CPE301 – SPRING 2019

Design Assignment 5A

Student Name: Ricky Perez
Student #: 5002297620
Student Email: perezr1@unlv.nevada.edu
Primary Github address: https://github.com/RickyPerez79/submission_da
Directory: 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 LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

- Atmega328p
- NRF24L01
- LM34

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

My Partner for this assignment is Saul Mendoza

```
#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>
unsigned int ADC_TEMP;
// Set up UART for printf();
#ifndef BAUD
#define BAUD 9600
#endif
#include "STDIO_UART.h"
// Include nRF24L01+ library
#include "nrf24l01.h"
#include "nrf24l01-mnemonics.h"
#include "spi.h"
void print_config(void);
void ADC_INIT(void);
void READ_ADC(void);
// Used in IRQ ISR
volatile bool message_received = false;
volatile bool status = false;
int main(void){
	// Set cliche message to send (message cannot exceed 32 characters)
	char tx_message[32]; // Define string array
	char *tx_ptr = tx_message;
	strcpy(tx_message,"Hi :) !"); // Copy string into array
	// Initialize UART
	uart_init();
	// Initialize nRF24L01+ and print configuration info
	nrf24_init();
	print_config();
	ADC_INIT();
	// Start listening to incoming messages
	printf("start listening\n");
	nrf24_start_listening();
	printf("Done listening\n");
	ADC_TEMP = 0;
	while (1){
		//printf("outside if\n");
		READ_ADC();
		tx_ptr = tx_message;
```

```
        //sprintf(tx_ptr, "%d", ADC_TEMP);
        nrf24_send_message(tx_ptr);
        delay_ms(100);
        //continue;
        //;
        if (message_received){
            printf("inside if condition\n");
            // 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(tx_message);
            if (status == true) printf("Message sent successfully\n");
        }
    }
}
// Interrupt on IRQ pin
ISR(INT0_vect) {
    message_received = true;
}
void ADC_INIT(void){
    ADMUX = (0<<REFS1)| // Reference Selection Bits
    (1<<REFS0)| // AVcc - external cap at AREF
    (0<<ADLAR)| // ADC Left Adjust Result
    (1<<MUX2)| // ANalog Channel Selection Bits
    (0<<MUX1)| //
    (0<<MUX0);
    ADCSRA = (1<<ADEN)| // ADC ENable
    (0<<ADSC)| // ADC Start Conversion
    (0<<ADATE)| // ADC Auto Trigger Enable
    (0<<ADIF)| // ADC Interrupt Flag
    (0<<ADIE)| // ADC Interrupt Enable
    (1<<ADPS2)| // ADC Prescaler Select Bits
    (0<<ADPS1)|
    (1<<ADPS0);
    // Timer/Counter1 Interrupt Mask Register
    TIMSK1 |= (1<<TOIE1); // enable overflow interrupt
    TCCR1B |= (1<<CS12)|(1<<CS10); // clock
    TCNT1 = 49911; //((16MHz/1024)*1)-1 = 15624
}
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(50);
    }
    ADC_TEMP = ADC_TEMP/8 ; // Average
}
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%02X\n",data);
```

```
	nrf24_read(EN_AA,&data,1);
	printf("EN_AA 0x%02X\n",data);
	nrf24_read(EN_RXADDR,&data,1);
	printf("EN_RXADDR 0x%02X\n",data);
	nrf24_read(SETUP_RETR,&data,1);
	printf("SETUP_RETR 0x%02X\n",data);
	nrf24_read(RF_CH,&data,1);
	printf("RF_CH 0x%02X\n",data);
	nrf24_read(RF_SETUP,&data,1);
	printf("RF_SETUP 0x%02X\n",data);
	nrf24_read(STATUS,&data,1);
	printf("STATUS 0x%02X\n",data);
	nrf24_read(FEATURE,&data,1);
	printf("FEATURE 0x%02X\n",data);
	printf("-------------------------------------------\n\n");
}
```

