CPE301 – SPRING 2019

MIDTERM 2

Student Name: Ron Joshua Recrio

Student #: 5003825419

Student Email: recrio@unlv.nevada.edu

Primary Github address: <https://github.com/recrio/submissions>

Directory: /Midterms/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**

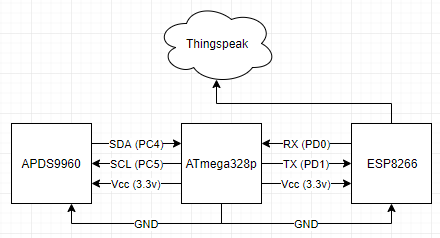
List of Components used

ATmega328p

APDS9960

ESP8266

Block diagram with pins used in the Atmega328P



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

#define *F\_CPU* 16000000UL

#define PRESCALAR 1024

#define BAUDRATE 115200

#define BAUD\_PRESCALAR (((*F\_CPU* / (BAUDRATE \* 16UL))) - 1)

#define I2C\_WRITE 0x00

#define I2C\_READ 0x01

#define APDS9960\_WRITE\_ADR (APDS9960\_I2C\_ADDR << 1) | I2C\_WRITE

#define APDS9960\_READ\_ADR (APDS9960\_I2C\_ADDR << 1) | I2C\_READ

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <stdint.h>

#include <stdlib.h>

#include <util/delay.h>

#include "i2c\_master.h"

#include "APDS9960\_def.h"

void USART\_init(void); // initializes USART settings

void USART\_sendChar(char ch); // sends a character

void USART\_sendString(char\* str); // sends a string

void AT\_init(void); // initializes AT settings

void APDS\_init(void); // initializes APDS settings

void getValues(void);

//global variables

uint16\_t cl, r, g, b; // holds clear, red, green, blue values

char sendValues[150]; // holds a string for sending values

int main(void)

{

i2c\_init(); // initialize i2c

APDS\_init(); // initialize APDS9960

USART\_init(); // initialize USART

AT\_init(); // initialize AT setttings

while (1) // Loop forever

{

getValues(); // get 0x94 to 0x9B

// convert line to string for sending

*snprintf*(sendValues, sizeof(sendValues),

"GET https://api.thingspeak.com/update?api\_key=LLCMG4KU5R9UWD9F&field1=%d&field2=%d&field3=%d&field4=%d\r\n", cl, r, g, b);

USART\_sendString("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n"); //connect to thingspeak

*\_delay\_ms*(3000);

USART\_sendString("AT+CIPSEND=150\r\n"); // send data 150 characters

*\_delay\_ms*(1000);

USART\_sendString(sendValues); // update channel using write key

*\_delay\_ms*(1000);

USART\_sendString("AT+CIPCLOSE\r\n"); // end of send

*\_delay\_ms*(5000);

}

}

void USART\_init( void )

{

UBRR0H = 0; // not needed

UBRR0L = 8; // used for 115200

UCSR0C = \_BV(UCSZ01) | \_BV(UCSZ00); /\* 8-bit data \*/

UCSR0B = \_BV(RXEN0) | \_BV(TXEN0); /\* Enable RX and TX \*/

}

void USART\_sendChar(char ch) {

while (!(UCSR0A & (1<<UDRE0))); // while data reg is not empty: hold

UDR0 = ch; // place character into reg

}

void USART\_sendString(char\* str) {

while ((\*str != '\0')) { // while not the end of the string

while (!(UCSR0A & (1<<UDRE0))); // while data reg is not empty: hold

USART\_sendChar(\*str); //take in character to reg

str++; // next character

}

}

void AT\_init(void) {

USART\_sendString("AT\r\n"); // Sends AT, expect OK

*\_delay\_ms*(1000);

USART\_sendString("AT+CWMODE=1\r\n"); // Sends mode set to station, expect OK

*\_delay\_ms*(1000);

USART\_sendString("AT+CWDCHP=1,1\r\n");

*\_delay\_ms*(1000);

USART\_sendString("AT+CWJAP=\"ATwifi\",\"ATpassword\"\r\n"); // Send command to join guest wifi, expect OK

*\_delay\_ms*(8000);

}

void getValues(void) {

i2c\_start(APDS9960\_WRITE\_ADR); // start writing

i2c\_write(APDS9960\_CDATAL); // point to CDATAL

i2c\_stop(); // stop

i2c\_start(APDS9960\_READ\_ADR); // read this time

//read all the values from CDATAL to BDATAH

cl =((int)i2c\_read\_ack()|((int)i2c\_read\_ack()<<8));

r = ((int)i2c\_read\_ack()|((int)i2c\_read\_ack()<<8));

g = ((int)i2c\_read\_ack()|((int)i2c\_read\_ack()<<8));

b = ((int)i2c\_read\_ack()|((int)i2c\_read\_ack()<<8));

i2c\_stop(); // stop

}

void APDS\_init(void) {

uint8\_t data; // holds configuration bits

// read device ID to see if it matches APDS9960

i2c\_readReg(APDS9960\_WRITE\_ADR | I2C\_WRITE, APDS9960\_ID, &data, 1);

if (data != APDS9960\_ID\_1) while(1); // if it does not match, loop forever

//Turn on Power and Enable from ENABLE register

data = APDS9960\_PON | APDS9960\_AEN;

i2c\_writeReg(APDS9960\_WRITE\_ADR, APDS9960\_ENABLE, &data, 1);

//Set Gain Control to default value

data = DEFAULT\_AGAIN; // ALS Gain 4X

i2c\_writeReg(APDS9960\_WRITE\_ADR, APDS9960\_CONTROL, &data, 1);

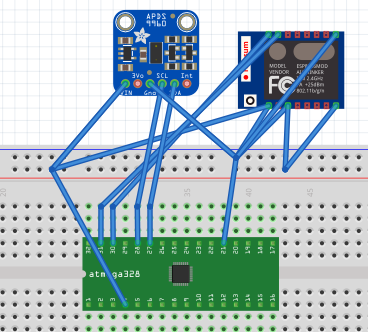
//Set ALS Time to default

data = DEFAULT\_ATIME; // default value = 219

i2c\_writeReg(APDS9960\_WRITE\_ADR, APDS9960\_ATIME, &data, 1);

}

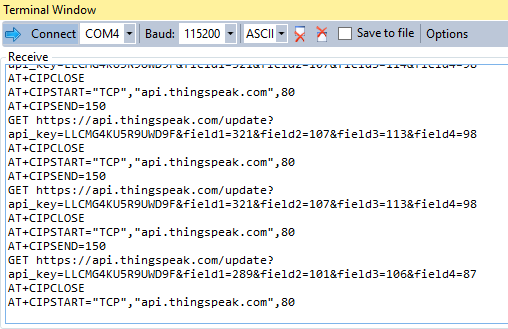
1. **SCHEMATICS**



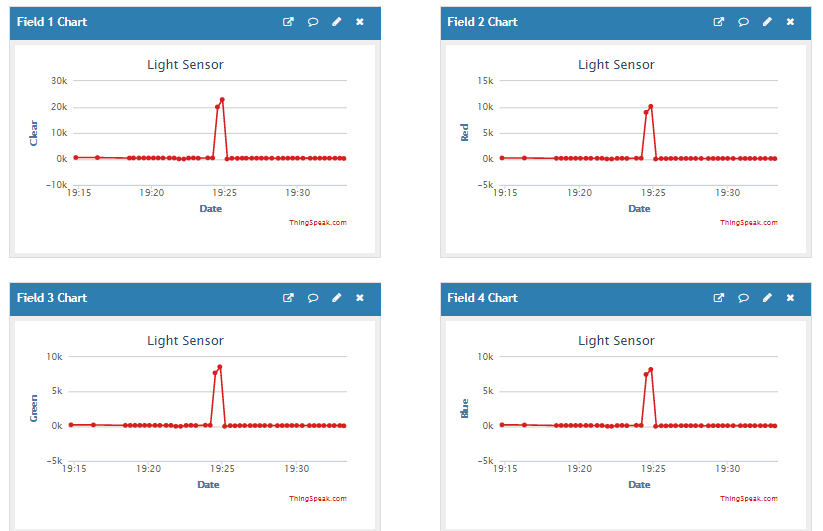
\*Not exact models

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

Terminal Output:

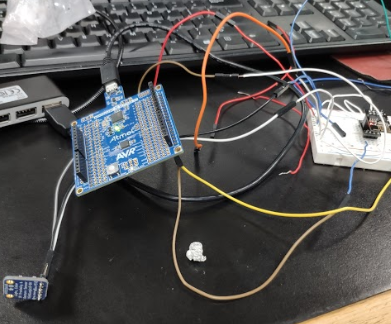


Thingspeak Charts (Flashing a light directly into it):



Thingspeak Charts (Covering it with my hand):

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



1. **VIDEO LINKS OF EACH DEMO**

https://youtu.be/rXC6fDnmPFI

1. **GITHUB LINK OF THIS DA**

<https://github.com/recrio/submissions/tree/master/Midterms/Midterm2>

**Student Academic Misconduct Policy**

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

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

Ron Joshua Recrio