CPE301 – SPRING 2020

Design Assignment 3B

Student Name: Mateo Markovic

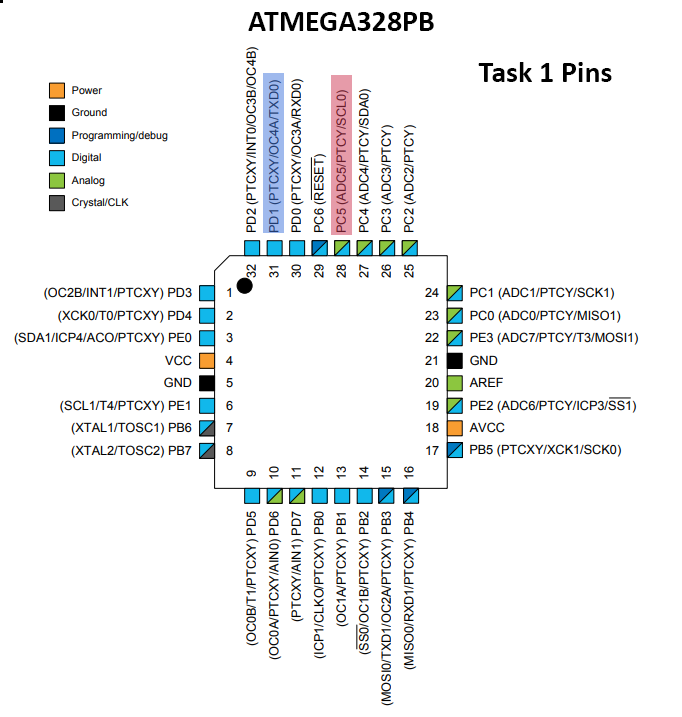
Student #: 2001338139

Student Email: Markom1@unlv.nevada.edu

Primary Github address: <https://github.com/mateom99/submission_da>

Directory: DesignAssignments/DA3B

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



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

/\*

\* DA3BT1.c

\*

\*

\* Author : Mateo Markovic

\*/

#define *F\_CPU* 16000000UL

#define BAUD 9600

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <util/setbaud.h>

#include <stdio.h>

#include <stdlib.h>

void UART\_Setup(); // UART\_Setup function prototype

void UART\_tx\_string(char \*data); // UART\_tx\_string function prototype

void readTemp(); // function to read temperature from LM35 on PC0

int OF = 0; // Global Variable to track overflow

float temp = 0; // Global Variable for temperature value

float Ftemp = 0; // Global Variable for Fahrenheit temperature value

int main(void) {

// timer setup

TCCR1A = 0; // Normal Operation

TCCR1B = 4; // set the pre-scaler to 256 and start timer

TCNT1=59285; // start the count at 59285

TIMSK1 = (1<<TOIE1); // Enable the timer interrupt

sei(); // Enable global interrupts

UART\_Setup(); // run UART setup

// ADC setup

DDRC = 0; // Make port c an input

PORTC = 0;

ADMUX |= (1<<REFS0); // 5v Voltage ref

// select ADC0 (PC5)

ADMUX |= (0<<MUX3);

ADMUX |= (1<<MUX2);

ADMUX |= (0<<MUX1);

ADMUX |= (1<<MUX0);

ADCSRA |= (1<<ADEN); // enable the ADC

ADCSRA |= (0<<ADSC); // don't do a conversion yet

// set pre-scaler as 128

ADCSRA |= (1<<ADPS2);

ADCSRA |= (1<<ADPS1);

ADCSRA |= (1<<ADPS0);

while (1);

}

// function to setup UART and Baud rate

void UART\_Setup(){

UBRR0H = *UBRRH\_VALUE*;

UBRR0L = *UBRRL\_VALUE*;

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

UCSR0B = \_BV(TXEN0) | \_BV(RXEN0); // Enable TX and RX

}

// function to send data to serial port

void UART\_tx\_string(char \*data){

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

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

UDR0 = \*data;

data++;

}

}

void readTemp(){

temp = 0; // reset the temperature

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

while(ADCSRA & (1<<ADSC)); // wait for conversion to finish

temp = ADC; // set the value from the temperature sensor to the temp variable

temp = (temp/1024) \* 500; // convert adc value to Centigrade

Ftemp = (temp \* 9/5) + 32; // convert C to F temp

}

// OF increments once every 0.1 seconds

ISR (TIMER1\_OVF\_vect){

OF++; // increment overflow

TCNT1 = 59285; // make sure to set the counter back to 59285

char tempChar [30]; // char array from temp float value

char fTempChar [30]; // char array from Fahrenheit temp float value

if (OF == 5){ // every 5 seconds (5 \* 0.1)

readTemp(); // get the temperature

UART\_tx\_string("\nCentigrade: ");

*snprintf*(tempChar,sizeof(tempChar),"%f\r\n", temp); // convert temp to a char \* (tempChar)

UART\_tx\_string(tempChar); // output the Centigrade temp

*snprintf*(fTempChar,sizeof(fTempChar),"%f\r\n", Ftemp); // convert temp to a char \* (tempChar)

UART\_tx\_string("Fahrenheit: ");

