CPE301 – SPRING 2020

MIDTERM 1

Student Name: Xianjie Cao

Student #: 5004222179

Student Email: caox2@unlv.nevada.edu

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

Directory: Mocha/Midterm

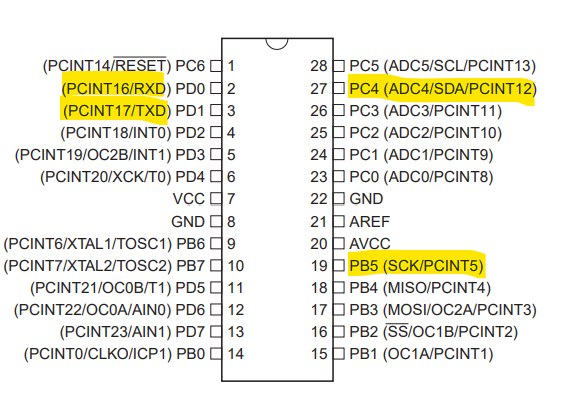
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

Atmel Studio 7: simulator, debugger, assembler, terminal.

LM 35

Atmega328PB-Xmini PC

LED



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

/\*

\* Midterm.c

\*

\* Created: 3/20/2020 10:57:17 AM

\* Author : c1029

\*/

#define *F\_CPU* 16000000UL

#define BAUD 9600

#include <avr/io.h>

#include <util/setbaud.h>

#include <stdlib.h>

#include <stdio.h>

#include <util\delay.h>

#include <time.h>

#include <avr/interrupt.h>

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 receiver and transmitter

}

void adc\_int(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 chanel selection bits

(0 << MUX1) | // ADC4(PC4)

(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);

}

void adc\_read(void)

{

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

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

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

}

//send some data to the serial port

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

{

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

{

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

UDR0 = \*data;

data++;

}

}

char USART\_Receiver()

{

//wait for data to be received

while(!(UCSR0A & (1<< RXC0)));

//get and return received data from buffer

return UDR0;

}

void display\_help()

{

USART\_tx\_string("\*\*\*'t' display temperature in Celsius.\n");

USART\_tx\_string("\*\*\*'T' display temperature in Farenthide.\n");

USART\_tx\_string("\*\*\*'o' turns ON LED at PB5.\n\*\*\*'O' turns OFF the LED at PB5.\n");

USART\_tx\_string("\*\*\*'s' sends a string to the terminal.\n");

USART\_tx\_string("\*\*\*'i' sends an integer to the terminal.\n");

USART\_tx\_string("Press 'h' key to display again the help table.\n");

}

void display\_tem\_c()

{

adc\_read();

char array1[20];

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

*sprintf*(array1,"%i", celsius ); //convert int to string

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

USART\_tx\_string(array1);

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

}

void display\_tem\_t()

{

adc\_read();

char array1[20];

*sprintf*(array1,"%i", temp); //convert int to string

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

USART\_tx\_string(array1);

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

}

void turn\_on()

{

PORTB &= ~(1 << 5); //Turn LED on

}

void turn\_off()

{

PORTB |= (1 << 5); //Turn LED off

}

void send\_str()

{

USART\_tx\_string("Hello World!\n"); //send "Hello World" to terminal

}

void my\_delay\_ms(int n) {

while(n--) {

*\_delay\_ms*(1);

}

}

void send\_int()

{

int delay = (*rand*() % 3 + 1) \* 1000; //generate random numbers of 1000, 2000, 3000 for delay in ms

char array1[20];

*sprintf*(array1, "%i", delay);

USART\_tx\_string(array1);

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

turn\_on();

my\_delay\_ms(delay);

turn\_off();

my\_delay\_ms(delay);

}

int main(void)

{

*srand*(*time*(*NULL*));

/\* Replace with your application code \*/

USART\_init(); //initialized USART module

adc\_int(); //initialized ADC module

DDRB |= (1 << 5); //set PORTB.5 as output

PORTB |= (1 << 5); //pull up;

display\_help();

while (1)

{

switch(USART\_Receiver())

{

case 'h':

display\_help(); //display help table

break;

case 't':

display\_tem\_c(); //display temperature in C

break;

case 'T':

display\_tem\_t(); //display temperature in F

break;

case 'o':

turn\_on(); //turn on LED

break;

case 'O':

turn\_off(); //turn off LED

break;

case 's':

send\_str(); //send string

break;

case 'i':

send\_int(); //send integer, and blink LED

break;

default:

break;

