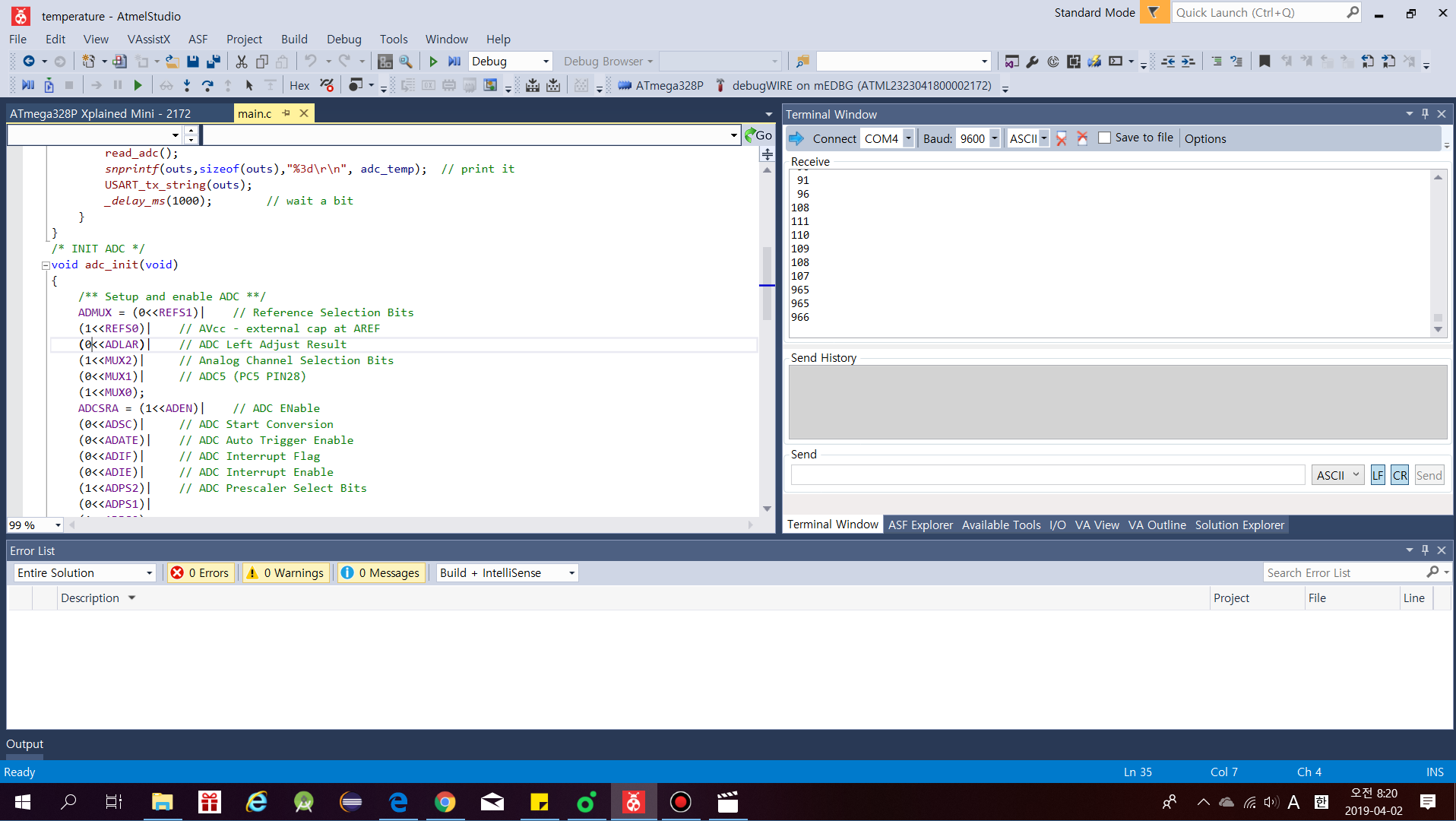
extraTime = 0;

