CPE301 – SPRING 2019

Design Assignment X

Student Name: Jeb Marinas

Student #:503376451

Student Email: marinj4@unlv.nevada.edu

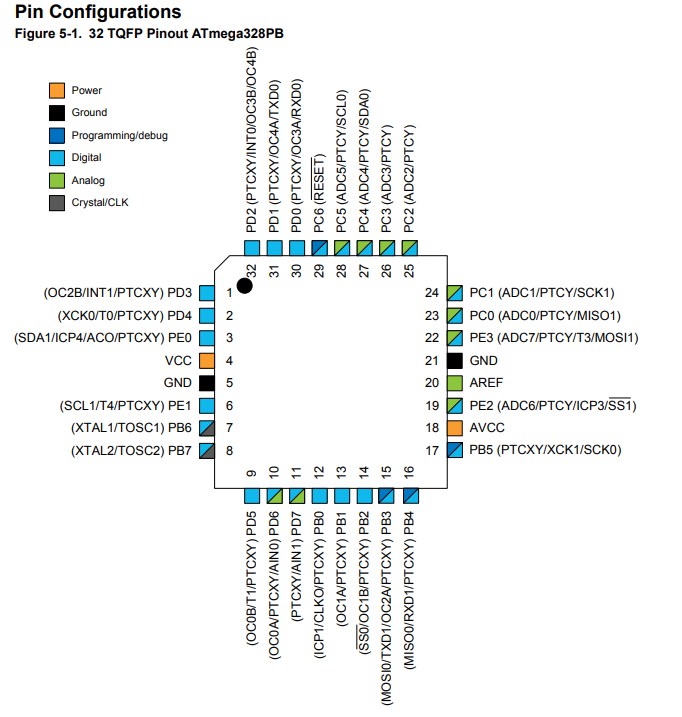
Primary Github address:github.com/jebmarinas

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**



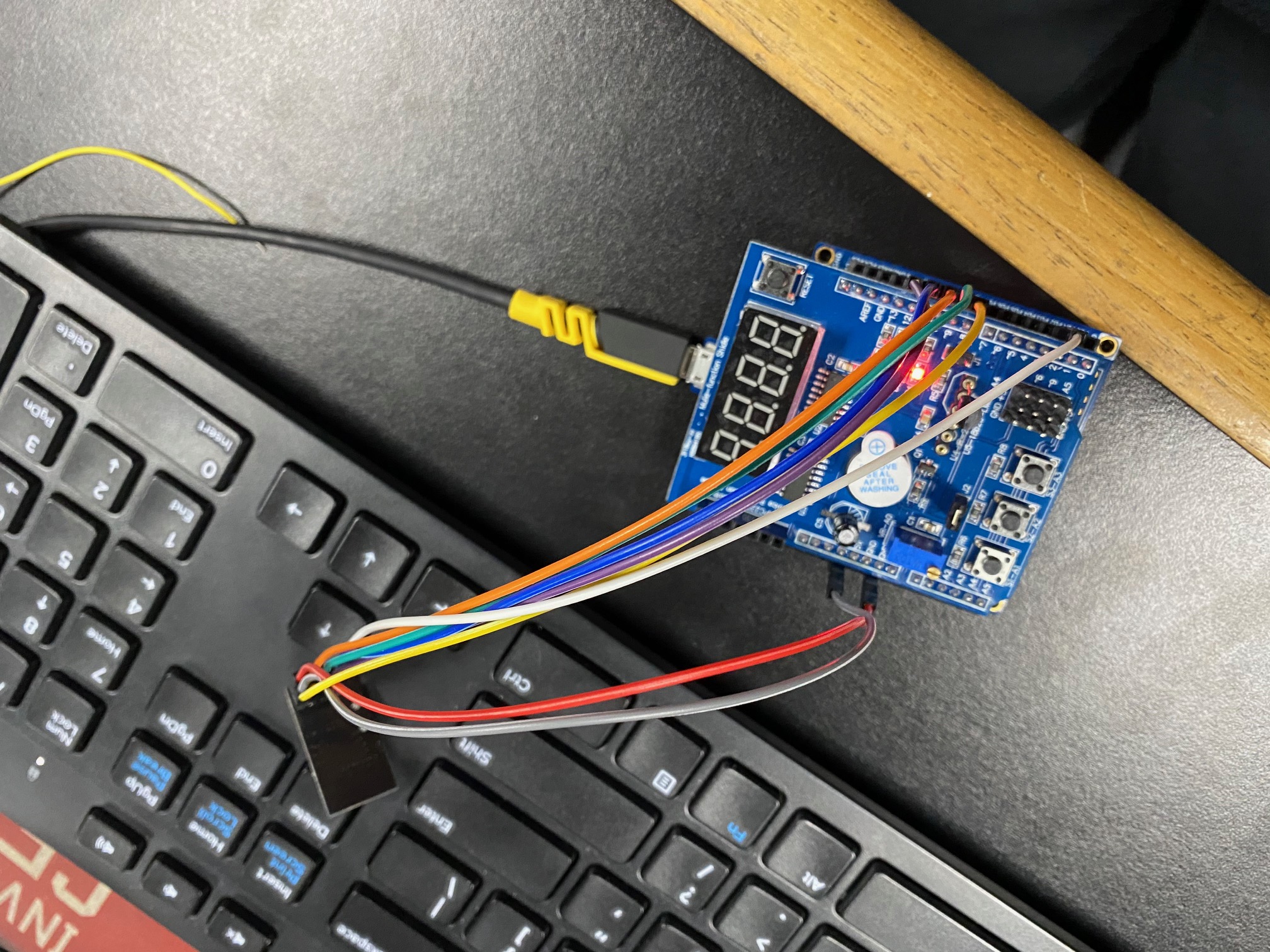
Atmega 328Pb, Multi-Function shield, Lm35 Temperature sensor, NRF24L01

