CPE301 – SPRING 2019

Design Assignment 3A

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

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

List of Components used

Xplained-Mini

Multifunction-Shield

Block diagram with pins used in the Atmega328PB

A screenshot of a cell phone

Description automatically generated

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

/\*

\* DesignAssignment3A.c

\*

\* Created: 10/16/2019 12:59:50 PM

\* Author : Henry Mesa

\*/

#define F\_CPU 16000000UL

#define UBRR\_9600 103

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

//Function declaration

void read\_adc(void);

void adc\_init(void);

void USART\_init( );

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

volatile float adc;

char ar[20];

int main(void)

{

adc\_init(); //intialize ADC

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

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

\_delay\_ms(500);

while (1)

{

read\_adc();

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

USART\_tx\_string("resistance = ");

USART\_tx\_string(ar);

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

}

}

/\* 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

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

(0 << MUX1)| // ADC0 (PC0 PIN23)

(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**

A circuit board

Description automatically generated

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)**

A circuit board

Description automatically generated

1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/uERYIGaK3jE>

1. **GITHUB LINK OF THIS DA**

<https://github.com/mesah1/submissions/tree/master/DA3A>

**Student Academic Misconduct Policy**

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

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

Henry Mesa