CPE301 – SPRING 2019

Midterm 2

Student Name: Tyler Gardenhire

Student #: 8000450294

Student Email: gardenhi@unlv.nevada.edu

Primary Github address: [gardenhi@unlv.nevada.edu](mailto:gardenhi@unlv.nevada.edu)

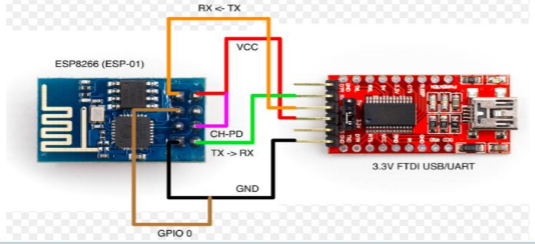
Directory: https://github.com/tylergardenhire/submission\_projects.git

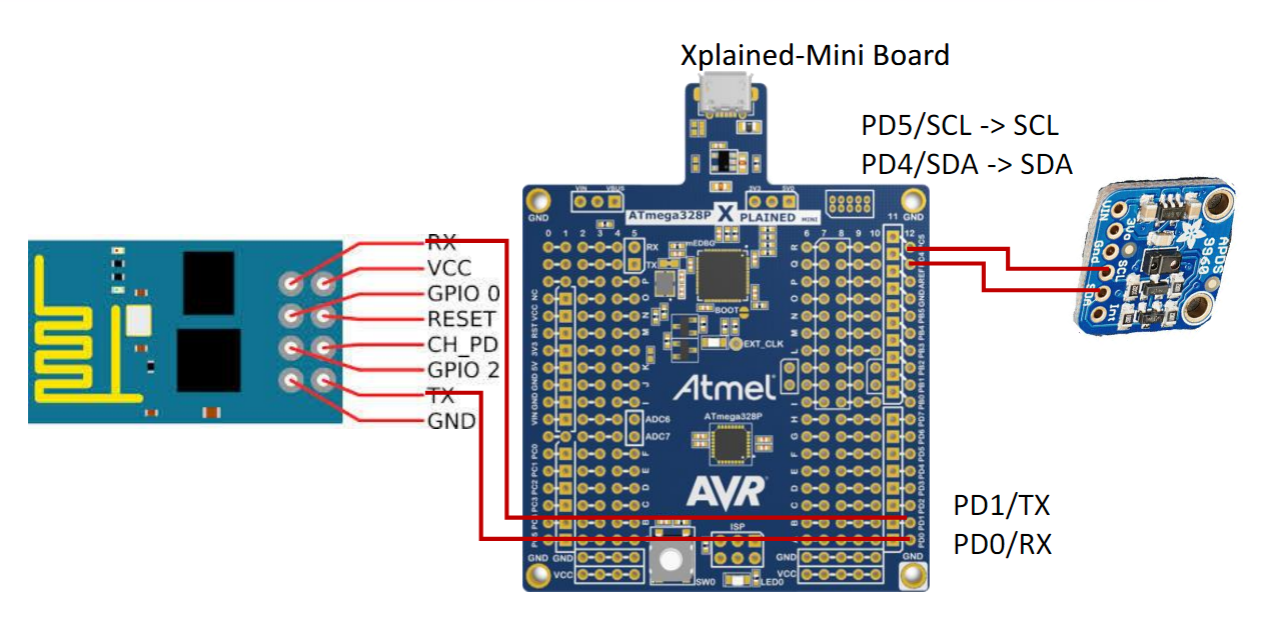
Submit the following for all Labs:

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/DA, 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**

Atmel Studio 7 w/ AVR assembly, Atmega328p board, FTDI chip, and ESP module used.





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

/\*

\* Midterm 2

\* Created: 5/13/2019 3:21:23 PM

\* Author : Tyler Gardenhire

\*/

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <stdlib.h>

#include <stdint.h>

#include "SparkFun\_APDS9960.h"

#include "i2c\_master.h"

#define F\_CPU 16000000UL

#define BAUD 9600

#define FOSC 16000000

#define UBRREQ FOSC/16/BAUD -1

#define APDS9960\_WRITE 0x72

#define APDS9960\_READ 0x73

void UART\_init (void);

void APDS\_init (void);

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

FILE str\_uart = FDEV\_SETUP\_STREAM(uart\_putchar, NULL , \_FDEV\_SETUP\_WRITE);

void getreading(void);

uint16\_t red;

uint16\_t green;

uint16\_t blue;

int main(void)

{

UART\_init(); //initialize uart values

APDS\_init(); //initialize APDS9960

i2c\_init(); //initialize I2C

stdout = &str\_uart;

red = 0;

green = 0;

blue = 0;