UART\_tx\_string(fTempChar); // output the Fahrenheit temp

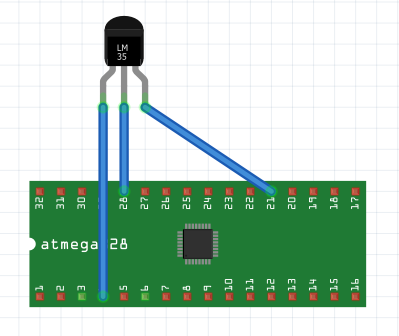
UART\_tx\_string("\n");

OF = 0; // reset the overflow

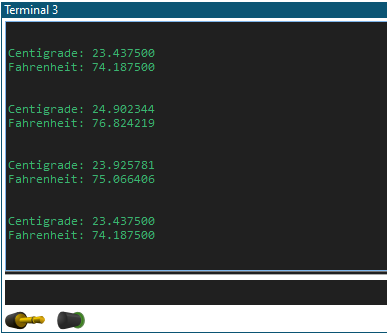
}

}

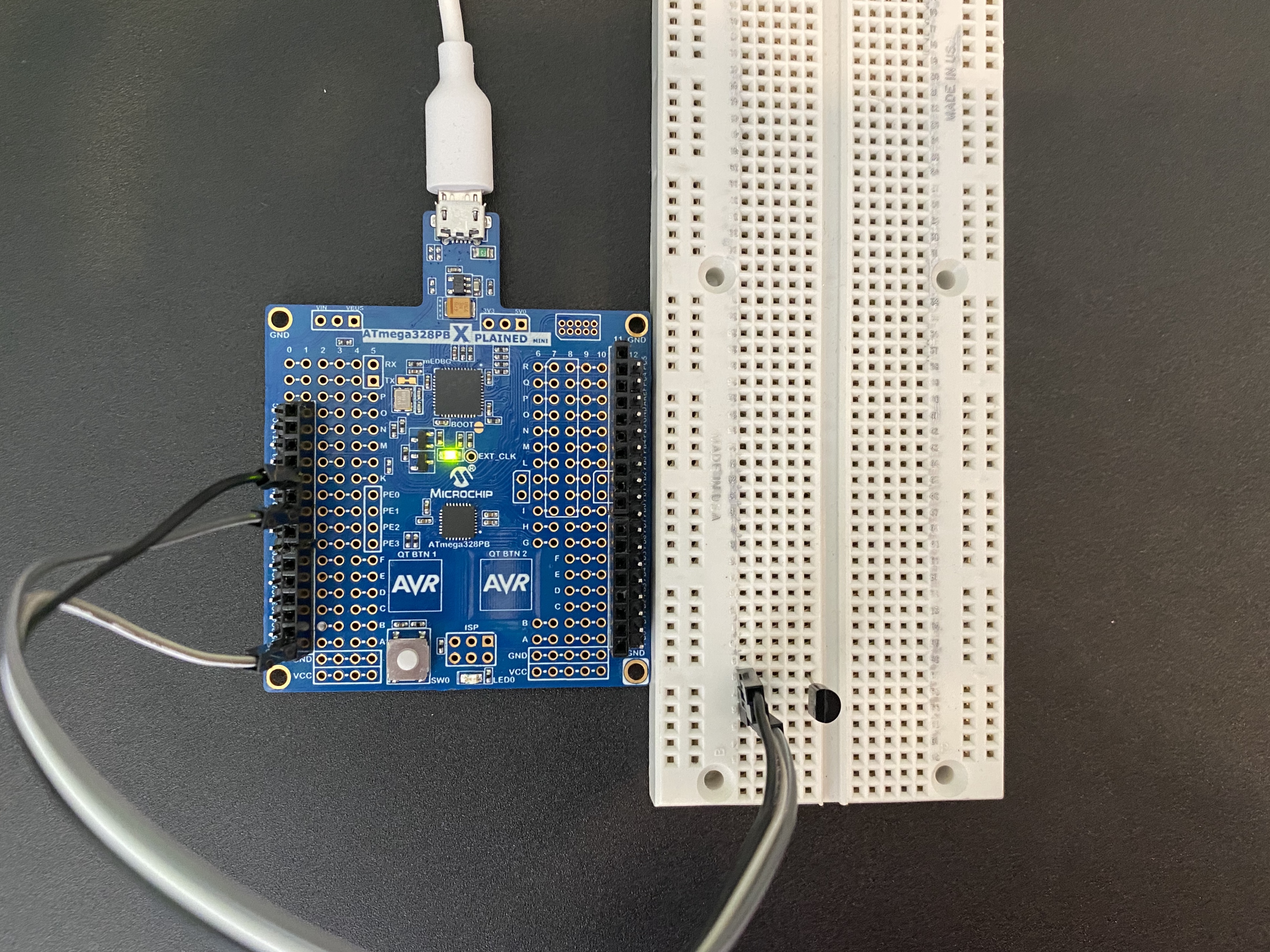
1. **SCHEMATICS**



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



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



1. **VIDEO LINKS OF EACH DEMO**

Task 1 Demonstration: <https://youtu.be/KTE6PXo8zBo>

1. **GITHUB LINK OF THIS DA**

<https://github.com/mateom99/submission_da/tree/master/DesignAssignments/DA3B>

**Student Academic Misconduct Policy**

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

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

Mateo Markovic