CPE301 – SPRING 2020

Design Assignment 3B

Student Name: Xianjie Cao

Student #: 5004222179

Student Email: [caox2@unlv.nevada.edu](mailto:caox2@unlv.nevada.edu)

Primary Github address: <https://github.com/c1029324620/Mocha.git>

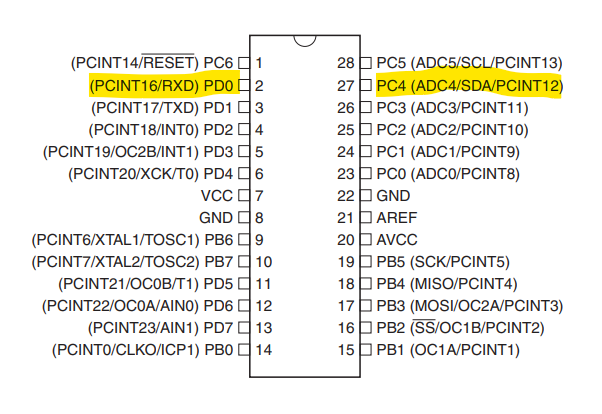
Directory: Mocha/DesignAssignments/LAB3/DA3B

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

Atmel studio 7: Simulator, Debugger, Assembler and Terminal

Atmega328PB-Xmini PC

LM35



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

/\*

\* DA3BT1.c

\*

\* Created: 3/13/2020 12:16:44 PM

\* Author : c1029

\*/

#define *F\_CPU* 16000000UL

#define BAUD 9600

#include <avr/io.h>

#include <util/setbaud.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdio.h>

volatile *uint8\_t* overflow\_cnt;

volatile *uint8\_t* temp;

void USART\_init(void)

{

UBRR0H = *UBRRH\_VALUE*;

UBRR0L = *UBRRL\_VALUE*;

UCSR0C = \_BV(UCSZ01) | \_BV(UCSZ00); //8-bit data

UCSR0B = \_BV(RXEN0) | \_BV(TXEN0); //enable transmitter

}

void set\_inter()

{

TCCR0A = 0x00; //normal mode

TCCR0B = 0x05; //prescaler 1024

TCNT0 = 0x00;

TIMSK0 = (1<< TOIE0); //enable timer interrupt

sei(); //enable global interrupt

}

void adc\_int(void)

{

ADMUX = (0 << REFS1) | (1 << REFS0) | (0 << ADLAR) | (0 << MUX3) | (1 << MUX2) | (0<< MUX1) | (0 << MUX0);

ADCSRA = (1 << ADEN) | (0 << ADSC) | (0 << ADATE) | (0 << ADIF) | (0 << ADIE) | (1<< ADPS2) | (1 << ADPS1) | (1<<ADPS0);

}

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

{

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

{

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

UDR0 = \*data;

data++;

}

}

void adc\_read(void)

{

ADCSRA |= (1 << ADSC); //enable start conversion

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

temp = ADC; // LM35 value stored in temp.

}

int main(void)

{

/\* Replace with your application code \*/

USART\_init();

adc\_int();

set\_inter();

int celsius;

char array1[20], array2[20];

while (1)

{

if (overflow\_cnt == 31)

{

adc\_read(); //Call Read Function

celsius = (temp - 32) \* 5 / 9; //convert to celsius

*sprintf*(array1,"%i", temp); //convert the number into a string type

*sprintf*(array2, "%i", celsius);

//Prints temperature value

USART\_tx\_string("Temperature: ");

USART\_tx\_string(array1);

USART\_tx\_string(" Fahrenheit or ");

USART\_tx\_string(array2);

USART\_tx\_string(" Celsius.");

USART\_tx\_string("\n");

overflow\_cnt = 0;

}

}

}

ISR(TIMER0\_OVF\_vect)