\_delay\_ms(5000);

printf("AT+CWMODE=3\r\n"); //set AP’s info

\_delay\_ms(5000);

printf("AT+CWJAP=\"eos2f8\",\"122gfh48\"\r\n"); //connect to Internet

while(1)//send values through the cloud

{

\_delay\_ms(5000);

printf("AT+CIPMUX=0\r\n"); //enable single connection

\_delay\_ms(5000);

printf("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n"); //start the connection to the cloud

//set length of data to be sent

\_delay\_ms(5000);

getreading();

printf("AT+CIPSEND=104\r\n");

printf("GET https://api.thingspeak.com/update?api\_key=thingspeakkey&field1=0%05u&field2=%05u&field3=%05u\r\n", red, green, blue);

\_delay\_ms(3000);

}

}

void getreading()

{

uint8\_t redH, redL;

uint8\_t greenH, greenL;

uint8\_t blueH, blueL;

//red

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

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

//green

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

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

//blue

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;

// THRESHOLD

if (red > 255)

red = 255;

if (green > 255)

green = 255;

if (blue > 255)

blue = 255;

}

void APDS\_init(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 USART\_putstring(char \*StringPtr)

{

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

{

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

UDR0 = \*StringPtr; //UDR0 register grabs the value given from the parameter

StringPtr++;

}

}

void UART\_init(void)

{

uint16\_t baud\_rate = UBRREQ;

UBRR0H = baud\_rate >> 8;

UBRR0L = baud\_rate & 0xFF;

UCSR0B = ( 1 <<RXEN0)|( 1 <<TXEN0); //enable receiver and transmitter

UCSR0C = (3 <<UCSZ00); // Set frame format: 8data, 1stop bit

}

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

{

while ( !( UCSR0A & ( 1 <<UDRE0)) ); //wait until buffer is empty

UDR0 = c; //data goes into buffer

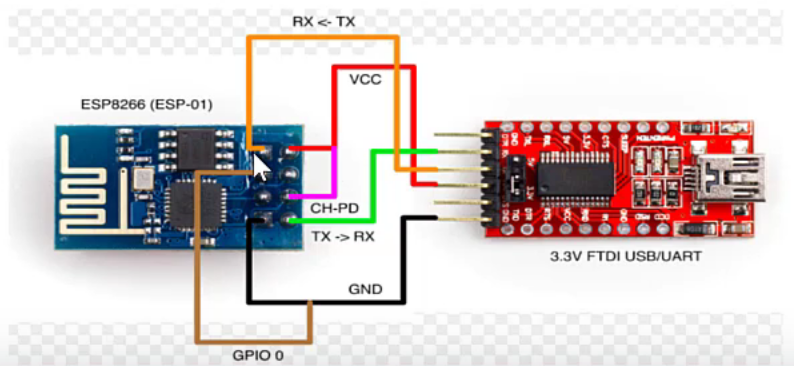
return 0;

