CPE301 – SPRING 2019

Design Assignment 4A

Student Name: Geovanni Portillo

Student #: 8000603824

Student Email: [portig1@unlv.nevada.edu](mailto:portig1@unlv.nevada.edu)

Primary Github address: <https://github.com/portig1/submissions_E>

Directory: submissions\_E/DA/LAB4A/

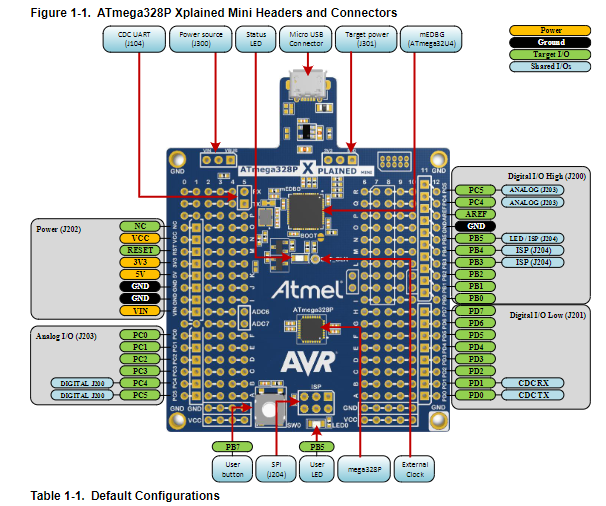
Submit the following for all Labs:

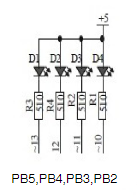
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

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

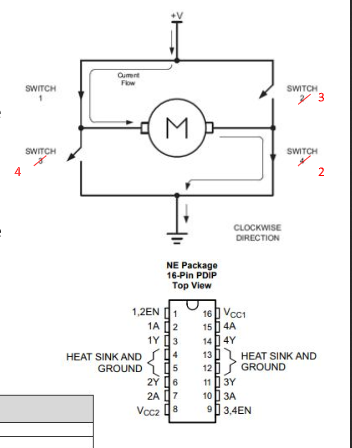
Atmel Studio 7

ATmega328PB Xplained mini





Schematic for shield LEDs (PB5:2) and Potentiometer (PC0)



Schematic for the driver L293D

1. **INITIAL CODE OF TASK 1 (Replacing Motor With LED)**

/\*

I didn't realize I don’t have the driver chip for the motor in time for this assignment.

There wasn't one included in the lecture kit and the lab kit only has one which my partner has at the

time of writing this. So to still demonstrate the application of the potentiometer being used to vary the duty cycle

I’m using the LEDs on the multifunction shield instead. Though the only difference in the code would be where

I turn the LEDs off and on, I would change that with turning the 1,2 enable pin on the driver on and off.

\*/

#define *F\_CPU* 16000000UL

#define BAUD\_RATE 9600

#define BAUD\_PRESCALLER (((*F\_CPU* / (BAUD\_RATE \* 16UL))) - 1)

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <util/delay.h>

void usart\_init ();

void USART\_send(unsigned char data);

void USART\_putstring(char\* StringPtr);

int MTR\_Status = 0; //Will be used for the pin change interrupt to toggle whether the motor can be on or off

int psuedoFallingEdgeDetector = 0; //Will be used so that only on the falling edge of the pin change process will the MTR \_Status be toggled

int main (void)

{

DDRB = 0xFF;

PORTB = 0xFF;

usart\_init ();

PCICR = (1 << PCIE1);

PCMSK1 = (1 << PCINT9); //Enable pin change interrupt vector 1 then enable pin change interrupt 9

sei();

/\*\* Setup and enable ADC \*\*/

ADMUX = (0<<REFS1)| // Reference Selection Bits

(1<<REFS0)| // AVcc - external cap at AREF

(0<<ADLAR)| // ADC Left Adjust Result

(0<<MUX2)| // Analog Channel Selection Bits

(0<<MUX1)| // ADC0 (PC0) Potentionmeter

(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)| // CLK/32

(1<<ADPS0);

while (1)

{

PORTC = (1 << 1); //Enable pull-up for PC1

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

while((ADCSRA&(1<<ADIF))==0);//wait for conversion to finish

ADCSRA |= (1<<ADIF);

int tempC = ADCL;

tempC = tempC | (ADCH<<8);

char output[20];

*snprintf*(output, sizeof(output), "%d\r\n", tempC); //prints out potentiometer value to serial terminal

USART\_putstring(output);

//When the ADC value from the potentiometer is below 10, a duty cycle of 0% will be output. Then above 10 and below 20, a duty cycle of 10%. This goes on until the value is above

//90 for which only a 95% duty cycle will be produced

if((0 <= tempC) & (tempC < 10))

{

if(MTR\_Status)

