CPE301 – SPRING 2019

MIDTERM 2

Student Name: Geovanni Portillo

Student #: 8000603824

Student Email: portig1@gmail.com

Primary Github address: <https://github.com/portig1/submissions_E>

Directory: portig1/submissions\_E/tree/master/Midterms/Midterm\_2

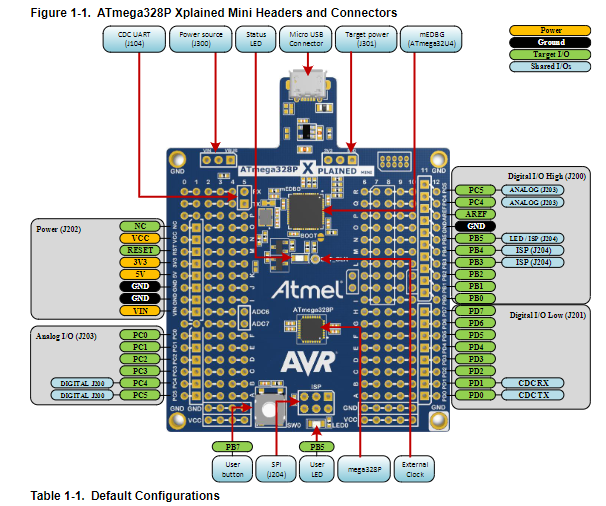
Submit the following for all Labs:

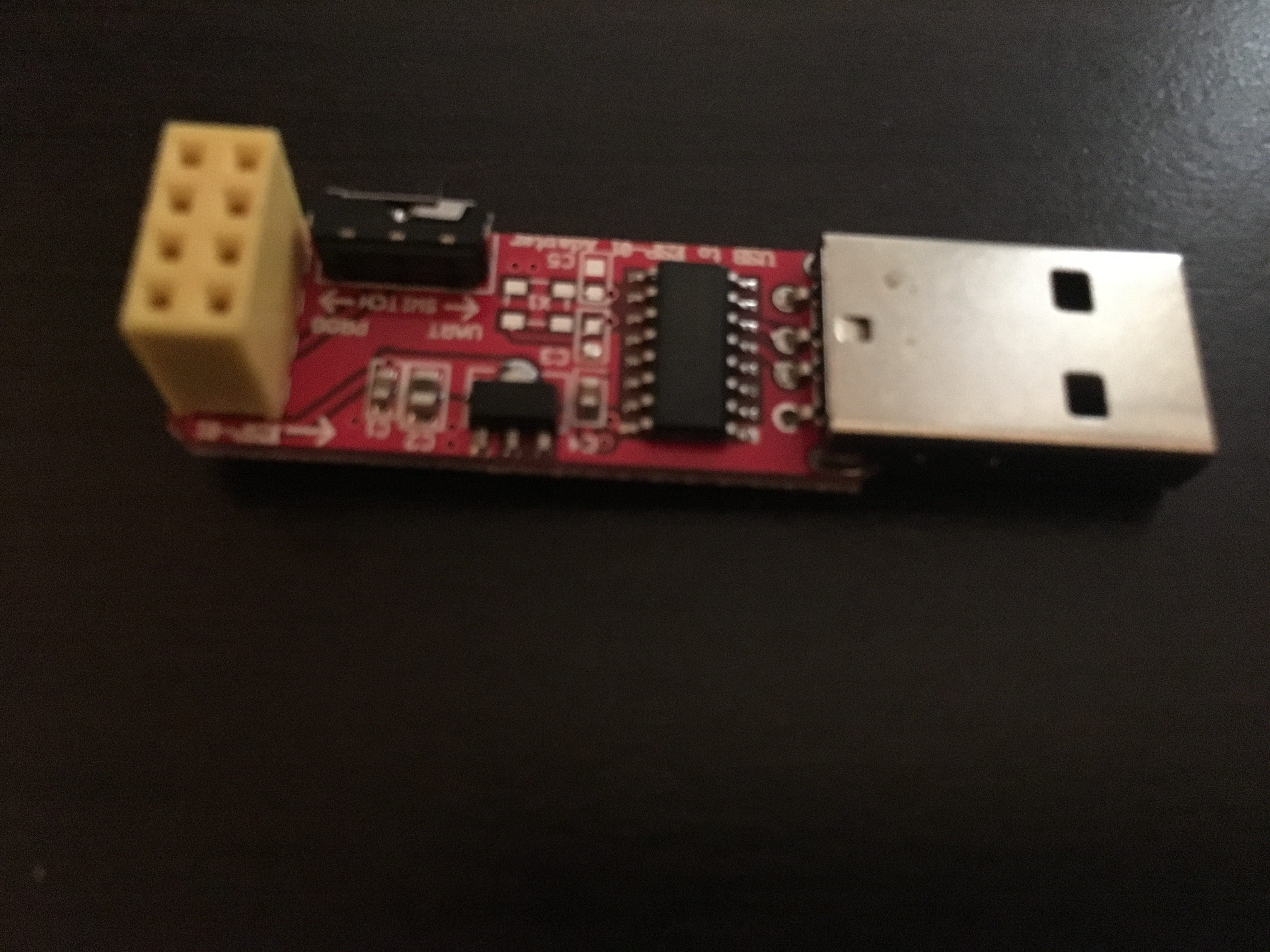
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder 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**

