CPE301 – SPRING 2019

Design Assignment X

Student Name: Jeb Marinas

Student #:5003376451

Student Email: marinj4@unlv.nevada.edu

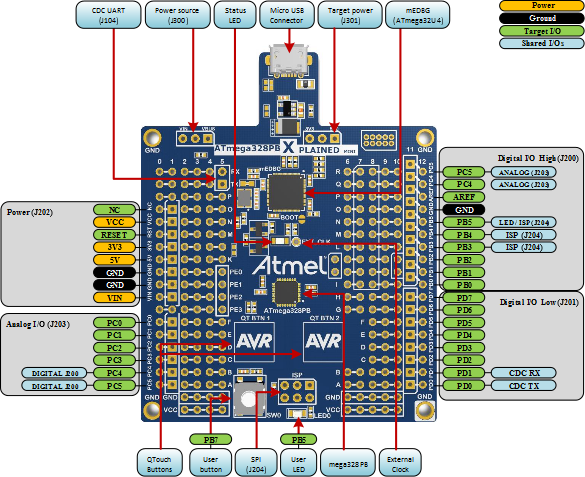
Primary Github address:github.com/jebmarinas

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**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

void adc\_init(void);

void read\_adc(void);

volatile int control = 0; //this will control the motor

volatile unsigned int speed; // this will control how was the motor go

volatile unsigned int adc;

void adc\_init(void)

{

ADMUX = (0<<REFS1)| // reference selection bit

(1<<REFS0)| //AVCC EXTERNAL CAP AT REF

(1<<ADLAR)| //ADC LEFT ADJUST RESULT

(0<<MUX2)| //ANALOG CHANNEL SELECTION BITS

(0<<MUX1)| //ADC

(0<<MUX0);

ADCSRA = (1<<ADEN)| // ADC ENABLE

(0<<ADSC)| // ADC START CONVERSION

(0<<ADATE)| // ADC AUTO TRIGGER ENABLE

(0<<ADIF)| // ADC INTERRUPT ENABLE

(0<<ADIE)|// ADC PRESCALER SELECT BIT

(1<<ADPS2)|

(1<<ADPS1)|

(1<<ADPS0); // SELECT CHANNEL

}

void read\_adc(void)

{

unsigned char i = 4;

adc = 0;

while(i--)

{

ADCSRA |=(1<<ADSC);

while(ADCSRA & (1<<ADSC));

adc += ADC;

*\_delay\_ms*(50);

}

adc = (adc/8);

}

int main(void)

{

DDRB = 0x0F;//will be used as an output

PORTB = 0x00;// the voltage in it will be 0

TCCR1B = (1 << WGM13)|(1 << WGM12)|(1 << CS11);// THE PRESSCALAR WILL BE AT 8

ICR1 = 62258;

adc\_init();

while(1)

{

read\_adc();

*\_delay\_ms*(30);

}

while(1)

{

PORTD = 0b01100110;

*\_delay\_ms*(50);

PORTD = 0b11001100;

*\_delay\_ms*(50);

PORTD = 0b10011001;

*\_delay\_ms*(50);

PORTD = 0b00110011;

*\_delay\_ms*(50);

if ((adc >= 981) && (adc < 1024 )) //when adc value is at its max

{

control = 1;

*\_delay\_ms*(20);

}

else if ((adc < 982) && (adc >= 758)) //will decrease pwm

{

control = 0;

speed = 0x1869;

*\_delay\_ms*(50);

}

else if ((adc < 770) && (adc >= 67)) //will decreasing pwm

{

control = 0;

speed = 0x124F;

*\_delay\_ms*(100);

}

else

{

PORTD = 0b00000000;//if resistor value is less than 5%

*\_delay\_ms*(100);

