CPE301 – SPRING 2019

MIDTERM 1

Student Name: David Flores

Student #: 2000881655

Student Email: flored6@unlv.nevada.edu

Primary Github address: https://github.com/David-Floress/submission\_da.git

Directory: https://github.com/David-Floress/submission\_da

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/Midterm, 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**

Atmega328p Xplained mini

FTDI Basic

LM35 (Celsius)

ESP01

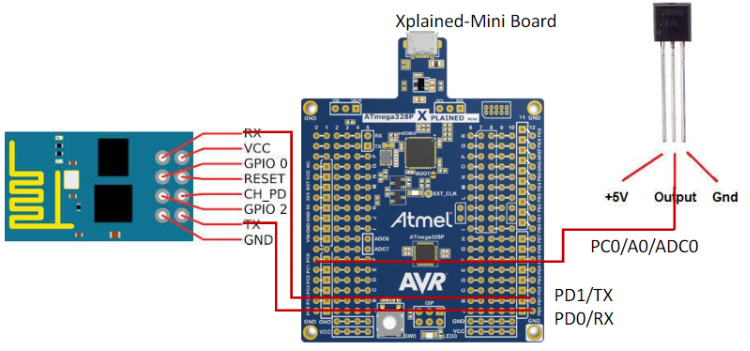
Block diagram with pins used in the Atmega328P

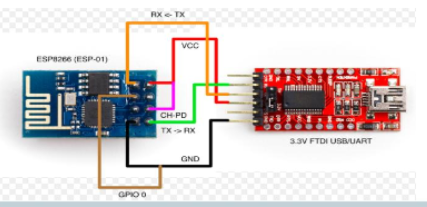
Wires (no female headers)

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A**
2. /\*
3. \* Created: 4/5/2019 3:02:41 PM
4. \* Author : David Flores
5. \*/
6. #include <avr/io.h>
7. #include <avr/interrupt.h>
8. #include <util/delay.h>
9. #include <stdlib.h>
10. #include <stdint.h>
11. #define *F\_CPU* 16000000UL
12. #define BAUD 115200
13. #define FOSC 16000000
14. #define THEUBRR FOSC/8/BAUD -1
15. volatile *uint8\_t* ADCvalue;
16. volatile unsigned char ADCtemp[5];
17. volatile unsigned char CWMODE[] = "AT+CWMODE=3\r\n"; //Set to three for the wifi
18. volatile unsigned char WIFI[] = "AT+CWJAP=\"EC4234\", \"G2C26B2B89582\"\r\n"; //Connects to Internet
19. volatile unsigned char ENABLE[] = "AT+CIPMUX=0\r\n"; //Connect to WiFi
20. volatile unsigned char CIPSTART[] = "AT+CIPSTART=\"TCP\",\"184.106.153.149\",80\r\n"; //TCP strip remote IP, 80 is the port
21. volatile unsigned char CIPSEND[] = "AT+CIPSEND=45\r\n"; //Length of data
22. volatile unsigned char SEND\_DATA[] = "GET /update?key=AHDBMTLB6ZCBMG40&field1="; //Writes API key
23. volatile unsigned char PAUSE[] = "\r\n\r\n";
24. void send\_AT (volatile unsigned char c[]); //sends the arrays to this function
25. int main( void )
26. {
27. /\* ADC \*/
28. ADMUX = 0; // uses ADC0
29. ADMUX |= (1 << REFS0); // use AVcc as the reference
30. ADMUX |= (1 << ADLAR); // Right adjusts for 8 bit resolution
31. ADCSRA |= (1 << ADATE); // Set ADC Auto Trigger Enable THEUBRR
32. ADCSRB = 0; // 0 for free running ode
33. ADCSRA |= (1 << ADEN); // Enable ADC
34. ADCSRA |= (1 << ADIE); // Enables Interrupts
35. ADCSRA |= (1 << ADSC); // Start ADC conversion
36. ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0); // 128 prescaler for 16Mhz
37. /\* UART \*/
38. UBRR0H = ((THEUBRR) >> 8); //Shift right to store upper 8 bits
39. UBRR0L = THEUBRR; //Store lower bits
40. UCSR0A |= (1 << U2X0); //Double the USART transmission speed
41. UCSR0B |= (1 << TXEN0); // Enables USART transmitter
42. UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00); // Set frame: 8data, 1 stop, no parity bit
43. *\_delay\_ms*(1000);
44. send\_AT(CWMODE);
45. *\_delay\_ms*(1000);
46. send\_AT(WIFI);
47. *\_delay\_ms*(2000);
48. send\_AT(ENABLE);
49. sei();
50. while(1)
51. {
52. *\_delay\_ms*(1000);
53. send\_AT(CIPSTART); //Connect to thingspeak, port 80
54. *\_delay\_ms*(1000);
55. send\_AT(CIPSEND);
56. *\_delay\_ms*(1000);
57. send\_AT(SEND\_DATA);
58. send\_AT(ADCtemp); //Sends the temperature data
59. send\_AT(PAUSE); //A break in the data display
60. }
61. }
62. ISR(ADC\_vect)
63. {
64. volatile unsigned int j=0;
65. char temp[5];
66. ADCvalue = (ADCH << 1); //Shifts the value left one place
67. *itoa*(ADCvalue, temp, 10); //Converts integers to string
68. while (j<5) { //This transfers the temprature string from itoa() to ADCtemp
69. ADCtemp[j] = temp[j];
70. j++;
71. }
72. }
73. void send\_AT(volatile unsigned char c[]) {
74. volatile unsigned int i=0, j=0;
75. j = 0; //initialize the counter
76. while (c[j] != 0x00) { //while not at end of string
77. j++;
78. }
79. while (i<j) {
80. while(!(UCSR0A & (1 << UDRE0))); //If UDRE0 is 1, buffer is empty an can be written to
81. UDR0 = c[i];
82. i++;
83. }
84. }
85. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

