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:

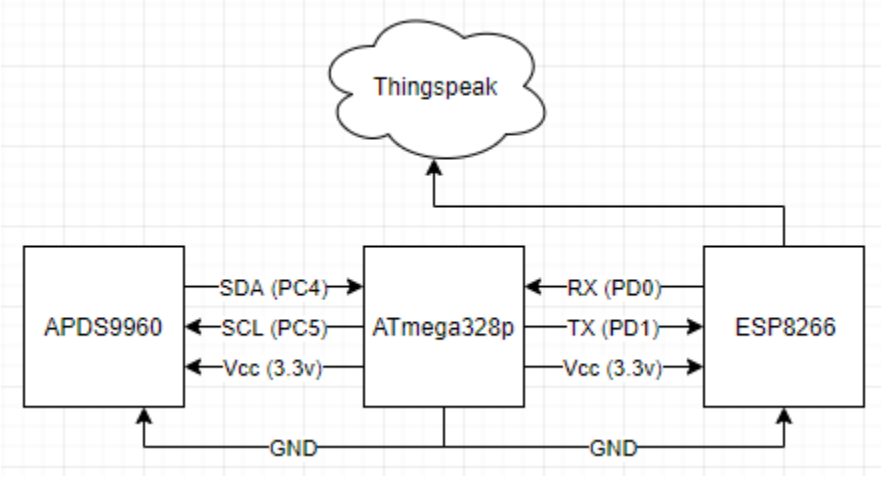
1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder 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



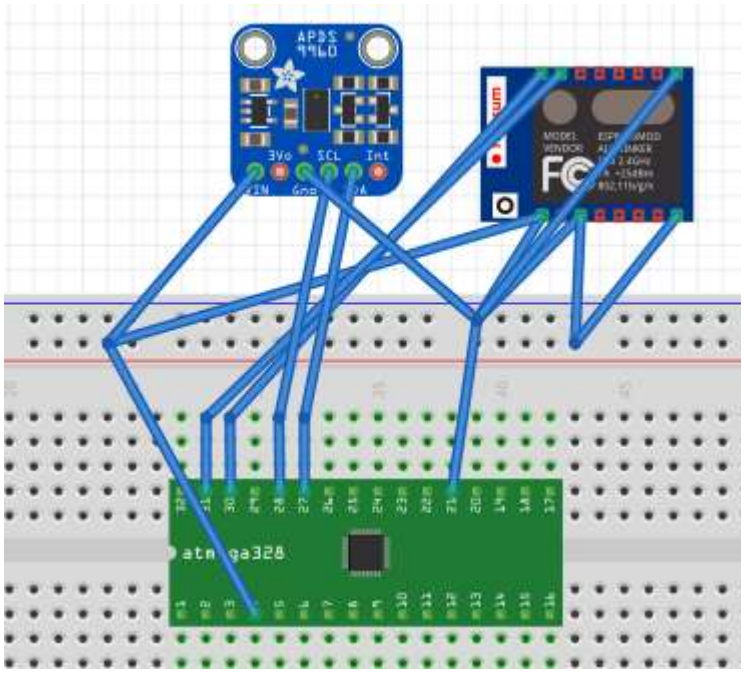
2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

```
#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);

}
```

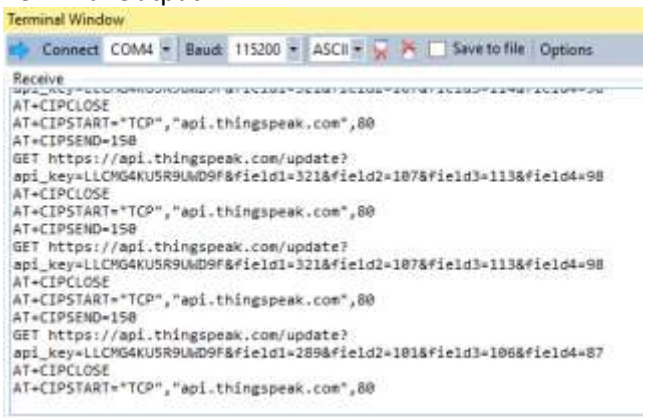
3. SCHEMATICS



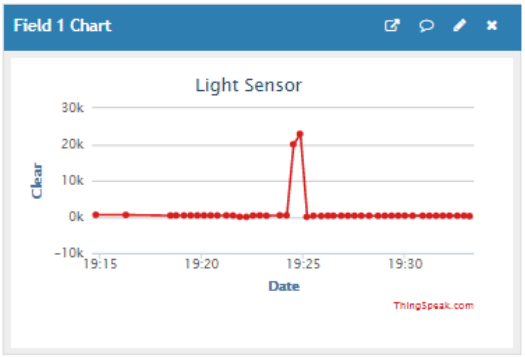
*Not exact models

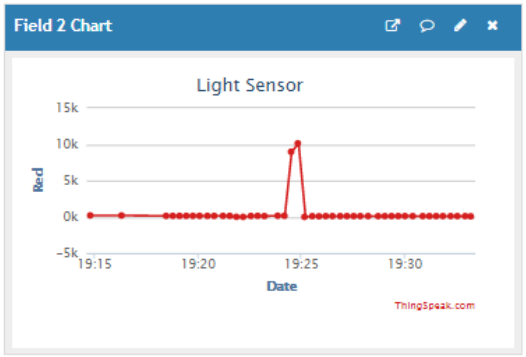
4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