}

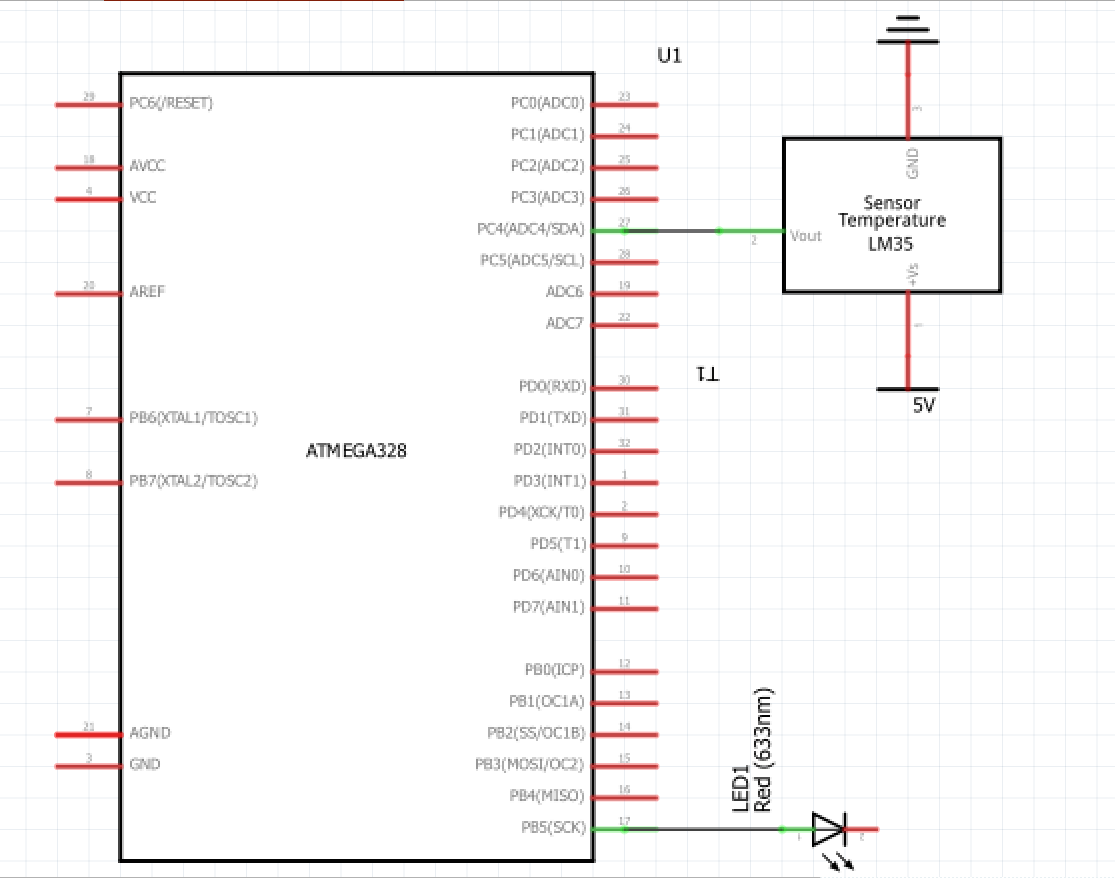
}

}

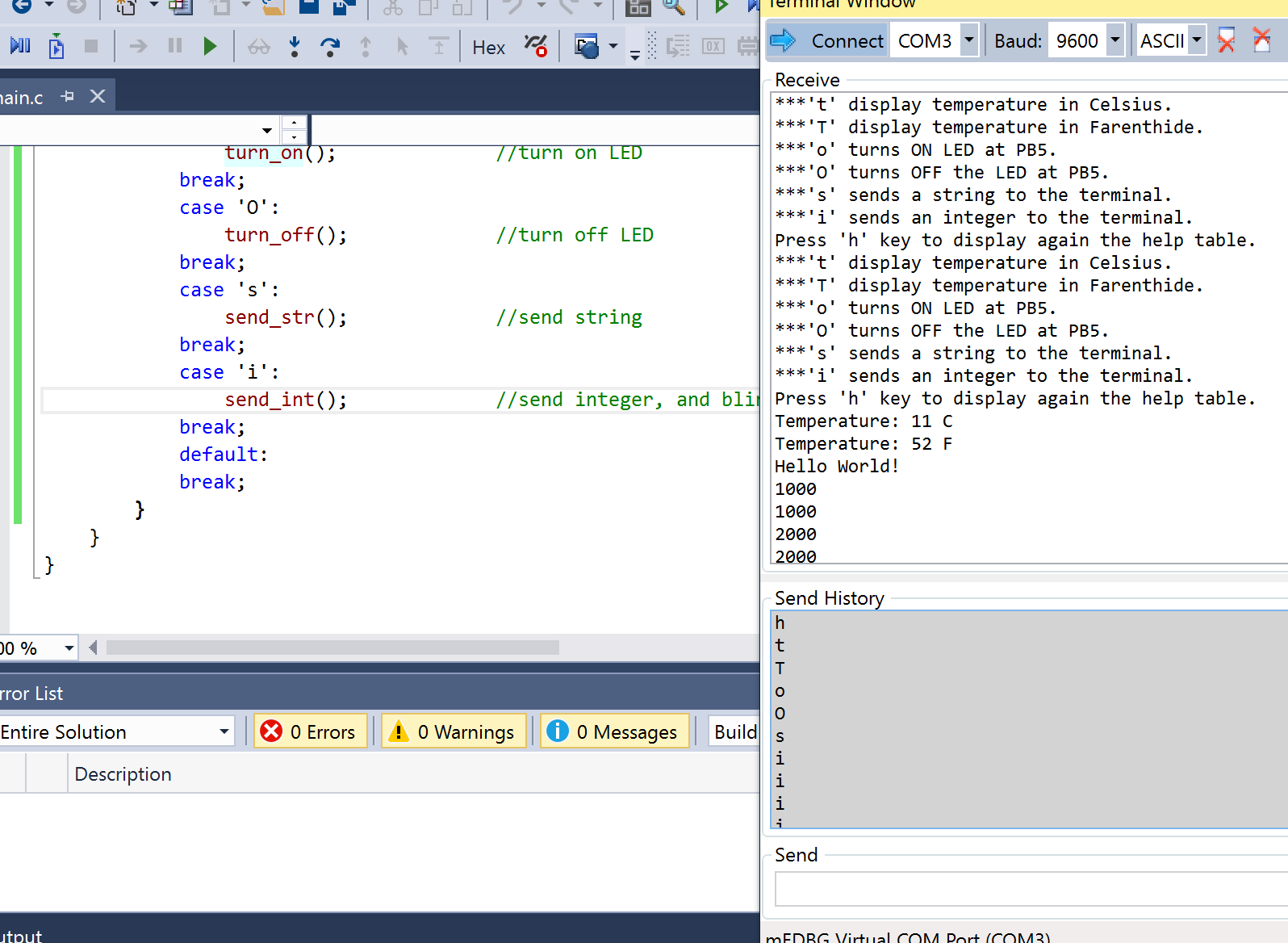
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

**N/A**

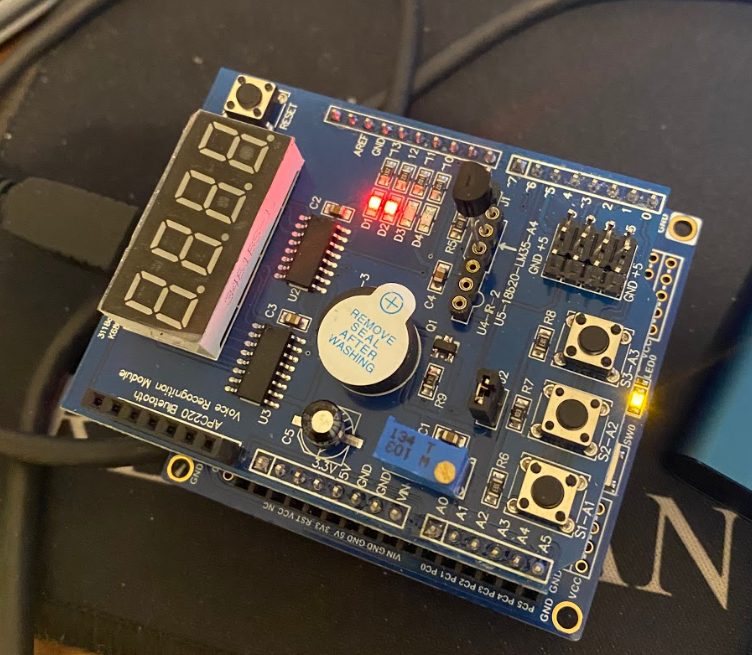
1. **SCHEMATICS**



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



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



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/_FAkqU4I3m8>

1. **GITHUB LINK OF THIS DA**

<https://github.com/c1029324620/Mocha/tree/master/Midterm>

**Student Academic Misconduct Policy**

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

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

Xianjie Cao