CPE301 – SPRING 2019

MIDTERM 2

Student Name: Victor Martinez

Student #: 5003131674

Student Email: [martiv6@unlv.nevada.edu](mailto:martiv6@unlv.nevada.edu)

Primary Github address: <https://github.com/martiv6/>

Directory: <https://github.com/martiv6/>midterms/Midterm\_2

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**

ESP-01

APDS

9960

BREAD

BOARD

328P

Mini

POWERSUPPLY

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

/\*

\* MIDTERM\_2.c

\*

\* Created: 5/01/2019 10:06:39 PM

\* Author : victor

\*/

#define *F\_CPU* 16000000UL

#define BAUD 9600

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

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#include <avr/interrupt.h>

#include <stdlib.h>

#include "I2C\_MASTER\_C.h"

#include "APDS\_9960\_LIB.h"

void USART\_tx\_string(char \*data); // where it prints string USART

void USART\_init( unsigned int ubrr ); // where USART initializes

void RGB\_dect(); // used to read the colors and put their values

void APDS\_9960\_init(); // begins to use APDS sensor

*uint16\_t* baud\_rate = MYUBRR; // Int declaration

*uint8\_t* red\_low, red\_high; //

*uint8\_t* blue\_low, blue\_high; //

*uint8\_t* green\_low, green\_high; //

*uint8\_t* config; //

char outs[256]; // room we have to print

int main(void)

{

*uint16\_t* red = 0; // Declare red

*uint16\_t* green = 0; // Declare green

*uint16\_t* blue = 0; // Declare blue

i2c\_init(); // i2C function call

USART\_init(MYUBRR); // uart initialization function call

APDS\_9960\_init(); // initialize 9960 sensor function call

unsigned char AT\_CHECK[] = "AT\r\n"; // AT Commands

unsigned char CWMODE\_SET[] = "AT+CWMODE=1\r\n"; // Set MODE

unsigned char CWJAP\_LOGIN[] = "AT+CWJAP=\"iPXSMax\",\"12345678\"\r\n"; // MUST CHANGE WIFI AND PASSWORD

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

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

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

*\_delay\_ms*(2500);

USART\_tx\_string(AT\_CHECK); //send commands

*\_delay\_ms*(6000);

USART\_tx\_string(CWMODE\_SET); //set mode

*\_delay\_ms*(6000);

USART\_tx\_string(CWJAP\_LOGIN); //connect to Wifi

*\_delay\_ms*(16000);

while (1)

{

///////////////////////// SENDING RED, GREEN, BLUE READINGS/////////////////////////

USART\_tx\_string(CIPMUX\_SET); //select MUX

*\_delay\_ms*(10000);

USART\_tx\_string(CIPSTART\_SEND); //connect TCP

*\_delay\_ms*(10000);

USART\_tx\_string(CIPSEND\_FINISH); //send size

*\_delay\_ms*(6000);

RGB\_dect(&red, &blue, &green); // Call colors function

*snprintf*(outs,sizeof(outs),"GET https://api.thingspeak.com/update?api\_key=5HICIFGYJQ1XUU90&field1=0%05u&field2=%05u&field3=%05u\r\n", red,green,blue); // print it

USART\_tx\_string(outs); //send data

*\_delay\_ms*(10000);

}

}

/\* INIT USART (RS-232) \*/

void USART\_init( unsigned int ubrr ) {

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

UCSR0B |= (1 << TXEN0) | (1 << RXEN0)| ( 1 << RXCIE0); // Enable receiver, transmitter & RX interrupt

UCSR0C |= (1<<UCSZ01) | (1 << UCSZ00);

}

void USART\_tx\_string( char \*data ) // used to print on to the screen

{

while ((\*data != '\0')) {

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

UDR0 = \*data;

data++;

}

}

void RGB\_dect(*uint16\_t* \*red, *uint16\_t* \*blue, *uint16\_t* \*green) // Function used to read the colors for sensor (red, blue and green)

{

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

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

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

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

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

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

\*red = red\_high << 8 | red\_low;

\*blue = blue\_high << 8 | blue\_low;

\*green = green\_high << 8 | green\_low;

}

void APDS\_9960\_init() // Function used to initialize the Sensor only for the RGB, no gesture or proximity used

{

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;

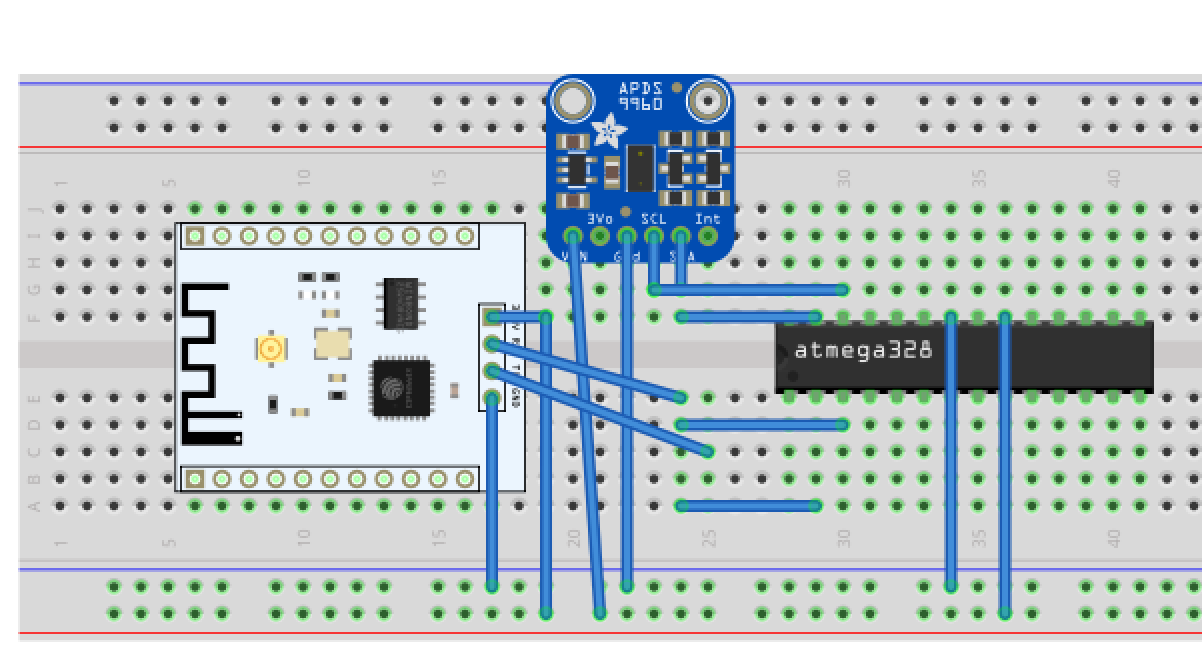
i2c\_writeReg(APDS\_WRITE, APDS9960\_CONFIG2, &config, 1);

config = DEFAULT\_CONFIG3;

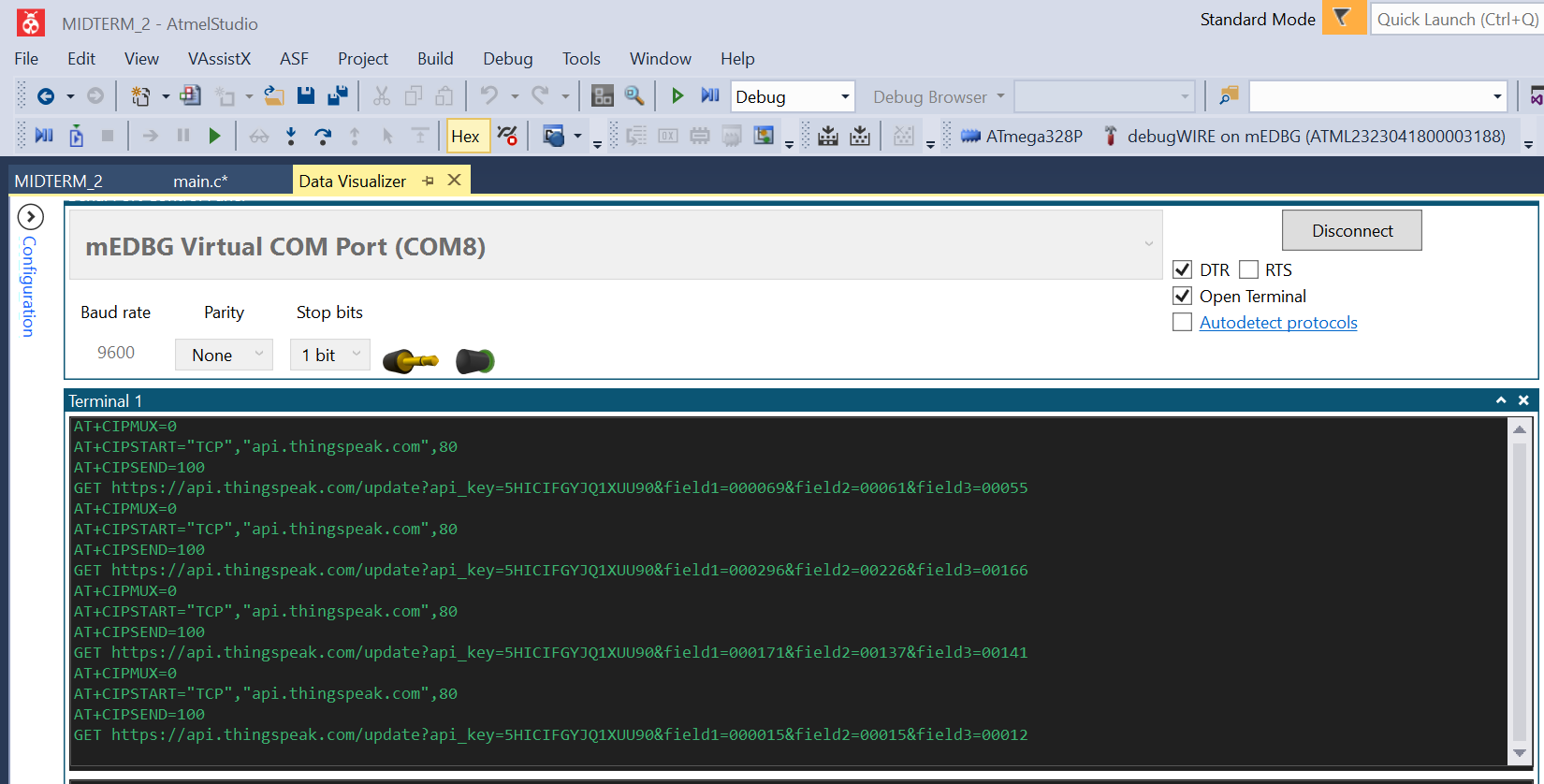
i2c\_writeReg(APDS\_WRITE, APDS9960\_CONFIG3, &config, 1);

