# CPE301 – SPRING 2019

# Midterm 2

Student Name: Rimon Sawa

Student #: 5001529281

Student Email: sawar1@unlv.nevada.edu

Primary Github address: <https://github.com/sawar1/UNLV301>

Directory: Midterm 2

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

List of Components used

* Breadboard
* Atmega328P
* Male wires
* Xplained Mini
* Esp32
* ADPS 9960

# 2. DEVELOPED C CODE

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include "i2c\_master.h"

#include "uart.h"

#include "apds.h"

*FILE* str\_uart = *FDEV\_SETUP\_STREAM*(uart\_putchar, *NULL* , *\_FDEV\_SETUP\_WRITE*); char results[256];

int main(void)

{

*uint16\_t* red = 0, green = 0, blue = 0; //set up variables

i2c\_init(); //call i2c init init\_UART(); //call uart init *stdout* = &str\_uart;

apds\_init(); //adps initialization function

*\_delay\_ms*(2000);

*printf*("AT\r\n");//at commands

*\_delay\_ms*(5000);

*printf*("AT+CWMODE=1\r\n"); //at commands

*\_delay\_ms*(5000);

*printf*("AT+CWJAP=\"XXXXX\",\"XXXXX\"\r\n");//at commands

while (1)

{

*\_delay\_ms*(5000);

*printf*("AT+CIPMUX=0\r\n");//atcommands

*\_delay\_ms*(5000);

*printf*("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n");//at commands

*\_delay\_ms*(5000);

readColor(&red, &green, &blue); //reads color *printf*("AT+CIPSEND=104\r\n"); //at command

*printf*("GET

https://api.thingspeak.com/update?api\_key=1V8WUUJNEHZGA9L7&field1=%05u&field2=%05u&field3 =%05u\r\n", red, green, blue); //send to thinkspeak

*\_delay\_ms*(3000);

} }

void init\_UART(void){

//Set baud rate *uint16\_t* baud\_rate = BRGVAL; UBRR0H = baud\_rate >> 8;

UBRR0L = baud\_rate & 0xFF;

//Enable receiver and transmitter

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

// Set frame format: 8data, 1stop bit

UCSR0C = (3 <<UCSZ00);

}

int uart\_putchar(char c, *FILE* \*stream){ //wait until buffer empty while ( !( UCSR0A & ( 1 <<UDRE0)) );

//Put data into buffer UDR0 = c; return 0;

}

# UART.h

#ifndef UART\_328P\_H

#define UART\_328P\_H

#ifndef F\_CPU

#define F\_CPU 16000000UL

#endif

#include <stdio.h>

#include <avr/io.h>

#include <avr/interrupt.h> #define BAUD 9600

#define BRGVAL (F\_CPU/16/BAUD) - 1

void init\_UART();

int uart\_putchar( char c, *FILE* \*stream);

#endif APDS.C

#include <avr/io.h>

#include "i2c\_master.h"

#include "apds.h"

void apds\_init(){

*uint8\_t* setup;

//read and write commands

i2c\_readReg(APDS\_WRITE, APDS9960\_ID, &setup,1); if(setup != APDS9960\_ID\_1) while(1); setup = 1 << 1 | 1<<0 | 1<<3 | 1<<4; i2c\_writeReg(APDS\_WRITE, APDS9960\_ENABLE, &setup, 1); setup = DEFAULT\_ATIME;

i2c\_writeReg(APDS\_WRITE, APDS9960\_ATIME, &setup, 1); setup = DEFAULT\_WTIME;

i2c\_writeReg(APDS\_WRITE, APDS9960\_WTIME, &setup, 1); setup = DEFAULT\_PROX\_PPULSE;

i2c\_writeReg(APDS\_WRITE, APDS9960\_PPULSE, &setup, 1); setup = DEFAULT\_POFFSET\_UR;

i2c\_writeReg(APDS\_WRITE, APDS9960\_POFFSET\_UR, &setup, 1); setup = DEFAULT\_POFFSET\_DL;

i2c\_writeReg(APDS\_WRITE, APDS9960\_POFFSET\_DL, &setup, 1); setup = DEFAULT\_CONFIG1;

i2c\_writeReg(APDS\_WRITE, APDS9960\_CONFIG1, &setup, 1); setup = DEFAULT\_PERS; i2c\_writeReg(APDS\_WRITE, APDS9960\_PERS, &setup, 1); setup = DEFAULT\_CONFIG2; i2c\_writeReg(APDS\_WRITE, APDS9960\_CONFIG2, &setup, 1); setup = DEFAULT\_CONFIG3; i2c\_writeReg(APDS\_WRITE, APDS9960\_CONFIG3, &setup, 1);

}

void readColor(*uint16\_t* \*red, *uint16\_t* \*green, *uint16\_t* \*blue){ *uint8\_t* redl, redh; //declare these variables

*uint8\_t* greenl, greenh; *uint8\_t* bluel, blueh; //read i2c vlaues

i2c\_readReg(APDS\_WRITE, APDS9960\_RDATAL, &redl, 1); i2c\_readReg(APDS\_WRITE, APDS9960\_RDATAH, &redh, 1); i2c\_readReg(APDS\_WRITE, APDS9960\_GDATAL, &greenl, 1); i2c\_readReg(APDS\_WRITE, APDS9960\_GDATAH, &greenh, 1); i2c\_readReg(APDS\_WRITE, APDS9960\_BDATAL, &bluel, 1); i2c\_readReg(APDS\_WRITE, APDS9960\_BDATAH, &blueh, 1);

