CPE301 – SPRING 2019

MIDTERM II

Student Name: John Duriman
Student #: 5002373995
Student Email: duriman@unlv.nevada.edu
Primary Github address: https://github.com/johnduriman/pirahnaplant.git
Directory: pirahnaplant/Midterms/Midterm II

Submit the following for all Labs:

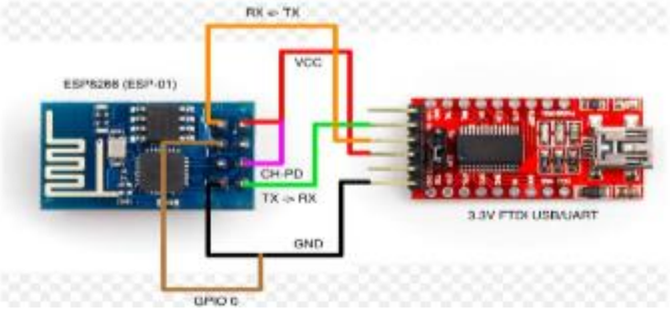
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder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

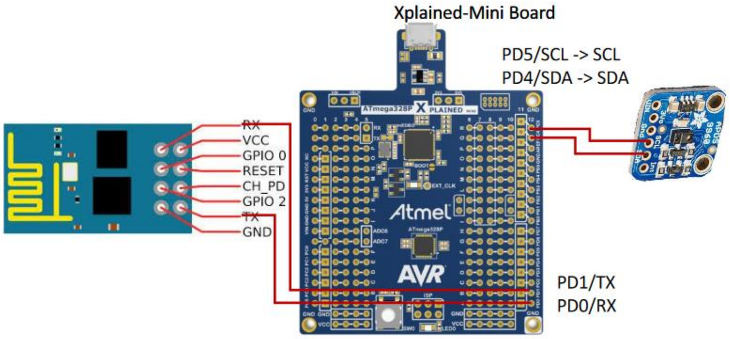
Parts used:

ESP8266 ESP01 WiFi module
5V Serial to USB Converter
Atmega 328P
Breadboard
Jumper wires
APDS 9960

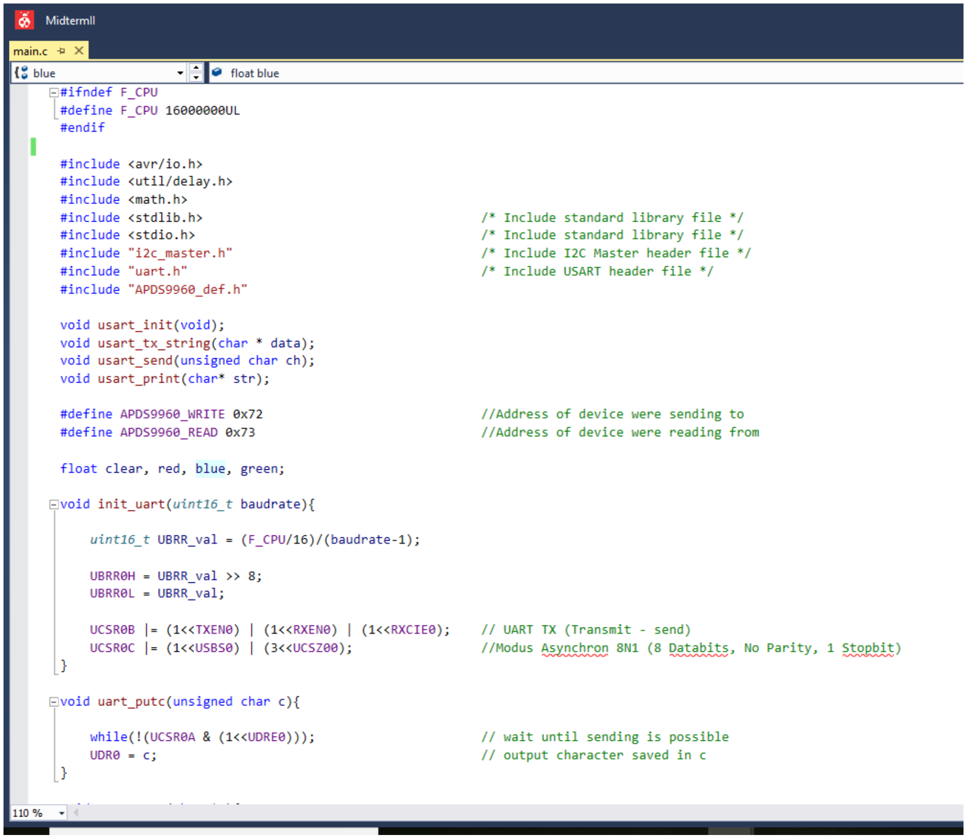
To program the ESP01, I wired it with the converter using this schematic



