CPE301 – FALL 2019

DA4B

Student Name: Andrew Buchanan

Student #:5003154346

Student Email: buchaa2@unlv.nevada.edu

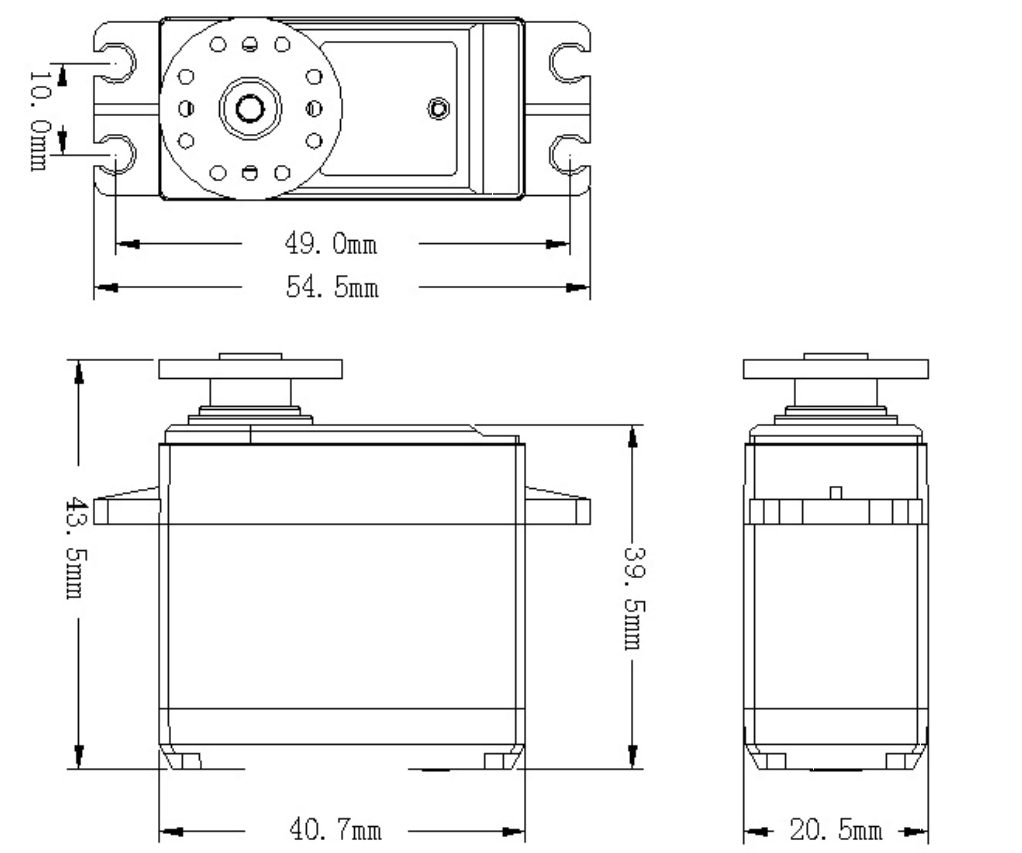
Primary Github address: <https://github.com/buchaa2/103EPC>

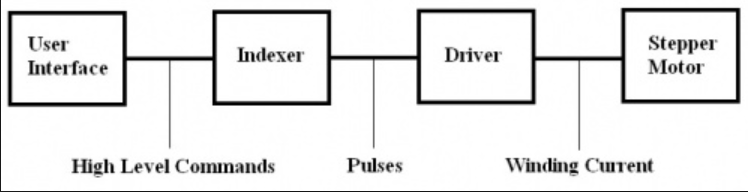
Directory:

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**





1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**
2. #define *F\_CPU* 16000000UL
3. #include <avr/io.h>
4. #include <avr/interrupt.h>
5. #include <util/delay.h>
6. volatile unsigned int ADCVal;
7. void ADC\_init();
8. void timer\_init();
9. int main(void){
10. // initialize ADC
11. ADC\_init();
12. //set timer
13. TCCR1B |=(1<<WGM12)|(1<<CS11); //CTC MODE prescaler = 8
14. // set DDR's
15. DDRB = 0xFF; //all outputs
16. DDRC = 0; //all input
17. PORTB = 0; //start at 0
19. while(1){
20. ADCSRA|=(1<<ADSC);
21. while ((ADCSRA&(1<<ADIF))==0); //wait until done converting
22. ADCVal = ADC & 0x03FF; //take only 10 bits
23. OCR1A = 10\*ADCVal; //scale for proper delay
25. //procedure for PB0
26. PORTB |= (1<<PB0);
27. while(!(TIFR1 & (1<<OCF1A))); //wait until timer flag is set. repeat
28. TIFR1 |= (1<<OCF1A);
29. PORTB &= ~(1<<PB0);
30. while(!(TIFR1 & (1<<OCF1A)));
31. TIFR1 |= (1<<OCF1A);
33. //procedure for PB1
34. PORTB |= (1<<PB1);
35. while(!(TIFR1 & (1<<OCF1A)));
36. TIFR1 |= (1<<OCF1A);
37. PORTB &= ~(1<<PB1);
38. while(!(TIFR1 & (1<<OCF1A)));
39. TIFR1 |= (1<<OCF1A);
41. //procedure for PB2
42. PORTB |= (1<<PB2);
43. while(!(TIFR1 & (1<<OCF1A)));
44. TIFR1 |= (1<<OCF1A);
45. PORTB &= ~(1<<PB2);
46. while(!(TIFR1 & (1<<OCF1A)));
47. TIFR1 |= (1<<OCF1A);
49. //procedure for PB3
50. PORTB |= (1<<PB3);
51. while(!(TIFR1 & (1<<OCF1A)));
52. TIFR1 |= (1<<OCF1A);
53. PORTB &= ~(1<<PB3);
54. while(!(TIFR1 & (1<<OCF1A)));
55. TIFR1 |= (1<<OCF1A);
57. }
58. }
59. void ADC\_init(void){
60. DIDR0 = 0x1;
61. ADMUX = (1<<REFS0); //input is PC0
63. ADCSRA |=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
64. ADCSRB = 0x0;
65. }

