CPE301 – FALL 2019

Design Assignment 5

Student Name: Henry Mesa

Student #: 1008099273

Student Email: mesah1@unlv.nevada.edu

Primary Github address: https://github.com/mesah1

Directory: https://github.com/mesah1/submissions

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

List of Components used :

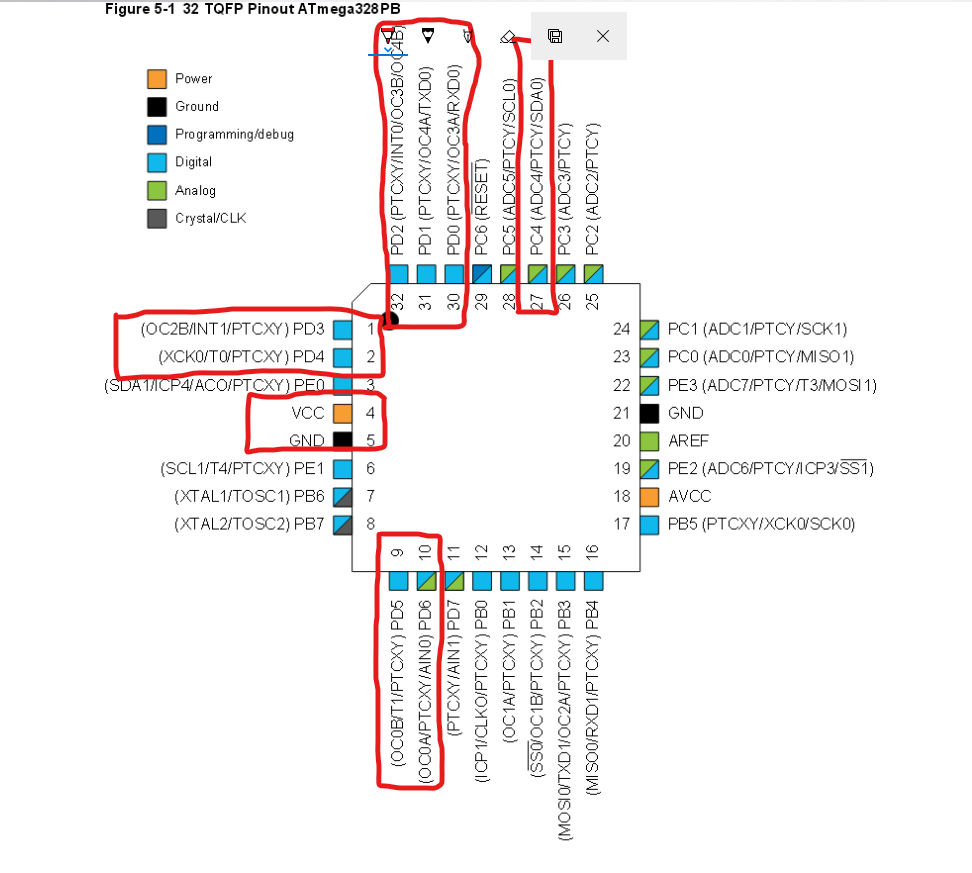
Atmega328PB

NRF24L01

LM35

Multi-Function Shield

Block diagram with pins used in the Atmega328PB:



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

/\*

\* DA5.c

\*

\* Created: 12/3/2019 6:04:07 PM

\* Author : Henry Mesa

\*/

// 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();

#define UBRR\_9600 103

#include "STDIO\_UART.h"

// Include nRF24L01+ library

#include "nrf24l01.h"

#include "nrf24l01-mnemonics.h"

#include "spi.h"

void print\_config(void);

// Used in IRQ ISR

volatile bool message\_received = false;

volatile bool status = false;

void read\_adc(void);

void adc\_init(void);

void USART\_init( );

void USART\_tx\_string(char \*data);

volatile float adc;

char ar[20];

*uint8\_t* rx\_address[5] = {0x44,0x44,0x44,0x44,0x44}; //receiver address

*uint8\_t* rx\_address[5] = {0x72,0x72,0x72,0x72,0x72}; //transmit address

#define READ\_PIPE;

int main(void)

{

adc\_init(); //intialize ADC

USART\_init(UBRR\_9600); //initialize USART

USART\_tx\_string("Hello there!\n");

*\_delay\_ms*(500);

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

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

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

// Initialize UART

USART\_init();

// Initialize nRF24L01+ and print configuration info

nrf24\_init();

print\_config();

// Start listening to incoming messages

nrf24\_start\_listening();

while (1)

{

if (message\_received)

{

// Message received, print it

message\_received = false;

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

// Send message as response

*\_delay\_ms*(500);

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

if (status == true) *printf*("Transmitted Temperature");

}

read\_adc();

*snprintf*(ar, sizeof(ar),"%f\r\n",adc); //print

USART\_tx\_string(ar);

*\_delay\_ms*(1000); //wait 1s

}

}

// 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");

}

/\* INITIALIZING ADC \*/

void adc\_init(void){

//ADC ENABLE AND SET-UP

ADMUX = (0 << REFS1)| // external cap at AREF

(1 << REFS0)| // AVcc - Reference Selection INTERNAL

(0 << ADLAR)| // ADC Left Adjust Result

(1 << MUX2)| // Analog Channel Selection Bits

(0 << MUX1)| // ADC4 (PC4)

(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 = 0;

while (i--)

{

ADCSRA |= (1 << ADSC);

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

adc+= ADC;

*\_delay\_ms*(50);

}

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

}

/\* INITIALIZING USART (RS-232) \*/

void USART\_init(unsigned int ubrr){

UBRR0H = (unsigned char)(ubrr >> 8);