**N/A**

1. **SCHEMATICS**

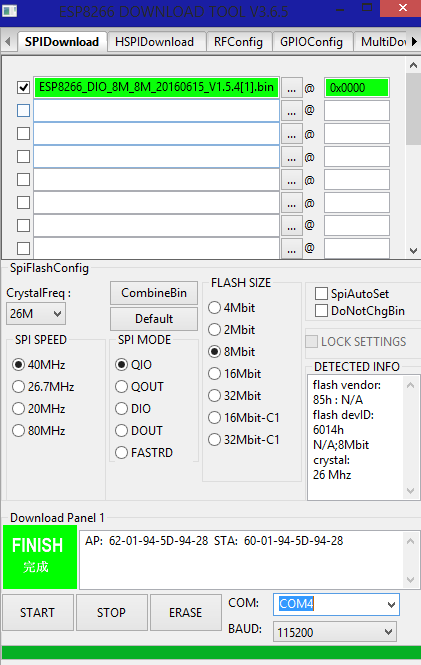




1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

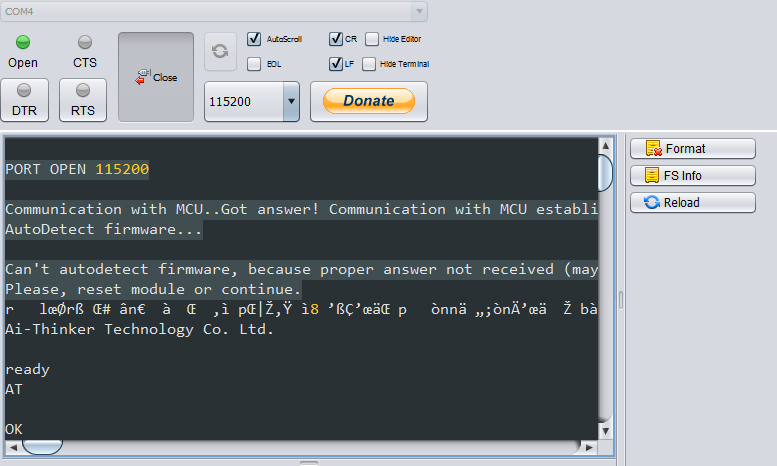
Part 1:

