

CPE301 – SPRING 2019
MIDTERM II

Student Name: Rocky Yasuaki Gonzalez

Student #: 5003229733

Student Email: gonzar14@unlv.nevada.edu

Primary Github address: <https://github.com/rockyg1995/ihswwppdar>

Directory: https://github.com/rockyg1995/ihswwppdar/tree/master/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/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

Atmega328PB Xplained Mini
Micro USB Cable (Power Supply)
Wire Connectors
ADPS-9960
ESP-01

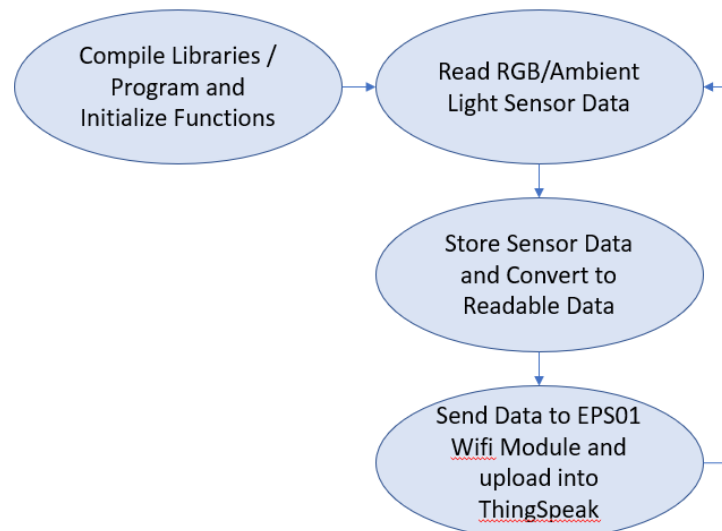


Figure 1 – Flow Diagram for displaying RGB Values to ThingSpeak

2. INITIAL/MODIFIED/DEVELOPED CODE

```
/*
 * MIDTERM_II.c
 *
 * Created: 5/10/2019 11:02:21 PM
 * Author: RYG95
 */

#ifndef F_CPU
#define F_CPU 16000000UL
#endif

#include <avr/io.h>          /* Include AVR input/output file */
#include <util/delay.h>      /* Include delay functions file */
#include <math.h>            /* Include math functions file */
#include <stdlib.h>          /* Include standard library file */
#include <stdio.h>           /* Include standard input/output file */
#include "APDS9960_def.h"    /* Include MPU6050 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
```

```

float C_Light;
float R_Light;
float G_Light;
float B_Light;

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<<RXCIF0); // UART TX (Transmit - senden)
    einschalten
    UCSR0C |= (1<<USBS0) | (3<<UCSZ00); //Modus Asynchron 8N1 (8 Datenbits, 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);

    i2c_start(APDS9960_WRITE);
    i2c_write(APDS9960_ENABLE);
    i2c_write(0x00);
    i2c_stop();

    i2c_start(APDS9960_WRITE);
    i2c_write(APDS9960_ATIME);
    i2c_write(DEFAULT_ATIME);
    i2c_stop();

    i2c_start(APDS9960_WRITE);
    i2c_write(APDS9960_CONTROL);
    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_CDATAH); // set pointer
    i2c_stop();
    i2c_start(APDS9960_READ);
    C_Light = (((int)i2c_read_ack() << 8) | (int)i2c_read_ack());
    i2c_stop();

    i2c_start(APDS9960_WRITE);
    i2c_write(APDS9960_RDATAH); // set pointer
    i2c_stop();
    i2c_start(APDS9960_READ);
    R_Light = (((int)i2c_read_ack() << 8) | (int)i2c_read_ack());
    i2c_stop();

    i2c_start(APDS9960_WRITE);
    i2c_write(APDS9960_GDATAH); // set pointer
    i2c_stop();
    i2c_start(APDS9960_READ);
    G_Light = (((int)i2c_read_ack() << 8) | (int)i2c_read_ack());
    i2c_stop();

    i2c_start(APDS9960_WRITE);
    i2c_write(APDS9960_BDATAH); // set pointer
    i2c_stop();
    i2c_start(APDS9960_READ);
    B_Light = (((int)i2c_read_ack() << 8) | (int)i2c_read_ack());
    i2c_stop();
}

int main(void) {
    char buffer[20], float_[10];
    float C1;
    float R1;
    float G1;
    float B1;

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

    uart_puts("TEST\r\n");

    uart_puts("AT+CWJAP=\"WIFI_NAME_HERE\", \"PASSWORD_HERE\""); // Log in WiFi

    while(1) {
        getreading();
        C1 = C_Light; // Divide raw value by sensitivity scale factor
        to get real values */
        R1 = R_Light;
        G1 = G_Light;
        B1 = B_Light;

        dtostrf(C1, 3, 2, float_); // Take values in buffer to send all
        parameters over USART */
        sprintf(buffer, "%s C1, ", float_);
        uart_puts(buffer);
    }
}

```

```

        dtostrf( R1, 3, 2, float_ );      /* Take values in buffer to send all
parameters over USART */
        sprintf(buffer,"%s R1, ",float_);
        uart_puts(buffer);

        dtostrf( G1, 3, 2, float_ );      /* Take values in buffer to send all
parameters over USART */
        sprintf(buffer,"%s G1, ",float_);
        uart_puts(buffer);

        dtostrf( B1, 3, 2, float_ );      /* Take values in buffer to send all
parameters over USART */
        sprintf(buffer,"%s B1, ",float_);
        uart_puts(buffer);

        uart_puts("AT+CIPSTART=\"TCP\", \"api.thingSpeak.com\",80");      //
Connect API Key
        uart_puts("AT+CIPSEND=51");
        // Send Serial Data
        uart_puts("GET /update?key=LMPV6R4U5HWZLME7&field1=outs\r\n");// Send Value
        uart_puts("AT+CIPCLOSE");
        // Close Data

        uart_puts("\r\n");
        _delay_ms(1000);
    }

    return 0;
}

```

3. SCHEMATICS

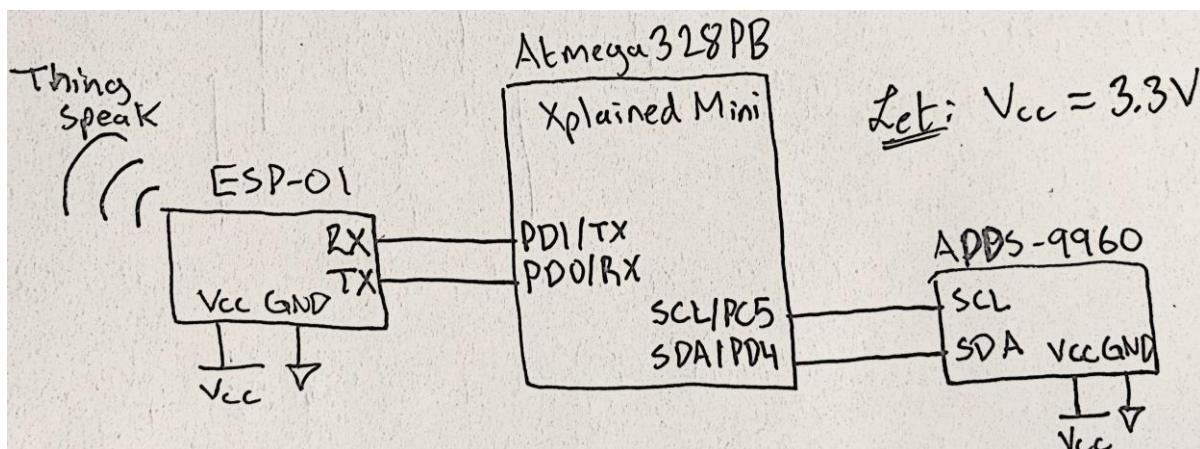


Figure 2 – Schematic of APDS-9960 Connected to ESP-01

4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

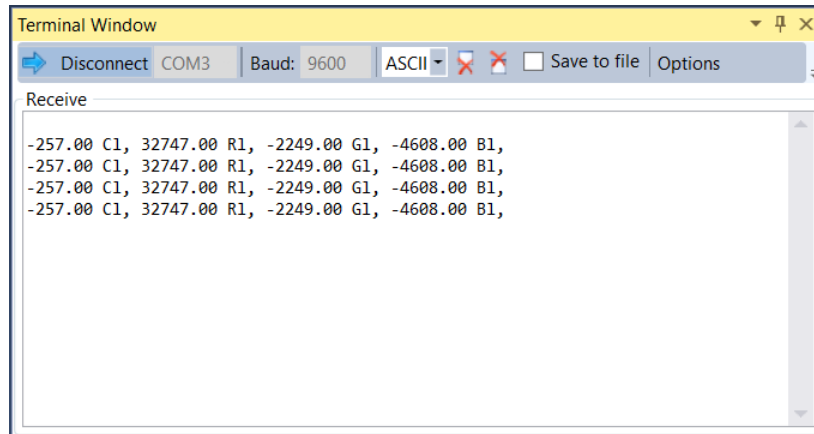


Figure 3 – Output Terminal RGBC Values

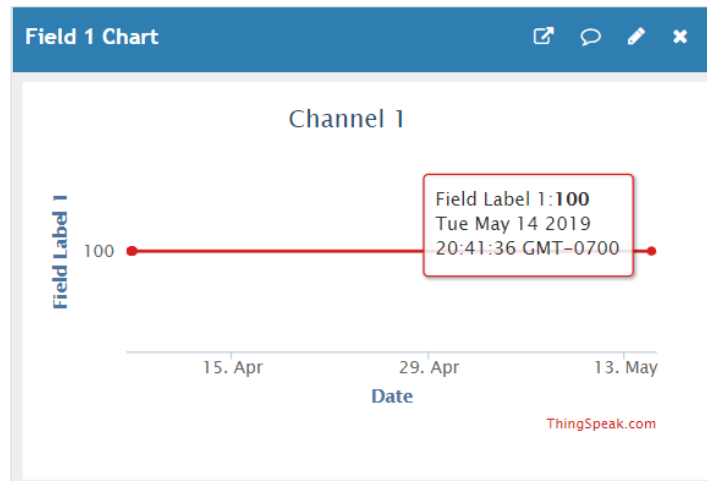


Figure 4 – Output through ThingSpeak Practice Using Explorer

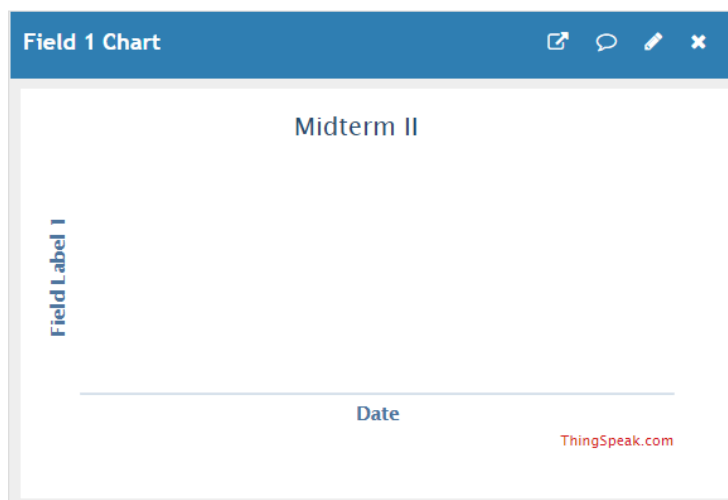


Figure 5 – Output through ThingSpeak RGBC Reading

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

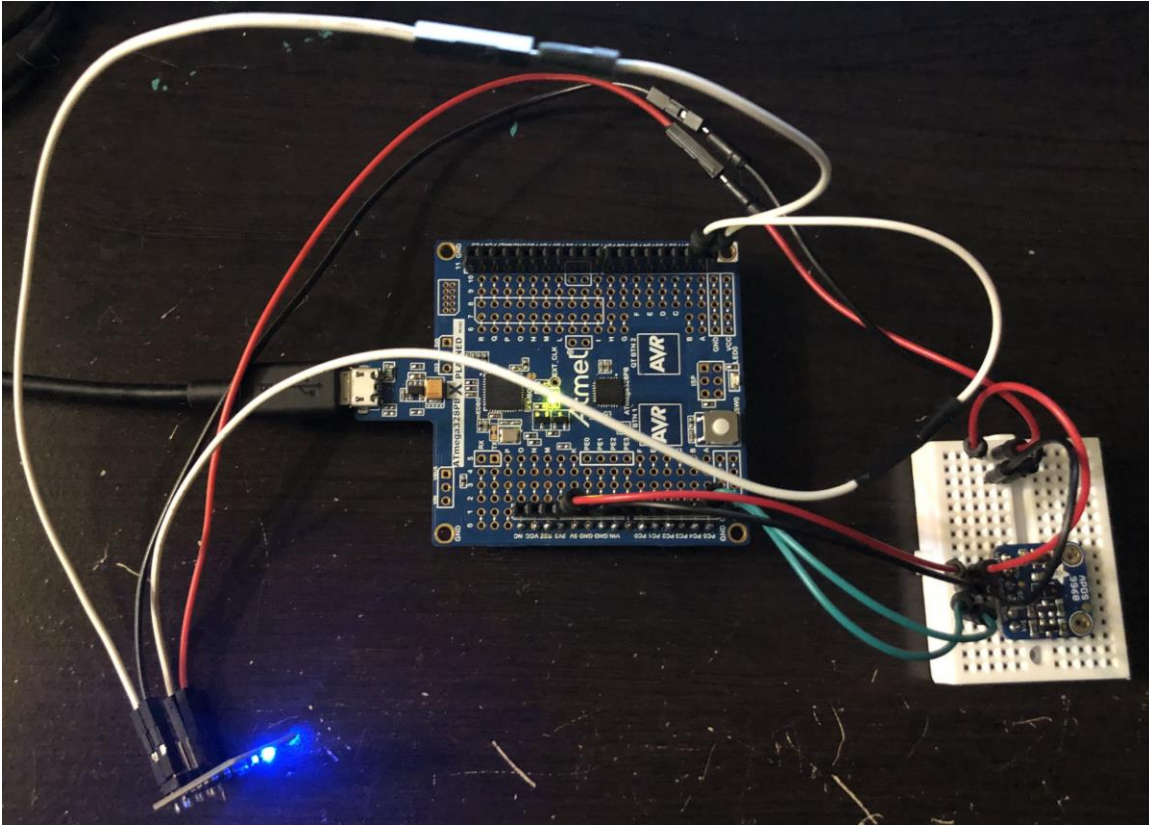


Figure 6 – Board Setup for APDS-9960 Connected to ESP-01 Module

6. VIDEO LINKS OF EACH DEMO

<https://youtu.be/WiNJATyUzM0>

7. GITHUB LINK OF THIS DA

https://github.com/rockyg1995/ihswwpdar/tree/master/Midterms/Midterm_II

Student Academic Misconduct Policy

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

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

Rocky Yasuaki Gonzalez