CPE301 – SPRING 2019

Design Assignment 5

Student Name: Kyungseo Yun

Student #: 2001091216

Student Email: yunk93@unlv.nevada.edu

Primary Github address: <https://github.com/biscuit0x/submission_yun.git>

Directory: submission\_yun/DesignAssignments/DA5/

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

Atmega328, multi-shield, nrf24L01, LM35

2. C code

/\*

\* DA5.c

\* Author : jayne

\*/

#ifndef F\_CPU

#define F\_CPU 16000000UL

#endif

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

//#include <stdbool.h>

#include <stdio.h>

#include <string.h>

// Set up UART for printf();

#ifndef BAUD

#define BAUD 9600

#endif

#include "inc\STDIO\_UART.c"

// Include nRF24L01+ library

#include "inc\nrf24l01.c"

#include "inc\nrf24l01-mnemonics.h"

#include "inc\spi.c"

void print\_config(void);

void print\_config(void)

volatile bool message\_received = false;

volatile bool status = false;

void ADC\_init (void);

volatile unsigned ADC\_temp;

volatile char temp[20];

void ADC\_init(void) {

ADMUX |= (1<<REFS0)|

(1<<MUX2) ;

ADCSRA |= (1<<ADEN)|

(0<<ADSC)|

(0<<ADATE)|// Auto Trigger off

(0<<ADIF)| // Interrupt Flag reset

(0<<ADIE)| // Interrupt Disabled

(1<<ADPS2)| // ADC Prescaler Select 32

(0<<ADPS1)|

(1<<ADPS0);

}

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");

}

void READ\_ADC(void) {

unsigned char i =4;

ADC\_TEMP = 0; //initialize

while (i--) {

ADCSRA |= (1<<ADSC);

while(ADCSRA & (1<<ADSC));

ADC\_TEMP+= ADC;

\_delay\_ms(100);

}

ADC\_TEMP = ADC\_TEMP/4; // Average

}

int main(void)

{

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

strcpy(tx\_message,"testing . . . ."); // Copy string into array

// Initialize UART

uart\_init();

// Initialize ADC

ADC\_init();

// Initialize nRF24L01+ and print configuration info

nrf24\_init();

print\_config();

// Start listening to incoming messages

nrf24\_start\_listening();

strcpy(tx\_message,"Initializing Chat Room..."); // Copy string into array

nrf24\_send\_message(tx\_message);

while (1)

{

if (message\_received)

{

message\_received = false;

READ\_ADC();

// Message received, print it

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

snprintf(temp,sizeof(temp),"%d\r\n",adc\_temp);

// Send message as response

\_delay\_ms(500);

status = nrf24\_send\_message(temp);

if (status == true) printf("Temperature Successfully Sent\n\n");

}

}

}

// Interrupt on IRQ pin

ISR(INT0\_vect)

{

message\_received = true;

}