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**
2. /\*
3. \* Assignment5A.c
4. \*
5. \* Created: 12/11/2019 2:43:15 PM
6. \* Author : marinj4
7. \*/
8. #ifndef *F\_CPU*
9. #define *F\_CPU* 16000000UL
10. #endif
11. #ifndef BAUD
12. #define BAUD 9600
13. #endif
14. #include <avr/io.h>
15. #include <util/delay.h>
16. #include <avr/interrupt.h>
17. #include <stdbool.h>
18. #include <stdio.h>
19. #include <string.h>
20. #include <util/setbaud.h>
21. #include "inc\STDIO\_UART.h"
22. #include "inc\nrf24l01.h"
23. #include "inc\nrf24l01-mnemonics.h"
24. #include "inc\spi.h"
25. void print\_config(void);
26. // Used in IRQ ISR
27. volatile bool message = false;
28. volatile bool condition = false;
29. volatile float adc\_temp = 0;
30. *uint8\_t* rx\_address[5] = {0x72, 0x72, 0x72, 0x72, 0x72}; // This will Read address
31. *uint8\_t* tx\_address[5] = {0x44, 0x44, 0x44, 0x44, 0x44}; // This will Write address
32. void USART\_init(void) { // will initialized the UART
33. UBRR0H = *UBRRH\_VALUE*; // Baud register will be set to high
34. UBRR0L = *UBRRL\_VALUE*; // Baud register will be set to low
35. UCSR0C = \_BV(UCSZ01) | \_BV(UCSZ00); //This is the 8 bit data
36. UCSR0B = \_BV(RXEN0) | \_BV(TXEN0); //this will enable the rx and tx
37. }
38. void USART\_tx\_string(char\*data) { // will send the string
39. while((\*data!='\0')) {
40. while(!(UCSR0A & (1 << UDRE0))); // buffer
41. UDR0 = \*data; // loads the buffer
42. data++; // will increment data
43. }
44. }
45. void adc\_init(void) {
46. ADMUX = (1<<REFS0) | // 5 Volts
47. (0<<REFS1) |
48. (1<<MUX2) |
49. (0<<ADLAR);
50. ADCSRA = (1<<ADEN) | enables the adc
51. (1<<ADPS2) |
52. (1<<ADPS1) |
53. (1<<ADPS0);
54. }
55. void read\_adc(void) {
56. unsigned char i = 4; // Variable to loop
57. adc\_temp = 0; // Set starting value to zero
58. while(i--) {
59. ADCSRA |= (1<<ADSC); // Start conversion
60. while(ADCSRA & (1<<ADSC)); // Wait for conversion
61. adc\_temp += ADC; // Read value from ADC
62. *\_delay\_ms*(50);
63. }
64. adc\_temp = adc\_temp / 4; //
65. }
66. void print\_config(void){
67. *uint8\_t* data;
68. *printf*("Startup successful\n\n nRF24L01+ configured as:\n");
69. *printf*("-------------------------------------------\n");
70. nrf24\_read(CONFIG,&data,1);
71. *printf*("CONFIG 0x%02X\n",data);
72. nrf24\_read(EN\_AA,&data,1);
73. *printf*("EN\_AA 0x%02X\n",data);
74. nrf24\_read(EN\_RXADDR,&data,1);
75. *printf*("EN\_RXADDR 0x%02X\n",data);
76. nrf24\_read(SETUP\_RETR,&data,1);
77. *printf*("SETUP\_RETR 0x%02X\n",data);
78. nrf24\_read(RF\_CH,&data,1);
79. *printf*("RF\_CH 0x%02X\n",data);
80. nrf24\_read(RF\_SETUP,&data,1);
81. *printf*("RF\_SETUP 0x%02X\n",data);
82. nrf24\_read(STATUS,&data,1);
83. *printf*("STATUS 0x%02X\n",data);
84. nrf24\_read(FEATURE,&data,1);
85. *printf*("FEATURE 0x%02X\n",data);
86. *printf*("-------------------------------------------\n\n");
87. }
88. ISR(INT0\_vect) {
89. message = true;
90. }
91. int main(void){
93. char tx\_message[64];
94. char \*ptr = text;