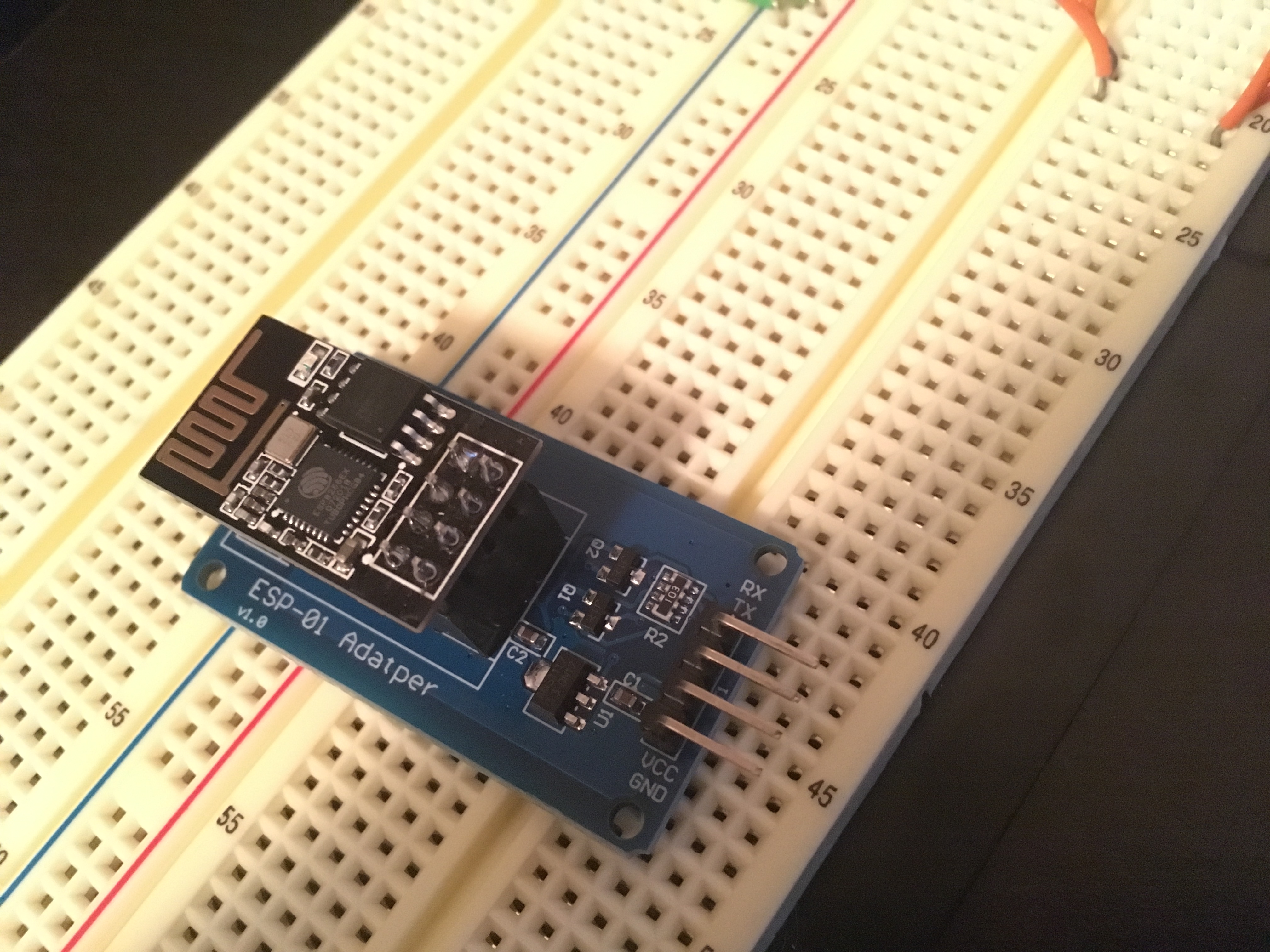
Atmel Studio 7

ATmega328PB Xplained mini





ESP-01 to USB Programmer/Adapter



ESP-01 Adapter with ESP8266 plugged in



APDS9960 with pin labels

1. **DEVELOPED CODE OF TASK 1/2/5/6**

#ifndef F\_CPU

#define F\_CPU 16000000UL

#endif

#include <avr/io.h>

#include <util/delay.h>

#include <math.h>

#include <stdlib.h> /\* Include standard library file \*/

#include <stdio.h> /\* Include standard library file \*/

#include "APDS9960\_def.h" /\* Include APDS9960 register define file \*/

#include "i2c\_master.h" /\* Include I2C Master header file \*/

#include "uart.h" /\* Include USART header file \*/

#define APDS9960\_WRITE 0x72

#define APDS9960\_READ 0x73

*uint16\_t* Clear\_data, Red\_data, Green\_data, Blue\_data;

void init\_uart(*uint16\_t* baudrate){

*uint16\_t* UBRR\_val = (F\_CPU/16)/(baudrate-1);

UBRR0H = UBRR\_val >> 8;

UBRR0L = UBRR\_val;

UCSR0B |= (1<<TXEN0) | (1<<RXEN0) | (1<<RXCIE0); // UART TX (Transmit - senden) einschalten

UCSR0C |= (1<<USBS0) | (3<<UCSZ00); //Modus Asynchron 8N1 (8 Datenbits, No Parity, 1 Stopbit)

}

void usart\_init (void)

{

//UBRR0H = (uint8\_t)(BAUD\_PRESCALLER >> 8);

//UBRR0L = (uint8\_t)(BAUD\_PRESCALLER);

UBRR0 = 8; //Manually setting as the formula give a float value that has too much error.

//UCSR0A = (1 << U2X0);

