CPE301 – SPRING 2019

Design Assignment 3B

Student Name: Tanner Tindall

Student #: 8000733733

Student Email: tindat1@unlv.nevada.edu

Primary Github address: <https://github.com/TannerTindall51>

Directory:

<https://github.com/TannerTindall51/tindalltannerm_submission/tree/master/Midterm>

1. Program the UART Module to interact with the ATmega328/p:

2. On transmitting the following keys from the host terminal, the following actions will be performed:

1. On-reboot or ‘h’ key – help screen (list all keys and functionalities)

2. ‘t’ display temperature in C of LM34/35 on terminal, ‘T’ - display temperature in F of LM34/35 on terminal.

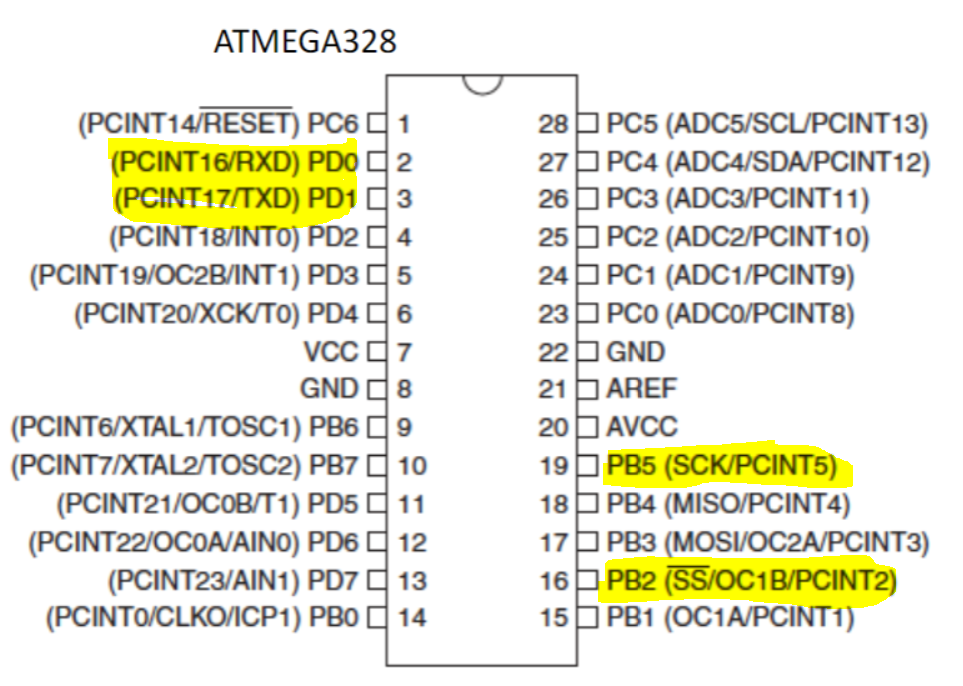
3. ‘o’ turns ON LED at PB5, ‘O’ turns OFF the LED at PB5.

4. ‘s’ sends a string to the terminal.

5. ‘i’ sends an integer to the terminal. Use the integer as a delay in seconds to blink the LED PB2.

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

* Atmel Studio 7.0 (Assembler, Simulator, & Debugger)
* Atmega328PB-Xmini
* Micro USB
* LM35 Temperature Sensor
* Multi-Functional Shield



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

|  |
| --- |
|  |

//

//CPE301 - Midterm1

//Tanner Tindall

//

#define BAUD 9600

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/setbaud.h>

#include <avr/interrupt.h>

#include <stdlib.h>

#include <stdio.h>

#include <util/delay.h>

#include <stdbool.h>

volatile static char helpScreen[] = " h - Help \n t - Display temp. in C \n T - Display temp. in F \n o - Turn on LED at PB5 \n O - Turn off LED at PB5 \n s - Send a string \n i - Send an integer \n";

volatile static char string[] = "supercalifragilisticexpialidocious \n";

volatile static char array[20];

volatile int randomNumber;

void initializeUART()