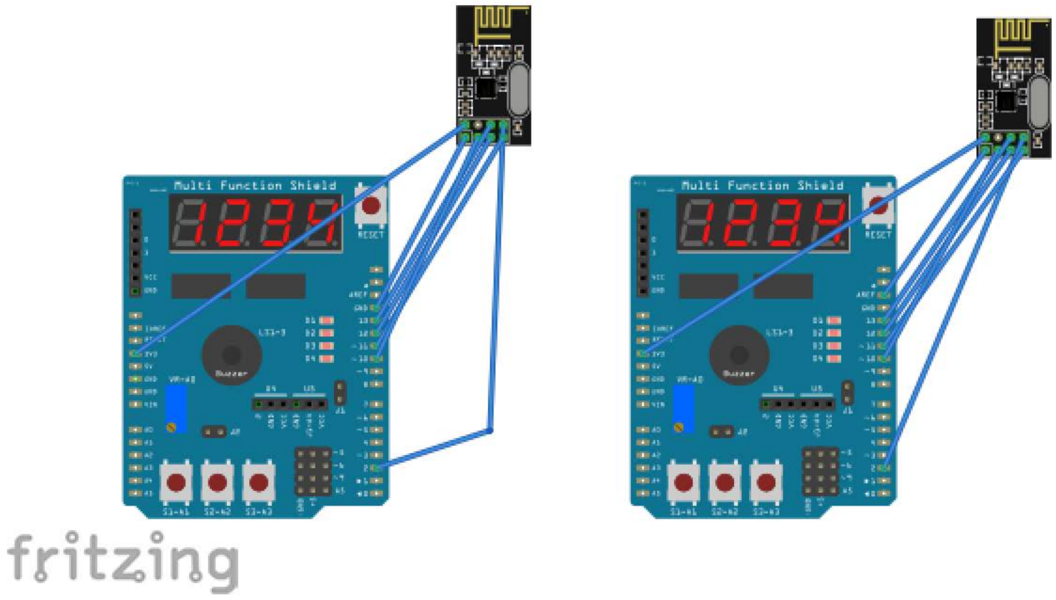
Partners Address

```
// Settings
uint8_t rx_address[5] = { 0x73, 0x73, 0x73, 0x73, 0x73 };      // Read pipe address
uint8_t tx_address[5] = { 0x42, 0x42, 0x42, 0x42, 0x42 };      // Write pipe address
#define READ_PIPE           0
```

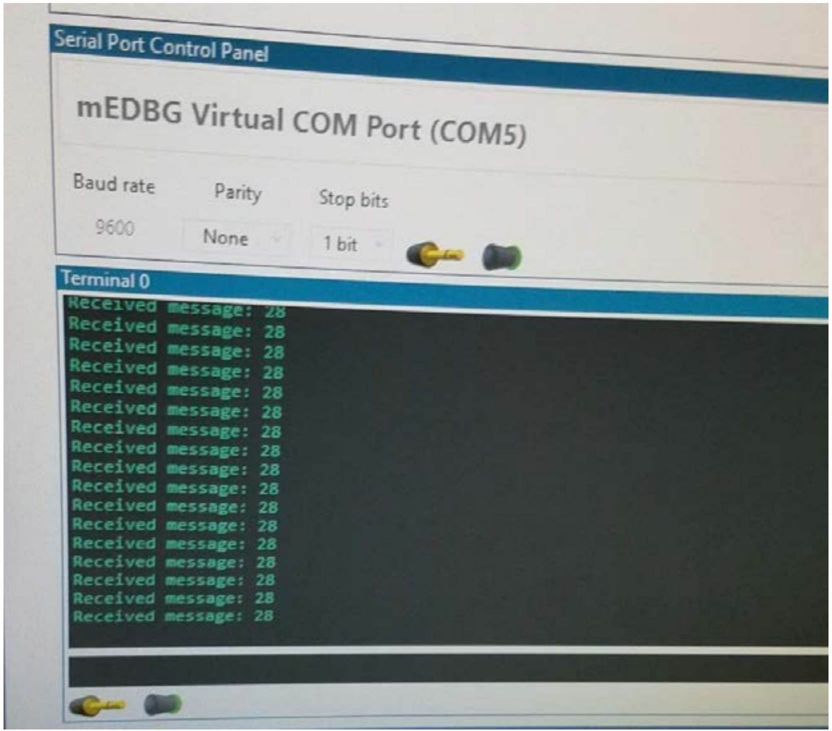
My Address

```
// Settings
uint8_t rx_address[5] = { 0x42, 0x42, 0x42, 0x42, 0x42 };
uint8_t tx_address[5] = { 0x73, 0x73, 0x73, 0x73, 0x73 };
#define READ_PIPE     0
```

