CPE301 – SPRING 2019

Design Assignment 5

Student Name: Allis Hierholzer

Student #: 2000160356

Student Email: hierholz@unlv.nevada.edu

Primary Github address: https://github.com/acexhp/submission\_da.git

Directory: Repository/cpe301/DesignAssignment/DA5

Task:

This DA is to be implemented in groups of two students. The goal of the assignment is to develop the above code to do the following:

1. Interface the provided NRF24L01+ RF Module to the ATmega328p using the SPI interface. Using the earlier developed code for ADC, transmit the ADC value of the internal temperature sensor, or LM35 sensor between two RF Modules. The ATmega328p interfacing the RF Modules should alternate between TX and RX modes every 0.5 secs (hopefully they are not both at TX and RX modes in the same interval). The temperature of both ATmega328p’s should be displayed on both ATmega328p’s.

Submission:

The following are required for successful completion of the design assignment:

a. AVR C code that has been compiled and working.

b. The C code should be well documented with explanation of every instruction.

c. A word document that contains the flow chart of the assembly code along with the snapshots of the schematics, components connected on the breadboard and screenshots.

d. Specify your assignment partner’s name on the submission doc.

🡪Partner: Francisco Mata Carlos

1. **COMPONENTS LIST**

* Breadboard
* Wires
* FTDI Chip
* nRF24L01 Single Chip 2.4 GHz Transceiver
* ATMEGA328P XPLAINED MINI
* ATMEL STUDIO 7.0

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

/\*

\* DA5.c

\*/

//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 set\_adc (void);

volatile unsigned char temperature\_value[3];

volatile *uint8\_t* adc\_num;

int main(void)

{

set\_adc();

//Set message to send (message < 32 characters)

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

*strcpy*(tx\_message,"Hello World!"); // String into array

//Initialize UART

uart\_init();

//Initialize nRF24L01+ and print configuration info

nrf24\_init();

print\_config();

//Start listening to incoming messages

nrf24\_start\_listening();

*strcpy*(tx\_message,"Hello World!"); //Copying string to array

nrf24\_send\_message(tx\_message);

while (1)

{

if (message\_received)

{

//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(temperature\_value);

if (status == true) *printf*("Message sent successfully\n");

}

}

}

//Interrupt on IRQ pin

ISR(INT0\_vect)

{

message\_received = true;

}

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");

}

// Interrupt used to follow instructions below when conversion is done

ISR(ADC\_vect)

{

volatile unsigned int i=0; // From characters to string

char temperature[3];

adc\_num = (ADCL >> 1); //Shifts the value right to one place

*itoa*(adc\_num, temperature, 10); //Int to string

while (i<3) //Making string from characters of

temperature\_value

{

temperature\_value[i] = temperature[i];

i++;

}

}

void set\_adc (void)

