CPE301 – FALL 2019

Midterm 1

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**

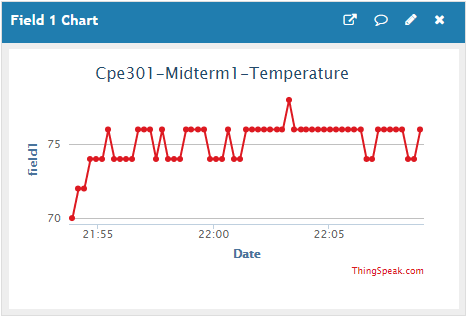
Atmega328PB, FTDI, ESP01, LM35

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 115200
10. #define BAUD\_PRESCALLER (((*F\_CPU* / (BAUDRATE \* 8UL))) - 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. *uint8\_t* OVF\_COUNT = 0; //initialize the overflow count for interrupt
17. *uint8\_t* OVF\_LIMIT = 250; //set the limit the count can reach to set 1 sec delay
18. char outs[20];
19. int main(void)
20. {
21. USART\_init(); //initialize USART
22. ADC\_init (); //initialize ADC
24. USART\_putstring("AT\r\n");
25. *\_delay\_ms*(500);
26. USART\_putstring("AT+CWMODE=3\r\n");
27. *\_delay\_ms*(500);
29. USART\_putstring("AT+CWJAP=\"WIFI\_NAME\_HERE\",\"WIFI\_PASSWORD\_HERE\""); // Log in WiFi
31. *\_delay\_ms*(500);
32. USART\_putstring("AT+CIPMUX=0\r\n");
33. *\_delay\_ms*(500);
35. TCCR0A = 0x00; //normal operation
36. TCCR0B |= (1 << CS02); //set prescalar to 256
37. TCNT0 = 16; //TOP = 256-240 = 16
38. TIMSK0 |= (1 << TOIE0);
39. sei(); //enable interrupt
41. while (1)
42. { //sit here waiting for interrupt
43. }
44. }
45. void USART\_init(void)
46. {
47. UBRR0H = (*uint8\_t*)(BAUD\_PRESCALLER >> 8);
48. UBRR0L = (*uint8\_t*)(BAUD\_PRESCALLER);
49. UCSR0A = (1 << U2X0); // double mode for more accuracy
50. UCSR0B = (1 << RXEN0) | (1 << TXEN0);
51. UCSR0C = (1 << UCSZ00) | (1 << UCSZ10);
52. }
53. unsigned char USART\_receive(void)
54. {
55. while(!(UCSR0A & (1 << RXC0)));
56. return UDR0;
57. }
58. void USART\_transmit(unsigned char data) { // Function to transmit ASCII value into UDR0
59. while (!(UCSR0A & (1 << UDRE0))); // Keep Checking until UDRE0 data register 'High' to break loop
60. UDR0 = data; // Store unsigned char serial data into UDR0
61. }
62. void USART\_putstring(char\* StringPtr)
63. {
64. while(\*StringPtr != 0x00)
65. {
66. USART\_transmit(\*StringPtr);
67. StringPtr++;
68. }
69. }
70. void ADC\_init (void)
71. {
72. ADMUX = (0<<REFS1)| // reference selection bits
73. (1<<REFS0)| // AVcc - external cap at AREF
74. (0<<ADLAR)| // ADC Left Adjust Result
75. (1<<MUX2)| // Analog Channel Selection Bits
76. (0<<MUX1)| // ADC4 (PC4)
77. (0<<MUX0);
78. ADCSRA = (1<<ADEN)| // ADC Enable
79. (0<<ADSC)| // ADC Start Conversion
80. (0<<ADATE)| // ADC Auto Trigger Enable
81. (0<<ADIF)| // ADC Interrupt Flag
82. (0<<ADIE)| // ADC Interrupt Enable
83. (1<<ADPS2)| // ADC Prescaler Select Bits
84. (0<<ADPS1)|
85. (1<<ADPS0);
87. }
88. ISR (TIMER0\_OVF\_vect)
89. {
90. OVF\_COUNT++; //increment the overflow counter
91. if (OVF\_COUNT == OVF\_LIMIT) //check to see if the limit was reached
92. {
93. ADCSRA|=(1<<ADSC); //start conversion
94. while((ADCSRA&(1<<ADIF))==0);//wait for conversion to finish
96. ADCSRA |= (1<<ADIF);
97. int a = ADCL; //a is temperature
98. a = a | (ADCH<<8);
99. a = (a/1024.0) \* 5000/10;
100. a = (a\*(2))+32; //equation to convert celsius to farenheit. use 2 instead of 9/5
102. //snprintf(outs,sizeof(outs),"%3f\r\n", a);// Stores integer 'adc\_temp' into the string 'outs'

