CPE301 – SPRING 2019

Design Assignment 5

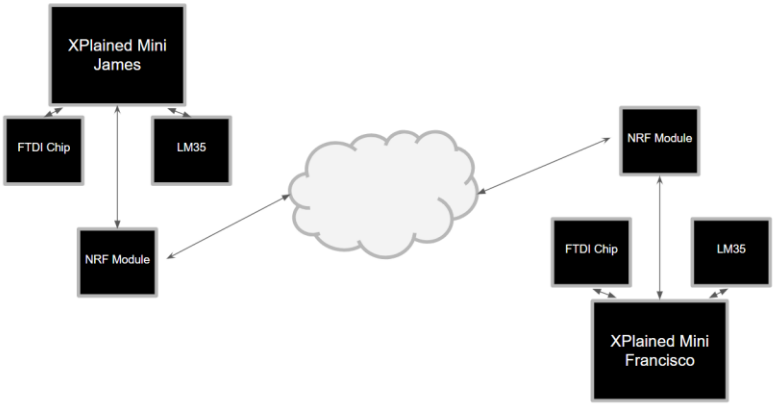
Student Name: James Skelly
Student #: 2000945485
Student Email: skellj1@unlv.nevada.edu
Primary Github address: https://github.com/skellj1/submission_da
Directory: skellj1/submission_da

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

The components used for this DA include Atmel Studio 7, the NRF module, the FTDI chip module (UART), the Xplained mini, iphone for recording, jumper wire, LM35 temperature sensor, and fritzing.org.



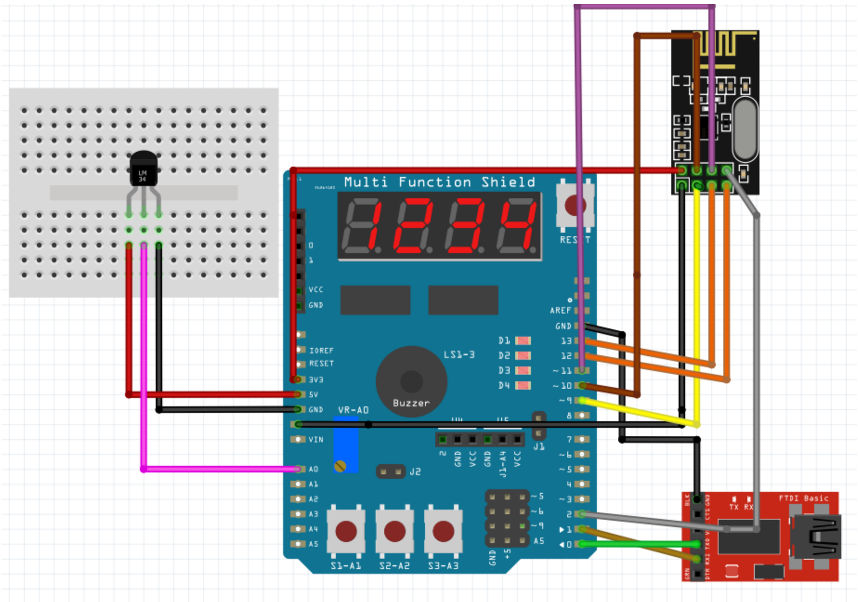
2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1/A

```
/*
 * DA5.c
 * Created: 4/25/2019 4:20:53 PM
 * Author : James Skelly
 */

// Set clock frequency for delay function
#ifndef F_CPU
#define F_CPU 16000000UL
#endif

#ifndef BAUD
#define BAUD 9600                                                // set baud rate to 9600
#endif

// Include necessary C, NRF, UART libraries
#include "inc\STDIO_UART.c"
#include "inc\nrf24l01.c"
#include "inc\nrf24l01-mnemonics.h"
#include "inc\spi.c"
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <stdbool.h>
#include <stdio.h>
#include <string.h>

// function prototyping
void print_config(void);
void ADC_init (void);

volatile unsigned char ADCtemp[5];               // initialize variable for ADC temp
volatile uint8_t ADCvalue;                       // initialize variable for the ADC value
volatile bool received;                          // initialize variable for message reception
volatile bool status = false;                    // initialize variable for status of //transmission

// Main code body

int main(void)
{
        sei();                                   // globally enable interrupts
        ADC_init();                              // initialize the ADC
        uart_init();                             // initialize the UART

        char tx_message[32];                     // character array to output string

        // copy string "Lets get it started" into array
        strcpy(tx_message,"Lets get it started");

        nrf24_init();                            // initialize NRF
        print_config();                  // run print config function
        nrf24_start_listening(); // start listening for transmission from other user

        strcpy(tx_message,"Lets get it started");        // Copy string into array
        nrf24_send_message(tx_message);  // send the string message to the other user

        while (1)
        {// if a message is received, send a message back with a success message for assurance
                if (received == true)
                {
                        received = false;                // reset received variable to false
                        printf("Received message: %s\n",nrf24_read_message());
                        _delay_ms(500);

                        status = nrf24_send_message(ADCtemp);
                        if (status == true) printf("Message sent successfully\n");
                }
        }
}

// Interrupts

//      Interrupt subroutine (IRQ)
ISR(INT0_vect)
{
        received = true;
}

//  Interrupt subroutine for ADC
ISR(ADC_vect)
{
        volatile unsigned int j=0;
        char temp[5];

        ADCvalue = (ADCH << 1);   // Shifts the left adjusted ADCH value left by 1
        itoa(ADCvalue, temp, 10);        // Converts integer to string

        while (j<5)                     // Transfers the temp string from itoa() to ADCtemp
        {
                ADCtemp[j] = temp[j];
                j++;
        }
}

// Functions

void ADC_init(void)
{
        ADMUX |= (1 << REFS0)|(1 << ADLAR);// set AVcc (reference voltage for ADC) and
                                                // left justify value in ADC (10-bit register)

        ADCSRA = (1 << ADEN)|           // enable the ADC
        (1 << ADSC) |                           // start converting
        (1 << ADATE) |                  // enable ADC auto-trigger
        (1 << ADIE) |                           // enable ADC interrupt
        (1 << ADPS2)|(1 << ADPS1)|(1 << ADPS0);         // set ADC prescaler of 128
}

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");
}
```