}

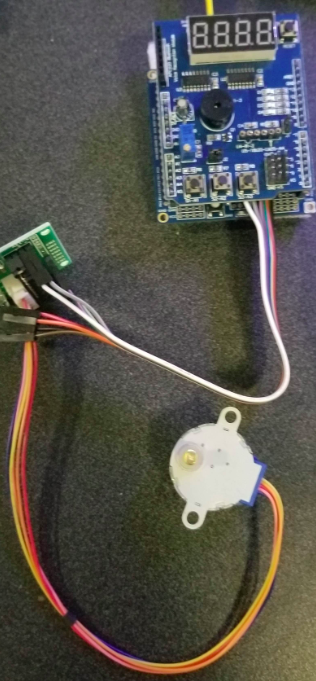
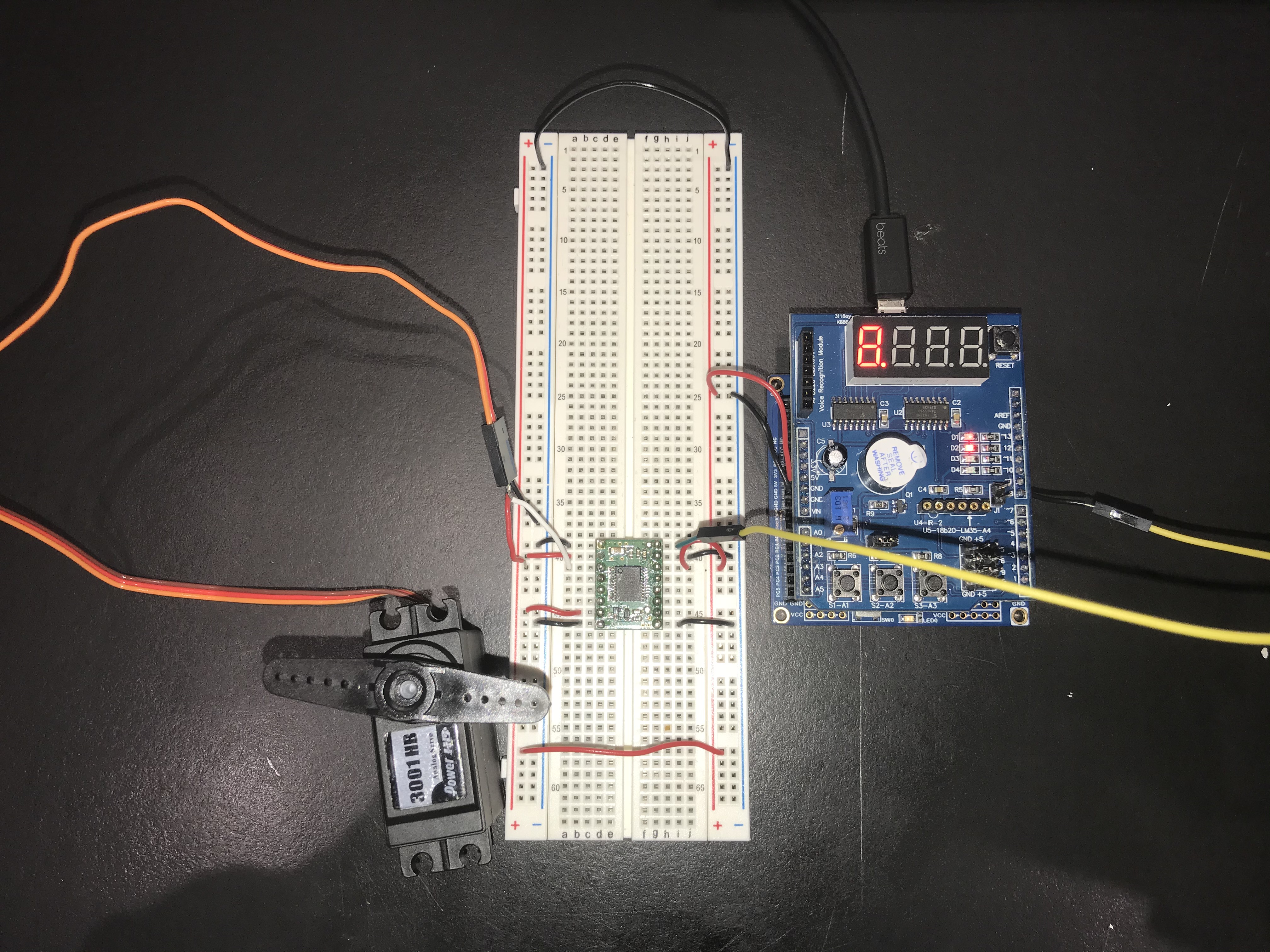
}

}

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**
2. #define *F\_CPU* 16000000UL
3. #include <avr/io.h>
4. #include <util/delay.h>
5. #include <avr/interrupt.h>
6. void adc\_init(void);
7. void read\_adc(void);
8. volatile float adc;
9. int check = 0;
10. void adc\_init(void)
11. {
12. ADMUX = (0<<REFS1)| // reference selection bit
13. (1<<REFS0)| //AVCC EXTERNAL CAP AT REF
14. (1<<ADLAR)| //ADC LEFT ADJUST RESULT
15. (0<<MUX2)| //ANALOG CHANNEL SELECTION BITS
16. (0<<MUX1)| //ADC
17. (0<<MUX0);
18. ADCSRA = (1<<ADEN)| // ADC ENABLE
19. (0<<ADSC)| // ADC START CONVERSION
20. (1<<ADATE)| // ADC AUTO TRIGGER ENABLE
21. (0<<ADIF)| // ADC INTERRUPT ENABLE
22. (0<<ADIE)|// ADC PRESCALER SELECT BIT
23. (1<<ADPS2)|
24. (1<<ADPS1)|
25. (0<<ADPS0); // SELECT CHANNEL
26. }
27. void read\_adc(void)
28. {
29. unsigned char i = 4;
30. adc = 0;
31. while(i--)
32. {
33. ADCSRA |=(1<<ADSC);
34. while(ADCSRA & (1<<ADSC));
35. adc += ADC;
36. *\_delay\_ms*(50);
37. }
38. adc = (adc/8);
39. }
40. int main(void)
41. {
42. adc\_init(); //Initializing ADC
43. PORTC |= 0b00000001;
44. DDRD = 0xFF; DDRB = 0xFF;//outputs
45. //Configuration of Timer1
46. TCCR1A |= (1 << COM1A1) |(1 << COM1B1) |(1 << WGM11);
47. TCCR1B |= (1 << WGM13) |(1 << WGM12) |(1 << CS11) |(1 << CS10);
48. ICR1 = 62258;
49. while(1)
50. {
51. read\_adc();
52. *\_delay\_ms*(50);
53. while(1)
54. {
55. ADCSRA |= (1<< ADSC);
56. while((ADCSRA & (1 << ADIF)) == 0);
57. check = ADCH;
58. if(check == 0)
59. {
60. OCR1A = 0;
61. *\_delay\_ms*(200);
62. }
63. if(check == 255)
64. {
65. OCR1A = 500;
66. *\_delay\_ms*(200);
67. }
68. }
69. }
70. }
71. **SCHEMATICS**
72. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



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



1. **VIDEO LINKS OF EACH DEMO**
2. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT