CPE301 – SPRING 2019

MIDTERM II

Student Name: Alex Cater
Student #: 2000583031
Student Email: cater@unlv.nevada.edu
Primary Github address: https://github.com/katmando/asdfghjkl

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

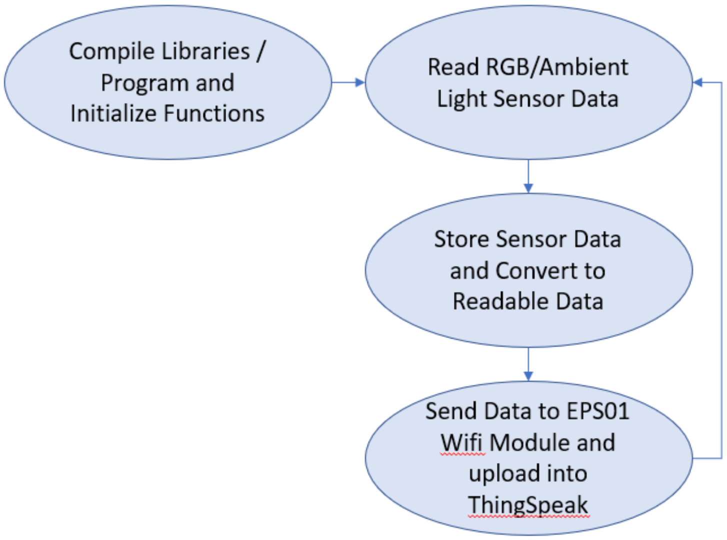

Compile Libraries / Program and Initialize Functions
Read RGB/Ambient Light Sensor Data
Store Sensor Data and Convert to Readable Data
Send Data to EPS01 Wifi Module and upload into ThingSpeak

Figure 1 – Flow Diagram for displaying RGBC Values to ThingSpeak

2. INITIAL/MODIFIED/DEVELOPED CODE

```
/*
 * MIDTERM_II.c
 *
 * Created: 5/08/2019 10:42:31 PM
 * Author: acate
 */

#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<<RXCIE0);
       UCSR0C |= (1<<USBS0) | (3<<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);

       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 Cl;
       float Rl;
       float Gl;
       float Bl;

       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();
              Cl = C_Light;              /* Divide raw value by sensitivity scale factor
to get real values */
              Rl = R_Light;
              Gl = G_Light;
              Bl = B_Light;

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

              dtostrf( Rl, 3, 2, float_ );      /* Take values in buffer to send all
parameters over USART */
```

```
                sprintf(buffer,"%s Rl, ",float_);
                uart_puts(buffer);

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

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

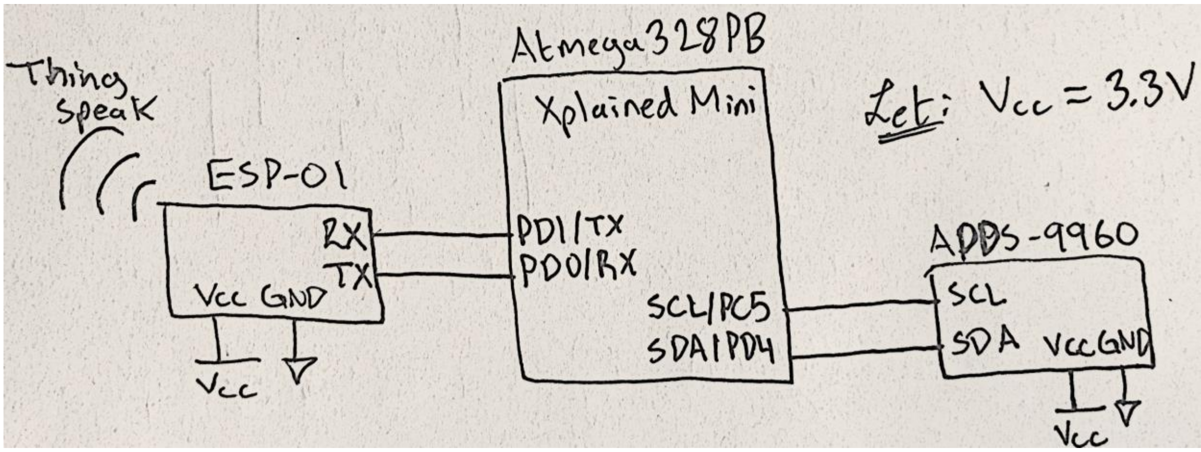
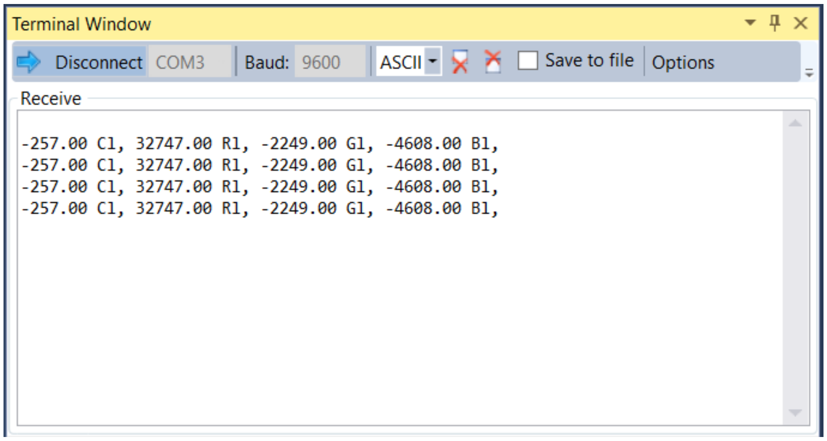


Figure 2 – Schematic of APDS-9960 Connected to ESP-01

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



