CPE301 – SPRING 2019

MIDTERM II

Student Name: Prachi Patel

Student #: 5002380222

Student Email: patelp3@unlv.nevada.edu

Primary Github address: <https://github.com/prachi173>

Directory: <https://github.com/prachi173/da_sp18>

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 XPlained Mini

APDS 9960

ESP-01

A picture containing text, map

Description automatically generated

1. **DEVELOPED MODIFIED CODE OF TASK**

#ifndef F\_CPU

#define F\_CPU 16000000UL

#endif

#define BAUD 9600

#define MYUBRR F\_CPU/16/BAUD-1

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <math.h>

#include <stdlib.h>

#include <stdio.h>

// include i2c, uart and APDS9960 library files

// APDS9960 library file was edited to add the following

//#define APDS\_WRITE (0x39 << 1) | 0

//#define APDS\_READ (0x39 << 1) | 1

#include "i2c\_master.h"

#include "i2c\_master.c"

#include "uart.h"

#include "uart.c"

#include "APDS9960\_def.h"

void color\_rgb();

void APDS9960\_init();

*uint8\_t* baud\_rate = MYUBRR;

*uint8\_t* RL, RH;

*uint8\_t* BL, BH;

*uint8\_t* GL, GH;

*uint8\_t* config;

char outs[255];

int main(void){

char buffer[20], float\_[10];

*uint8\_t* red = 0;

*uint8\_t* green = 0;

*uint8\_t* blue = 0;

init\_uart(9600); //initialize UART

*\_delay\_ms*(500);

USART\_SendString("We are live!"); //message to confirm uart is working

*\_delay\_ms*(5000);

i2c\_init(); //initialize i2c

APDS9960\_init(); //initialize sensor

unsigned char AT\_CHECK[] = "AT\r\n"; //AT command working check

unsigned char CWMODE\_SET[] = "AT+CWMODE=1\r\n"; //Mode set to 1

unsigned char CWJAP\_LOGIN[] = "AT+CWJAP=\"mywifi\",\"mypass\"\r\n"; //login personal wifi

unsigned char CIPMUX\_SET[] = "AT+CIPMUX=0\r\n"; // CIPMUX set to 0

unsigned char CIPSTART\_SEND[] = "AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n"; //connect to thingspeak

unsigned char CIPSEND\_FINISH[] = "AT+CIPSEND=100\r\n"; //end communication

//send strings to print and delays added to establish proper communication

*\_delay\_ms*(3000);

USART\_SendString(AT\_CHECK);

*\_delay\_ms*(10000);

USART\_SendString(CWMODE\_SET);

*\_delay\_ms*(10000);

USART\_SendString(CWJAP\_LOGIN);

*\_delay\_ms*(20000);

while(1){

//send strings to print and delays added to establish proper communication

USART\_SendString(CIPMUX\_SET);

*\_delay\_ms*(100000);

USART\_SendString(CIPSTART\_SEND);

*\_delay\_ms*(10000);

USART\_SendString(CIPSEND\_FINISH);

*\_delay\_ms*(10000);

color\_rgb(&red, &blue, &green);

//print to update field1 with red, field2 with green and field3 with blue

*snprintf*(outs,sizeof(outs),"GET <https://api.thingspeak.com/update?api_key=7JVWL1NQ7916EIRM&field1=0%05u&field2=%05u&field3=%05u\r\n>", red,green,blue);

USART\_SendString(outs);

*\_delay\_ms*(10000);

}

return 0;

}

void color\_rgb(*uint8\_t* \*red, *uint8\_t* \*blue, *uint8\_t* \*green){

//read colors

i2c\_readReg(APDS\_WRITE, APDS9960\_RDATAL, &RL, 1);

i2c\_readReg(APDS\_WRITE, APDS9960\_RDATAH, &RH, 1);

i2c\_readReg(APDS\_WRITE, APDS9960\_GDATAL, &GL, 1);

i2c\_readReg(APDS\_WRITE, APDS9960\_GDATAH, &GH, 1);

i2c\_readReg(APDS\_WRITE, APDS9960\_BDATAL, &BL, 1);

i2c\_readReg(APDS\_WRITE, APDS9960\_BDATAH, &BH, 1);

\*red = RH << 8 | RL;

\*blue = BH << 8 | BL;

\*green = GH << 8 | GL;

}

void APDS9960\_init(){

//reading and writing colors to register

i2c\_readReg(APDS\_WRITE, APDS9960\_ID, &config,1);

if(config != APDS9960\_ID\_1)

while(1)

{

config = 1 << 1 | 1 << 0 | 1 << 3 | 1 << 4;

}

i2c\_writeReg(APDS\_WRITE, APDS9960\_ENABLE, &config, 1);

config = DEFAULT\_ATIME;

i2c\_writeReg(APDS\_WRITE, APDS9960\_ATIME, &config, 1);

config = DEFAULT\_WTIME;

i2c\_writeReg(APDS\_WRITE, APDS9960\_WTIME, &config, 1);

config = DEFAULT\_PROX\_PPULSE;

i2c\_writeReg(APDS\_WRITE, APDS9960\_PPULSE, &config, 1);

config = DEFAULT\_POFFSET\_UR;

i2c\_writeReg(APDS\_WRITE, APDS9960\_POFFSET\_UR, &config, 1);

config = DEFAULT\_POFFSET\_DL;

i2c\_writeReg(APDS\_WRITE, APDS9960\_POFFSET\_DL, &config, 1);

config = DEFAULT\_CONFIG1;

i2c\_writeReg(APDS\_WRITE, APDS9960\_CONFIG1, &config, 1);

config = DEFAULT\_PERS;

i2c\_writeReg(APDS\_WRITE, APDS9960\_PERS, &config, 1);

config = DEFAULT\_CONFIG2;

i2c\_writeReg(APDS\_WRITE, APDS9960\_CONFIG2, &config, 1);

config = DEFAULT\_CONFIG3;

i2c\_writeReg(APDS\_WRITE, APDS9960\_CONFIG3, &config, 1);

}

void init\_uart(*uint16\_t* baudrate){

//initialize uart

*uint16\_t* UBRR\_val = (F\_CPU/16)/(baudrate-1);

UBRR0H = UBRR\_val >> 8;

UBRR0L = UBRR\_val;

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

UCSR0C |= (1<<USBS0) | (3<<UCSZ00);

}

void uart\_outchar(unsigned char c){

//output uart data

while(!(UCSR0A & (1<<UDRE0))); // wait until sending is possible

UDR0 = c; // output character saved in c

}

void uart\_outstring(char \*s){

while(\*s){

uart\_outchar(\*s);

s++;

}

}

1. **SCHEMATICS**

A circuit board

Description automatically generated

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

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

For all 10 feeds

A screenshot of a computer screen

Description automatically generated

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

A circuit board

Description automatically generated

1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/HJeCSGr0u58>

1. **GITHUB LINK OF THIS DA**

<https://github.com/prachi173/da_sp18/tree/master/Midterms/MIDTERM2>

**Student Academic Misconduct Policy**

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

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

PRACHI PATEL