

**CPE301 – SPRING 2019**

# MIDTERM 2/FINAL

---

Student Name: Jett Guerrero

Student #: 5002377713

Student Email: guerrj1@unlv.nevada.edu

Primary Github address: <https://github.com/guerrj1>

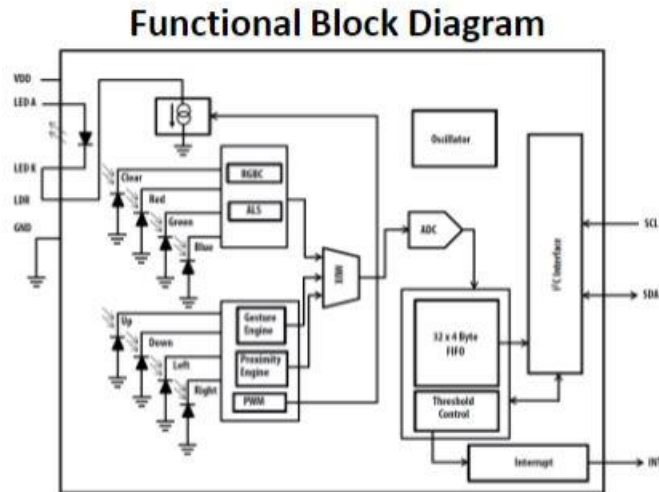
Directory: [https://github.com/guerrj1/Submission\\_Midterms/tree/master/Midterm2](https://github.com/guerrj1/Submission_Midterms/tree/master/Midterm2)

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
- Micro USB Cable
- APDS-9960 Module
- ESP-8266 Module
- Male to male wire
- Breadboard



Functional Block Diagram of the APDS-9960

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

//Midterm2

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <stdio.h>
#include <util/delay.h>
#include <stdlib.h>
#include <stdint.h>
#include <avr/interrupt.h>
#include "SparkFun_APDS9960.h"
#include "i2c_master.h"

#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);
void getreading(void);
int uart_putchar( char c, FILE *stream);
FILE str_uart = FDEV_initialize_STREAM(uart_putchar, NULL , _FDEV_initialize_WRITE);

uint16_t red, green, blue;
```

```

int main( void )
{
    UART_init();      //uart initialization
    APDS_init();      //APDS initialization
    i2c_init();       //i2c initialization
    stdout = &str_uart;

    red = 0;          //color initialization
    green = 0;        //color initialization
    blue = 0;         //color initialization

    _delay_ms(2000);
    printf("AT\r\n");
    //Connection type
    _delay_ms(5000);
    printf("AT+CWMODE=3\r\n");
    //internet connection with ssid and password
    _delay_ms(5000);
    printf("AT+CWJAP=\"%ssid\", \"wifi password\"\r\n");

    while(1)
    {
        //single connection enabled
        _delay_ms(5000);
        printf("AT+CIPMUX=0\r\n");
        //connection to thingspeak
        _delay_ms(5000);
        printf("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", 80\r\n");
        _delay_ms(5000);
        getreading();
        printf("AT+CIPSEND=104\r\n");
        printf("GET
https://api.thingspeak.com/update?api_key=CXDQTMUMJGB6K44I&field1=0%05u&field2=%05u&field3=%05
u\r\n", red, green, blue); //API key
        _delay_ms(3000);
    }
}

void UART_init(void)
{
    //baud rate settings
    uint16_t baud_rate = UBRREQ;
    UBRRH = baud_rate >> 8;
    UBRRL = baud_rate & 0xFF;
    UCSRB = ( 1 <<RXEN0)|( 1 <<TXEN0); //enable reciver and transmitter
    UCSRC = (3 <<UCSZ00); //8 data bits
}

int uart_putchar(char c, FILE *stream)
{
    while ( !( UCSRA & ( 1 <<UDRE0)) ); //while empty buffer
    UDR0 = c;
    return 0;
}

void getreading()
{
    uint8_t redh, redl, greenh, greenl, blueh, blue1;

    //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;

//max value limit
if (red > 255)
{
    red = 255;
}
if (green > 255)
{
    green = 255;
}
if (blue > 255)
{
    blue = 255;
}
}

void APDS_init(void)
{
    uint8_t initialize ;

    i2c_readReg(APDS9960_WRITE, APDS9960_ID, &initialize ,1);
    if(initialize != APDS9960_ID_1) while(1);
    initialize = (1<<1) | (1<<0) | (1<<3) | (1<<4);

    i2c_writeReg(APDS9960_WRITE, APDS9960_ENABLE, &initialize , 1);
    initialize = DEFAULT_ETIME;

    i2c_writeReg(APDS9960_WRITE, APDS9960_ETIME, &initialize , 1);
    initialize = DEFAULT_WTIME;

    i2c_writeReg(APDS9960_WRITE, APDS9960_WTIME, &initialize , 1);
    initialize = DEFAULT_PROX_PPULSE;

    i2c_writeReg(APDS9960_WRITE, APDS9960_PPULSE, &initialize , 1);
    initialize = DEFAULT_POFFSET_UR;

    i2c_writeReg(APDS9960_WRITE, APDS9960_POFFSET_UR, &initialize , 1);
    initialize = DEFAULT_POFFSET_DL;

    i2c_writeReg(APDS9960_WRITE, APDS9960_POFFSET_DL, &initialize , 1);
    initialize = DEFAULT_CONFIG1;

    i2c_writeReg(APDS9960_WRITE, APDS9960_CONFIG1, &initialize , 1);
    initialize = DEFAULT_PERS;

    i2c_writeReg(APDS9960_WRITE, APDS9960_PERS, &initialize , 1);
    initialize = DEFAULT_CONFIG2;
}

```

```

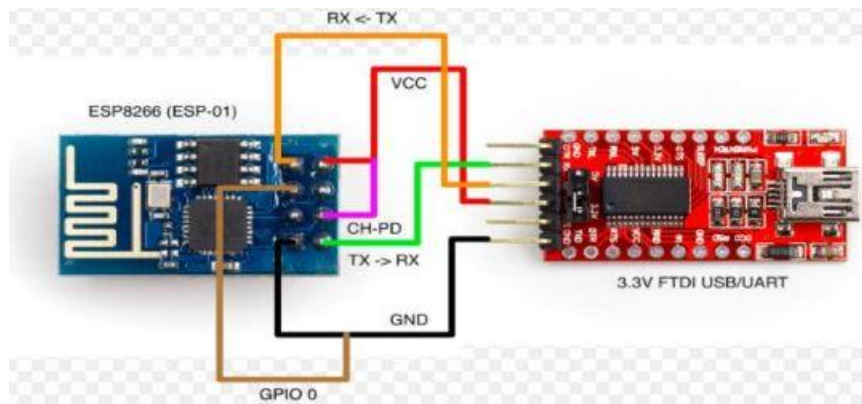
i2c_writeReg(APDS9960_WRITE, APDS9960_CONFIG2, &initialize , 1);
initialize = DEFAULT_CONFIG3;

