

Midterm 2

Student Name: Itzel Becerril

Student #: 2000478001

Student Email: becerri2@unlv.nevada.edu

Primary Github address: hadidbuilds

Directory: Final

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

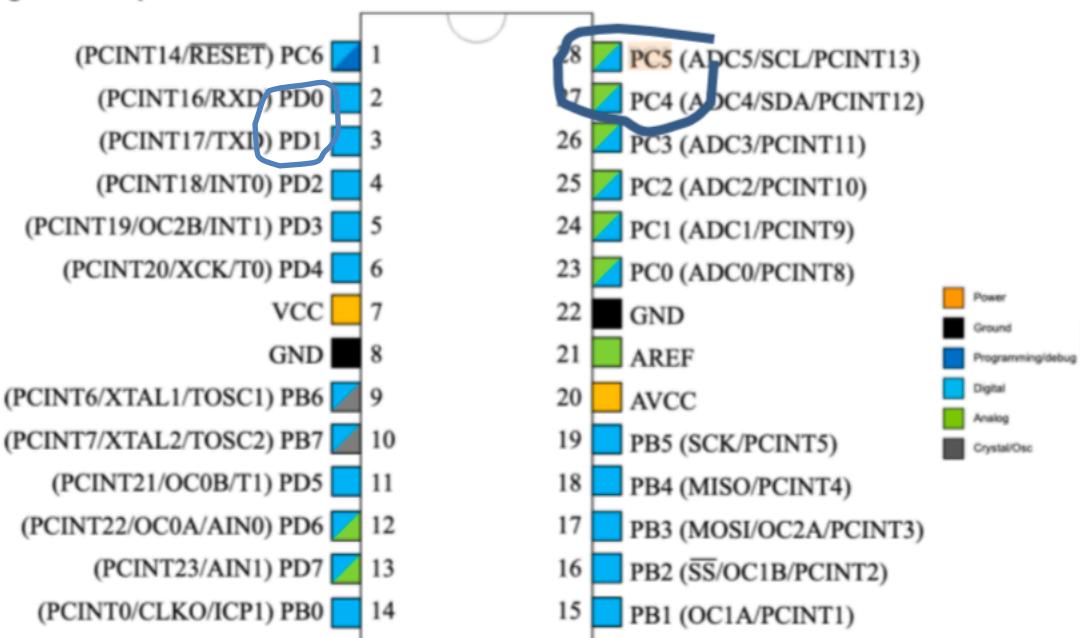
List of Components used

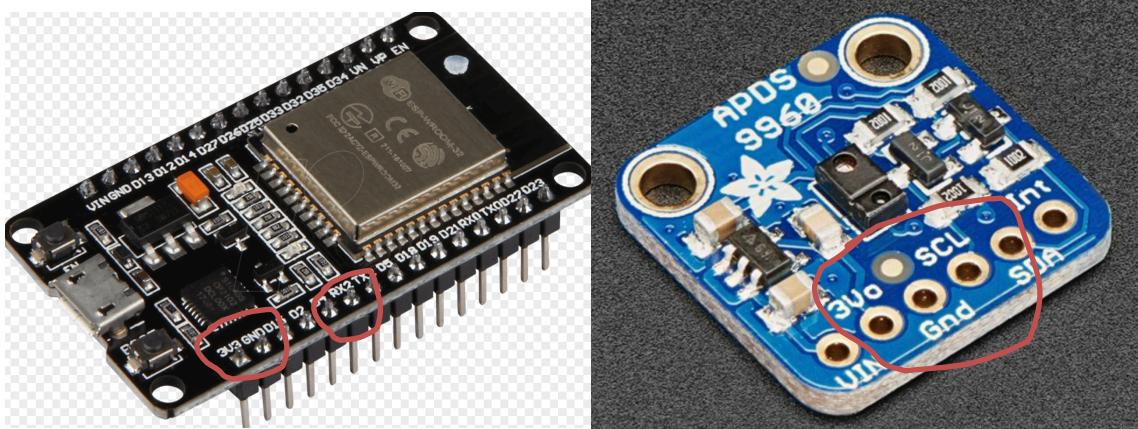
- Breadboard
- 2 MicroUSB
- Atmega328P
- Male wires
- Xplained Mini
- Esp32
- ADPS 9960
- Atmel studio
- Thinkspeak

Block diagram with pins used in the Atmega328P

5.1. Pin-out

Figure 5-1. 28-pin PDIP





2. DEVELOPED C CODE

Final.c

```

#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(); //apds 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"); //at commands

        _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

```

```

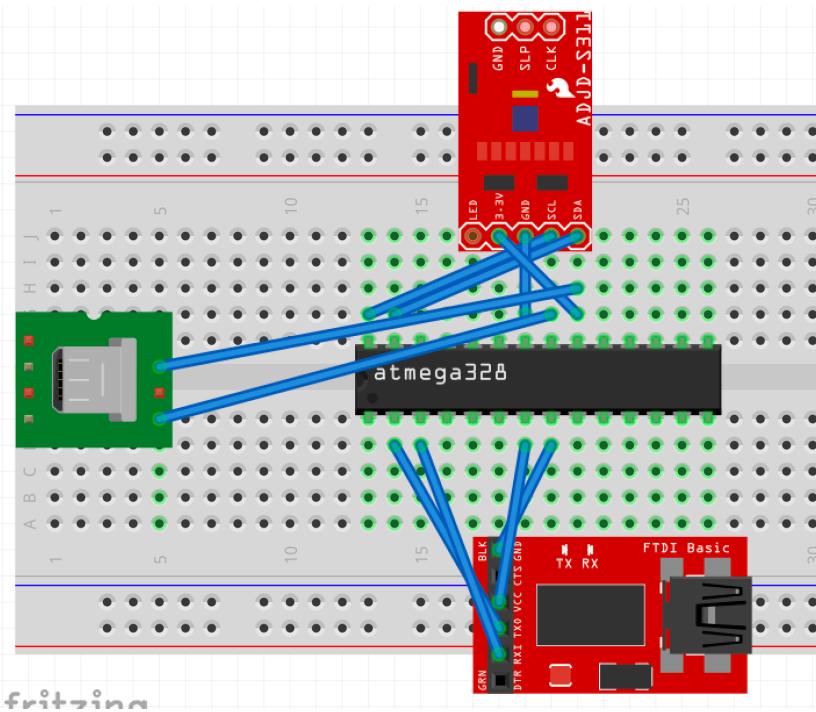
i2c_readReg(APDS_WRITE, APDS9960_ID, &setup,1);
if(setup != APDS9960_ID_1) while(1);
setup = 1 << 1 | 1<<0 | 1<<3 | 1<<4;
i2c_writeReg(APDS_WRITE, APDS9960_ENABLE, &setup, 1);
setup = DEFAULT_ATIME;
i2c_writeReg(APDS_WRITE, APDS9960_ATIME, &setup, 1);
setup = DEFAULT_WTIME;
i2c_writeReg(APDS_WRITE, APDS9960_WTIME, &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;
}

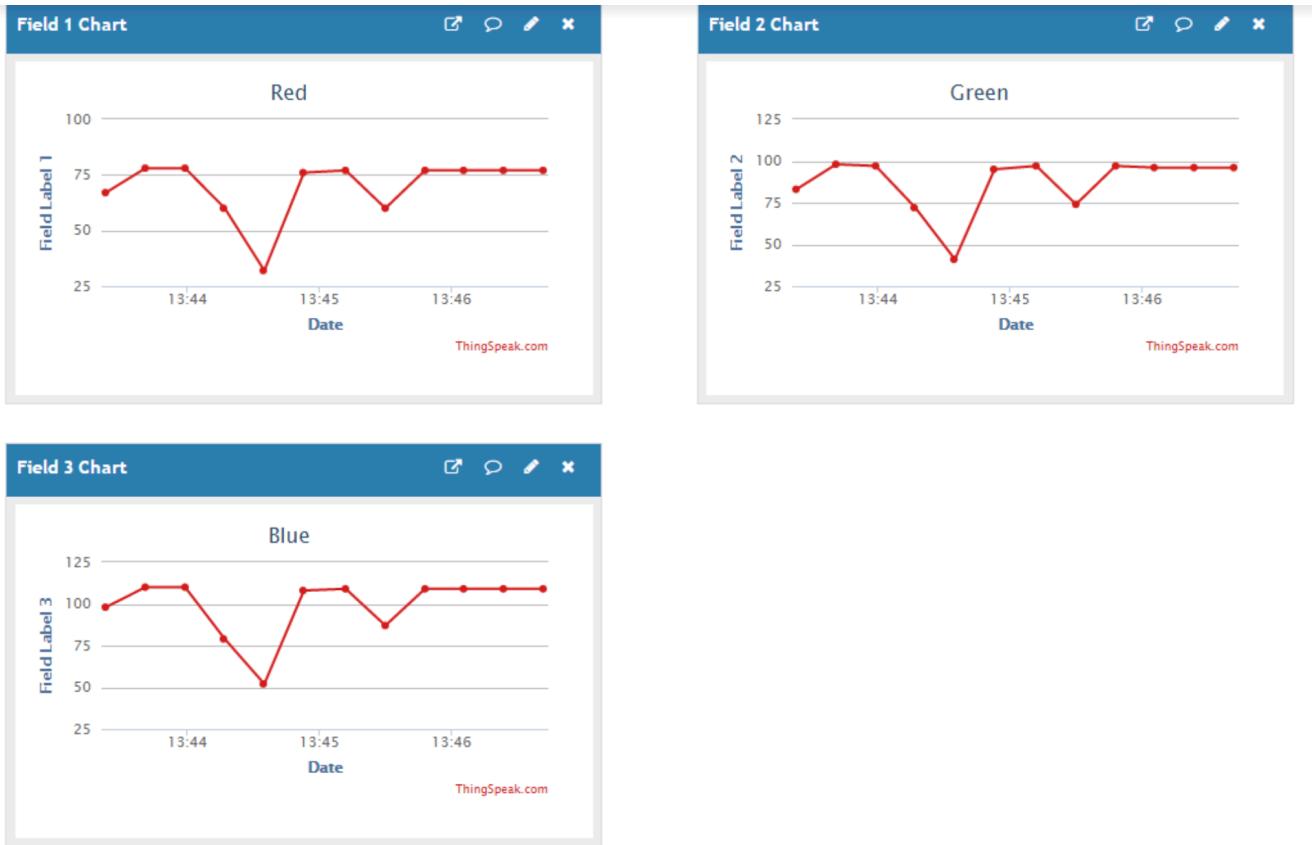
```