```
-257.00 C1, 32747.00 R1, -2249.00 G1, -4608.00 B1,
-257.00 C1, 32747.00 R1, -2249.00 G1, -4608.00 B1,
-257.00 C1, 32747.00 R1, -2249.00 G1, -4608.00 B1,
-257.00 C1, 32747.00 R1, -2249.00 G1, -4608.00 B1,
```

Figure 3 – Output Terminal RGBC Values

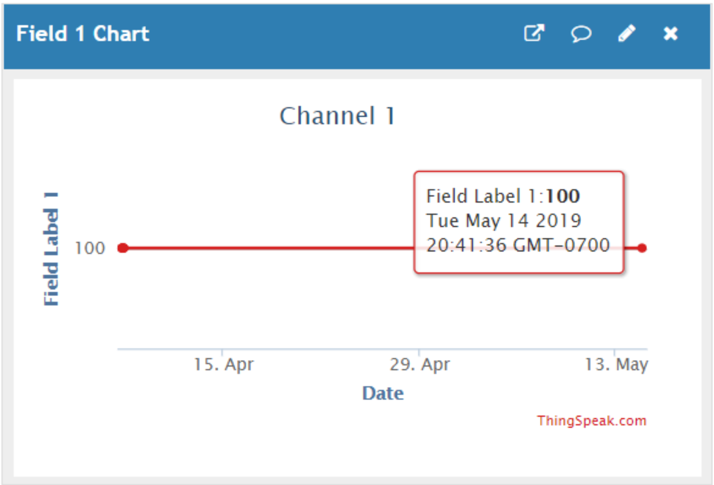


Figure 4 – Output through ThingSpeak Practice Using Esplorer

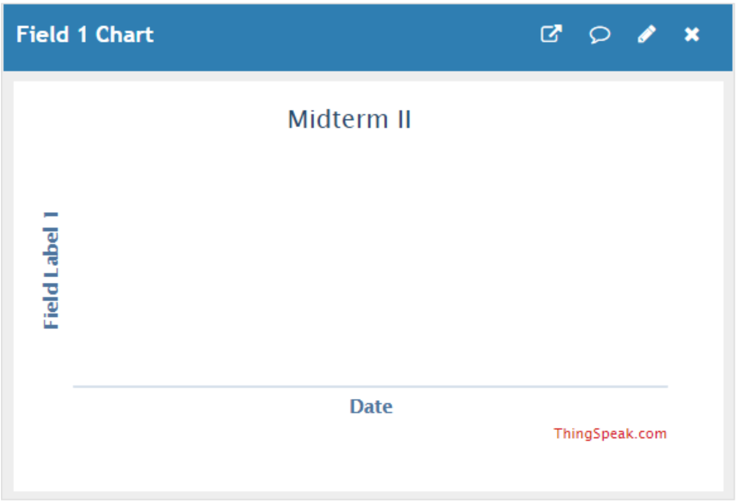


Figure 5 – Output through ThingSpeak RGBC Reading

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

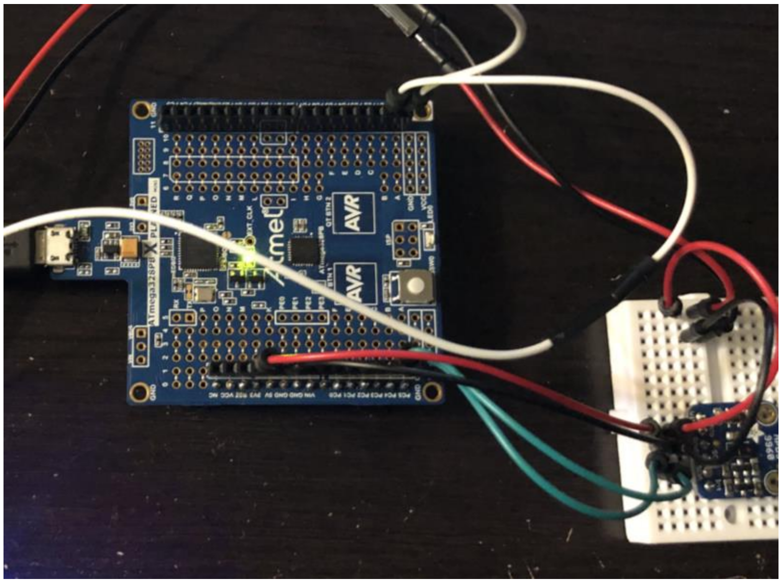


Figure 6 – Board Setup

6. GITHUB LINK OF THIS DA

https://github.com/katmando/asdfghjkl/tree/master/Midterms/midterm2

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

"This assignment submission is my own, original work".
Alex Cater