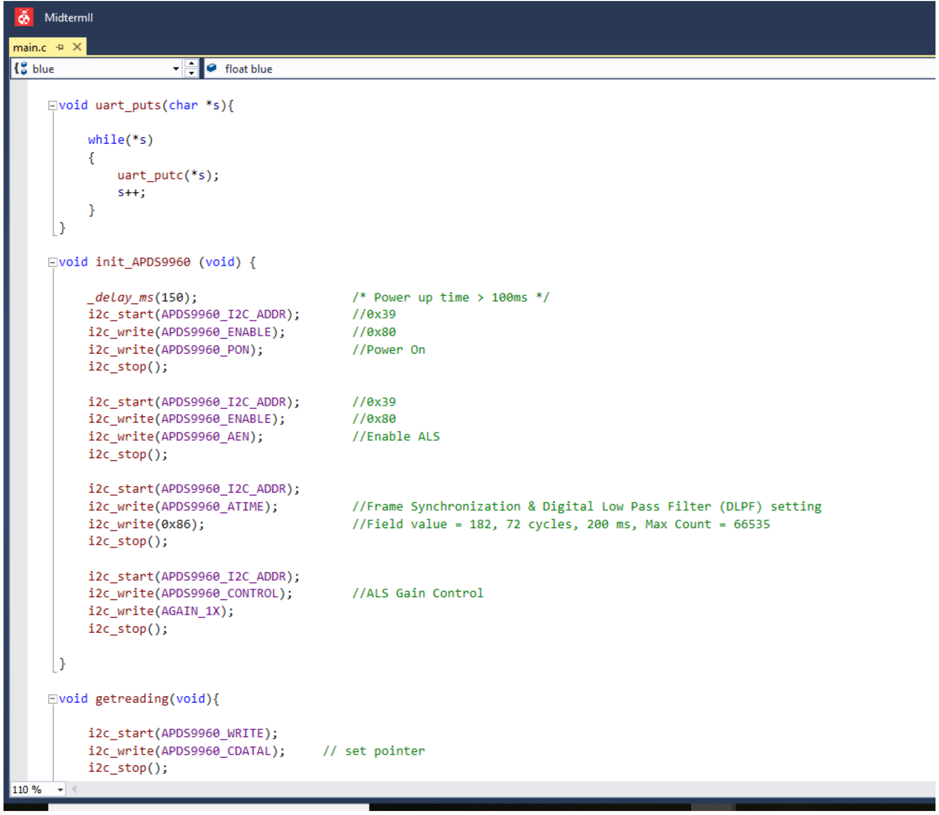
Once programmed, I hooked up the APDS 9960 to the ATMEGA328P and used PD5 for SCL and PD4 for SDA.