{

DDRD |= 0x02;

UBRR0 = 103;

UCSR0C |= (0<<UPM01) | (0<<UPM00) | (1<<UCSZ01) | (1<<UCSZ00); //disable parity and set to asynchronous

}

void startADC()

{

DDRC &= ~(1<<4); //set PC4 to input

ADMUX = (1<<REFS1)| (1<<REFS0)| (1<<MUX2); //set Vref and set ADC4 as input

ADCSRA = (1<<ADEN) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0); //enable ADC and set prescalar to 128

ADCSRB = 0x00; //set ADCSRB to free running

}

*uint16\_t* adcRead()

{

ADCSRA |= (1<<6); //enable ADC conversion

while (!(ADCSRA & (1<<4))); //waits for conversion to complete by ADIF flag

ADCSRA |= (1<<4); //reset when complete

return ADC; //return value once complete

}

void USART\_tx\_transmit()

{

UBRR0 = 103;

UCSR0B |= (1 << TXCIE0) | (1<<TXEN0); //enable data transmission

DDRD |= 0x02; //transmit through PD2

}

void USART\_tx(char\*data) //outputs data to terminal

{

while((\*data != '\0')) //loop until all data is emptied

{

while(!(UCSR0A & (1 << UDRE0)));//wait until data register in emptied

UDR0 = \*data; //once emptied, import into data register UDR0

data++; //increments pointer position for data

}

}

void USART\_rx\_receive()

{

UBRR0 = 103;

UCSR0B |= (1 << RXCIE0) | (1<<RXEN0); //enable data transmission

DDRD |= 0x01; //receive from PD1

}

char USART\_rx() //inputs data from terminal

{

while (!(UCSR0A & (1<<7))); //loop until all data is received

return UDR0; //returns received values to UDR0

}

int main(void)

{

startADC(); //initialize

USART\_tx\_transmit();

USART\_rx\_receive();

initializeUART();

DDRB = 0b00100100; //sets DDRB to OUTPUT

PORTB = 0b00100100; //initialize PB5

while (1)

{

char input = USART\_rx();

if (input == 'h')

{

USART\_tx(helpScreen);

}

else if (input == 't')

{

USART\_tx("\n");

tempC();

}

else if (input == 'T')

{

USART\_tx("\n");

tempF();

}

else if (input == 'o')

{

PORTB &= ~(1<<5);

}

else if (input == 'O')

{

PORTB |= (1<<5);

}

else if (input == 's')

{

USART\_tx("\n");

USART\_tx(string);

}

else if (input == 'i')

{

USART\_tx("\n");

randomValue();

blinkLED();

}

}

}

void tempC()

{

*uint16\_t* temperature = adcRead();

float tempC\_float = ((temperature)\*(110.0/1023.0));

*snprintf*(array,sizeof(array), "%f\r\n", tempC\_float);

USART\_tx(array);

return;

}

void tempF()

{

*uint16\_t* temperature = adcRead();

float tempF\_float = ((((temperature) \* (110/1023.0)) \* 9.0/5.0) + 32.0);

*snprintf*(array,sizeof(array), "%f\r\n", tempF\_float);

USART\_tx(array);

PORTB = (1<<2);

return;

}

void randomValue()

{

int max = 3;

int min = 0;

randomNumber = *rand*() % (max - min + 1) + min;

*snprintf*(array,sizeof(array), "%d\r\n", randomNumber);

USART\_tx(array);

return;

}

void blinkLED()

{

if(randomNumber == 3)

{

PORTB = (0<<2);

*\_delay\_ms*(3000);

PORTB = (1<<2);

}

else if(randomNumber == 2)

{

PORTB = (0<<2);

*\_delay\_ms*(2000);

PORTB = (1<<2);

}

else if(randomNumber == 1)

{

PORTB = (0<<2);

