CPE301 – FALL 2019

Design Assignment 3B

Student Name: Cody Jones

Student #: 5002863328

Student Email: Jonesc30@unlv.nevada.edu

Primary Github address: https://github.com/Jonesc30/Submission

Directory: Submission

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**

Atmega328PB and LM35

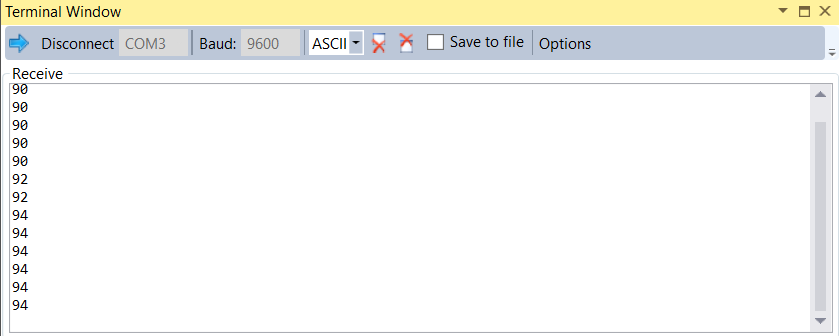
1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**
2. #define *F\_CPU* 16000000UL
3. #include <avr/io.h>
4. #include <avr/interrupt.h>
5. #include <stdio.h>
6. #include <stdlib.h>
7. #include <util/delay.h>
8. #include <math.h>
9. #define BAUDRATE 9600
10. #define BAUD\_PRESCALLER (((*F\_CPU* / (BAUDRATE \* 16UL))) - 1)
11. void USART\_init(void); //function to initialize USART
12. unsigned char USART\_receive(void); //function to receive through USART
13. void USART\_transmit(unsigned char data); //function to send through USART
14. void USART\_putstring(char\* StringPtr); //function to scan through the string
15. void ADC\_init (void); //function to initialize ADC
16. volatile unsigned int ADC\_temp;
17. char outs[20];
18. *uint8\_t* OVF\_COUNT = 0; //initialize the overflow count for interrupt
19. *uint8\_t* OVF\_LIMIT = 250; //set the limit the count can reach to set 1 sec delay
20. int main(void)
21. {
22. USART\_init(); //initialize USART
23. ADC\_init (); //initialize ADC
24. USART\_putstring("Connected\r\n");

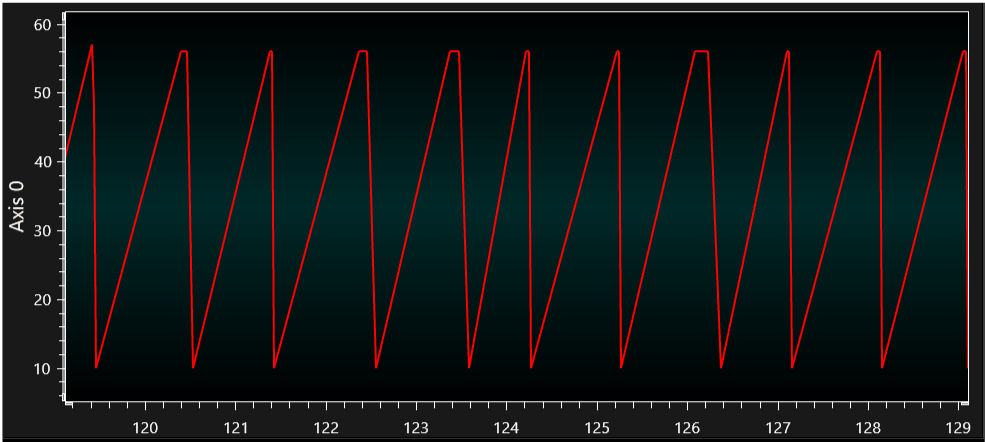
27. TCCR0A = 0x00; //normal operation
28. TCCR0B |= (1 << CS02); //set prescalar to 256
29. TCNT0 = 16; //TOP = 256-240 = 16
30. TIMSK0 |= (1 << TOIE0);
31. sei(); //enable interrupt
33. while (1)
34. { //sit here waiting for interrupt
35. }
36. }
37. void USART\_init(void)
38. {
39. UBRR0H = (*uint8\_t*)(BAUD\_PRESCALLER >> 8);
40. UBRR0L = (*uint8\_t*)(BAUD\_PRESCALLER);
41. UCSR0B = (1 << RXEN0) | (1 << TXEN0);
42. UCSR0C = (3 << UCSZ00);
43. }
44. unsigned char USART\_receive(void)
45. {
46. while(!(UCSR0A & (1 << RXC0)));
47. return UDR0;
48. }
49. void USART\_transmit(unsigned char data)
50. {
51. while(!(UCSR0A & (1 << UDRE0)));
52. UDR0 = data;
53. }
54. void USART\_putstring(char\* StringPtr)
55. {
56. while(\*StringPtr != 0x00)
57. {
58. USART\_transmit(\*StringPtr);
59. StringPtr++;
60. }
61. }
62. ISR (TIMER0\_OVF\_vect)
63. {
64. OVF\_COUNT++; //increment the overflow counter
65. if (OVF\_COUNT == OVF\_LIMIT) //check to see if the limit was reached
66. {
67. ADCSRA|=(1<<ADSC); //start conversion
68. while((ADCSRA&(1<<ADIF))==0);//wait for conversion to finish
70. ADCSRA |= (1<<ADIF);
71. int a = ADCL;
72. a = a | (ADCH<<8);
73. a = (a/1024.0) \* 5000/10;
74. a = (a\*2)+32; //equation to convert celsius to farenheit. can't use used 2 instead of 9/5
75. a = a % 100;
76. USART\_transmit((a/10)+'0');
77. a = a % 10;
78. USART\_transmit((a)+'0');
79. USART\_transmit('\n');
80. OVF\_COUNT = 0; //reset overflow counter
81. }
82. TCNT0 = 16; //reset TOP
83. }
84. void ADC\_init (void)
85. {
86. ADMUX = (0<<REFS1)| // reference selection bits
87. (1<<REFS0)| // AVcc - external cap at AREF
88. (0<<ADLAR)| // ADC Left Adjust Result
89. (1<<MUX2)| // Analog Channel Selection Bits
90. (0<<MUX1)| // ADC5 (PC5)
91. (0<<MUX0);
92. ADCSRA = (1<<ADEN)| // ADC Enable
93. (0<<ADSC)| // ADC Start Conversion
94. (0<<ADATE)| // ADC Auto Trigger Enable
95. (0<<ADIF)| // ADC Interrupt Flag
96. (0<<ADIE)| // ADC Interrupt Enable
97. (1<<ADPS2)| // ADC Prescaler Select Bits
98. (0<<ADPS1)|
99. (1<<ADPS0);
101. }
102. **SCHEMATICS**