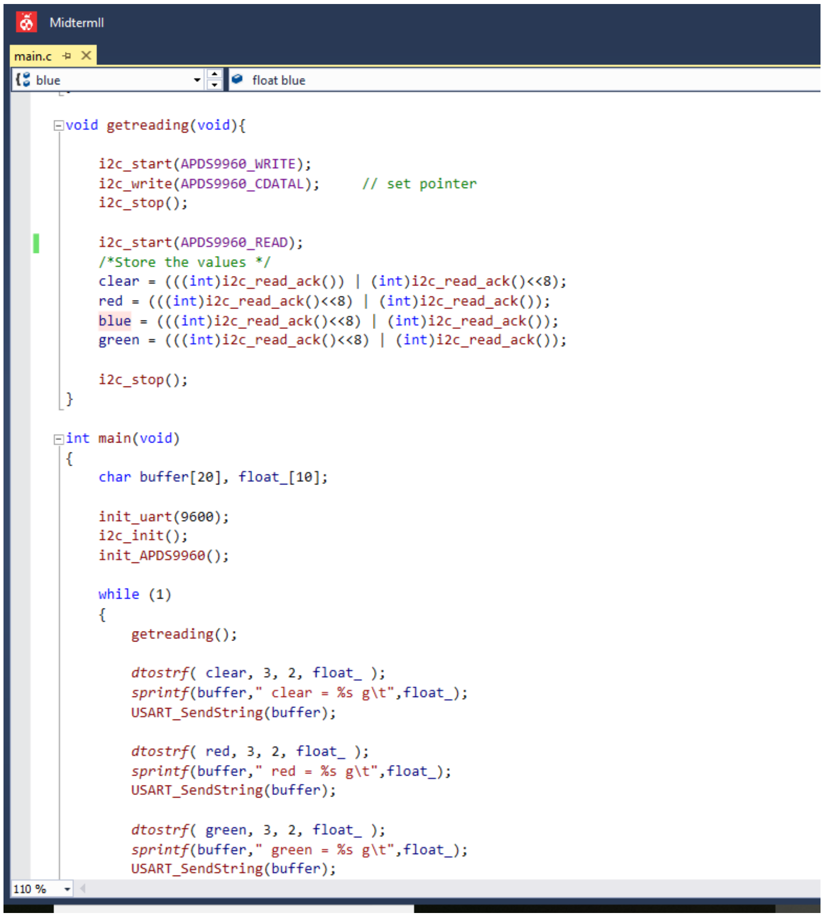
2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A



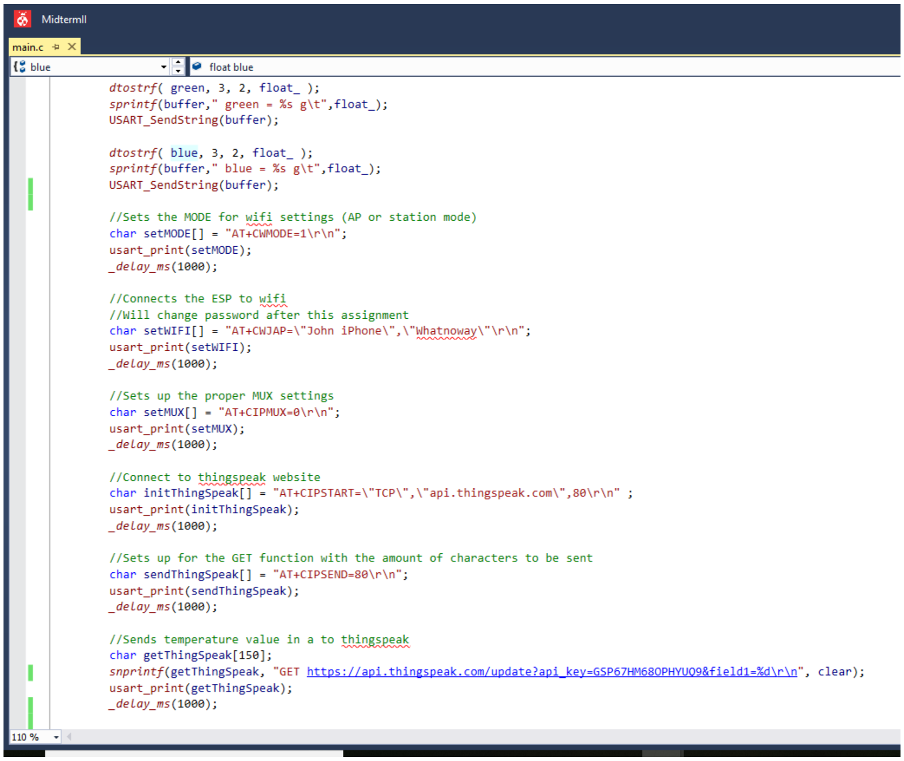
```
#ifndef F_CPU
#define F_CPU 16000000UL
#endif

#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 "i2c_master.h"                                 /* Include I2C Master header file */
#include "uart.h"                                       /* Include USART header file */
#include "APDS9960_def.h"

void usart_init(void);
void usart_tx_string(char * data);
void usart_send(unsigned char ch);
void usart_print(char* str);

#define APDS9960_WRITE 0x72                             //Address of device were sending to
#define APDS9960_READ 0x73                              //Address of device were reading from

float clear, red, blue, green;

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

void uart_putc(unsigned char c){

    while(!(UCSR0A & (1<<UDRE0)));                      // wait until sending is possible
    UDR0 = c;                                           // output character saved in c
}
```



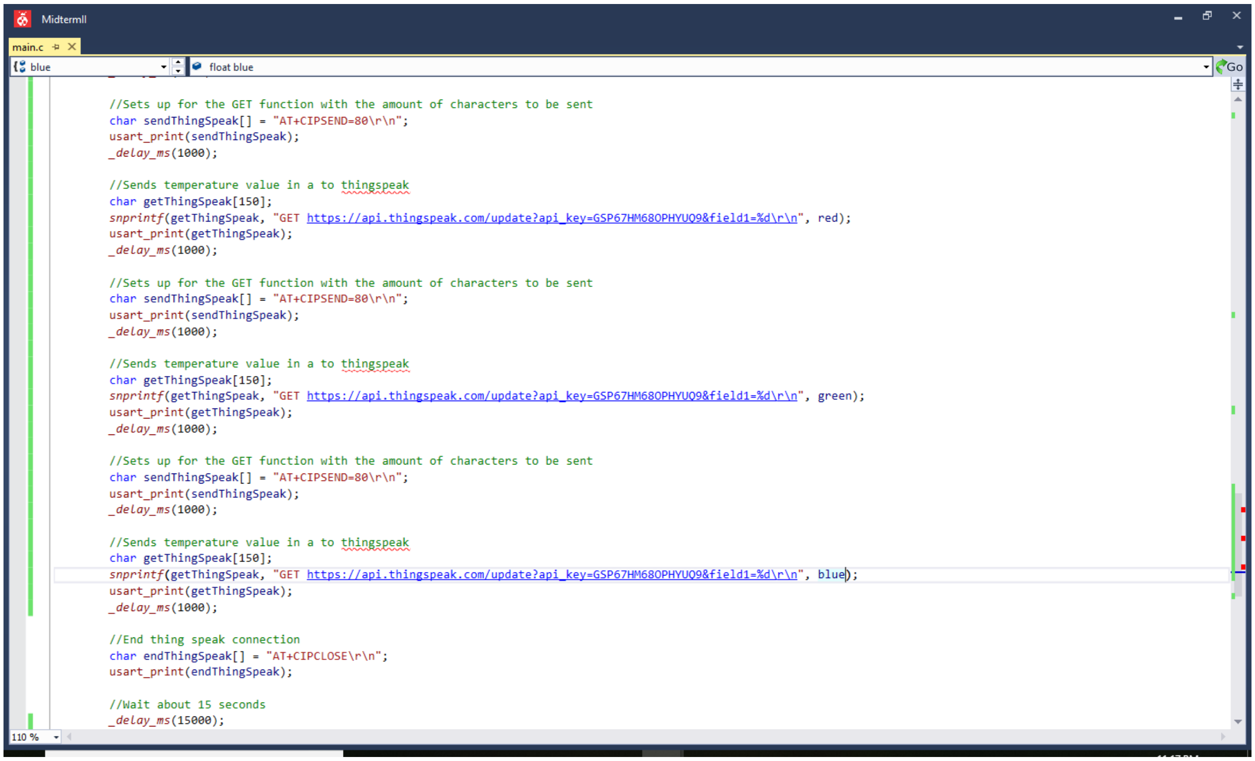
```
void uart_puts(char *s){

    while(*s)
    {
        uart_putc(*s);
        s++;
    }
}

void init_APDS9960 (void) {

    _delay_ms(150);                    /* Power up time > 100ms */
    i2c_start(APDS9960_I2C_ADDR);      //0x39
    i2c_write(APDS9960_ENABLE);        //0x80
    i2c_write(APDS9960_PON);           //Power On
    i2c_stop();

    i2c_start(APDS9960_I2C_ADDR);      //0x39
    i2c_write(APDS9960_ENABLE);        //0x80
    i2c_write(APDS9960_AEN);           //Enable ALS
    i2c_stop();

    i2c_start(APDS9960_I2C_ADDR);
    i2c_write(APDS9960_ATIME);         //Frame Synchronization & Digital Low Pass Filter (DLPF) setting
    i2c_write(0x86);                   //Field value = 182, 72 cycles, 200 ms, Max Count = 66535
    i2c_stop();

    i2c_start(APDS9960_I2C_ADDR);
    i2c_write(APDS9960_CONTROL);       //ALS Gain Control
    i2c_write(AGAIN_1X);
    i2c_stop();

}

void getreading(void){

    i2c_start(APDS9960_WRITE);
    i2c_write(APDS9960_CDATAL);    // set pointer
    i2c_stop();
```



```c
void getreading(void){

    i2c_start(APDS9960_WRITE);
    i2c_write(APDS9960_CDATAL);     // set pointer
    i2c_stop();

    i2c_start(APDS9960_READ);
    /*Store the values */
    clear = (((int)i2c_read_ack()) | (int)i2c_read_ack()<<8);
    red = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());
    blue = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());
    green = (((int)i2c_read_ack()<<8) | (int)i2c_read_ack());

    i2c_stop();
}

int main(void)
{
    char buffer[20], float_[10];

    init_uart(9600);
    i2c_init();
    init_APDS9960();

    while (1)
    {
        getreading();

        dtostrf( clear, 3, 2, float_ );
        sprintf(buffer," clear = %s g\t",float_);
        USART_SendString(buffer);

        dtostrf( red, 3, 2, float_ );
        sprintf(buffer," red = %s g\t",float_);
        USART_SendString(buffer);

        dtostrf( green, 3, 2, float_ );
        sprintf(buffer," green = %s g\t",float_);
        USART_SendString(buffer);
```

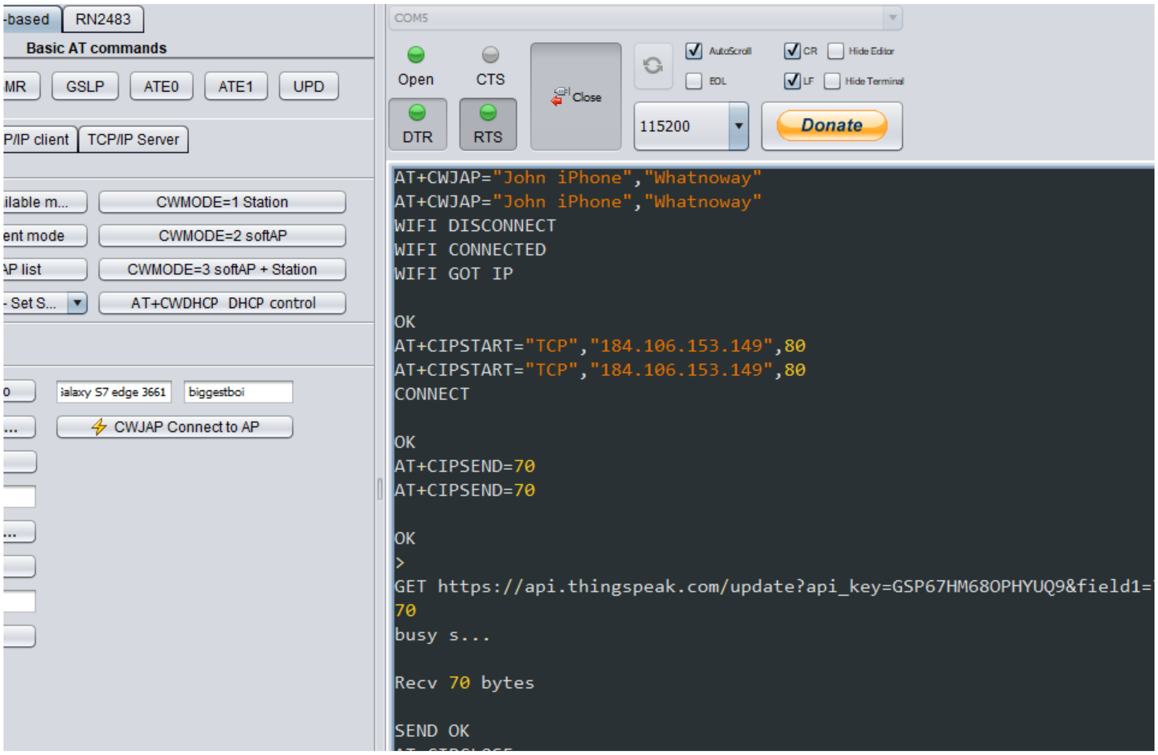


```c
        dtostrf( green, 3, 2, float_ );
        sprintf(buffer," green = %s g\t",float_);
        USART_SendString(buffer);

        dtostrf( blue, 3, 2, float_ );
        sprintf(buffer," blue = %s g\t",float_);
        USART_SendString(buffer);

        //Sets the MODE for wifi settings (AP or station mode)
        char setMODE[] = "AT+CWMODE=1\r\n";
        usart_print(setMODE);
        _delay_ms(1000);

        //Connects the ESP to wifi
        //Will change password after this assignment
        char setWIFI[] = "AT+CWJAP=\"John iPhone\",\"Whatnoway\"\r\n";
        usart_print(setWIFI);
        _delay_ms(1000);

        //Sets up the proper MUX settings
        char setMUX[] = "AT+CIPMUX=0\r\n";
        usart_print(setMUX);
        _delay_ms(1000);

        //Connect to thingspeak website
        char initThingSpeak[] = "AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n" ;
        usart_print(initThingSpeak);
        _delay_ms(1000);

        //Sets up for the GET function with the amount of characters to be sent
        char sendThingSpeak[] = "AT+CIPSEND=80\r\n";
        usart_print(sendThingSpeak);
        _delay_ms(1000);

        //Sends temperature value in a to thingspeak
        char getThingSpeak[150];
        snprintf(getThingSpeak, "GET https://api.thingspeak.com/update?api_key=GSP67HM68OPHYUQ9&field1=%d\r\n", clear);
        usart_print(getThingSpeak);
        _delay_ms(1000);
```

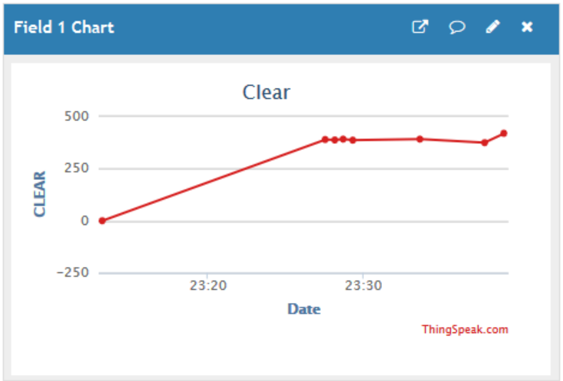


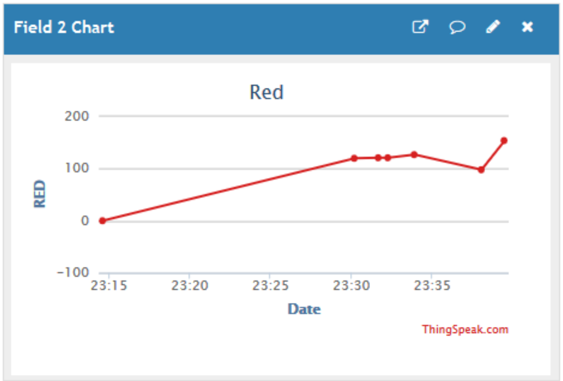
```c
        //Sets up for the GET function with the amount of characters to be sent
        char sendThingSpeak[] = "AT+CIPSEND=80\r\n";
        usart_print(sendThingSpeak);
        _delay_ms(1000);

        //Sends temperature value in a to thingspeak
        char getThingSpeak[150];
        snprintf(getThingSpeak, "GET https://api.thingspeak.com/update?api_key=GSP67HM68OPHYUQ9&field1=%d\r\n", red);
        usart_print(getThingSpeak);
        _delay_ms(1000);

        //Sets up for the GET function with the amount of characters to be sent
        char sendThingSpeak[] = "AT+CIPSEND=80\r\n";
        usart_print(sendThingSpeak);
        _delay_ms(1000);

        //Sends temperature value in a to thingspeak
        char getThingSpeak[150];
        snprintf(getThingSpeak, "GET https://api.thingspeak.com/update?api_key=GSP67HM68OPHYUQ9&field1=%d\r\n", green);
        usart_print(getThingSpeak);
        _delay_ms(1000);

        //Sets up for the GET function with the amount of characters to be sent
        char sendThingSpeak[] = "AT+CIPSEND=80\r\n";
        usart_print(sendThingSpeak);
        _delay_ms(1000);

        //Sends temperature value in a to thingspeak
        char getThingSpeak[150];
        snprintf(getThingSpeak, "GET https://api.thingspeak.com/update?api_key=GSP67HM68OPHYUQ9&field1=%d\r\n", blue);
        usart_print(getThingSpeak);
        _delay_ms(1000);

        //End thing speak connection
        char endThingSpeak[] = "AT+CIPCLOSE\r\n";
        usart_print(endThingSpeak);

        //Wait about 15 seconds
        _delay_ms(15000);
```

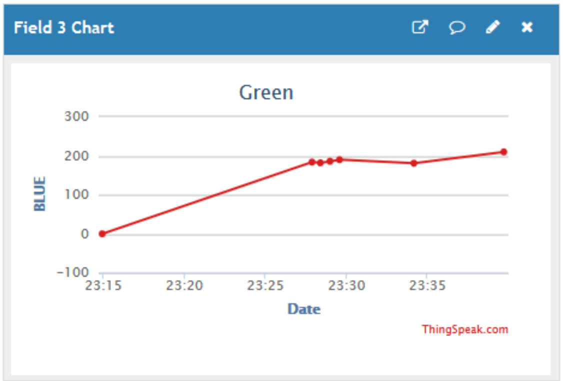
3. SCREENSHOTS OF EACH TASK OUTPUT

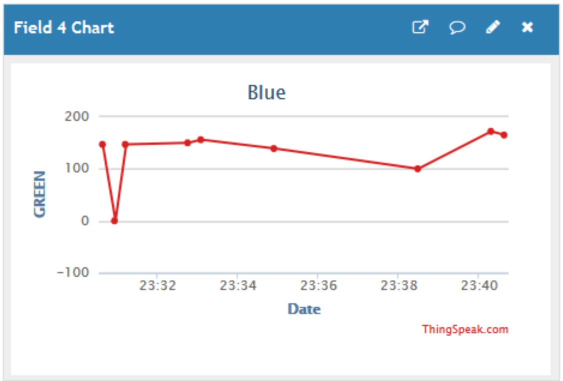


```
AT+CWJAP="John iPhone","Whatnoway"
AT+CWJAP="John iPhone","Whatnoway"
WIFI DISCONNECT
WIFI CONNECTED
WIFI GOT IP

OK
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSTART="TCP","184.106.153.149",80
CONNECT

OK
AT+CIPSEND=70
AT+CIPSEND=70

OK
>
GET https://api.thingspeak.com/update?api_key=GSP67HM68OPHYUQ9&field1=
70
busy s...

Recv 70 bytes

SEND OK
AT+CIPCLOSE
```



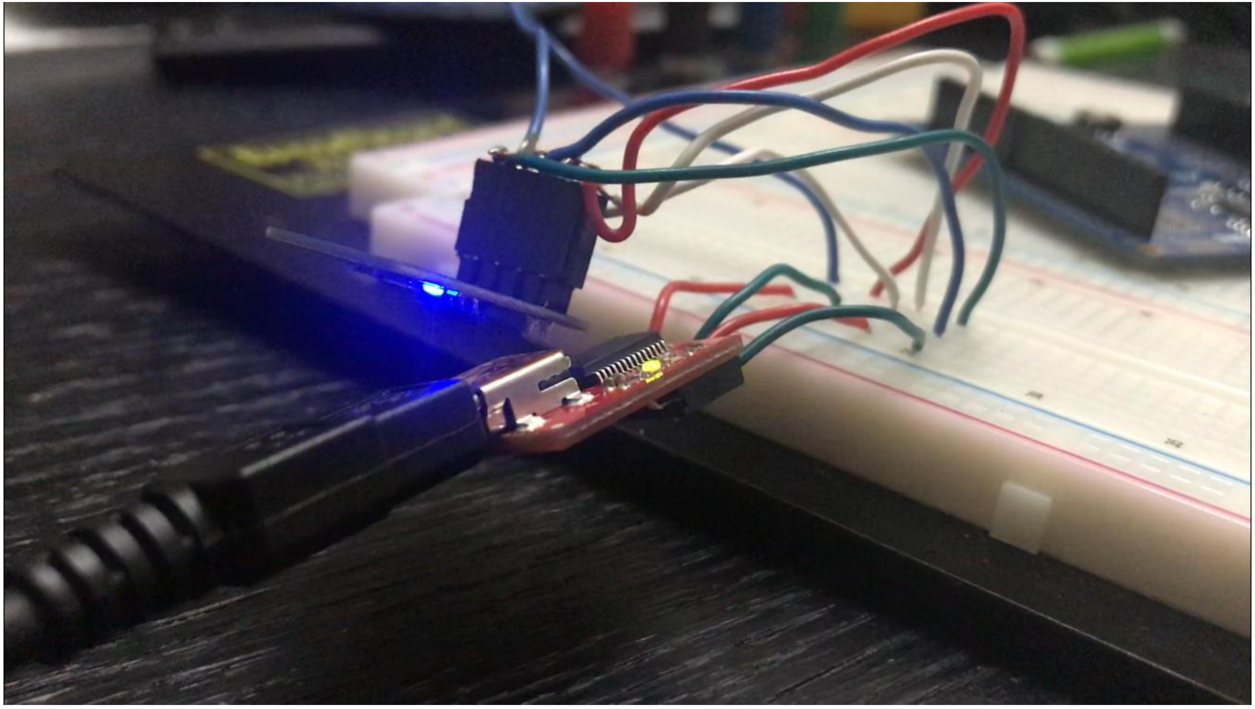




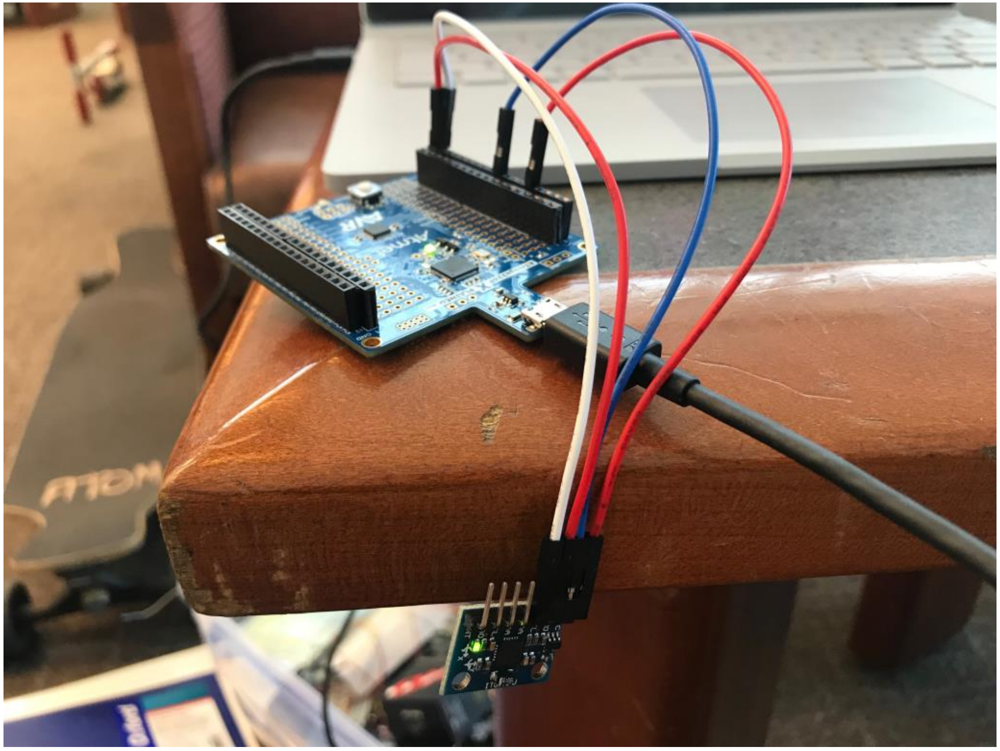


4. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Here is what the ESP looked like when it was getting programmed.



Here is the initial setup of the APDS 9960



5. VIDEO LINKS OF EACH DEMO

6. GITHUB LINK OF THIS DA

https://github.com/johnduriman/pirahnaplant.git

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

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