CPE301 – SPRING 2024

Design Assignment 4

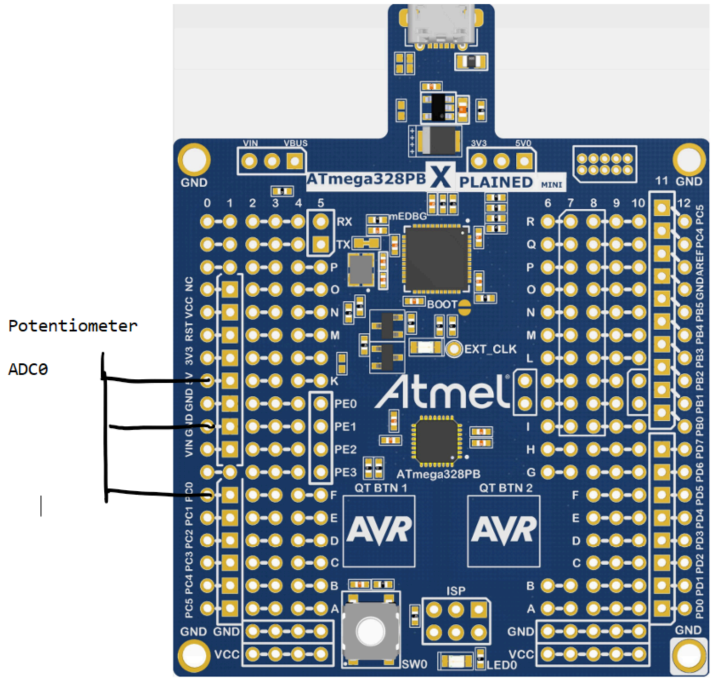
Student Name: Nathan Ramos
Student #: 5006437353
Student Email: ramosn8@unlv.nevada.edu
Primary Github address: https://github.com/n8ramos/
Directory: /atmega328pb
Video Playlist:
https://www.youtube.com/playlist?list=PL2RpCRW8TC6YOj-NnLPqqfRTV48RcUe48

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Components Used:

- potentiometer
- atmega328pb Xplained Mini
- microchip studio 7

Block diagram with pins used in the Atmega328PB (only)

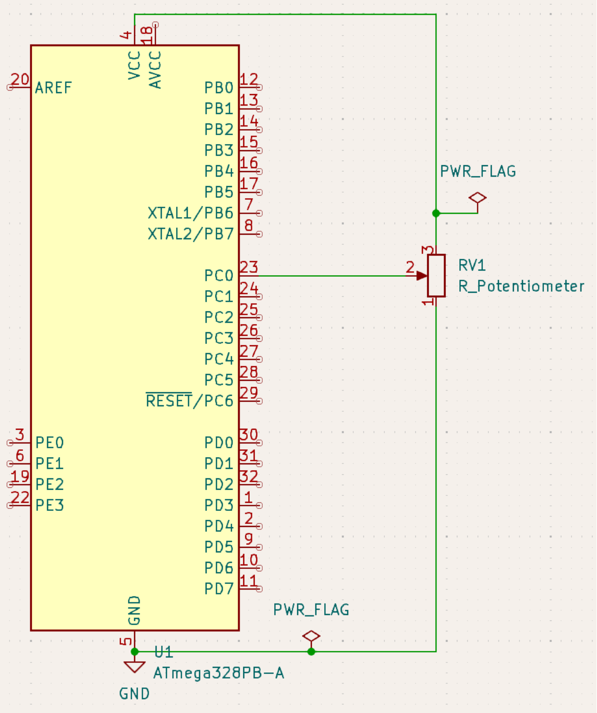


2. DEVELOPED CODE OF TASK 1

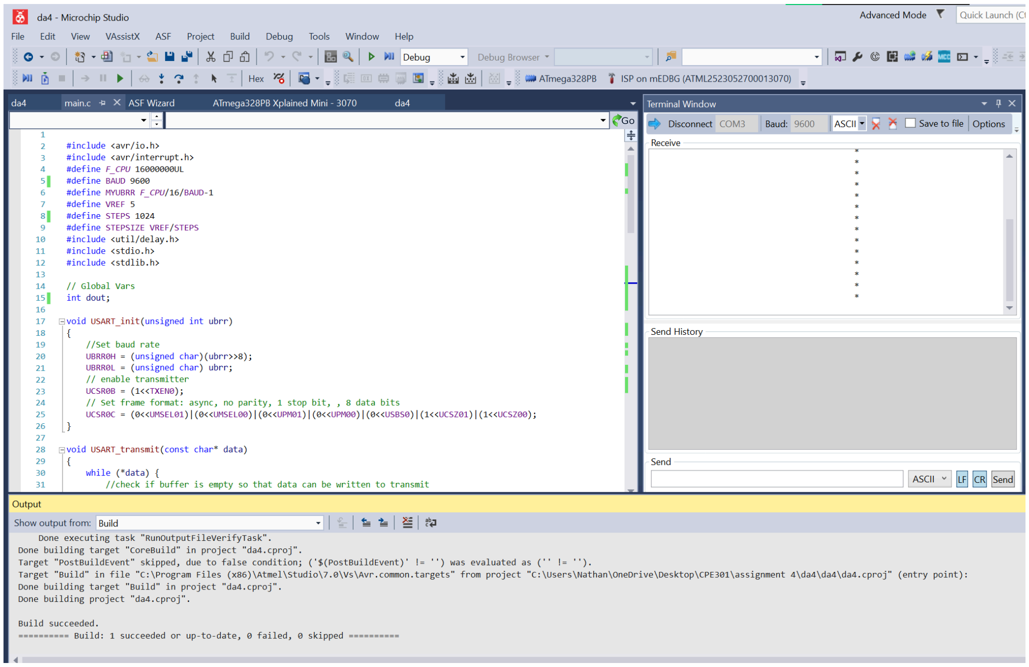
```
#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 16000000UL
#define BAUD 9600
```

```
#define MYUBRR F_CPU/16/BAUD-1
#define VREF 5
#define STEPS 1024
#define STEPSIZE VREF/STEPS
#include <util/delay.h>
#include <stdio.h>
#include <stdlib.h>

// Global Vars
int dout;

void USART_init(unsigned int ubrr)
{
        //Set baud rate
        UBRR0H = (unsigned char)(ubrr>>8);
        UBRR0L = (unsigned char) ubrr;
        // enable transmitter
        UCSR0B = (1<<TXEN0);
        // Set frame format: async, no parity, 1 stop bit, , 8 data bits
        UCSR0C =
(0<<UMSEL01)|(0<<UMSEL00)|(0<<UPM01)|(0<<UPM00)|(0<<USBS0)|(1<<UCSZ01)|(1<<UCSZ00);
}

void USART_transmit(const char* data)
{
        while (*data) {
                //check if buffer is empty so that data can be written to transmit
                while (!(UCSR0A & (1 << UDRE0)));
                UDR0 = *data; //copy “data” to be sent to UDR0
                ++data;
        }
}

void USART_transmitChar(const char data)
{
        //check if buffer is empty so that data can be written to transmit
        while (!(UCSR0A & (1 << UDRE0)));
        UDR0 = data; //copy character to be sent to UDR0
}

ISR(TIMER1_COMPA_vect) {
        cli();
        int volt = dout/20; // scale dout from 0-1023 to 0-50
        for (int i = 0; i <= volt; i++) {
                USART_transmitChar(' '); // output # of spaces representing # of 0.1v
        }
        USART_transmitChar('*');
        // display min and max of oscilloscope
        switch (dout) {
                case 0:
                        USART_transmit(" MIN");
                        break;
        case 1023:
            USART_transmit(" MAX");
            break;
    }
    USART_transmitChar('\n');

    sei();
}

ISR (ADC_vect)
{
    cli();
    dout = ADC;
    // start ADC conversion
    ADCSRA |=(1<<ADSC);
    sei();
}

void timer1_CTC_init() {
    TCCR1A |= (0 << WGM11) | (0 << WGM10); // set bits WGM1[1:0] for CTC
    TCCR1B |= (0 << WGM13) | (1 << WGM12) | (0 << CS12) | (1 << CS11) | (0 << CS10); // set
WGM[3:2] for CTC & CS[2:0] for prescaler = 8
    TIMSK1 |= (1 << OCIE1A); // sets bit to enable CTC comparator A
    TCNT1 = 0; // initialize counter to 0
    OCR1A = 19999; // counter for 0.01 seconds
}

int main(void) {
    // For the USART output
    USART_init(MYUBRR);
    // initialize timer1 to 0.01 seconds
    timer1_CTC_init();
    sei();
    //set channel to take input for ADC0,right justified, AVcc with external cap at AREF
    ADMUX = (1 << REFS0) | (0 << MUX0); // Also defaults ADC0 reading
    //set prescaler to 64, enable ADC interrupt,enable ADC,start conversion
    ADCSRA |= (1 << ADEN) | (1 << ADIE) | (1 << ADPS2) | (1 << ADPS1) | (0 << ADPS0) | (1 << ADSC);
    while (1) {
        // start ADC conversion
        ADCSRA |=(1<<ADSC);
    }
}
```

3. SCHEMATICS



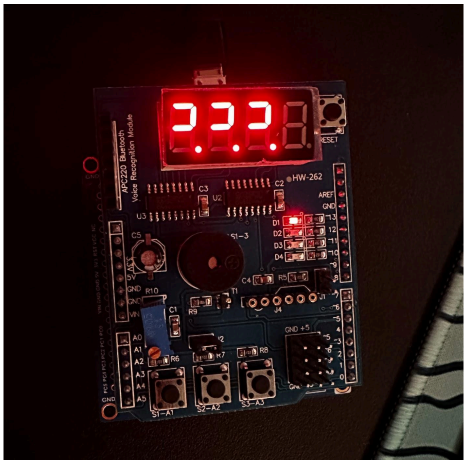
4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)



```

#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 16000000UL
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1
#define VREF 5
#define STEPS 1024
#define STEPSIZE VREF/STEPS
#include <util/delay.h>
#include <stdio.h>
#include <stdlib.h>

// Global Vars
int dout;

void USART_init(unsigned int ubrr)
{
    //Set baud rate
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char) ubrr;
    // enable transmitter
    UCSR0B = (1<<TXEN0);
    // Set frame format: async, no parity, 1 stop bit, , 8 data bits
    UCSR0C = (0<<UMSEL01)|(0<<UMSEL00)|(0<<UPM01)|(0<<UPM00)|(0<<USBS0)|(1<<UCSZ01)|(1<<UCSZ00);
}

void USART_transmit(const char* data)
{
    while (*data) {
        //check if buffer is empty so that data can be written to transmit
```

```
    Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "da4.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Users\Nathan\OneDrive\Desktop\CPE301\assignment 4\da4\da4\da4.cproj" (entry point):
Done building target "Build" in project "da4.cproj".
Done building project "da4.cproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



6. VIDEO LINKS OF EACH DEMO

https://youtu.be/rtKgkxnpQP4

7. GITHUB LINK OF THIS DA

https://github.com/n8ramos/atmega328pb/tree/main

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"This assignment submission is my own, original work".
Nathan Ramos