**Task 2**

#define *F\_CPU* 16000000L

#include <avr/io.h>

#include <util/delay.h>

volatile unsigned int ADCVal;

int main(void) {

//Set DDR's

DDRB = 0xFF; // Set port B as output

//Set ADC

DIDR0 = 0x1;

ADMUX = (1<<REFS0); //input pc0

ADCSRA |=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);

ADCSRB = 0x0;

TCCR1B = (1<<WGM13)|(1<<WGM12)|(1<<CS11)|(1<<CS10); //fast PWM mode and prescaler = 64

TCCR1A = (1<<COM1A1)|(1<<WGM11); // Non-inverting PWM

ICR1 = 2500; // Top of counter

//Main body loop

while (1)

{

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

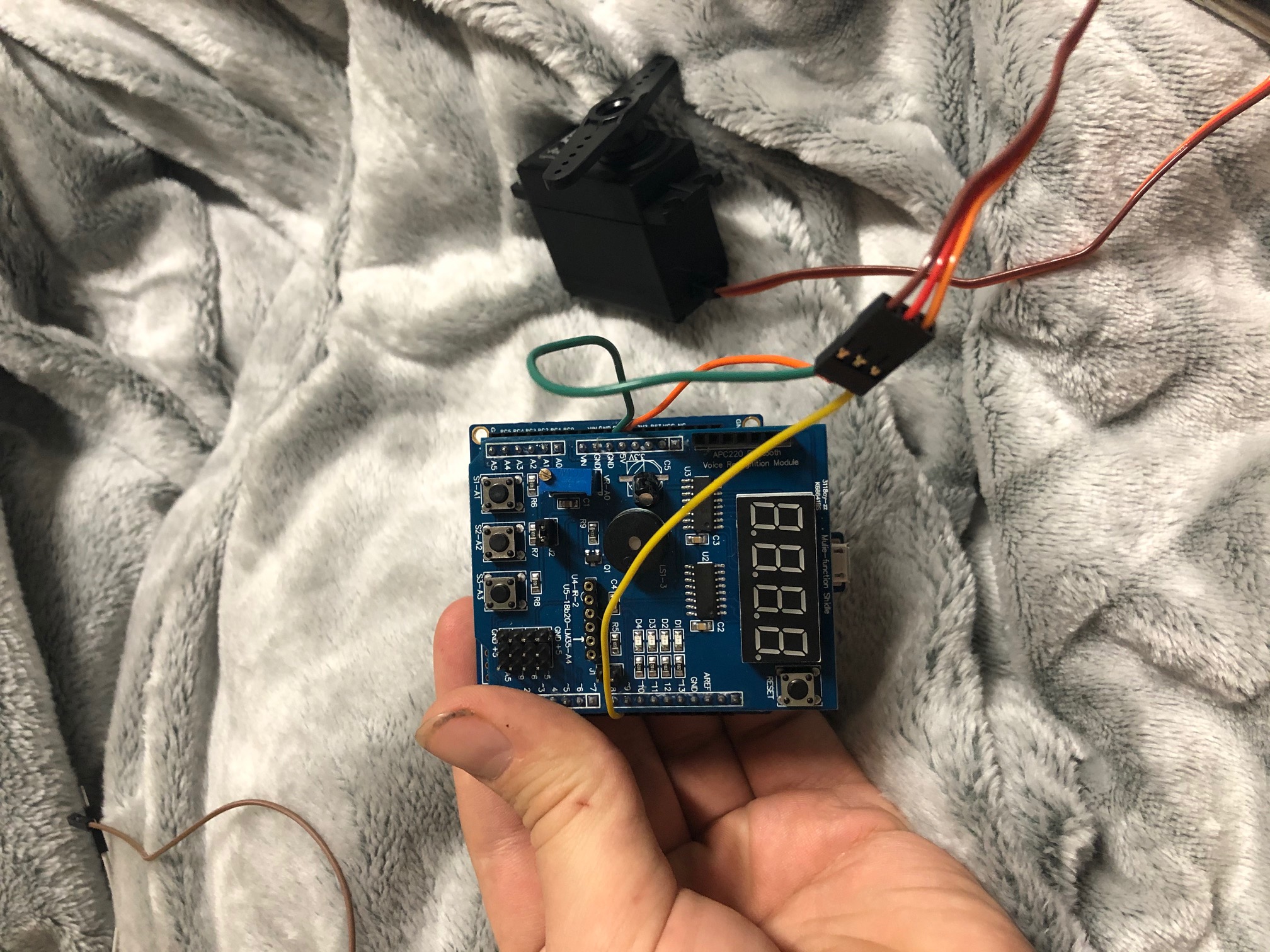
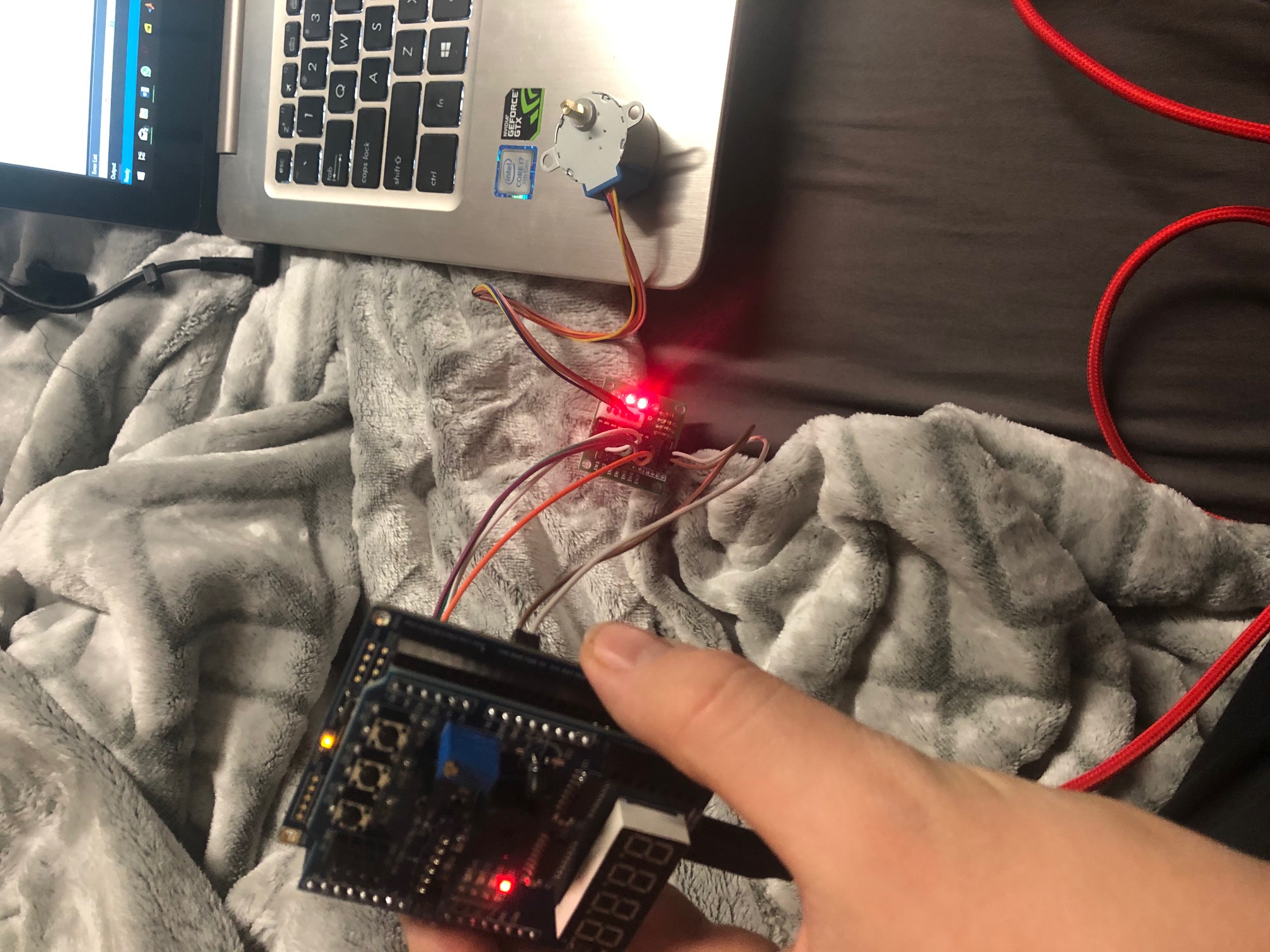
while((!ADCSRA) &(1<<ADIF)); // Wait for conversion

OCR1A = ADC/2 ; //Set pwm with ADCValue

}

}

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



1. **VIDEO LINKS OF EACH DEMO**

Task 1

<https://www.youtube.com/watch?v=Hp5TtBw-bvM>

task2

<https://www.youtube.com/watch?v=oKQtA2IMzQ0>

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

Andrew Buchanan