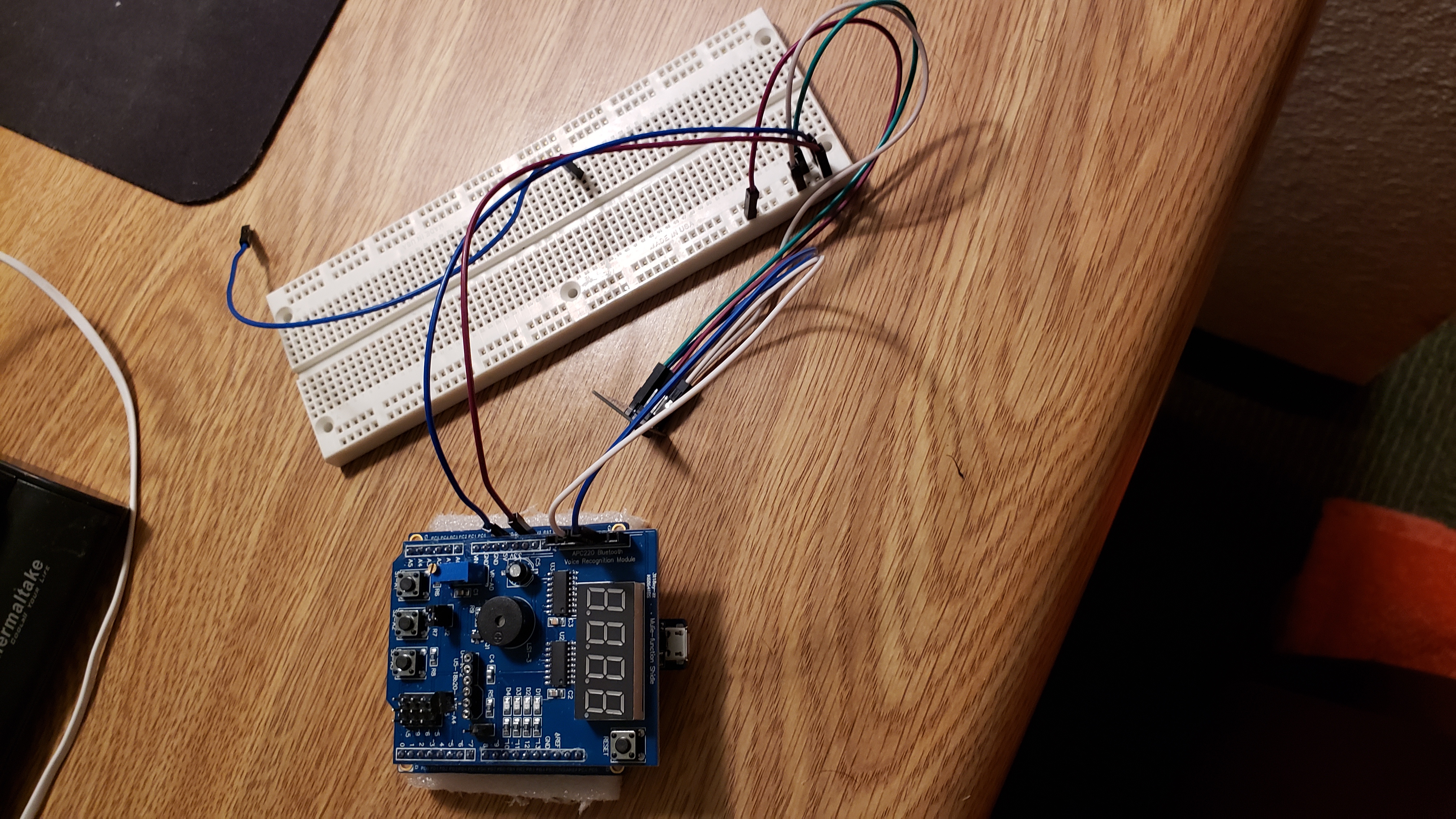
105. USART\_putstring("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n"); // Connect API KEy
106. *\_delay\_ms*(500);
107. USART\_putstring("AT+CIPSEND=51\r\n"); // Send Serial Data
108. *\_delay\_ms*(500);
109. USART\_putstring("GET /update?key=8031EGDV0KJXD8IU&field1=");// Send Value
110. *\_delay\_ms*(500);
111. a = a % 100;
112. USART\_transmit((a/10)+'0');
113. a = a % 10;
114. USART\_transmit((a)+'0');
115. *\_delay\_ms*(500);
116. USART\_putstring(" ");
117. *\_delay\_ms*(500);
118. USART\_putstring("\r\n");
119. USART\_putstring("\r\n");
121. OVF\_COUNT = 0; //reset overflow counter
122. }
123. TCNT0 = 16; //reset TOP
124. }
125. **SCHEMATICS**

Use fritzing.org

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



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



1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=XR4jNkcFTyQ>

1. **GITHUB LINK OF THIS DA**

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

**Student Academic Misconduct Policy**

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

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

Cody Jones