{

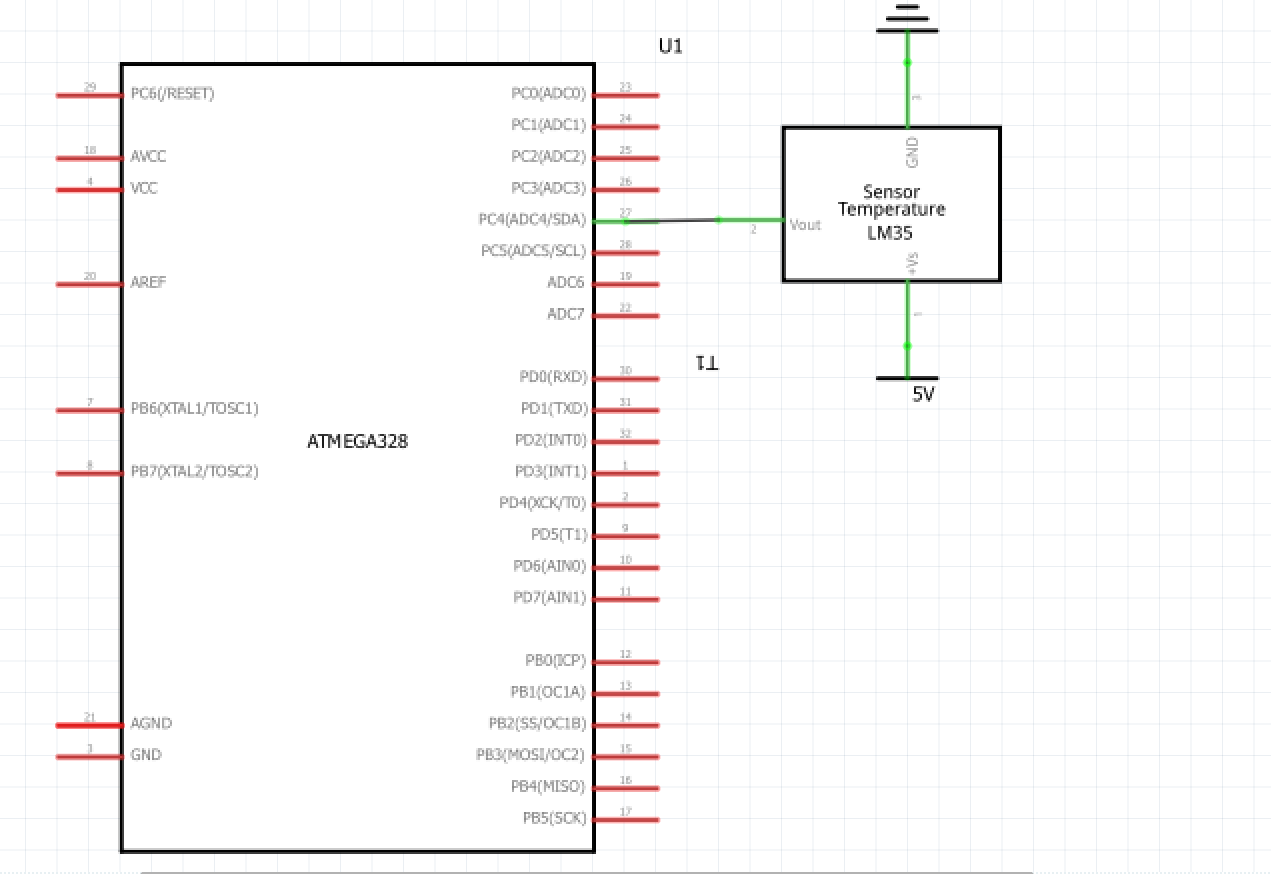
overflow\_cnt++;