i2c_writeReg(APDS9960_WRITE, APDS9960_CONFIG3, &initialize , 1);
}

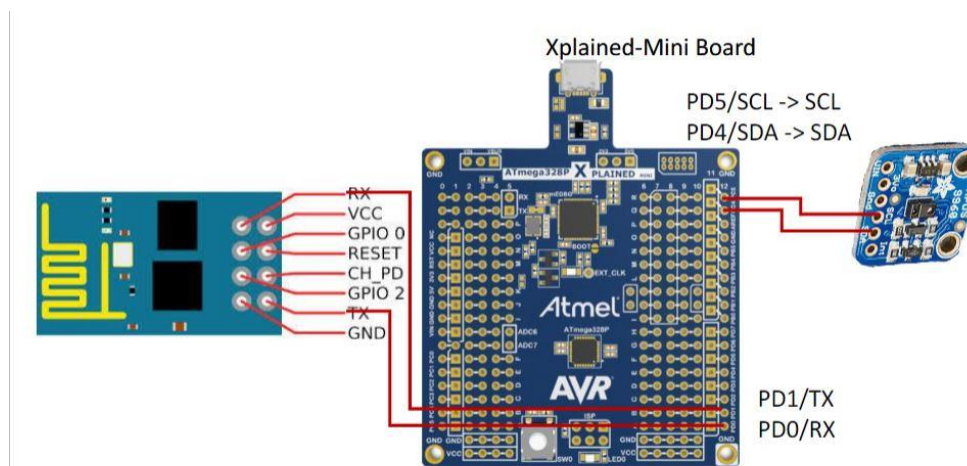
void USART_putstring(char *data)
{
    while ((*data != '\0')){
        while (!(UCSR0A & (1 << UDRE0)));
        UDR0 = *data;
        data++;
    }
}

```

### 3. SCHEMATICS

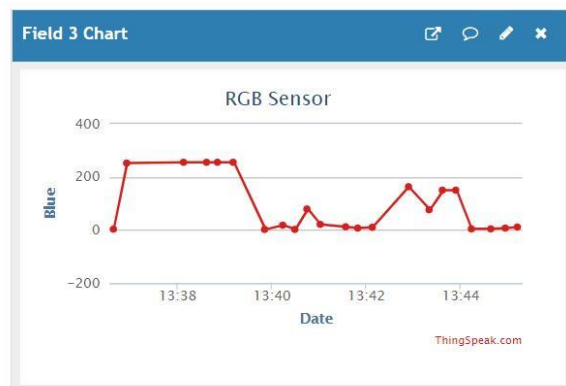
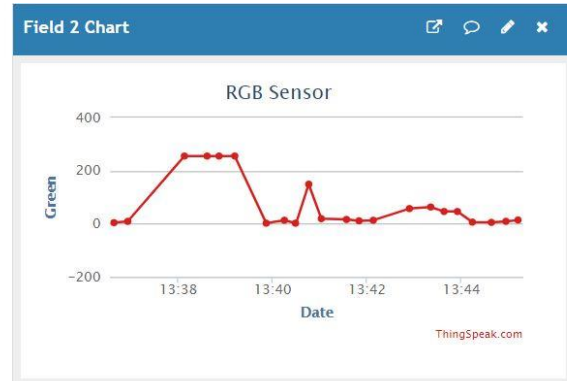
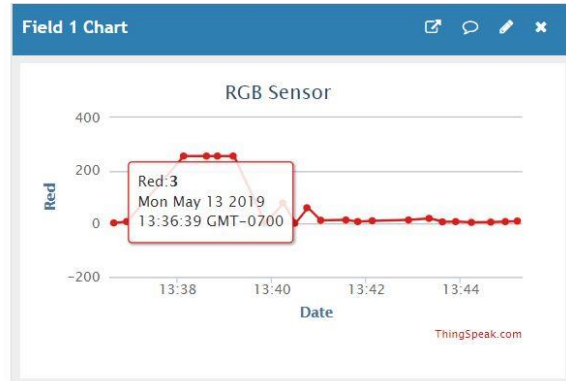


Schematic for flashing ESP module



Schematic of the ESP module and the APDS module to the Atmega328p

#### 4. SCREENSHOTS OF TASK OUTPUT (ATMEL STUDIO OUTPUT & THINGSPEAK)



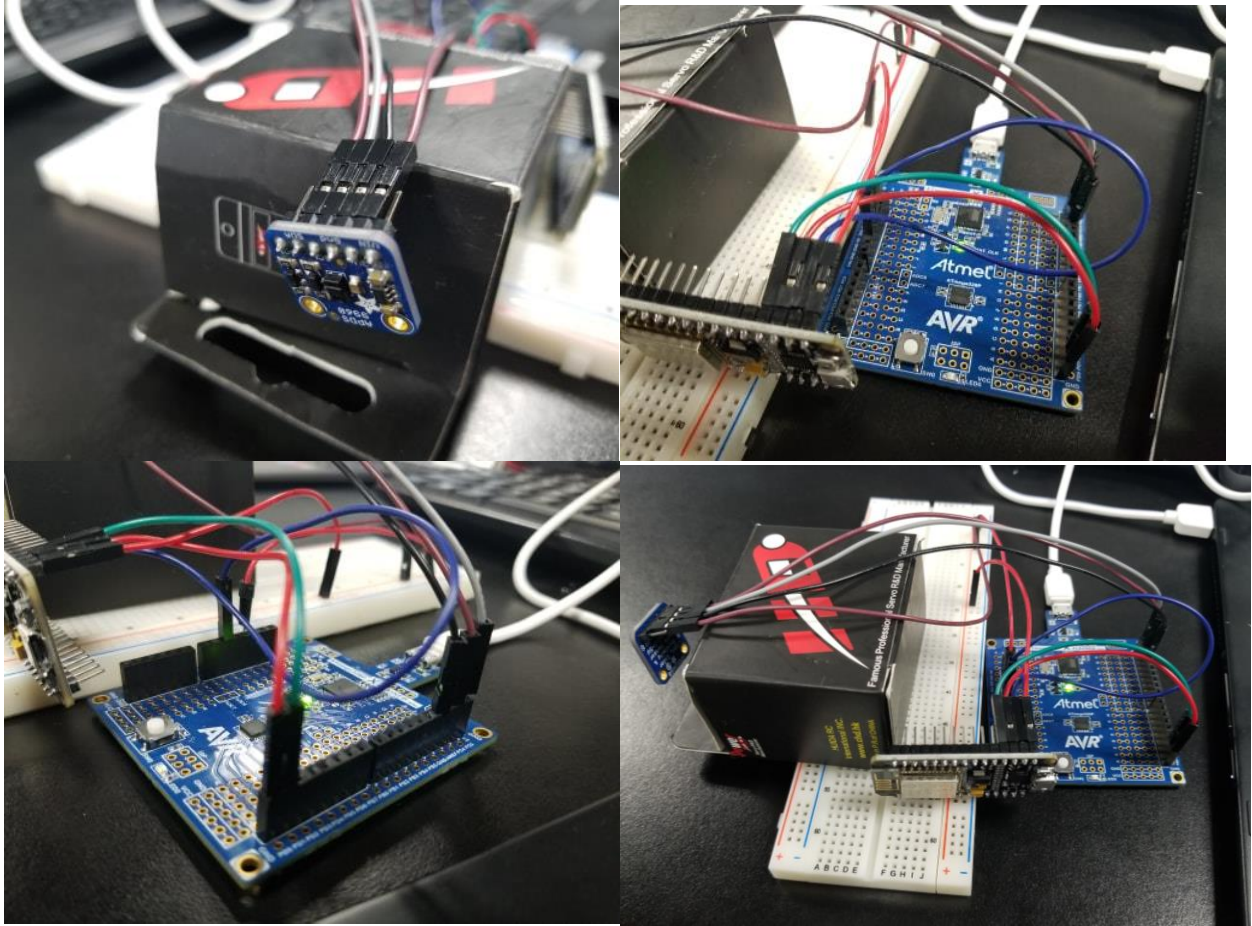
Graph of the RGB through Thingspeak Cloud

```
Terminal Window
Disconnect COM6 Baud: 9600 ASCII
Receive
AT+CIPMUX=0
AT+CIPSTART="TCP","api.thingspeak.com",80
AT+CIPSEND=104
date?
api_key=CXDQTMUMJGB6K44I&field1=000011&field2=00014&field3=00011
AT+CIPMUX=0
AT+CIPSTART="TCP","api.thingspeak.com",80
AT+CIPSEND=104
pi_key=CXDQTMUMJGB6K44I&field1=000011&field2=00014&field3=00010
AT+CIPMUX=0
AT+CIPSTART="TCP","api.thingspeak.com",80
AT+CIPSEND=104
ate?
api_key=CXDQTMUMJGB6K44I&field1=000010&field2=00014&field3=00010
AT+CIPMUX=0
AT+CIPSTART="TCP","api.thingspeak.com",80
AT+CIPSEND=104
i_key=CXDQTMUMJGB6K44I&field1=000010&field2=00013&field3=00010
AT+CIPMUX=0
AT+CIPSTART="TCP","api.thingspeak.com",80
AT+CIPSEND=104
GET https://api.thingspeak.com/up
```

Receiving and transmitting displayed on the terminal



5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



Board set up with connections with the Atmega328p to the two modules

6. VIDEO LINKS OF EACH DEMO

<https://youtu.be/BZanvXhKKgo>

7. GITHUB LINK OF THIS DA

[https://github.com/guerri1/Submission\\_Midterms/tree/master/Midterm2](https://github.com/guerri1/Submission_Midterms/tree/master/Midterm2)

Student Academic Misconduct Policy

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

*"This assignment submission is my own, original work".*

Jett Guerrero