Other files are in my Github

3. SCHEMATICS



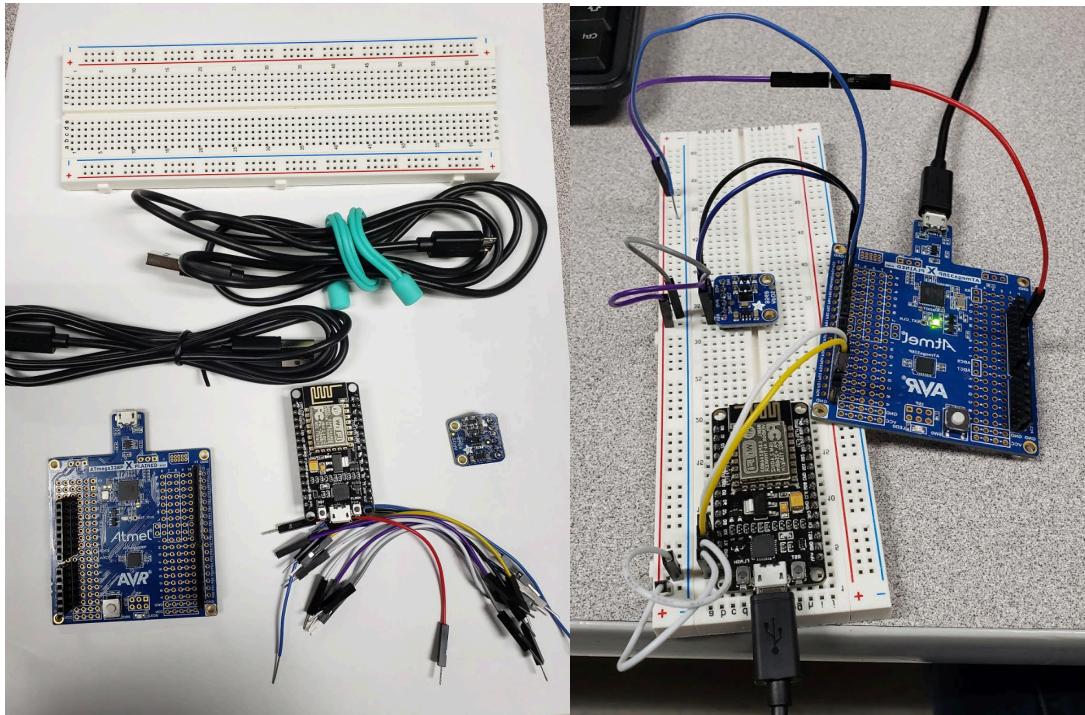
4. SCREENSHOTS OF EACH TASK OUTPUT



5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Before Setup:

After Setup:



6. VIDEO LINKS OF EACH DEMO

<https://www.youtube.com/watch?v=OYH3hHSesLE>

7. GITHUB LINK OF THIS DA

https://github.com/HadidBuilds/hw_sub_da1

Student Academic Misconduct Policy

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

"This assignment submission is my own, original work".

Itzel Becerril