{

//DC = 0%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(0);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(1000);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

else if((10 <= tempC) & (tempC < 20))

{

if(MTR\_Status)

{

//DC = 10%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(100);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(900);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

else if((20 <= tempC) & (tempC < 30))

{

if(MTR\_Status)

{

//DC = 20%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(200);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(800);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

else if((30 <= tempC) & (tempC < 40))

{

if(MTR\_Status)

{

//DC = 30%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(300);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(700);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

else if((40 <= tempC) & (tempC < 50))

{

if(MTR\_Status)

{

//DC = 40%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(400);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(600);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

else if((50 <= tempC) & (tempC < 60))

{

if(MTR\_Status)

{

//DC = 50%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(500);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(500);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

else if((60 <= tempC) & (tempC < 70))

{

if(MTR\_Status)

{

//DC = 60%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(600);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(400);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

else if((70 <= tempC) & (tempC < 80))

{

if(MTR\_Status)

{

//DC = 70%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(700);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(300);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

else if((80 <= tempC) & (tempC < 90))

{

if(MTR\_Status)

{

//DC = 80%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(800);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(200);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

else

{

if(MTR\_Status)

{

//DC = 95%

//Enable motor in clockwise direction;

PORTB = 0;

*\_delay\_us*(950);

//Disable motor

PORTB = 0xFF;

*\_delay\_us*(50);

}

else

{

//Disable motor

PORTB = 0xFF;

}

}

}

return 0;

}

ISR(PCINT1\_vect) //For Pin Change INT 9 on PC1

{

psuedoFallingEdgeDetector++;

if(psuedoFallingEdgeDetector > 1) //Only after the pin change interrupt occurs a second time will it toggle, needed since there isn't a default way of making it so the interrupt occurs on the falling edge.

{

MTR\_Status ^= 1; //Toggles MTR\_Status

psuedoFallingEdgeDetector = 0;

}

*\_delay\_ms*(200); //Delay for switch debouncing

}

void usart\_init (void)

{

UBRR0H = (*uint8\_t*)(BAUD\_PRESCALLER >> 8);

UBRR0L = (*uint8\_t*)(BAUD\_PRESCALLER);

UCSR0B = (1 << RXEN0) | (1 << TXEN0);

UCSR0C = (3 << UCSZ00);

}

void USART\_send( unsigned char data) {

while (!(UCSR0A & (1 << UDRE0))); //wait until UDR0 is empty

UDR0 = data; //transmit ch

}

void USART\_putstring(char\* StringPtr) {

while (\*StringPtr != 0x00) {

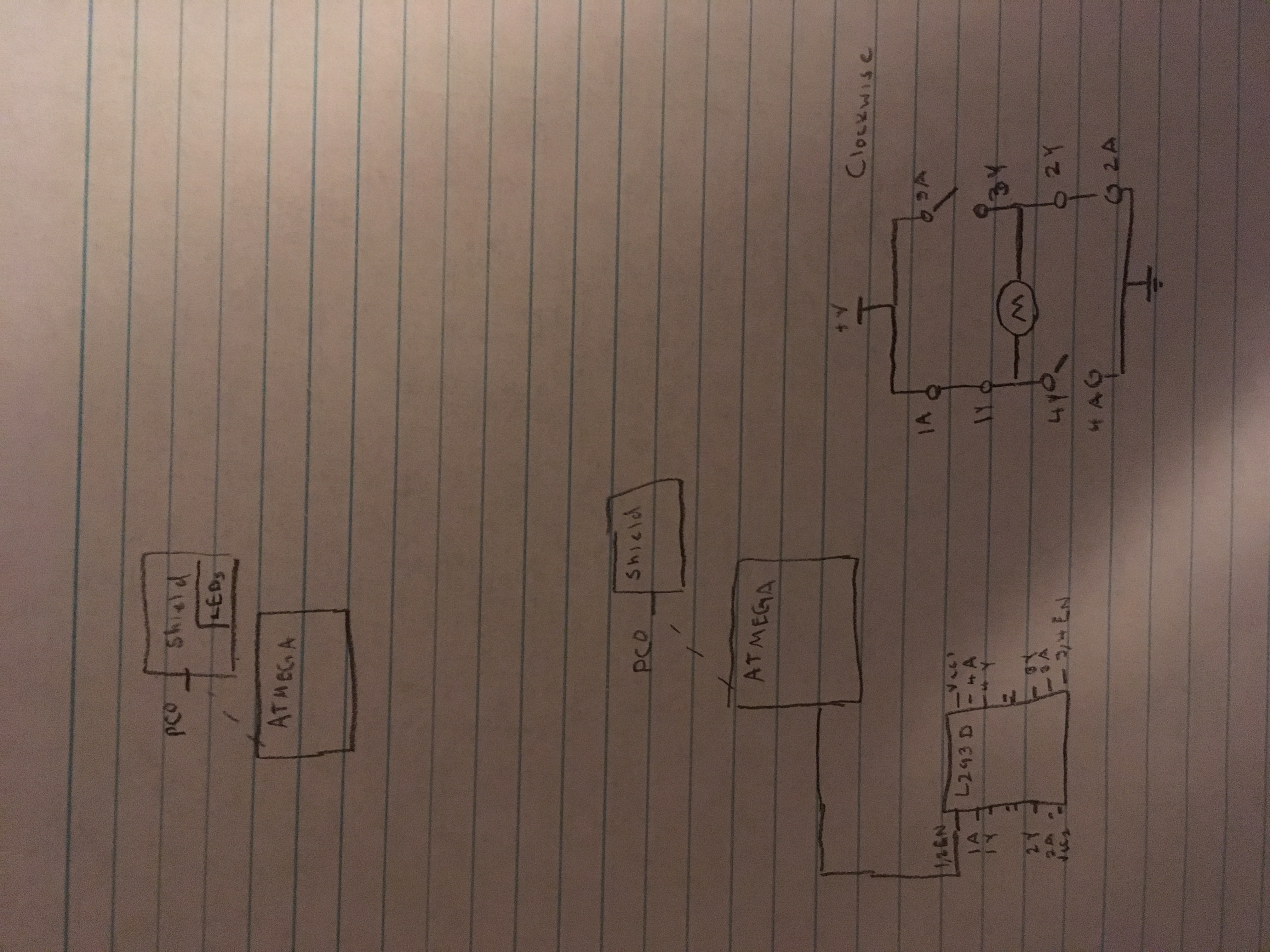
USART\_send(\*StringPtr);

