CPE301 –Fall 2019

Design Assignment 5

Student Name: Moriah Wingrove

Student #: 5000156247

Student Email: Wingrove@unlv.nevada.edu

Primary Github address: <https://github.com/windew/Tiny_Dragons.git>

Directory:

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**

Atmega 328pb qty2

nrf24l01 qty2

lm35 qty2

**INITIAL/MODIFIED/DEVELOPED CODE OF TASK 4B/1**

**C code**

/\*

\* Design\_Assignment\_5.c

\*

\* Created: 11/21/2019 10:52:57 AM

\* Author : Moriah Wingrove

\*/

//TASK FIVE: Interface the provided NRF24L01+ RF module to the ATmega328p using the SPI interface. Using the earlier developed code

// for ADC, transmit the ADC value of the internal temperature sensor, or LM35 sensor between the two RF modules. The ATmega328p

// interfacing the RF module should alternate between TX and RX modes in the same interval. The temperature of both ATmega328p's

// should be displayed on both ATmega328p's

#ifndef *F\_CPU*

#define *F\_CPU* 16000000UL

#endif

#include <string.h>

#include <avr/io.h>

#include <stdbool.h>

#include <util/delay.h>

#include <stdio.h>

#include <avr/interrupt.h>

// Set up UART for printf();

#ifndef BAUD

#define BAUD 9600

#endif

// Include library

#include "STDIO\_UART.h"

#include "nrf24l01.h"

#include "nrf24l01-mnemonics.h"

#include "spi.h"

void print\_config(void);

// Used in IRQ ISR

volatile bool message\_received = false;

volatile bool status = false;

// Global variables

volatile *uint8\_t* ADCdata;

volatile unsigned char temp[10];

int main(void)

{

// Set up ADC

ADMUX |= (1 << REFS0); // use AVcc

ADMUX |= (1 << ADLAR); // Right adjust

ADCSRA = (1 << ADEN) // Enable

|(1 << ADPS1)

|(1 << ADPS0) // 128 prescaler for 16Mhz

|(1 << ADATE) // ADC Auto Trigger

|(1 << ADSC); // Start ADC

// send message

char tx\_message[32]; // Define string array

unsigned char i;

char dummy[10];

// Initialize UART

uart\_init();

// Initialize nRF24L01+ and print configuration info

nrf24\_init();

print\_config();

// Start listening to incoming messages

nrf24\_start\_listening();

*strcpy*(tx\_message,"okay"); // Copy string into array

nrf24\_send\_message(tx\_message);

while (1)

{

ADCdata = (ADCH << 1) \* 2 + 32; // Convert Celsius to Fahrenheit

*itoa*(ADCdata, dummy, 10); //convert char to ascii

for(i = 0 ; i < 10 ; i++)

{

temp[i] = dummy[i]; //move converted ascii

}

if (message\_received)

{

// Message received, print it

message\_received = false;

*printf*("Received message: %s\n",nrf24\_read\_message());

// Send message as response

*\_delay\_ms*(500);

status = nrf24\_send\_message(temp);

if (status == true) *printf*("Message sent successfully\n");

}

}

}

// Interrupt on IRQ pin

ISR(INT0\_vect)

{

message\_received = true;

}

void print\_config(void)

{

*uint8\_t* data;

*printf*("Startup successful\n\n nRF24L01+ configured as:\n");

*printf*("-------------------------------------------\n");

nrf24\_read(CONFIG,&data,1);

*printf*("CONFIG 0x%x\n",data);

nrf24\_read(EN\_AA,&data,1);

*printf*("EN\_AA 0x%x\n",data);

nrf24\_read(EN\_RXADDR,&data,1);

*printf*("EN\_RXADDR 0x%x\n",data);

nrf24\_read(SETUP\_RETR,&data,1);

*printf*("SETUP\_RETR 0x%x\n",data);

nrf24\_read(RF\_CH,&data,1);

*printf*("RF\_CH 0x%x\n",data);

nrf24\_read(RF\_SETUP,&data,1);

*printf*("RF\_SETUP 0x%x\n",data);

nrf24\_read(STATUS,&data,1);

*printf*("STATUS 0x%x\n",data);

nrf24\_read(FEATURE,&data,1);

*printf*("FEATURE 0x%x\n",data);