UCSR0B = (1 << RXEN0) | (1 << TXEN0);

UCSR0C = (1<< UCSZ01)|(1<<UCSZ00);

}

void uart\_putc(unsigned char c){

while(!(UCSR0A & (1<<UDRE0))); // wait until sending is possible

UDR0 = c; // output character saved in c

}

void uart\_puts(char \*s){

while(\*s){

uart\_putc(\*s);

s++;

}

}

void init\_APDS9960(void){

*\_delay\_ms*(150); /\* Power up time >100ms \*/

i2c\_start(APDS9960\_WRITE); //

i2c\_write(APDS9960\_ENABLE); //

i2c\_write(0x00); //Turn off all features

i2c\_stop();

i2c\_start(APDS9960\_WRITE);

i2c\_write(APDS9960\_ATIME);

i2c\_write(DEFAULT\_ATIME); // Set default integration time

i2c\_stop();

i2c\_start(APDS9960\_WRITE);

i2c\_write(APDS9960\_CONTROL); //Set default gain value

i2c\_write(DEFAULT\_AGAIN);

i2c\_stop();

i2c\_start(APDS9960\_WRITE);

i2c\_write(APDS9960\_ENABLE);

i2c\_write((1 << POWER) | (1 << AMBIENT\_LIGHT)); //

i2c\_stop();

}

void getreading(void){

i2c\_start(APDS9960\_WRITE);

i2c\_write(APDS9960\_CDATAL); // set pointer

i2c\_stop();

i2c\_start(APDS9960\_READ);

Clear\_data = (((int)i2c\_read\_ack()) | (int)i2c\_read\_ack() << 8);

Red\_data = (((int)i2c\_read\_ack()) | (int)i2c\_read\_ack() << 8);

Green\_data = (((int)i2c\_read\_ack()) | (int)i2c\_read\_ack() << 8);

Blue\_data = (((int)i2c\_read\_ack()) | (int)i2c\_read\_ack() << 8);

i2c\_stop();