Firmware successfully installed onto ESP8226 using the FTDI Basic



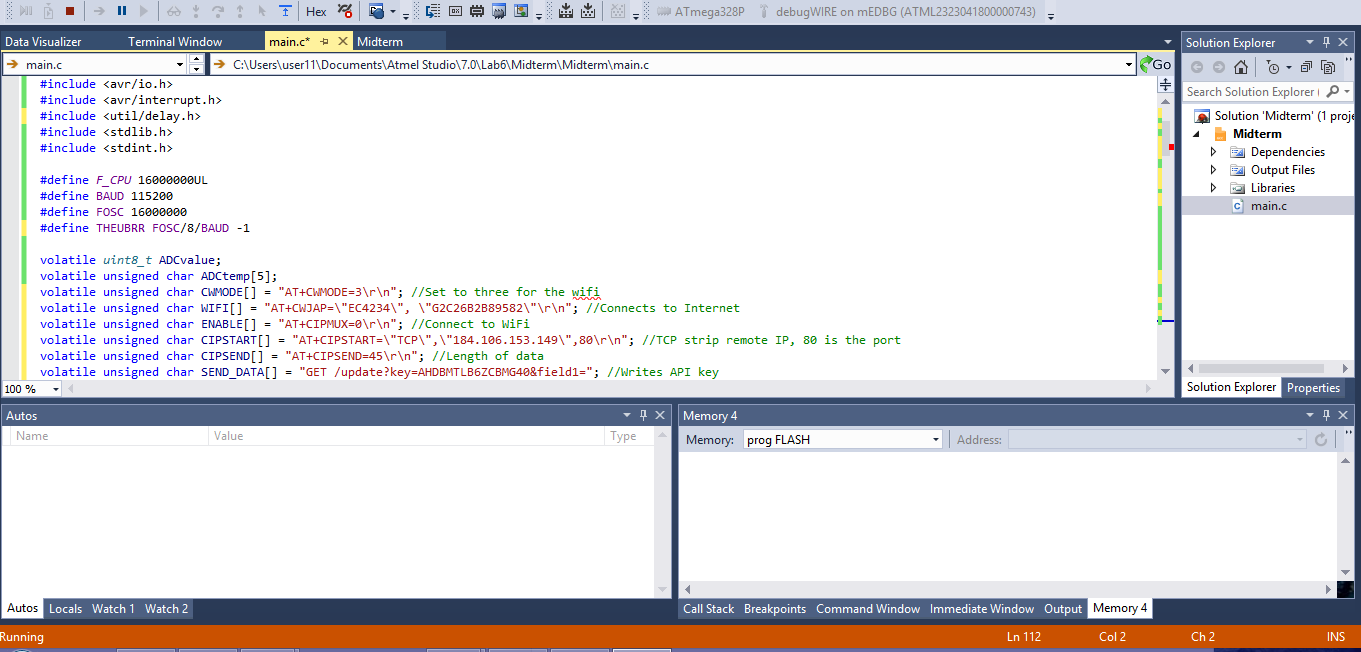
Part 2:

We check to see if the ESP8226 can connect to thingspeak we use ESPlorer to do so.



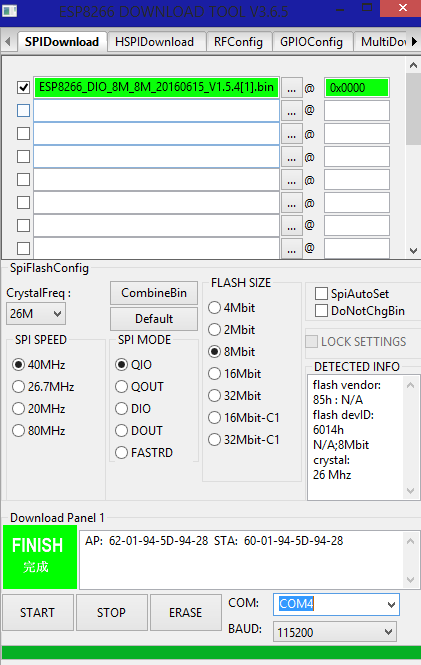
The board connects to the ESPlorer and then we can connect to the internet

Program Running:



