CPE301 – FALL 2019

Midterm 2

Student Name: Henry Mesa

Student #: 1008099273

Student Email: mesah1@unlv.nevada.edu

Primary Github address: https://github.com/mesah1

Directory: https://github.com/mesah1/submissions

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

List of Components used:

ATmega328PB

ESP8366

APDS-9960

Block diagram with pins used in the Atmega328P

A screenshot of a cell phone

Description automatically generated

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

/\*

\* Midterm2.c

\*

\* Created: 12/12/2019 2:15:01 AM

\* Author: Henry M

\*/

#define F\_CPU 16000000UL

# include <avr/io.h>

# include <stdlib.h>

# include <stdio.h>

# include <util/delay.h>

# include <math.h>

# include "APDS9960\_def.h"

# include "i2c\_master.h"

#define BAUD 115200

#define FOSC 16000000

#define UBRR FOSC/8/BAUD-1

#define APDS9960\_WRITE 0x72

#define APDS9960\_READ 0x73

//Functions

void getValues(void);

// void TIMER1\_init();

void init\_APDS9960(void);

void usart\_init();

void USART\_putstring(volatile unsigned char\* StringPtr);

//AT commands

volatile unsigned char AT[] = "AT\r\n"; // Test

volatile unsigned char CWMODE[] = "AT+CWMODE=3\r\n"; // Wi-Fi mode

volatile unsigned char CWJAP[] = "AT+CWJAP=\”WIFI\_ID\”,\”PASSWORD\”\r\n”; //Wi-Fi info

volatile unsigned char CIPSTART[] = "AT+CIPSTART=\"TCP\",\"184.106.153.149\",80\r\n"; // Connect to ThingSpeak.com

volatile unsigned char CIPSEND[] = "AT+CIPSEND=99\r\n”;

volatile unsigned char CIPMUX[] = "AT+CIPMUX=0\r\n"; // Enable Connection

volatile unsigned char SEND\_DATA[] = "GET /update?key=3JJZ3QRGR01RKJMQ="; // Write To

volatile unsigned char RESET[] = "AT+RST\r\n"; // AT Firmware info

volatile unsigned char LINEBREAK[] = "\r\n";

volatile unsigned char CLOSE[] = "AT+CIPCLOSE\r\n";// End Transmission

volatile unsigned char RedStr[20];

volatile unsigned char GreenStr[20];

volatile unsigned char BlueStr[20];

uint16\_t green, blue, red;

int main(void)

{

i2c\_init();

usart\_init(115200);

init\_APDS9960();

//Start communication

\_delay\_ms(10);

//send AT to the USART

USART\_putstring(AT);

//connect to WIFI

\_delay\_ms(10);

//reset ESP

USART\_putstring(RESET);

\_delay\_ms(10);

//confirm communication

USART\_putstring(AT);

\_delay\_ms(10);

//WIFI mode = 3

USART\_putstring(CWMODE);

\_delay\_ms(10);

//Login

USART\_putstring(CWJAP);

while (1)

{

getValues();

\_delay\_ms(20);

USART\_putstring(CIPMUX);

\_delay\_ms(10);

USART\_putstring(CIPSTART); // Connect to ThingSpeak

\_delay\_ms(10);

USART\_putstring(CIPSEND); // Declare send length

\_delay\_ms(10);

//AT Commands

USART\_putstring(SEND\_DATA);

USART\_putstring("&field1=");

USART\_putstring(RedStr);

USART\_putstring("&field2=");

USART\_putstring(GreenStr);

USART\_putstring("&field3=");

USART\_putstring(BlueStr);

\_delay\_ms(500);

}

return 0;

}

void init\_APDS9960(void)

{

uint8\_t setup;

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

if (setup != APDS9960\_ID\_1) while (1) ;

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

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

setup = DEFAULT\_ATIME;

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

setup = DEFAULT\_WTIME;

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

setup = DEFAULT\_PROX\_PPULSE;

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

setup = DEFAULT\_POFFSET\_UR;

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

setup = DEFAULT\_POFFSET\_DL;

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

setup = DEFAULT\_CONFIG1;

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

setup = DEFAULT\_PERS;

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

setup = DEFAULT\_CONFIG2;

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

setup = DEFAULT\_CONFIG3;

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

}

void getValues(void)

{

uint8\_t redH, redL;

uint8\_t greenH, greenL;

uint8\_t blueH, blueL;

unsigned char i;

char m[20];

// Read red value

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

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

// Read green value

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

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

// Read blue value

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;

// Set max threshold values

if (red > 255)

{

red = 255;

}

if (green > 255)

{

green = 255;

}

if (blue > 255)

{

blue = 255;

}

itoa(red, m, 20);

for (i = 0; i < 20; i++)

{

RedStr[i] = m[i];

}

itoa(green, m, 20);

for (i = 0; i < 20; i++)

{

GreenStr[i] = m[i];

}

itoa(blue, m, 20);

for (i = 0; i < 20; i++)

{

BlueStr[i] = m[i];

}

}

void usart\_init()

{

UBRR0H = ((UBRR) >> 8);

UBRR0L = UBRR;

UCSR0A |= (1 << U2X0); // div = 8

UCSR0B |= (1 << TXEN0); // Enable transmission

UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00); // 8 bits

}

void USART\_putstring(volatile unsigned char\* StringPtr)

{

while ((\*StringPtr != '\0'))

{

while (!(UCSR0A & (1 << UDRE0))) ;

UDR0 = \*StringPtr;

StringPtr++;

}

}

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

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

A close up of a device

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/zJhFLSEhODI>

1. **GITHUB LINK OF THIS DA**

<https://github.com/mesah1/submissions/tree/master/Midterm2>

**Student Academic Misconduct Policy**

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

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

Henry Mesa