}

int main(void){

char buffer[20];

//init\_uart(9600);

usart\_init(); //manually set for 115200 BAUD rate as formula has too much error

i2c\_init();

init\_APDS9960();

while(1){

getreading();

USART\_SendString("\n-----------------------\n");

*sprintf*(buffer,"Clear = %d, ",Clear\_data);

USART\_SendString(buffer);

*sprintf*(buffer,"Red = %d, ",Red\_data);

USART\_SendString(buffer);

*sprintf*(buffer,"Green = %d, ",Green\_data);

USART\_SendString(buffer);

*sprintf*(buffer,"Blue = %d\n",Blue\_data);

USART\_SendString(buffer);

USART\_SendString("\n-----------------------\n");

*\_delay\_ms*(1000);

char thingSpeakUpdate[150];

char setMUX[] = "AT+CIPMUX=0\r\n"; //Repeating the steps ESPlorer used

USART\_SendString(setMUX);

*\_delay\_ms*(1000);

char thingSpeakStart[] = "AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n" ;

USART\_SendString(thingSpeakStart);

*\_delay\_ms*(1000); //add in delay to allow for proper interactions

char thingSpeakSend[] = "AT+CIPSEND=119\r\n"; //saying we'll send more data than we actually will

USART\_SendString(thingSpeakSend);

*\_delay\_ms*(1000);

*snprintf*(thingSpeakUpdate, sizeof(thingSpeakUpdate), "GET https://api.thingspeak.com/update?api\_key=P7JD9OICCNIR59PL&field1=%d&field2=%d&field3=%d&field4=%d\r\n", Clear\_data, Red\_data, Green\_data, Blue\_data);

USART\_SendString(thingSpeakUpdate);

*\_delay\_ms*(1000);

char thingSpeakClose[] = "AT+CIPCLOSE\r\n";

USART\_SendString(thingSpeakClose);

*\_delay\_ms*(1000);

*\_delay\_ms*(1500); //delay of 15s

}

return 0;