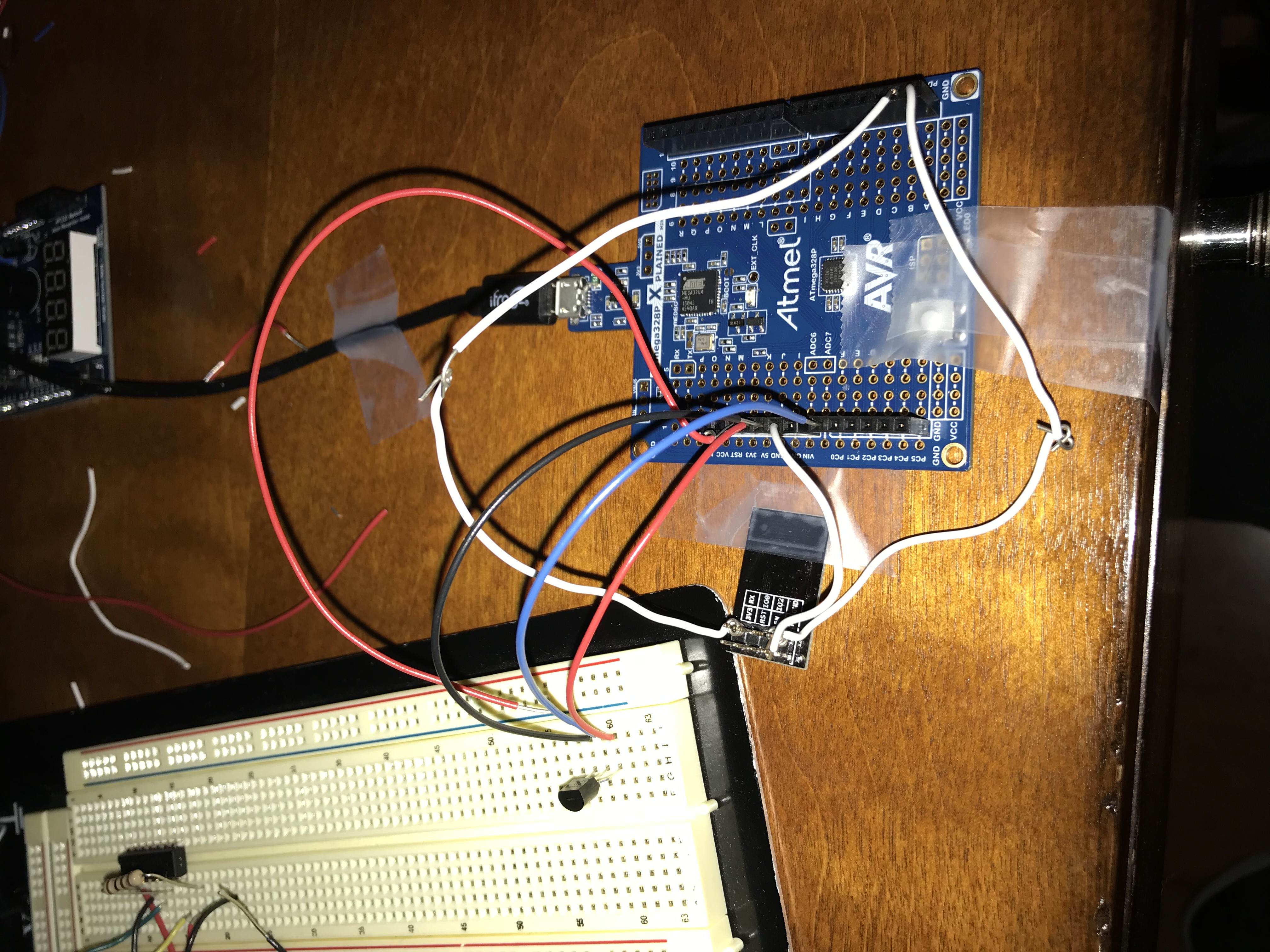
1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

I am not going to be including the setup of when I flashed. The only difference was that IO0 was grounded for this part. I did not have female to female wires, so I undid the setup and did the one shown in the next picture without taking the picture. It was a big pain doing it without female connections so I will just include the finished screen from the ESP8266 Download with the Addresses.



ATmega to ESP01 including sensor picture.

In the picture I left the 3.3V wire out during the process I was actually holding it down as seen in the video. That’s the only reason why its not connected.



1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=91Y0Box9ECA>

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”.

NAME OF THE STUDENT