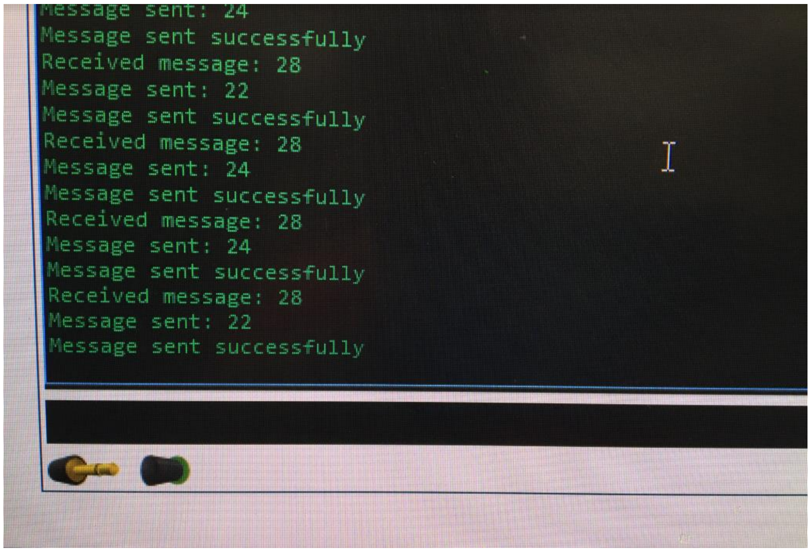
3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

Not applicable for this assignment.

