CPE301 – SPRING 2019

Design Assignment 5

Student Name: Ron Joshua Recrio

Student #: 5003825419

Student Email: recrio@unlv.nevada.edu

Primary Github address: <https://github.com/recrio/submissions>

Directory: /DesignAssignments/DA5

Partner: Steven Lee

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**

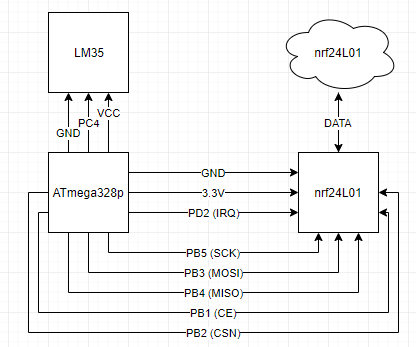
List of Components used:

ATmega328p Xplained Mini

NRF24L01

LM35

Block diagram with pins used in the Atmega328P



1. **INITIAL CODE OF TASK 1/B**

settings changed in lib file



main.c

// Set clock frequency

#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);

// Used in IRQ ISR

volatile bool message\_received = false;

volatile bool status = false;

void ADC\_init (void);

volatile unsigned adc\_temp;

volatile char temp[20];

int main(void)

{

// Set cliche message to send (message cannot exceed 32 characters)

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

*strcpy*(tx\_message,"It's working!"); // 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, print it

message\_received = false;

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

ADCSRA |= (1<<ADSC); // start conversion

while ((ADCSRA&(1<<ADIF))==0){} // Wait for conversion

ADCSRA |= (1<<ADIF); // Clear Interrupt Flag

adc\_temp = ADCL; // take in lower bits first

adc\_temp = adc\_temp | (ADCH<<8); // take in upper bits

adc\_temp = (adc\_temp/1024.0) \* 5000/10;

adc\_temp = adc\_temp\*1.8 + 32;

*snprintf*(temp,sizeof(temp),"%d\r\n",adc\_temp); // print to the buffer

// 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;

}

// Prints configuration

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 ADC\_init(void) {

ADMUX |= (1<<REFS0)| // Selected AVcc

(1<<MUX2) ;

ADCSRA |= (1<<ADEN)| // Enable ADC

(0<<ADSC)| // Do not start conversion

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

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

(0<<ADIE)| // Interrupt Disabled

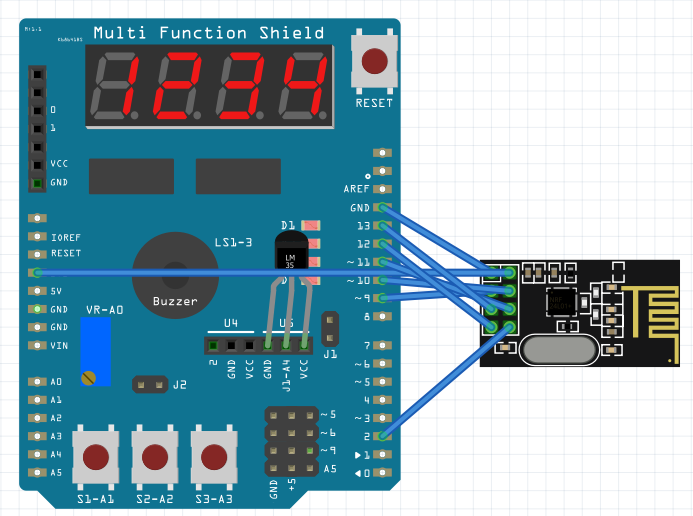
(1<<ADPS2)| // ADC Prescaler Select

(0<<ADPS1)| // Set to

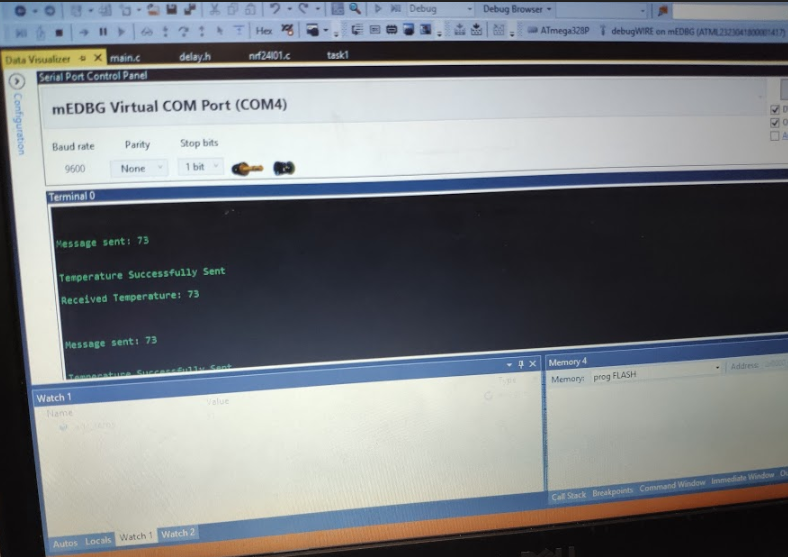
(1<<ADPS0); // 32

}

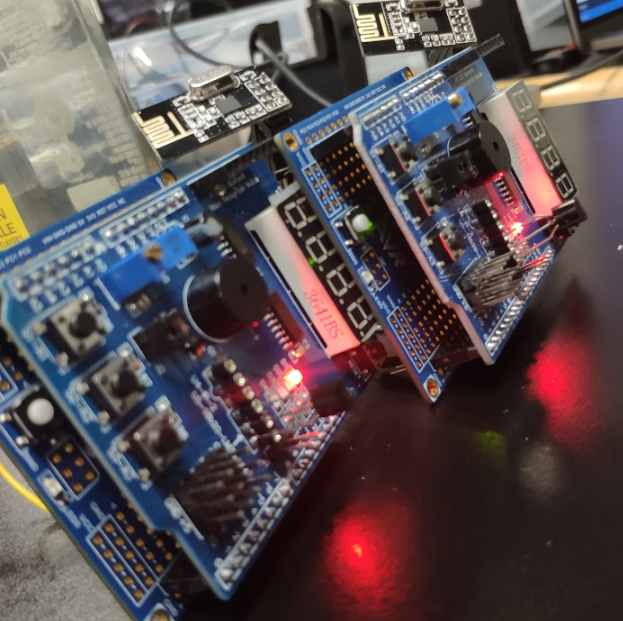
1. **SCHEMATICS**



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



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



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/ZMXrQmhXJXM>

1. **GITHUB LINK OF THIS DA**

<https://github.com/recrio/submissions/tree/master/DesignAssignments/DA5>

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.

Ron Joshua Recrio