}

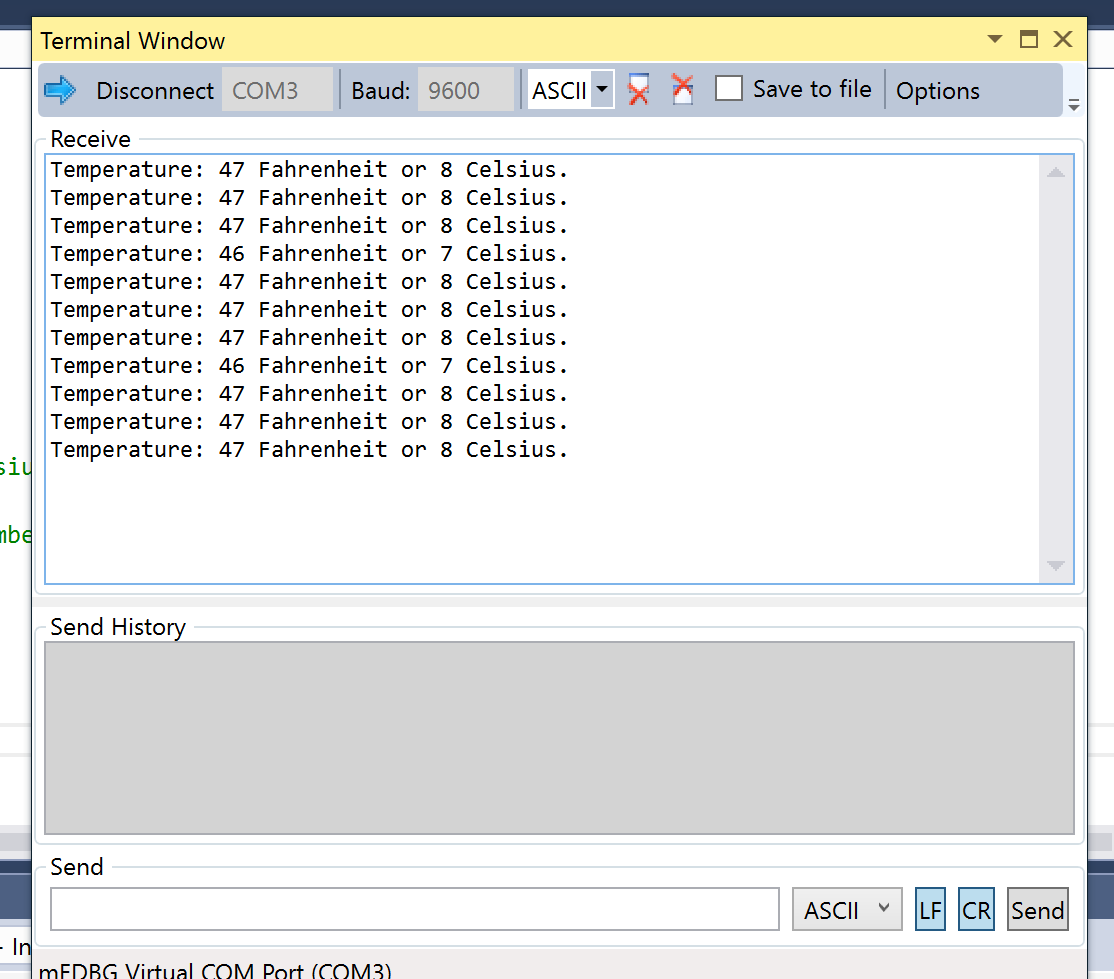
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

N/A

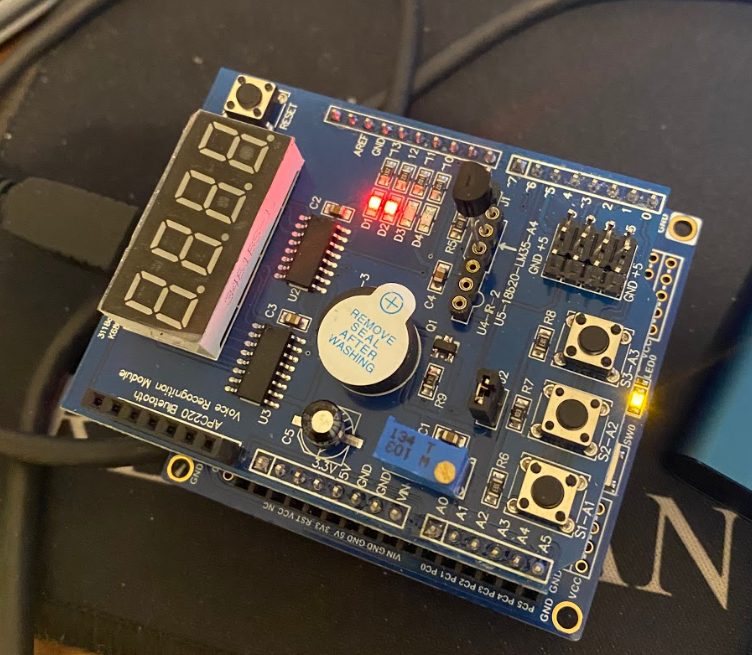
1. **SCHEMATICS**



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



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



1. **VIDEO LINKS OF EACH DEMO**

Video: <https://youtu.be/Mm9k7M7V9Y8>

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

Xianjie Cao