}

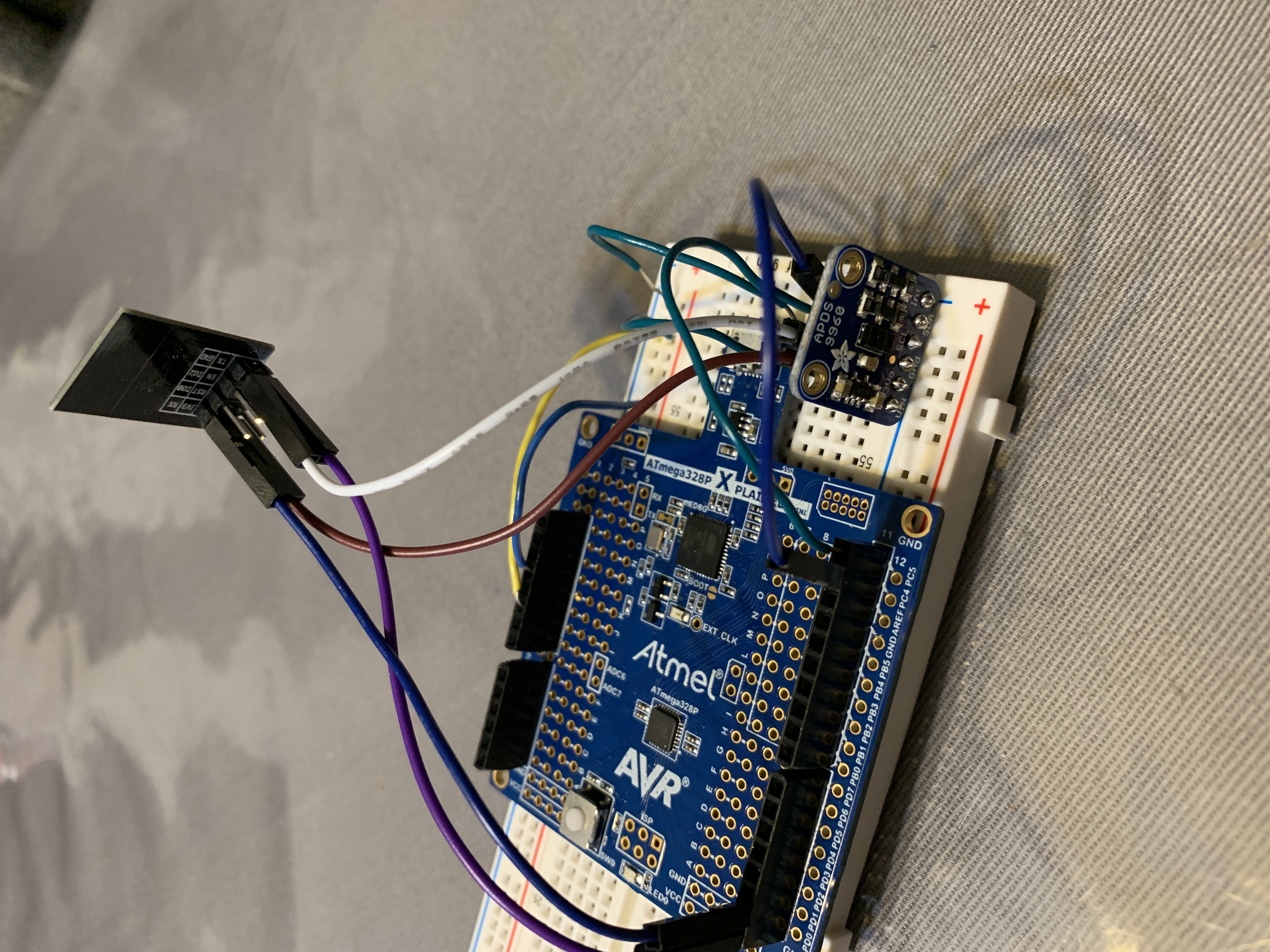
1. **SCHEMATICS**



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



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



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/Fwgln-hfr2Q>

1. **GITHUB LINK OF THIS DA**

<https://github.com/martiv6/>midterms/Midterm\_2

**Student Academic Misconduct Policy**

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

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

Victor Martinez