}

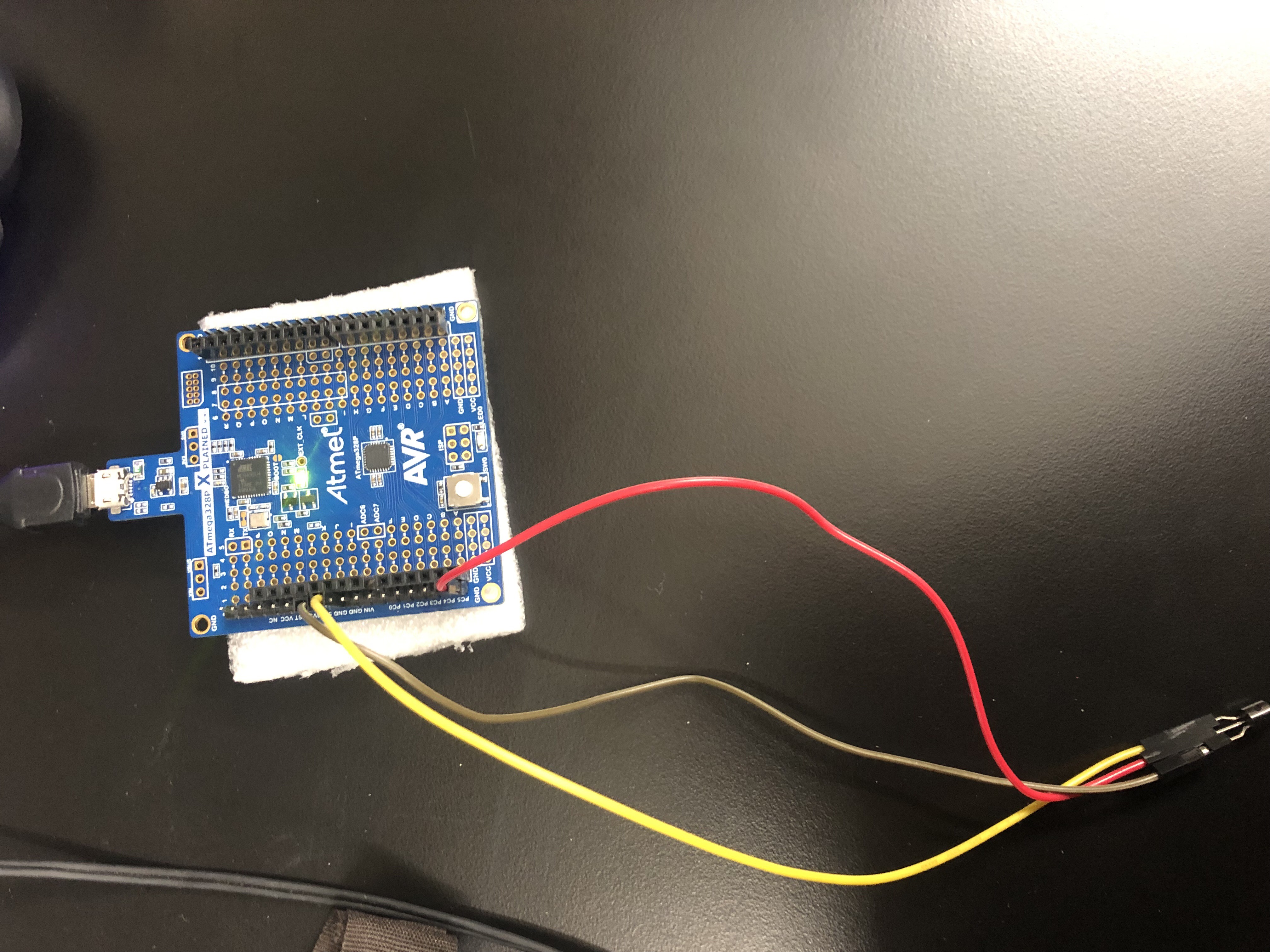
1. }**SCHEMATICS**



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



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



1. **VIDEO LINKS OF EACH DEMO**

https://www.youtube.com/watch?v=VRhC\_rOPMBw&feature=youtu.be

1. **GITHUB LINK OF THIS DA**

Primary Github address: https://github.com/yeeun219/submission\_da.git

Directory: cpe301\DesignAssignments\DA3B

**Student Academic Misconduct Policy**

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

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

YEEUNLEE