4. SCHEMATICS

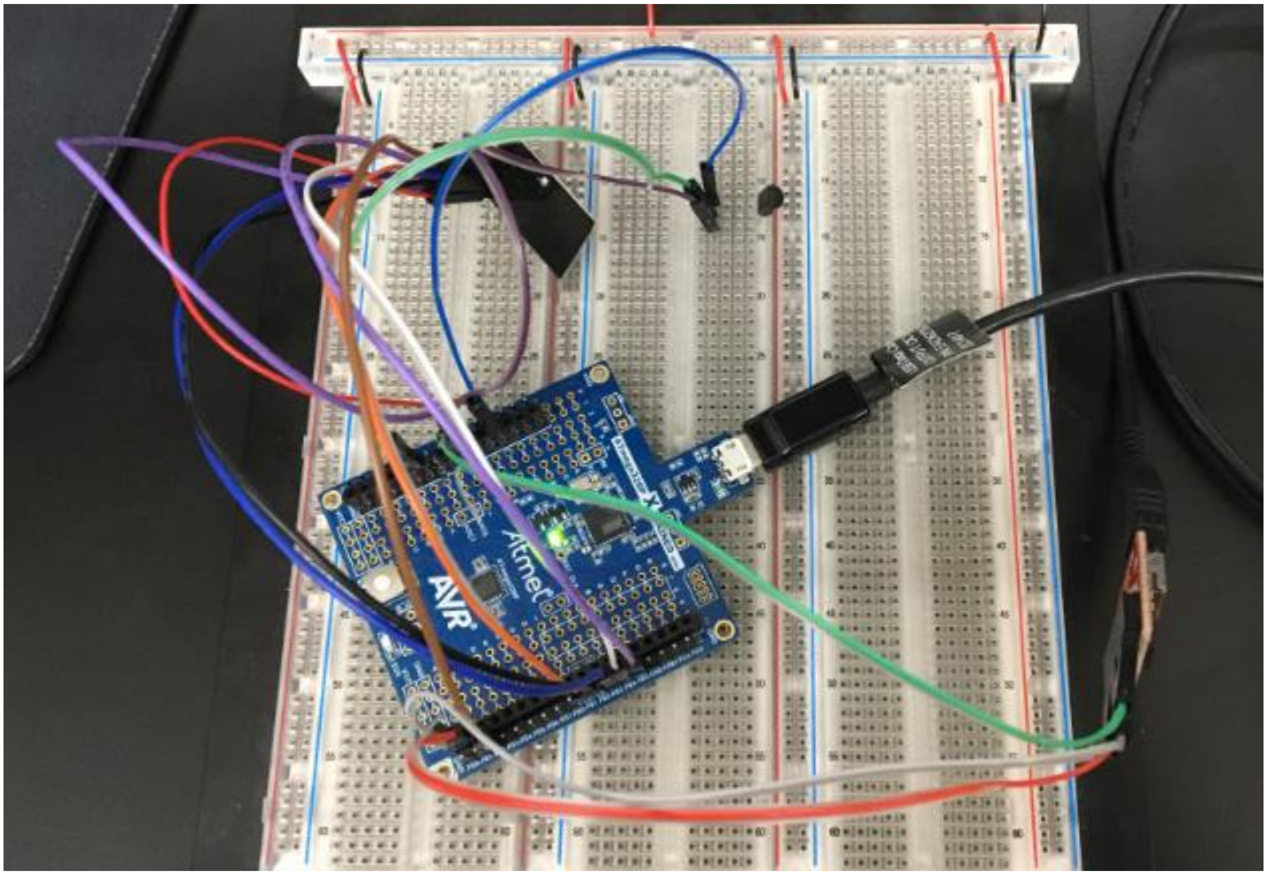


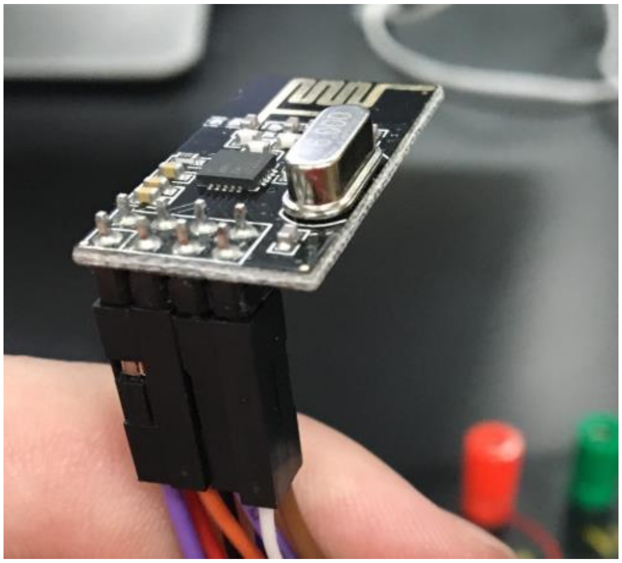
5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

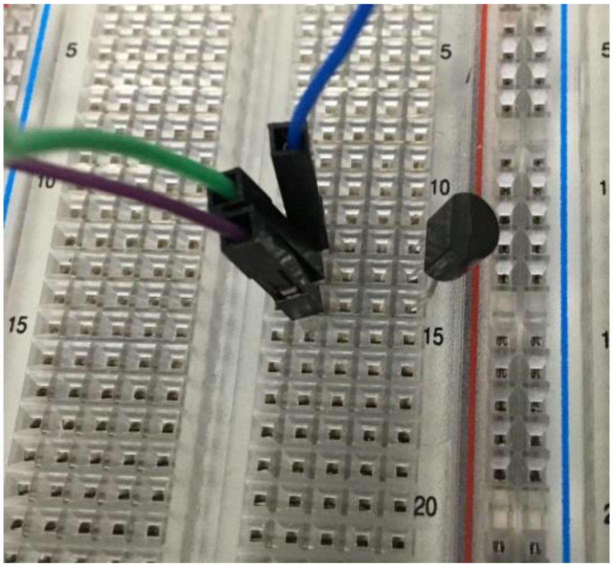


```
Message sent: 24
Message sent successfully
Received message: 28
Message sent: 22
Message sent successfully
Received message: 28
Message sent: 24
Message sent successfully
Received message: 28
Message sent: 24
Message sent successfully
Received message: 28
Message sent: 22
Message sent successfully
```

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)







7. VIDEO LINKS OF EACH DEMO

https://www.youtube.com/watch?v=wWM-BwBbP4s

8. GITHUB LINK OF THIS DA

https://github.com/skellj1/submission_da

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

“*This assignment submission is my own, original work*”.
James W. Skelly