StringPtr++;

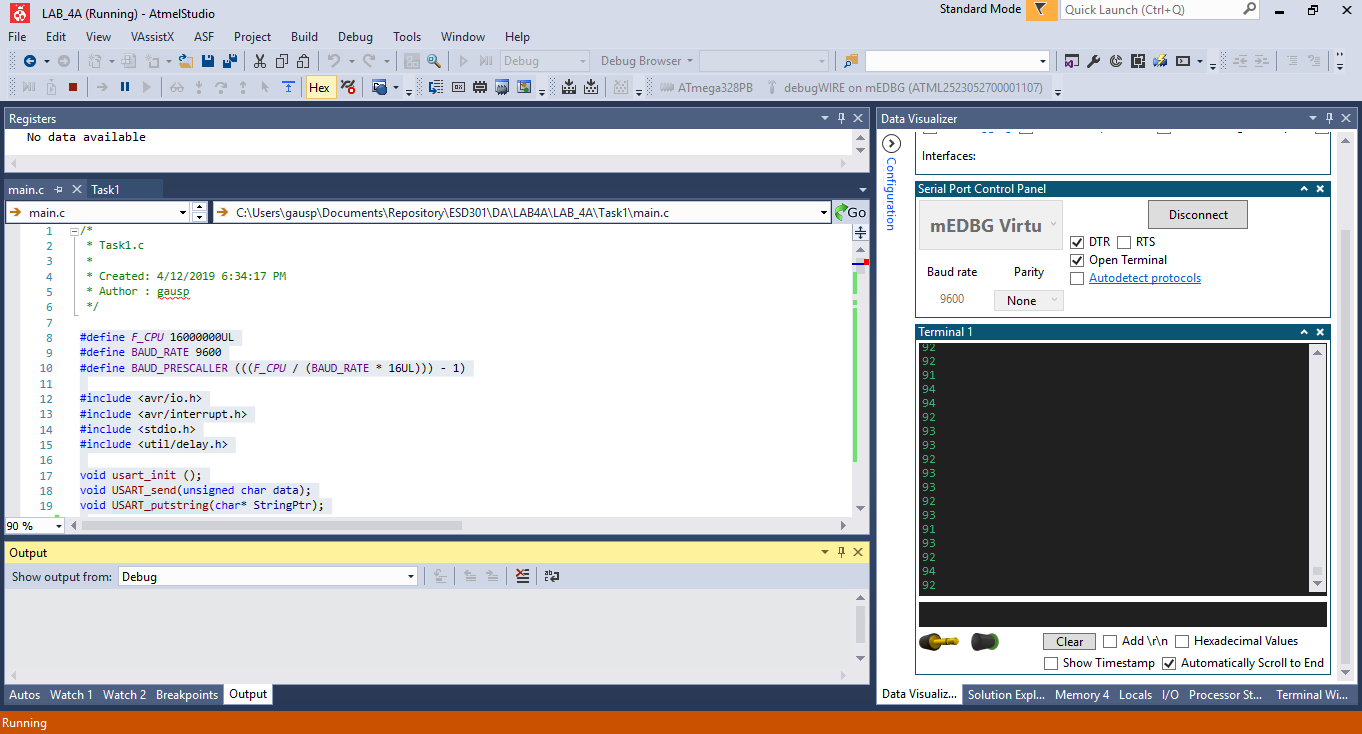
}

}

1. **SCHEMATICS**

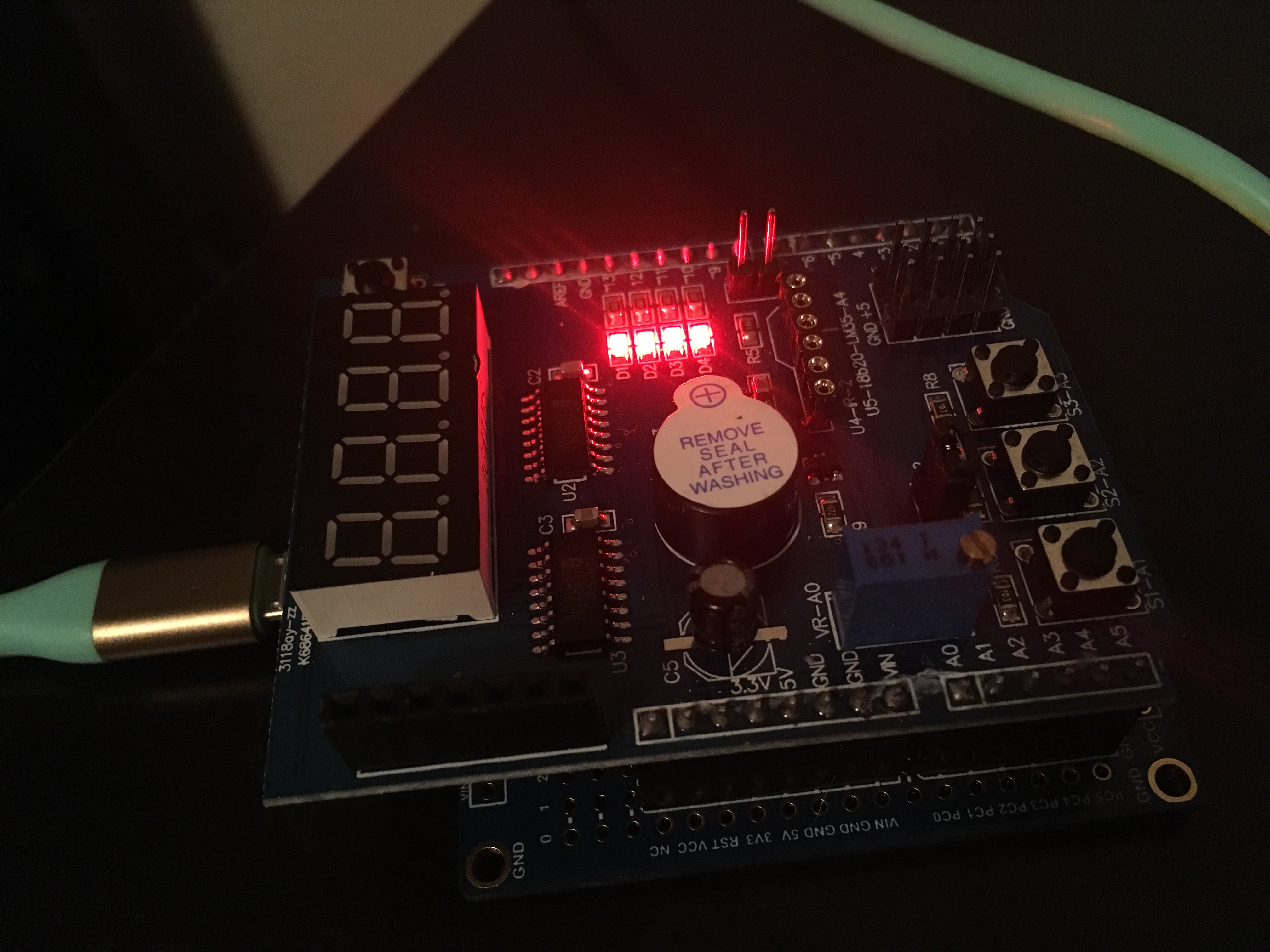


1. **SCREENSHOTS OF EACH TASK OUTPUT**



Output of potentiometer value on serial terminal

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



Board setup for PWM with LEDs

1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/KiNvxbXI3i0>

1. **GITHUB LINK OF THIS DA**

https://github.com/portig1/submissions\_E/tree/master/DA/LAB4A

**Student Academic Misconduct Policy**

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

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

Geovanni Portillo