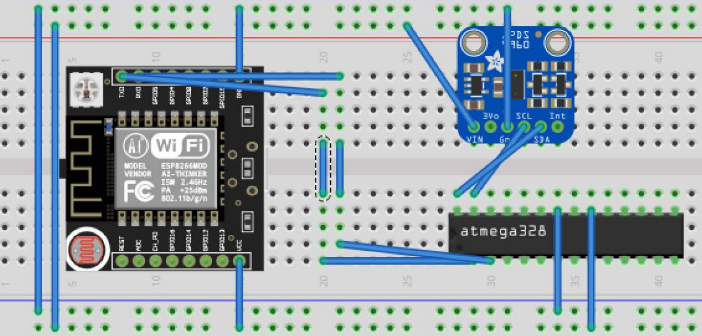
\*red = redh << 8 | redl;

\*green = greenh << 8 | greenl;

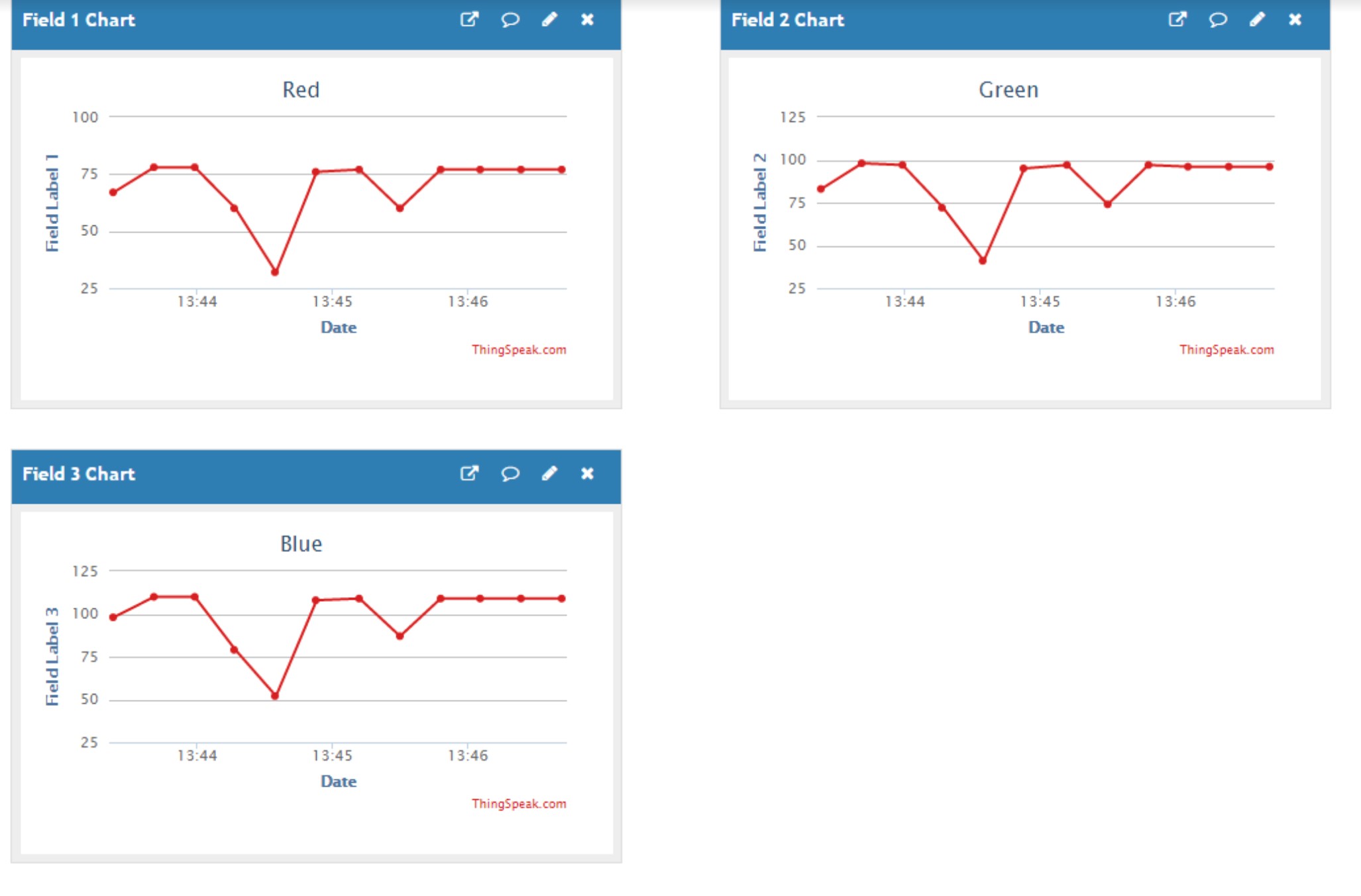
\*blue = blueh << 8 | bluel;

}

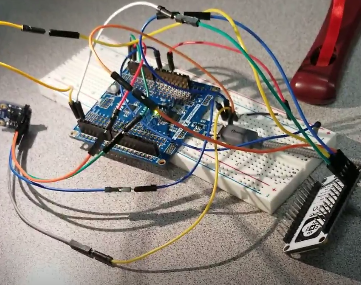
## 3. SCHEMATICS



## 4. SCREENSHOTS OF EACH TASK OUTPUT



## 5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



**6. VIDEO LINKS OF EACH DEMO**

<https://youtu.be/3O47TrK1IFs>

## 7. GITHUB LINK OF THIS DA

<https://github.com/sawar1/UNLV301>

**Student Academic Misconduct Policy**

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

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