Use fritzing.org

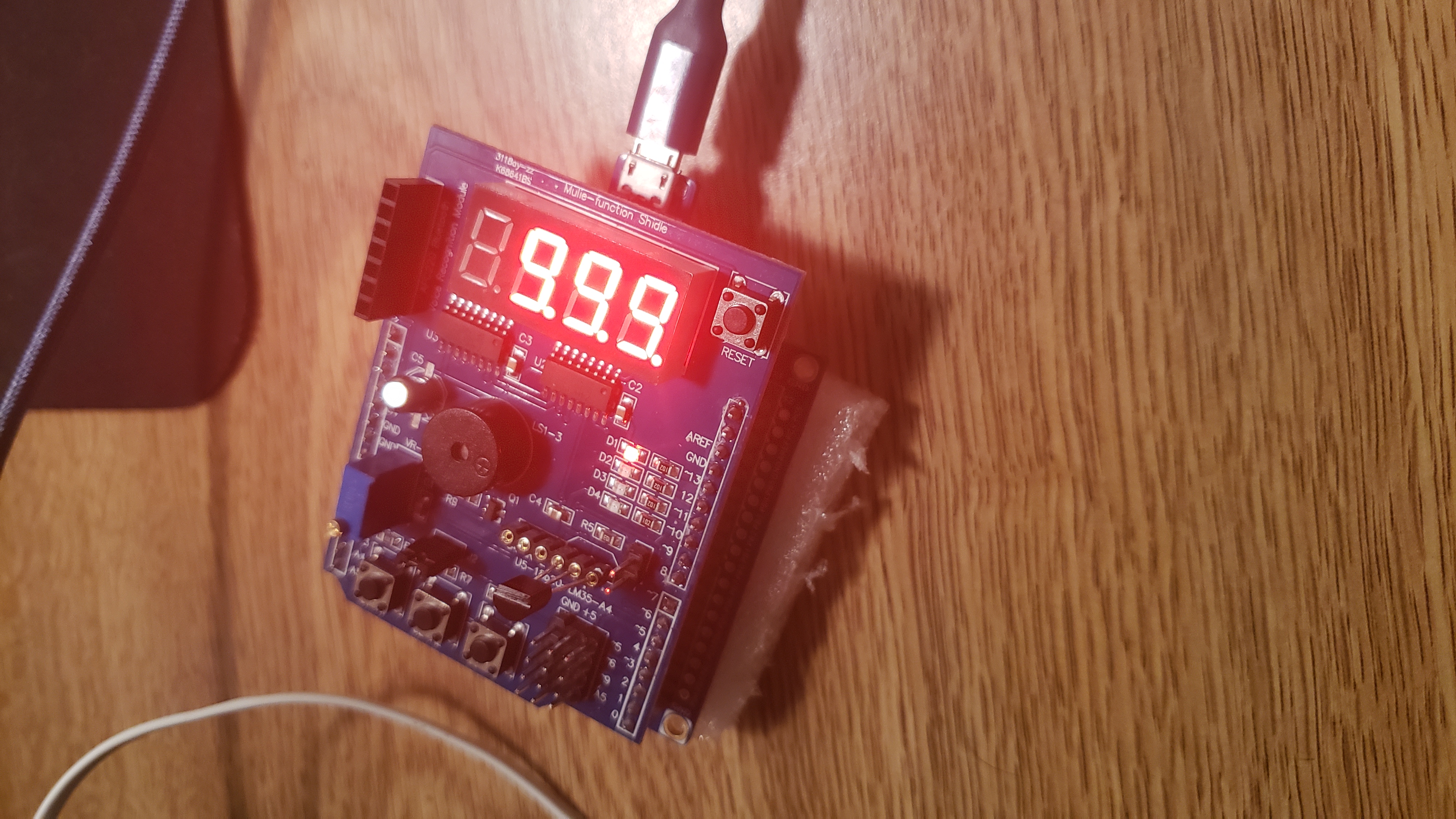
1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

I started to touch it so it got hotter





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



1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=Mfd6kXiqiSo>

1. **GITHUB LINK OF THIS DA**

<https://github.com/Jonesc30/Submission/tree/master/DesignAssignments>

**Student Academic Misconduct Policy**

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

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

Cody Jones