*\_delay\_ms*(1000);

PORTB = (1<<2);

}

return;

}

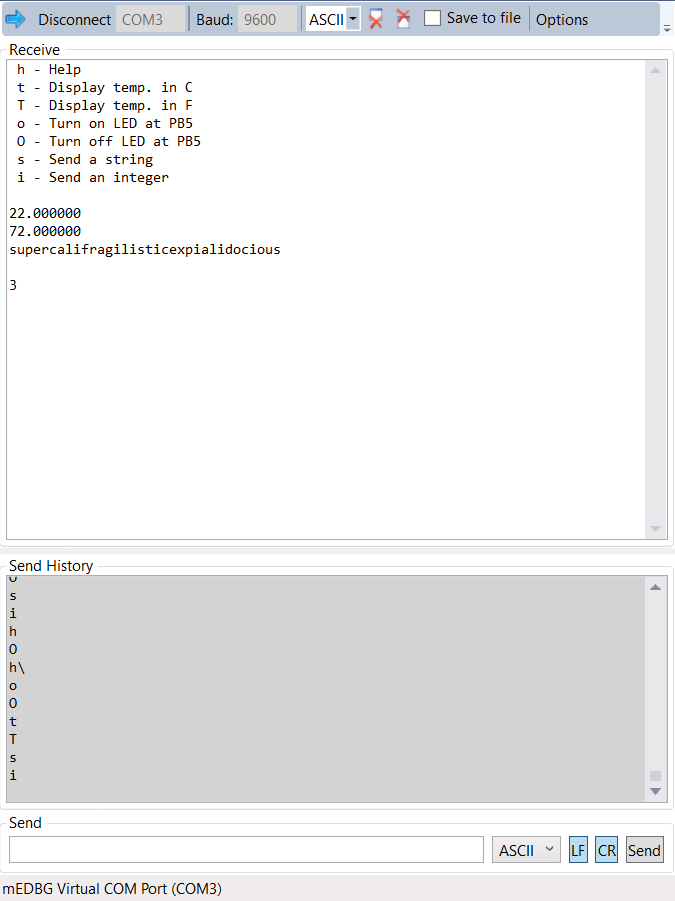
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

N/A

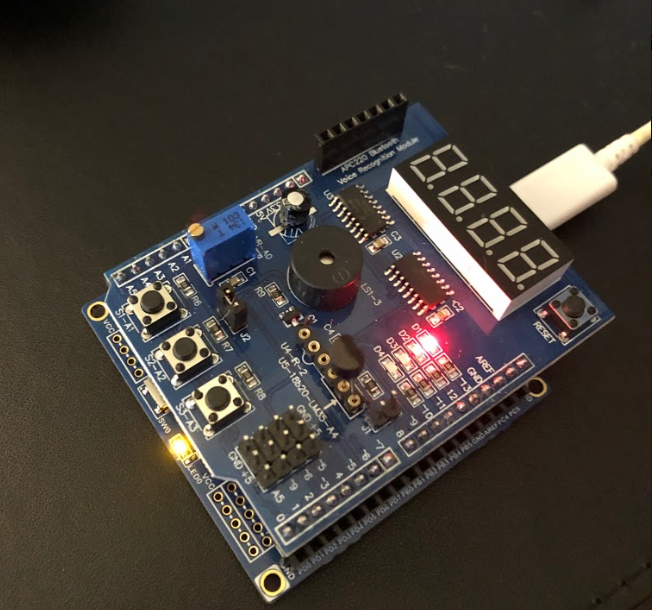
1. **SCHEMATICS**

N/A

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



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



1. **VIDEO LINKS OF EACH DEMO**

Demo Video: https://youtu.be/aZcA05d\_xK4

1. **GITHUB LINK OF THIS DA**

<https://github.com/TannerTindall51/tindalltannerm_submission/tree/master/Midterm>

**Student Academic Misconduct Policy**

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

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

Tanner Tindall