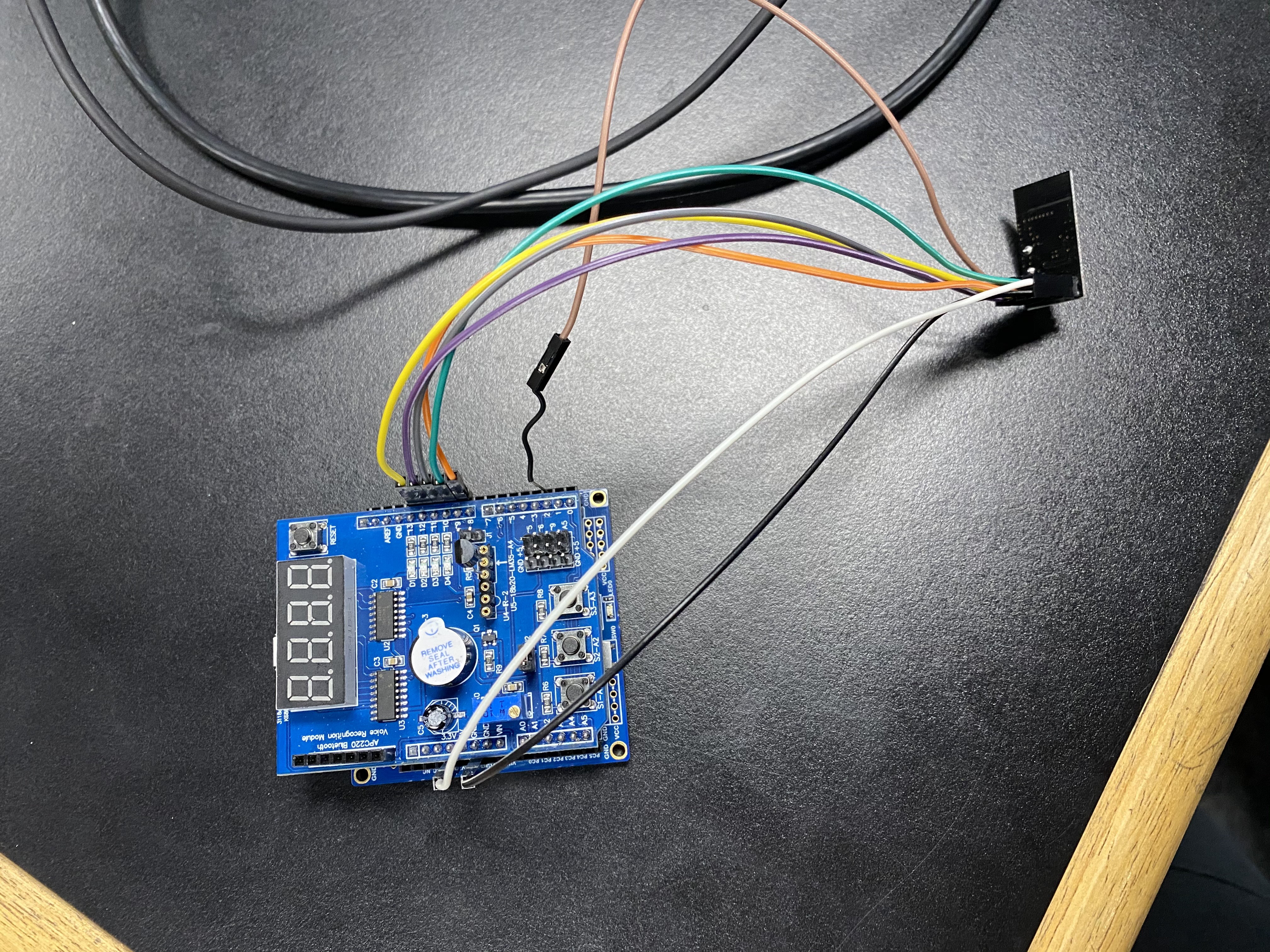
97. USART\_init();

100. nrf24\_init();
101. print\_config();
102. adc\_init();

105. *printf*("start listening\n");
106. nrf24\_start\_listening();
107. *printf*("Done listening\n");
109. while (1){
110. read\_adc();
111. ptr = text;
112. *snprintf*(text, sizeof(text), "%f\r\n", adc\_temp); //
113. nrf24\_send\_message(ptr);
114. *\_delay\_ms*(100);
115. if (message){
116. *printf*("inside if condition\n");
117. // Message received, print it
118. message = false;
119. *printf*("Message Received: %s\n",nrf24\_read\_message());
120. // Send message as response
121. *\_delay\_ms*(500);
122. condition = nrf24\_send\_message(tx\_message);
123. if (status == true)
124. *printf*("Message Transmitted\n");
125. }
126. }
127. }
128. **SCHEMATICS**
129. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**
130. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**
131. **First Board:**

****

**Second Board:**



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/mxIQu5hmI_4>

1. **GITHUB LINK OF THIS DA**

<https://github.com/jebmarinas/Assignment_projects/tree/master/Assignment5A>

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT