CPE301 – SPRING 2019

MIDTERM 2

Student Name: Chris Barr

Student #: 2000682859

Student Email: barrc1@unlv.nevada.edu

Primary Github address: https://github.com/BarrChris

Directory: https://github.com/BarrChris/submission\_da.git

Submit the following for all Labs:

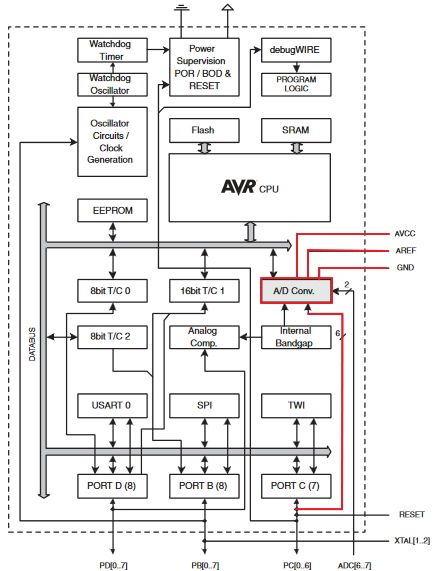
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder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

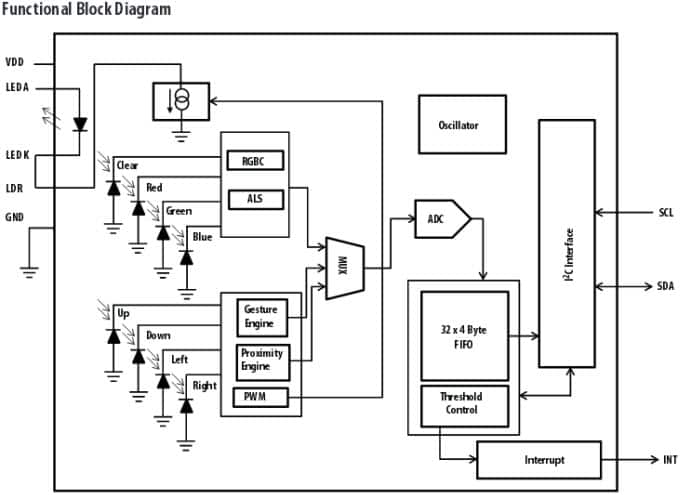
List of Components used

* 10 Wires
* Breadboard
* Atmega 328p
* FTDI Basic (testing purposes)
* ESP8266
* APDS9960

Block diagram with pins used in the Atmega328P

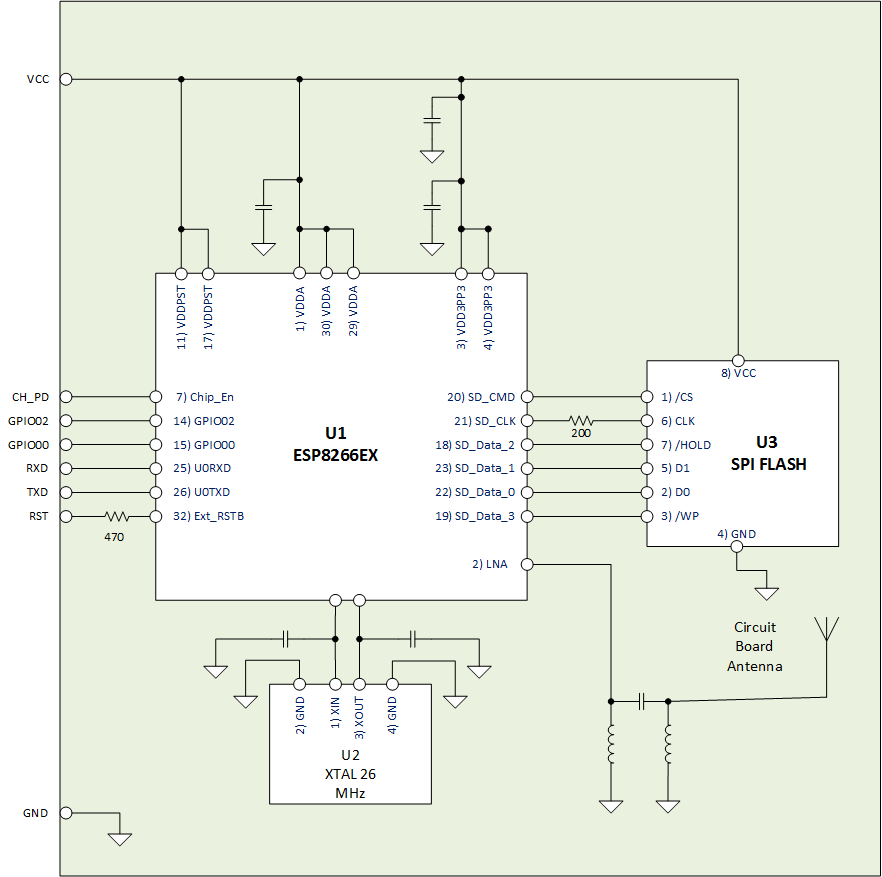


Used pins PC5 and PC4, used 3.3V for both the ESP and the APDS, and used PD0 and PD1 for TX and RX respectively.



Used SCL for the clock to be inputted to the APDS chip on PC5

Used SDA for the RGB data outputs on PC4



Used TX and RX to transmit and receive from the 328P to the ESP

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

/\*

\* Midterm 2

\*

\* Created: 5/10/2019 3:23:52 PM

\* Author : Chris

\*/

// NOTE: Did not tamper with the given 4 files, this is the main code!

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdlib.h>

#include <stdint.h>

#include "SparkFun\_APDS9960.h"

#include "i2c\_master.h"

#define F\_CPU 16000000UL

#define BAUD 9600

#define FOSC 16000000

#define UBRREQ FOSC/16/BAUD -1

#define APDS9960\_WRITE 0x72

#define APDS9960\_READ 0x73

void UART\_init **(**void**);**

void APDS\_init **(**void**);**

int uart\_putchar**(** char c**,** FILE **\***stream**);**

FILE str\_uart **=** FDEV\_SETUP\_STREAM**(**uart\_putchar**,** **NULL** **,** \_FDEV\_SETUP\_WRITE**);**

void getreading**(**void**);**

uint16\_t red**;**

uint16\_t green**;**

uint16\_t blue**;**

char sred**[**5**];**

char sgreen**[**5**];**

char sblue**[**5**];**

int main**(** void **)**

**{**

UART\_init**();** // Initializes UART values

APDS\_init**();** // Initializes APDS9960

i2c\_init**();** // Initializes I2C

stdout **=** **&**str\_uart**;**

red **=** 0**;**

green **=** 0**;**

blue **=** 0**;**

// Checks AT commands (not needed)

\_delay\_ms**(**2000**);**

printf**(**"AT\r\n"**);**

// Set AP’s info which will be connect by ESP8266. (AP + Station Mode)

\_delay\_ms**(**5000**);**

printf**(**"AT+CWMODE=3\r\n"**);**

// Connect to Internet

\_delay\_ms**(**5000**);**

printf**(**"AT+CWJAP=\"[WIFI NAME]\",\"[WIFI P/W]\"\r\n"**);** // WIFI SETTINGS LOCATED HERE

**while(**1**)** // Constantly send values through the cloud until device turns off

**{**

// ========================================================================================================

// Calls functions to connect to thingspeak, sets length of data to be sent, sends the data values to cloud,

// pauses till data goes through cloud

// ========================================================================================================

// Enable Single Connection

\_delay\_ms**(**5000**);**

printf**(**"AT+CIPMUX=0\r\n"**);**

// Start the connection to the cloud

\_delay\_ms**(**5000**);**

printf**(**"AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n"**);**

// Grab values from APDS chip

// Set length of data to be sent (type of data)

// Send values red, green, blue (in respective order) to field 1, field 2, and field 3

\_delay\_ms**(**5000**);**

getreading**();**

printf**(**"AT+CIPSEND=104\r\n"**);**

printf**(**"GET https://api.thingspeak.com/update?api\_key=[API\_WRITE\_KEY]&field1=0%05u&field2=%05u&field3=%05u\r\n"**,** red**,** green**,** blue**);** // KEY LOCATED HERE

\_delay\_ms**(**3000**);**

// ========================================================================================================

**}**

**}**

void getreading**(){**

uint8\_t redH**,** redL**;**

uint8\_t greenH**,** greenL**;**

uint8\_t blueH**,** blueL**;**

// RED

i2c\_readReg**(**APDS9960\_WRITE**,** APDS9960\_RDATAH**,** **&**redH**,** 1**);**

i2c\_readReg**(**APDS9960\_WRITE**,** APDS9960\_RDATAL**,** **&**redL**,** 1**);**

// GREEN

i2c\_readReg**(**APDS9960\_WRITE**,** APDS9960\_GDATAH**,** **&**greenH**,** 1**);**

i2c\_readReg**(**APDS9960\_WRITE**,** APDS9960\_GDATAL**,** **&**greenL**,** 1**);**

// BLUE

i2c\_readReg**(**APDS9960\_WRITE**,** APDS9960\_BDATAH**,** **&**blueH**,** 1**);**

i2c\_readReg**(**APDS9960\_WRITE**,** APDS9960\_BDATAL**,** **&**blueL**,** 1**);**

red **=** **(**redH **<<** 8**)** **|** redL**;**

green **=** **(**greenH **<<** 8**)** **|** greenL**;**

blue **=** **(**blueH **<<** 8**)** **|** blueL**;**

// THRESHOLD

**if** **(**red **>** 255**)**

red **=** 255**;**

**if** **(**green **>** 255**)**

green **=** 255**;**

**if** **(**blue **>** 255**)**

blue **=** 255**;**

**}**

void APDS\_init**(**void**){**

uint8\_t setup**;**

i2c\_readReg**(**APDS9960\_WRITE**,** APDS9960\_ID**,** **&**setup**,**1**);**

**if(**setup **!=** APDS9960\_ID\_1**)** **while(**1**);**

setup **=** 1 **<<** 1 **|** 1**<<**0 **|** 1**<<**3 **|** 1**<<**4**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_ENABLE**,** **&**setup**,** 1**);**

setup **=** DEFAULT\_ATIME**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_ATIME**,** **&**setup**,** 1**);**

setup **=** DEFAULT\_WTIME**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_WTIME**,** **&**setup**,** 1**);**

setup **=** DEFAULT\_PROX\_PPULSE**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_PPULSE**,** **&**setup**,** 1**);**

setup **=** DEFAULT\_POFFSET\_UR**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_POFFSET\_UR**,** **&**setup**,** 1**);**

setup **=** DEFAULT\_POFFSET\_DL**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_POFFSET\_DL**,** **&**setup**,** 1**);**

setup **=** DEFAULT\_CONFIG1**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_CONFIG1**,** **&**setup**,** 1**);**

setup **=** DEFAULT\_PERS**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_PERS**,** **&**setup**,** 1**);**

setup **=** DEFAULT\_CONFIG2**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_CONFIG2**,** **&**setup**,** 1**);**

setup **=** DEFAULT\_CONFIG3**;**

i2c\_writeReg**(**APDS9960\_WRITE**,** APDS9960\_CONFIG3**,** **&**setup**,** 1**);**

**}**

void USART\_putstring**(**char **\***StringPtr**)**

**{**

**while** **((\***StringPtr **!=** '\0'**)){** // Until it reaches the end of the line, it will keep looping

**while** **(!(**UCSR0A **&** **(**1 **<<** UDRE0**)));** // Until UDRE0 goes high, it will keep looping

UDR0 **=** **\***StringPtr**;** // UDR0 register grabs the value given from the parameter

StringPtr**++;** // but it does it by every character as shown here

**}**

**}**

void UART\_init**(**void**)**

**{**

//Set baud rate

uint16\_t baud\_rate **=** UBRREQ**;**

UBRR0H **=** baud\_rate **>>** 8**;**

UBRR0L **=** baud\_rate **&** 0xFF**;**

//Enable receiver and transmitter

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

// Set frame format: 8data, 1stop bit

UCSR0C **=** **(**3 **<<**UCSZ00**);**

**}**

int uart\_putchar**(**char c**,** FILE **\***stream**)**

**{**

//wait until buffer empty

**while** **(** **!(** UCSR0A **&** **(** 1 **<<**UDRE0**))** **);**

//Put data into buffer

UDR0 **=** c**;**

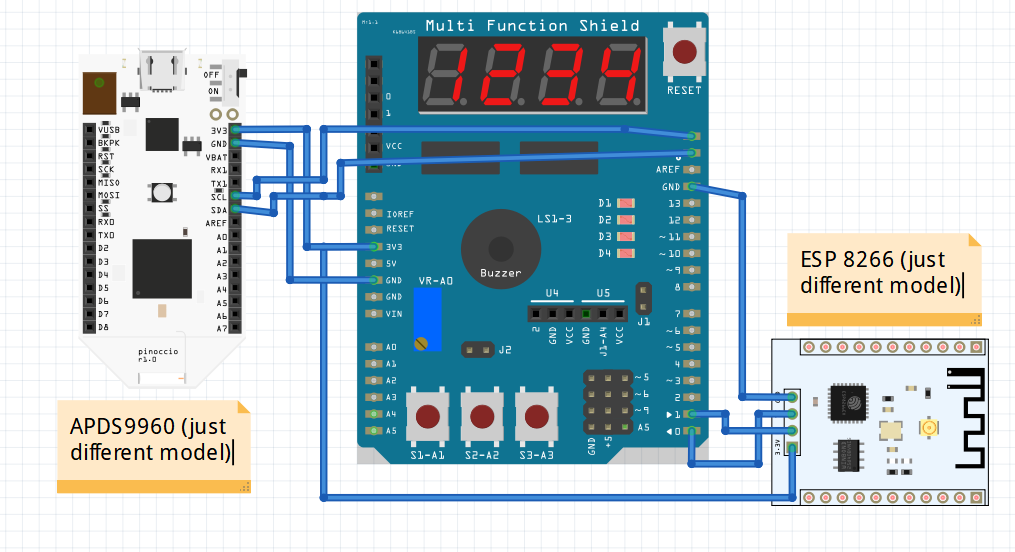
**return** 0**;**

**}**

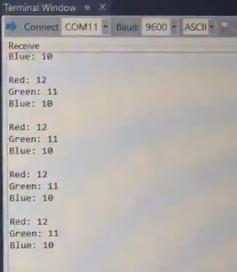
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

n/a

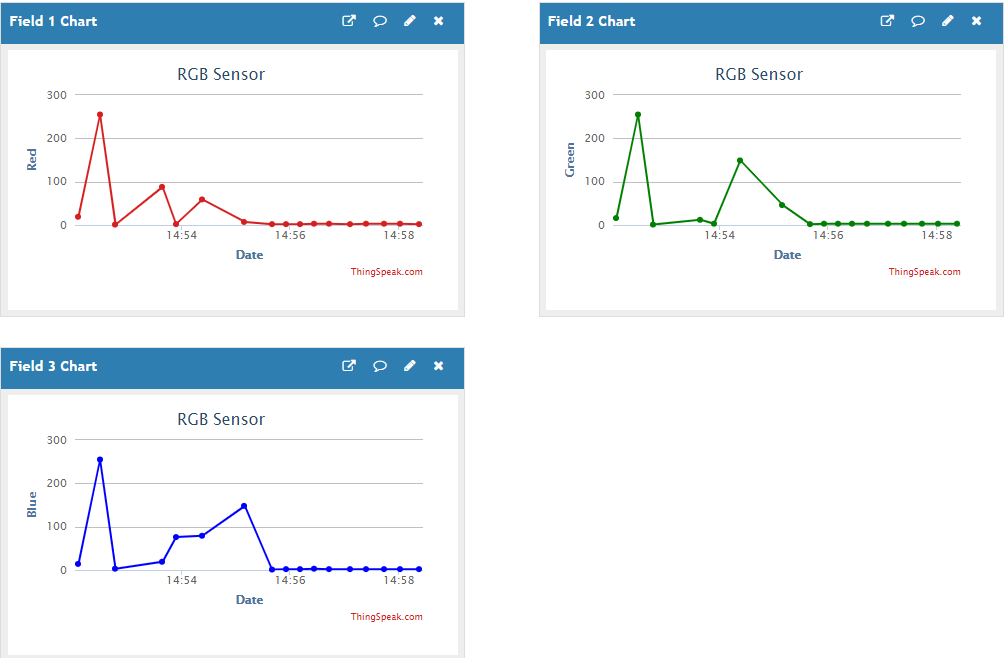
1. **SCHEMATICS**



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



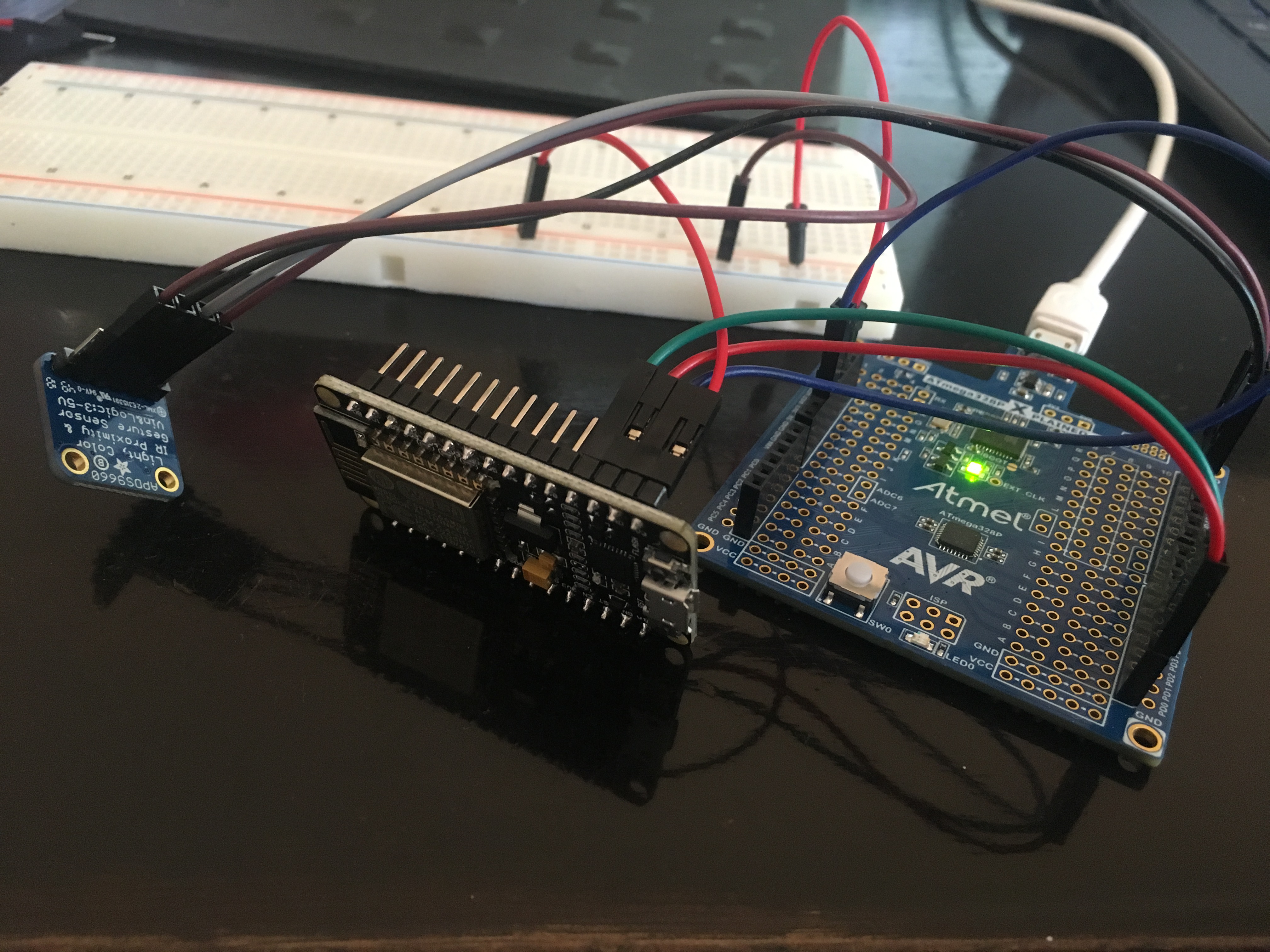
Terminal Output for testing the sensor



ThingSpeak output

Note: Tops off at 255 for each RGB which shows at the beginning of each graph. A white light was shined into it which gave 255 for all three. **255 is the Threshold** for the device.

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



Board setup with APDS (blue) on the left, ESP in the center, and the Atmega328P on the right

1. **VIDEO LINKS OF EACH DEMO**

<https://www.youtube.com/watch?v=Ky9lEbHhfRw>

1. **GITHUB LINK OF THIS DA**

<https://github.com/BarrChris/submission_da/tree/master/Midterms>

**Student Academic Misconduct Policy**

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

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

Chris Barr