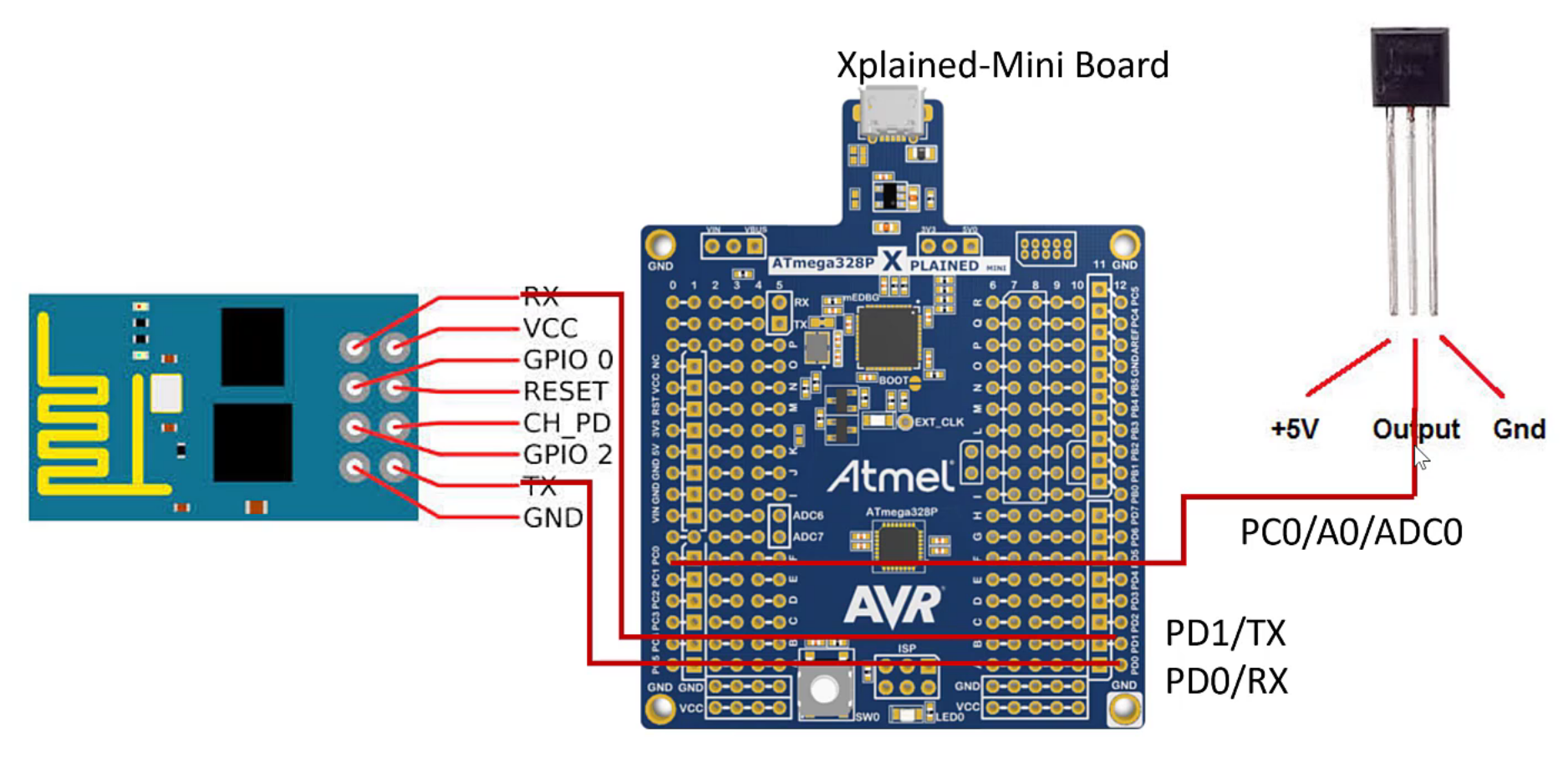
}

1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

N/A

1. **SCHEMATICS**

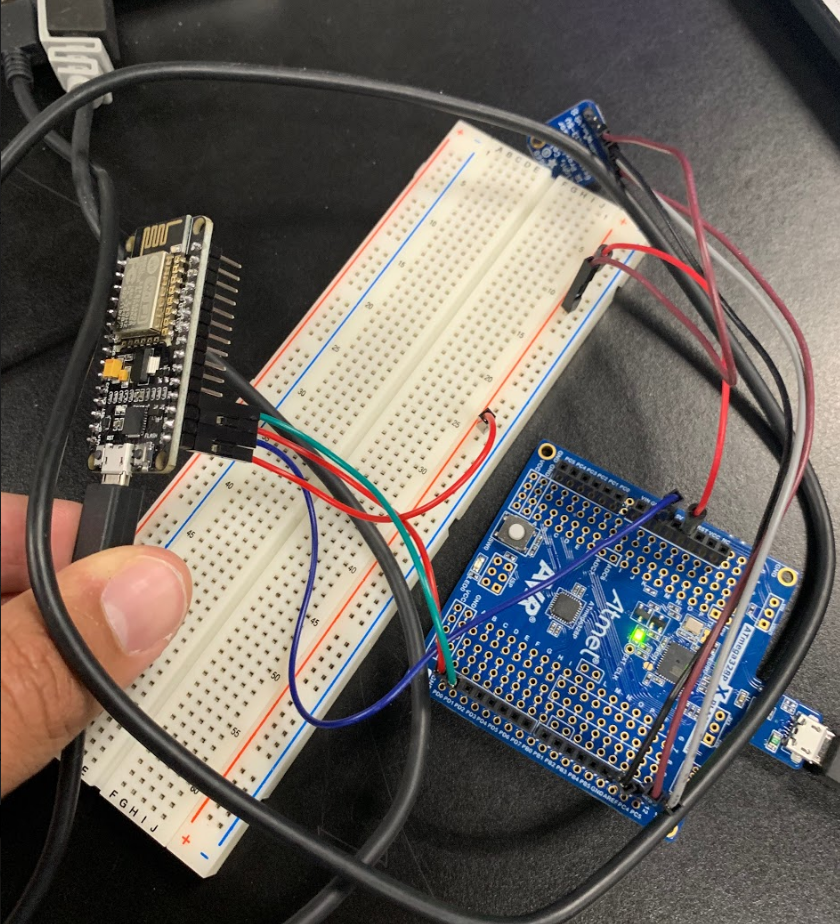




1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

N/A

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



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/J3iwdaao_Ws>

1. **GITHUB LINK OF THIS DA**

https://github.com/tylergardenhire/submission\_projects.git

**Student Academic Misconduct Policy**

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

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

TYLER GARDENHIRE