CPE301 – SPRING 2019

Design Assignment 5

Student Name: James Skelly

Student #: 2000945485

Student Email: skellj1@unlv.nevada.edu

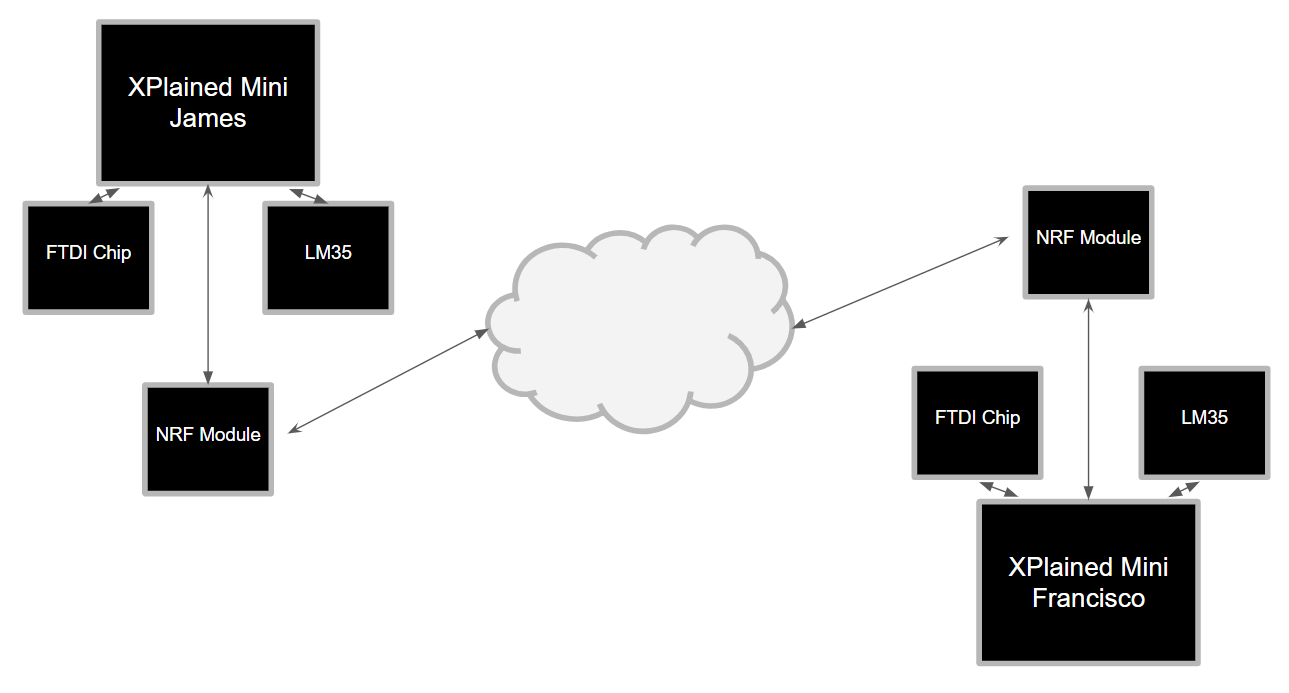
Primary Github address: <https://github.com/skellj1/submission_da>

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).
5. **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.



1. **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);

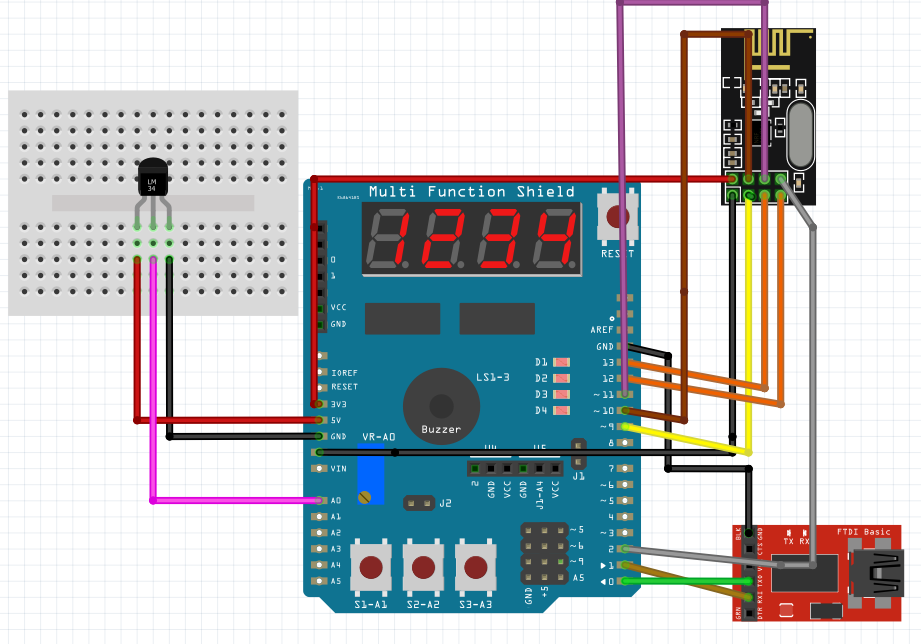
*printf*("-------------------------------------------\n\n");