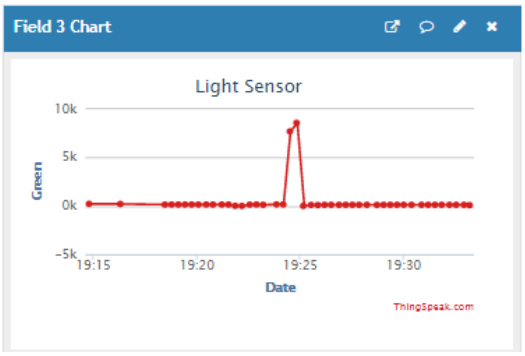
Terminal Output:

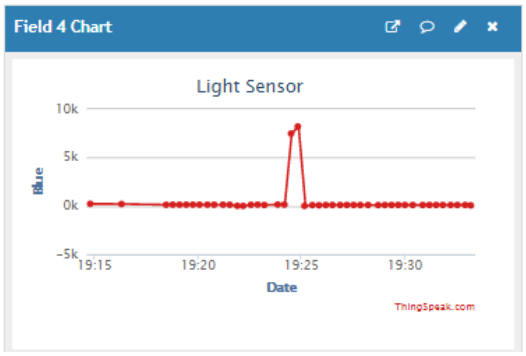


Thingspeak Charts (Flashing a light directly into it):

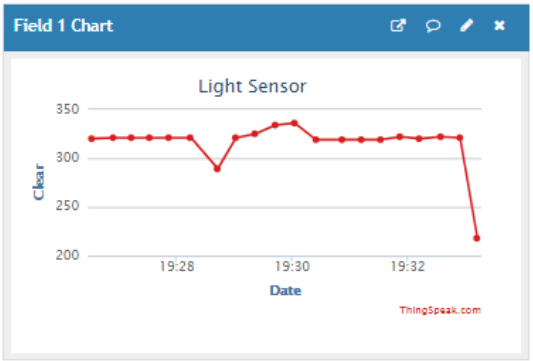


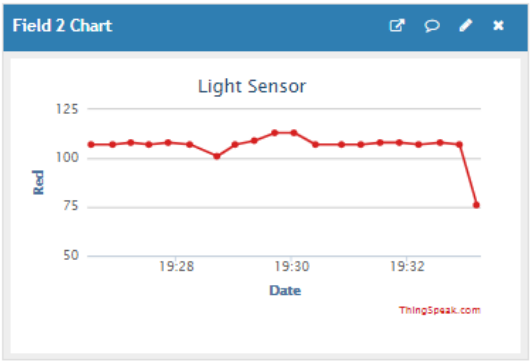


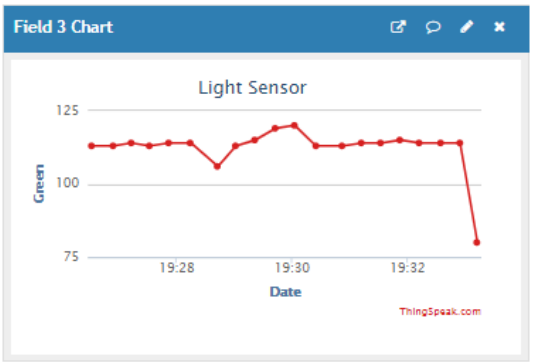


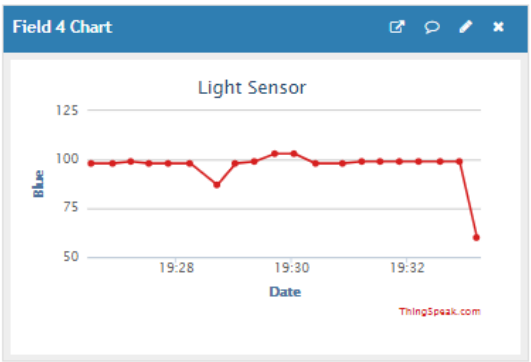


Thingspeak Charts (Covering it with my hand):

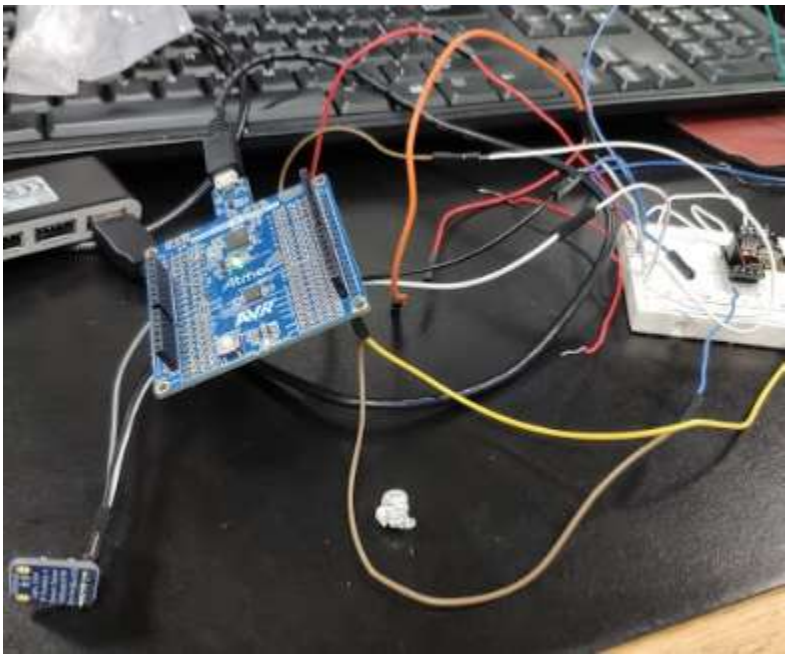








5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



6. VIDEO LINKS OF EACH DEMO

https://youtu.be/rXC6fDnmPFI

7. 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