}

1. **SCHEMATICS**

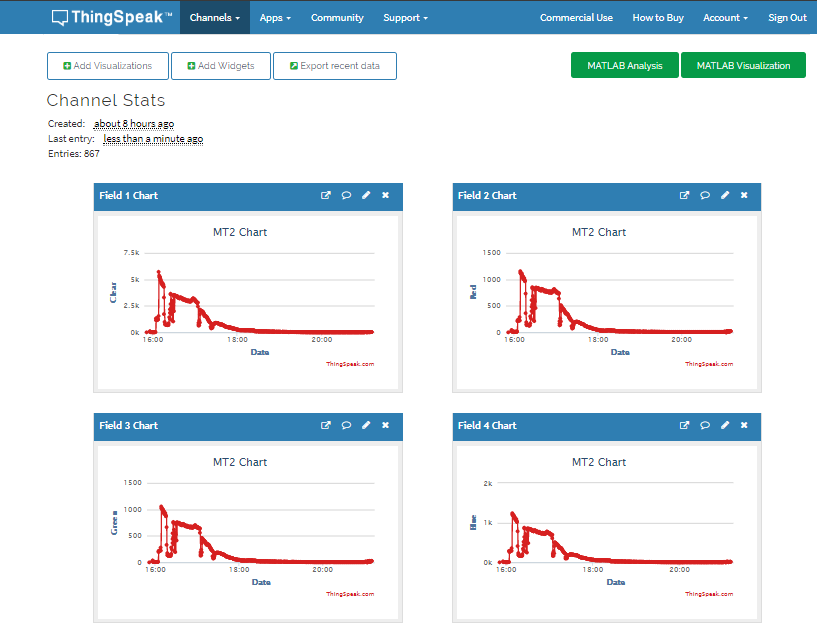


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

****

****

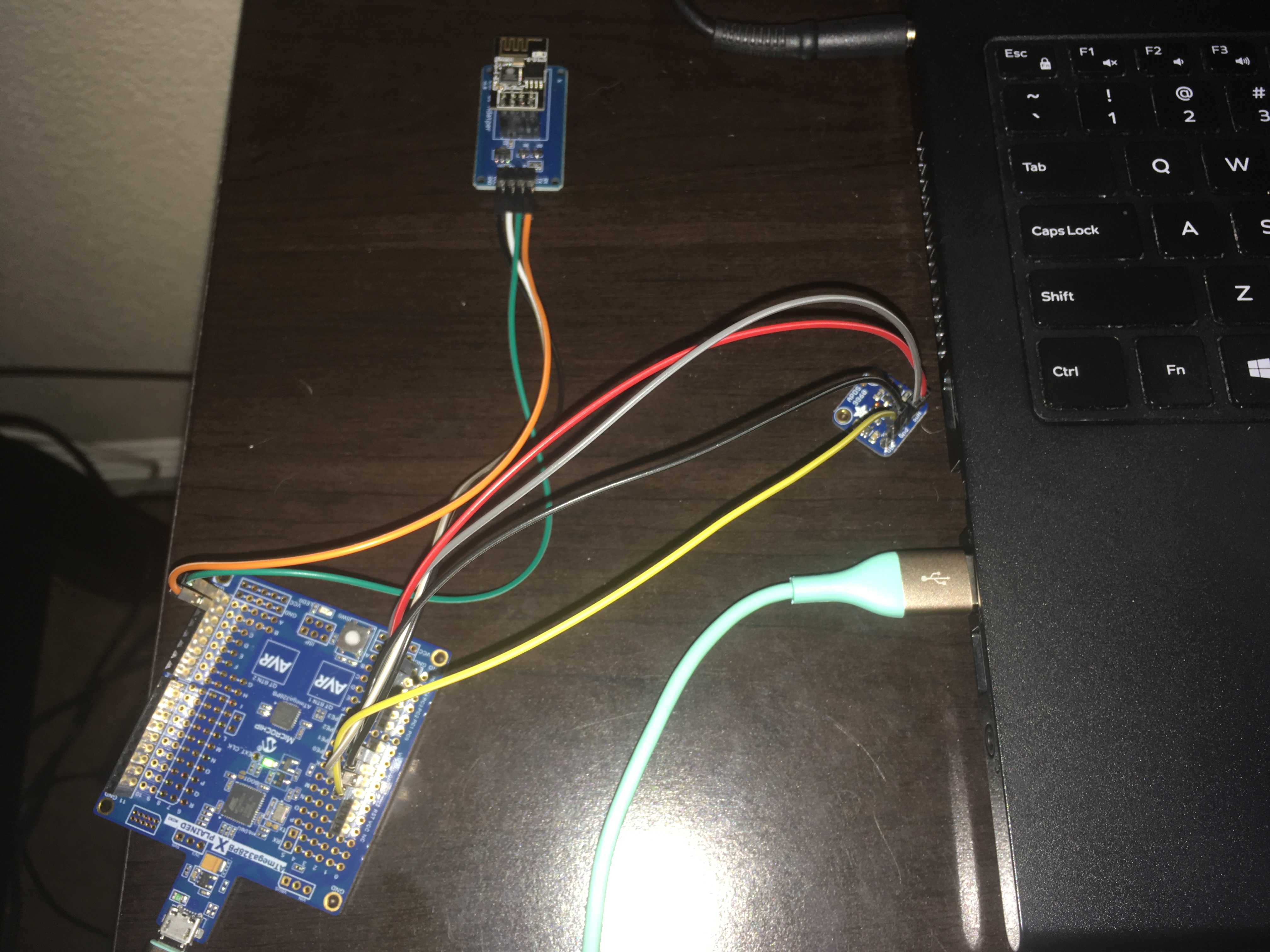
Output over serial at 115200 baudrate, displays RGBC values to user then sends commands to esp module (write key has since been changed).



ThingSpeak graphs light values from a window starting just before 4pm and ending some time in the night.

Can also be viewed at <https://thingspeak.com/channels/778393>

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



Board Setup, ESP8266 is plugged into ESP01-Adapter which is then connected to 5V and GND on Xplained mini while the TX line from the adapter is connected to RX (PD1) on the Xplained mini and then RX from the adapter to TX of the mini. The ASP9960 is connected to SCL (PC5) and SDA (PC4) on the mini as well as to 3.3V and GND.

1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/EYj7FFeU6o0>

1. **GITHUB LINK OF THIS DA**

[https://github.com/portig1/submissions\_E/tree/master/Midterms/Midterm\_](https://github.com/portig1/submissions_E/tree/master/Midterms/Midterm_1)2

**Student Academic Misconduct Policy**

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

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

Geovanni Portillo