CPE301 – SPRING 2019

Design Assignment 5A

Student Name: Mohamad Jundi

Student #: 8000321867

Student Email: jundi@unlv.nevada.edu

Primary Github address: https://github.com/MohamedJundi1994/Submission\_DA.git

Directory: Documents\School\CPE 301\Repository\CPE\_301\DesignAssignments\DA5A

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

USB port => Xplained Mini => NRF24L01+ RF

USB port => Xplained Mini => 5V (left) => PC5 output (middle) => GND (right) => LM34 (Temp Sensor)

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

My main code:

#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>

#ifndef BAUD // Used to set UART for printf();

#define BAUD 9600

#endif

#include "STDIO\_UART.h"

#include "nrf24l01.h" // Must include nRF24L01+ library

#include "nrf24l01-mnemonics.h"

#include "spi.h"

void print\_config(void);

void read\_adc(void);

void adc\_init(void);

volatile unsigned int adc\_temp;

char outs[20];

volatile bool message\_received = false; // Used in IRQ ISR

volatile bool status = false;

int main(void)

{

char tx\_message[32]; // Used to declare string

*strcpy*(tx\_message,"This is the message from Moe"); // Copy this string into the array

uart\_init(); // Call function to initialize UART

adc\_init(); // Call function to initialize ADC

nrf24\_init(); // Call function to initialize NRF24L01+

print\_config(); // Call function to print configure

nrf24\_start\_listening(); // Call function to start listening

while (1)

{

read\_adc();

adc\_temp = (adc\_temp\*500)/1023 + 20;

*snprintf*(outs,sizeof(outs),"%3d\r\n", adc\_temp);

*strcpy*(tx\_message,outs);

nrf24\_send\_message(tx\_message);

*\_delay\_ms*(1500);

if (message\_received)

{

message\_received = false;

*printf*("Message received: %s\n",nrf24\_read\_message()); // When message received, print the message

*\_delay\_ms*(500);

status = nrf24\_send\_message(tx\_message);

if (status == true) *printf*("Message sent successfully\n"); // When message sent, print this message

}

}

}

ISR(INT0\_vect) // Interrupt on IRQ pin

{

message\_received = true;

}

// INIT ADC

void adc\_init(void)

{

/\*\* Setup and enable ADC \*\*/

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)| // ADC4 (PC4 PIN27)

(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);

}

// READ ADC PINS

void read\_adc(void)

{

unsigned char i = 4;

adc\_temp = 0;

while (i--)

{

ADCSRA |= (1<<ADSC);

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

adc\_temp+= ADC;

*\_delay\_ms*(50);

}

adc\_temp = adc\_temp / 4; // Average a few samples

}

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);

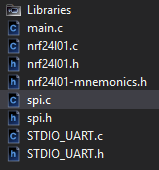
nrf24\_read(FEATURE,&data,1);

*printf*("FEATURE 0x%02X\n",data);

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

}

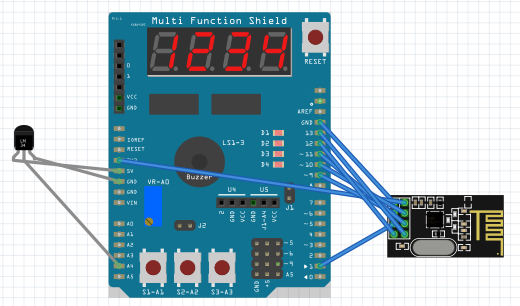
Libraries I used (included in the Github file):