{

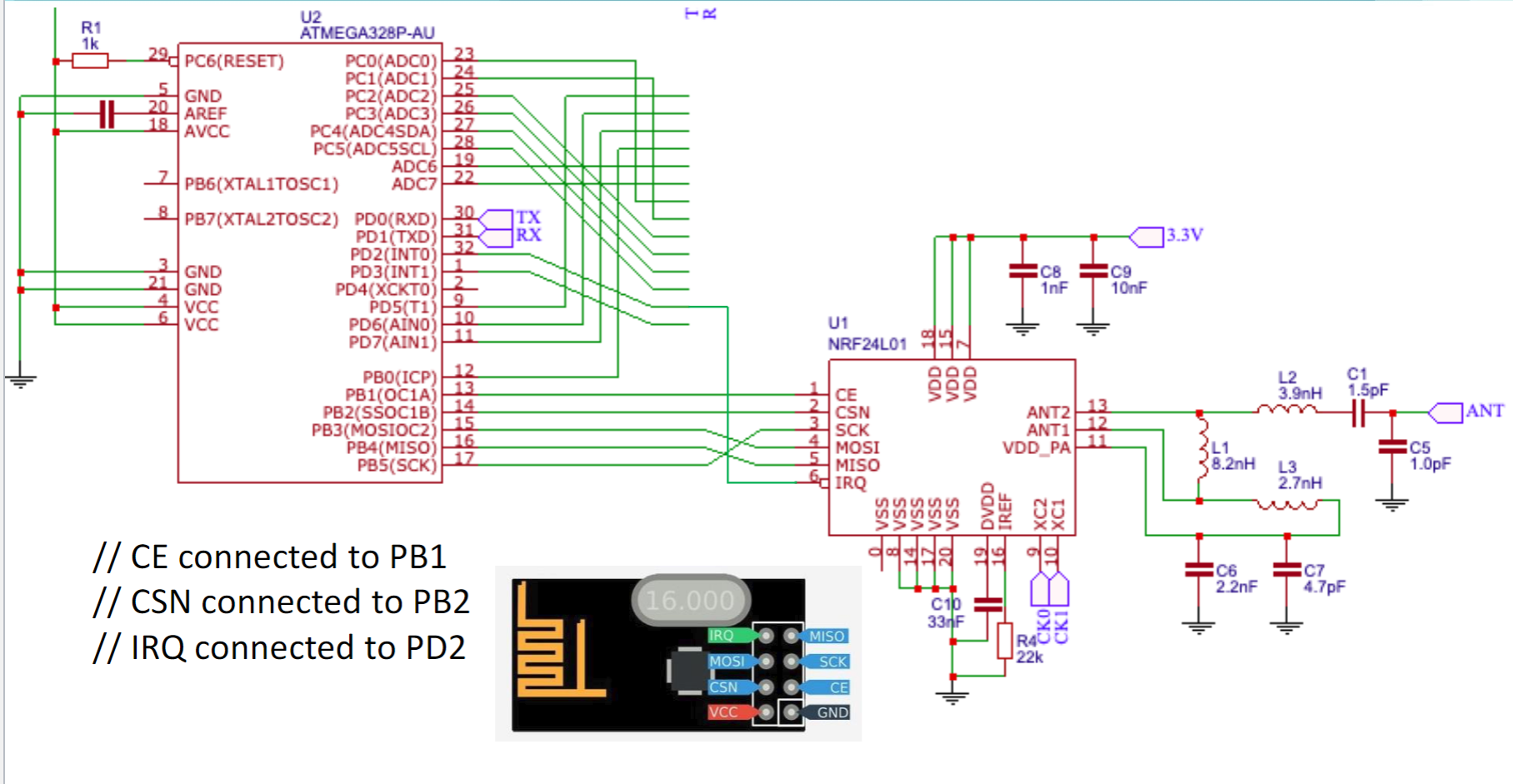
ADMUX = 0x40; //ADC shifts right

ADCSRA = 0xEF; //ADC enable, start conversion, auto trig enable, interrupt

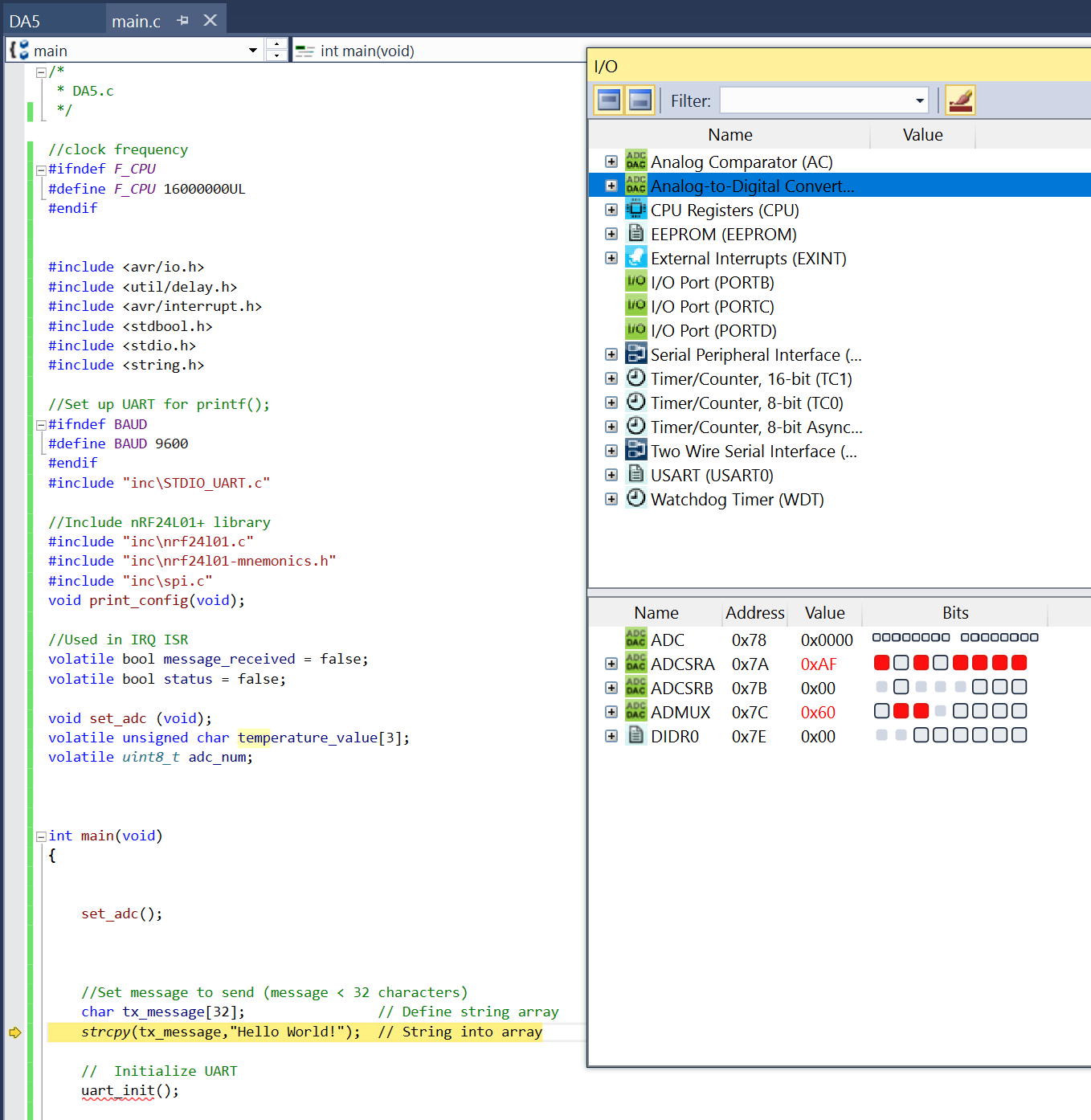
enable, set prescaler

}

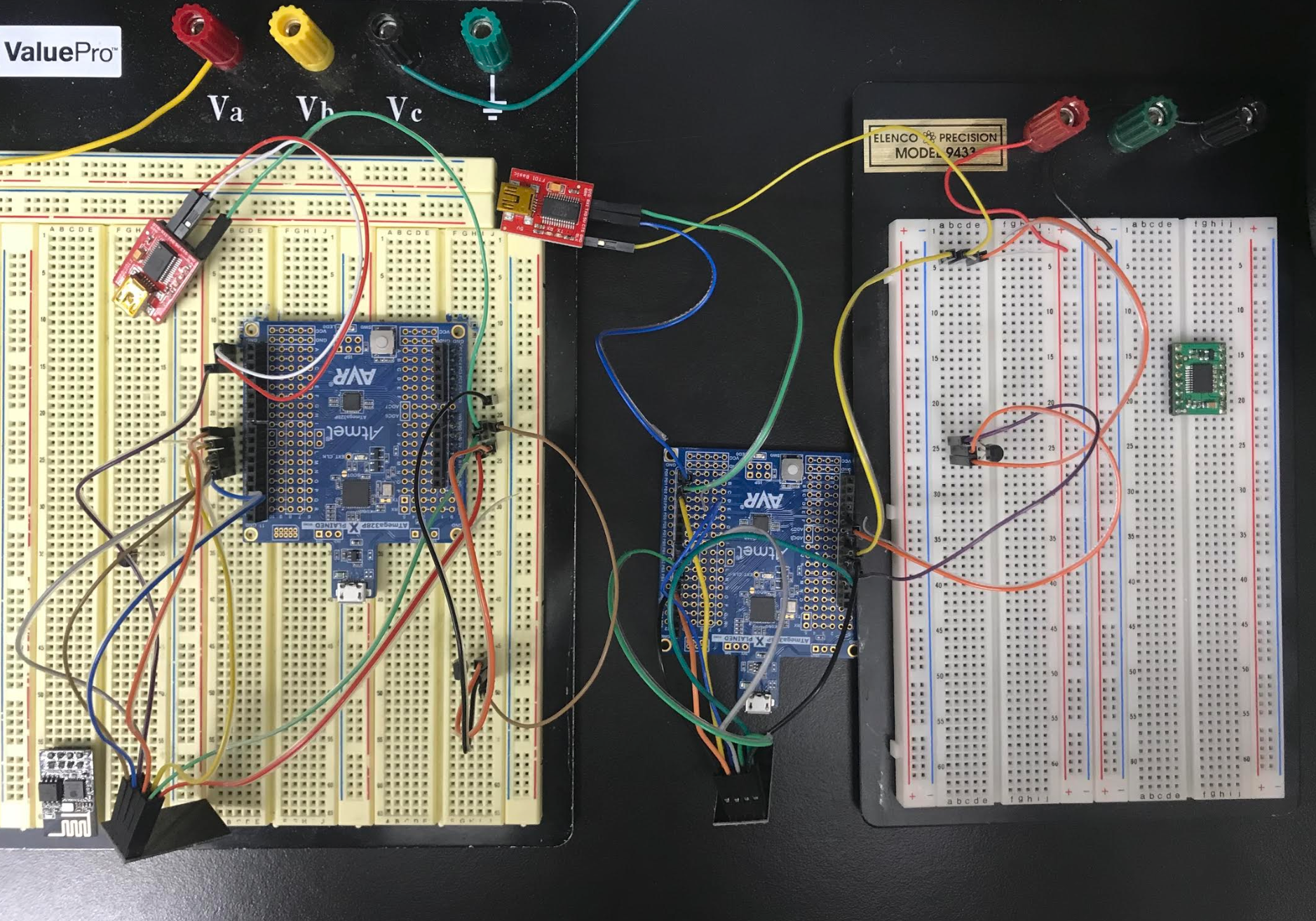
1. **SCHEMATICS**



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/x6FYeoPLdWY**](https://youtu.be/x6FYeoPLdWY)

1. **GITHUB LINK OF THIS DA**

<https://github.com/acexhp/submission_da.git>

**Student Academic Misconduct Policy**

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

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

Allis Hierholzer