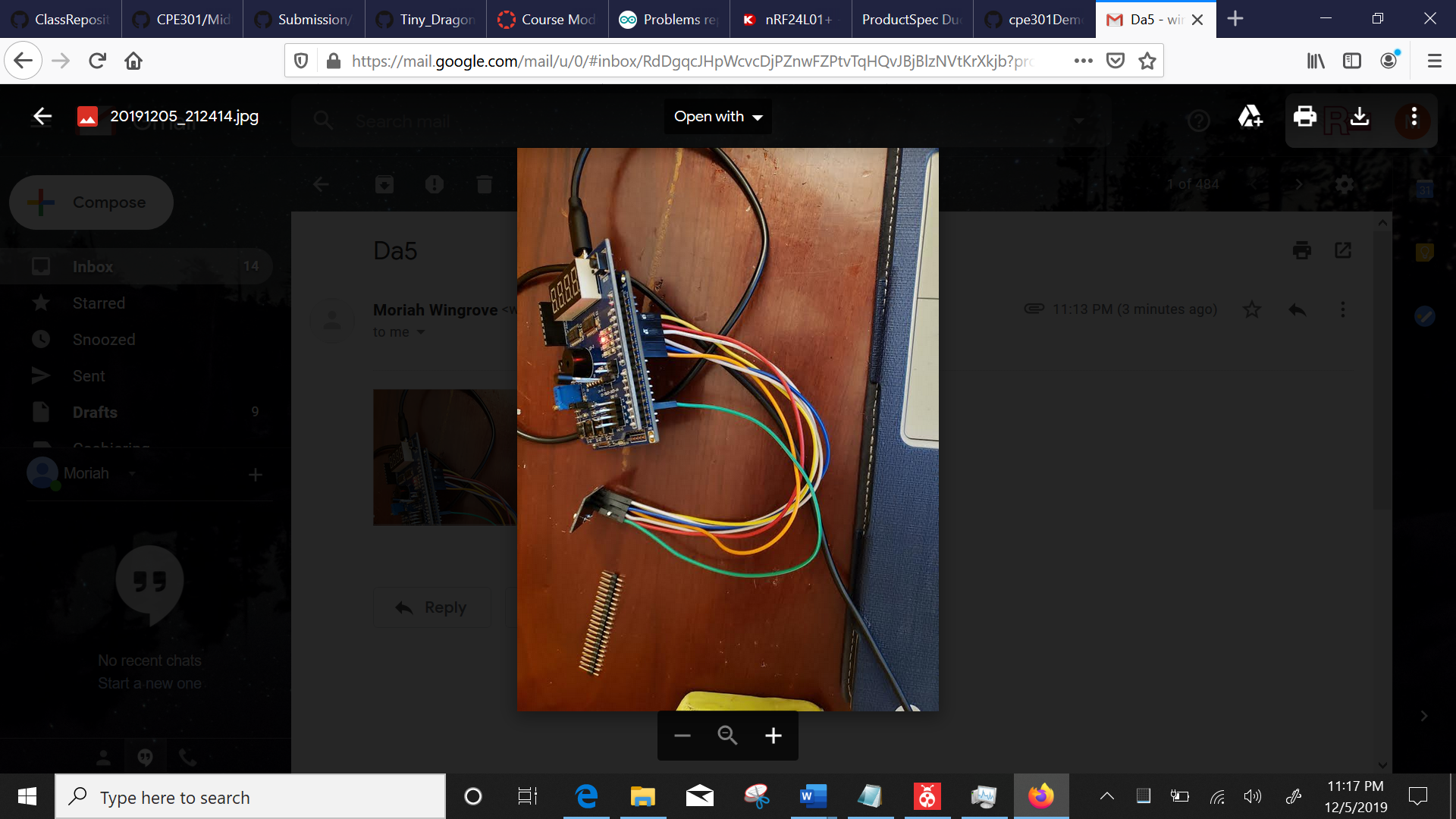
*printf*("-------------------------------------------\n\n");

}

1. **SCHEMATICS**

Use fritzing.org

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**
2. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**



1. **VIDEO LINKS OF EACH DEMO**
2. **GITHUB LINK OF THIS DA**

<https://github.com/windew/Tiny_Dragons>

**Student Academic Misconduct Policy**

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

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

NAME OF THE STUDENT