UBRR0L = (unsigned char)ubrr;

UCSR0B = (1 << TXEN0); //transmitter enabled

UCSR0C = (1 << UCSZ01) | (1 << UCSZ00); //5-bit characters

}

/\* SEND STRING TO RS-232 \*/

void USART\_tx\_string(char \*data){

while ((\*data != '\0')){

while (!(UCSR0A & (1 << UDRE0)));

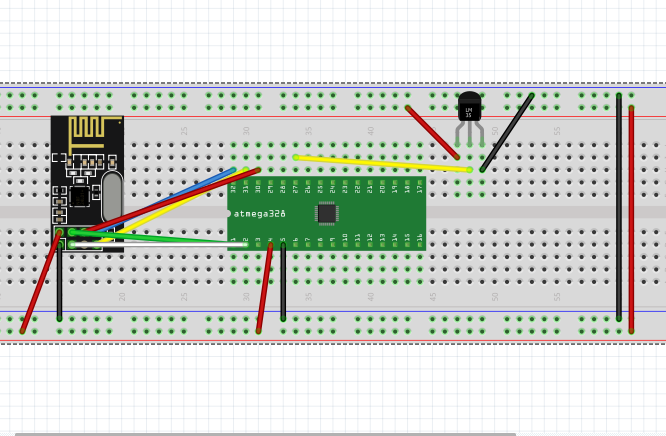
UDR0 = \*data;

data++;

}

}

1. **SCHEMATICS**

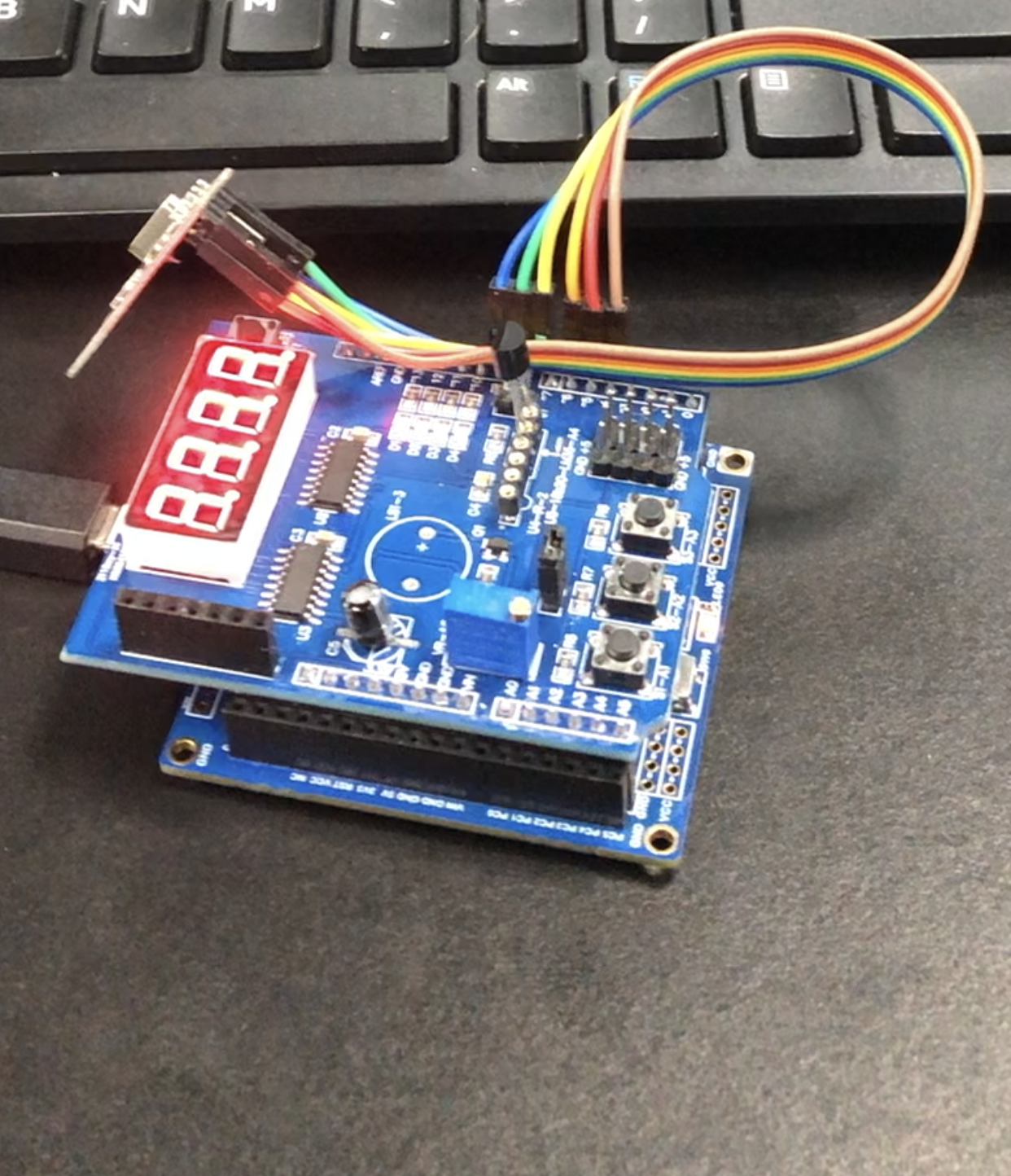


1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

A screenshot of a cell phone

Description automatically generated

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



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/OPBZ5fp9fO4>

1. **GITHUB LINK OF THIS DA**

<https://github.com/mesah1/submissions/tree/master/DA5>

**Student Academic Misconduct Policy**

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

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

Henry Mesa