3. SCHEMATICS

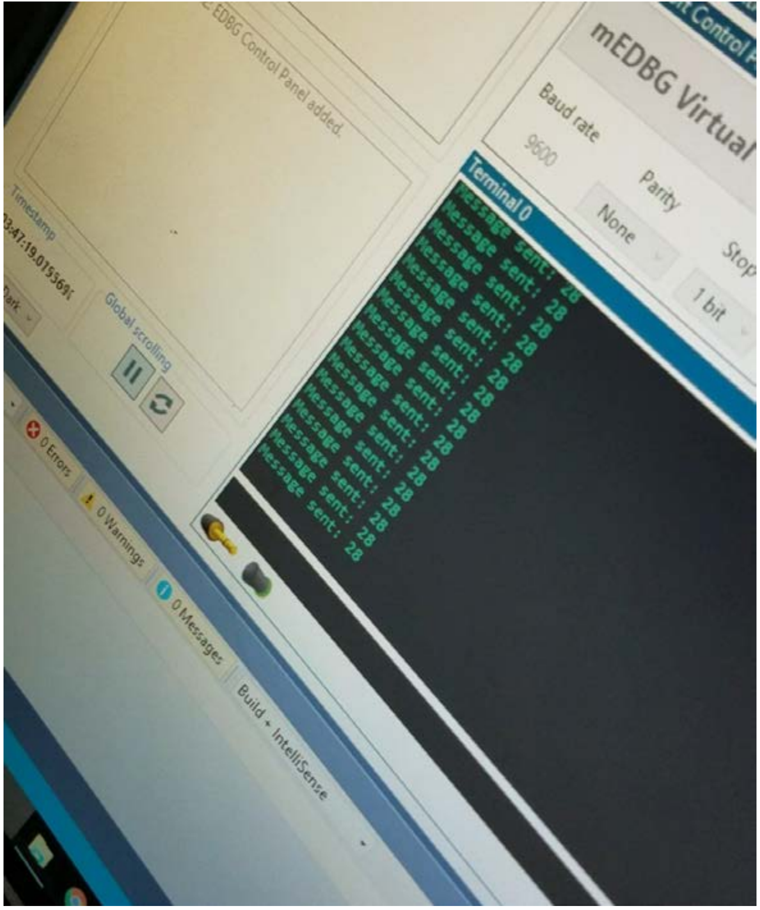


4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)



Baud rate: 9600
Parity: None
Stop bits: 1 bit

```
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
Received message: 28
```



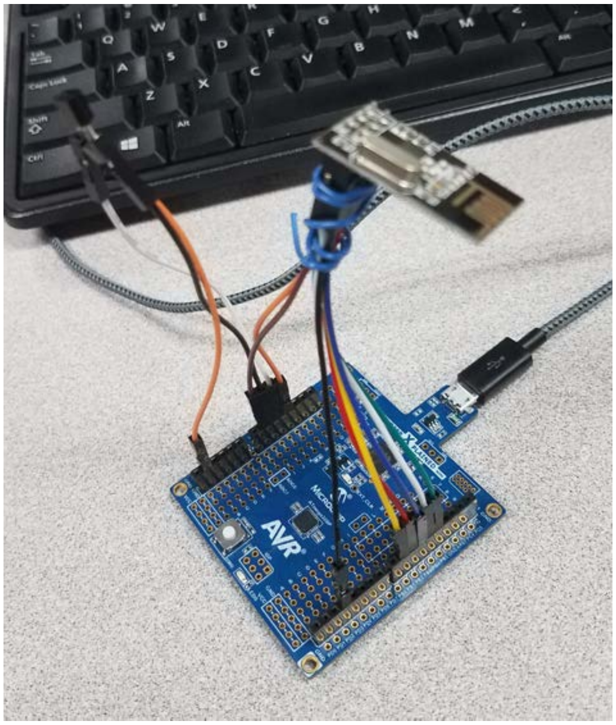
EDBG Control Panel added.

3:47:19.015569

Baud rate: 9600
Parity: None
Stop: 1 bit

```
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
Message sent: 28
```

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)



6. VIDEO LINKS OF EACH DEMO

https://youtu.be/v86b5uXF0AU

7. GITHUB LINK OF THIS DA

https://github.com/RickyPerez79/submission_da

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

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