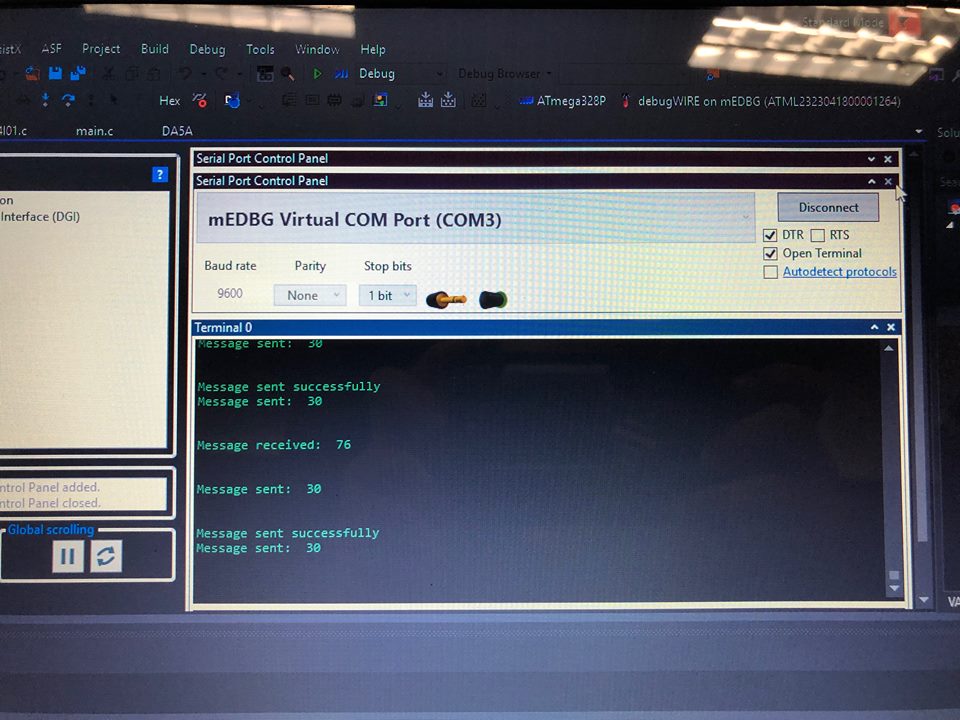
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

My main code, and libraries I’ve used are in number 1.

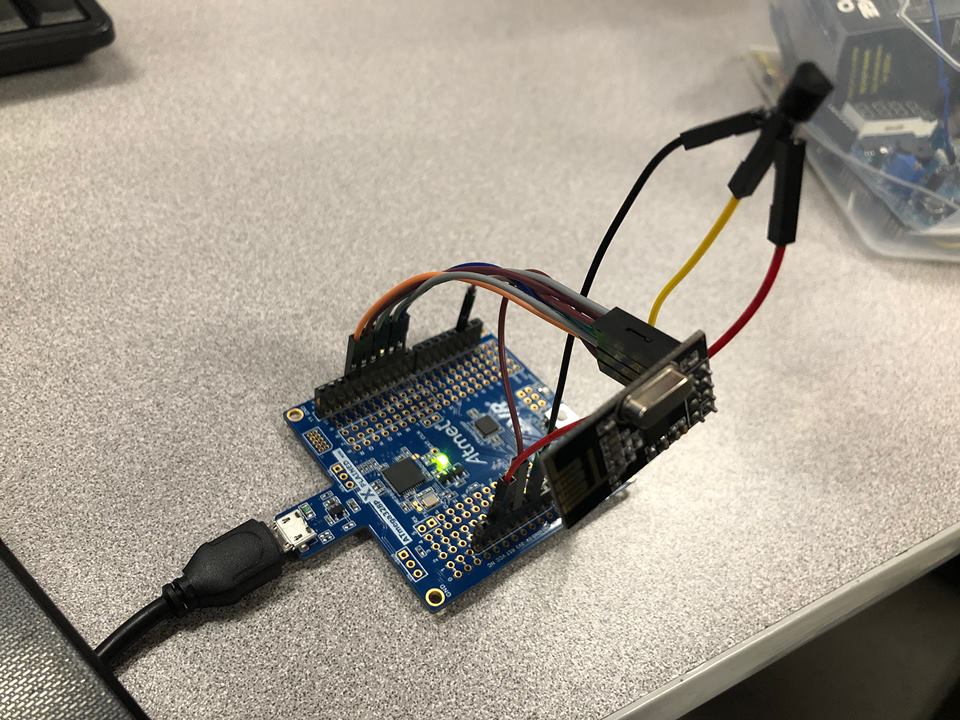
1. **SCHEMATICS**



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



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



1. **VIDEO LINKS OF EACH DEMO**

Link: <https://www.youtube.com/watch?v=dqNswmNOvpw&feature=share&fbclid=IwAR37TU6sWenlgLN_Ds_nobo9i-kGodNLiMgo-3xiFnyTcooY7fnSqgOyFwk>

1. **GITHUB LINK OF THIS DA**

Link: https://github.com/MohamedJundi1994/Submission\_DA.git

This assignment submission is my own, original work.

MOHAMAD JUNDI