}

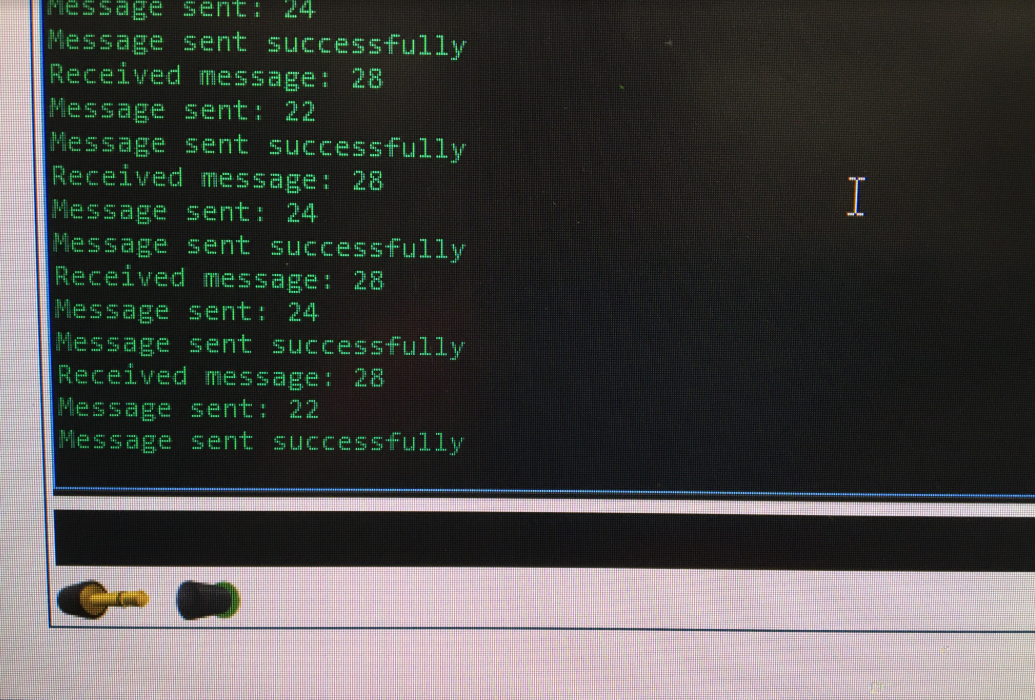
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

**Not applicable for this assignment.**

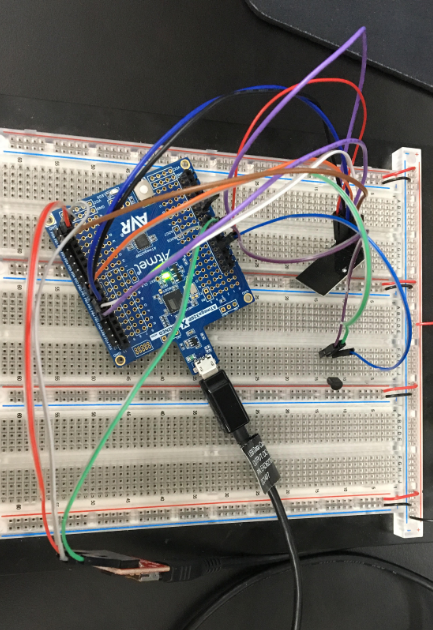
1. **SCHEMATICS**

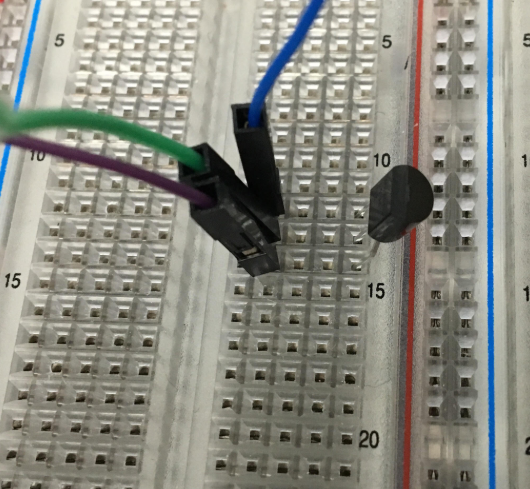
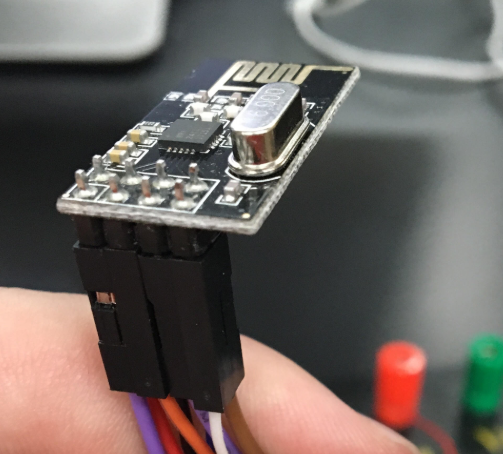


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**



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





1. **VIDEO LINKS OF EACH DEMO**

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

1. **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