CPE301 – SPRING 2025

Design Assignment 4

Student Name: Carlos Funes

Student #: 5007995832

Student Email: [funesc2@unlv.nevada.edu](mailto:funesc2@unlv.nevada.edu)

Primary Github address: <https://github.com/CaFu0320>

Directory: <https://github.com/CaFu0320/submission_da/tree/main/DesignAssignments/DA4>

Video Playlist: <https://youtube.com/playlist?list=PLa3xS6s509hhSrDtIiRlyynXMlhipu13Z&si=vMDsBcMr7TgJm_Hq>

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**

* ATMEGA328PB MINI BOARD
* ARDUINO SHIELD
* UART
* POTENTIOMETER
* CAPACITOR
* ADC CONVERTER

1. **C CODE TASK 1**

/\*

\* DA4

\* Created: 3/19/2025 11:30:20 AM

\* Author : Carlos Funes

\*/

#define *F\_CPU* 16000000UL //defining CPU frequency

#include <avr/io.h> //including AVR I/O register definitions

#include <avr/interrupt.h> //including interrupt handling functions

#include <stdio.h> //including standard I/O functions for snprintf

//UART configuration

void uart\_init(unsigned int ubrr) {

UBRR0H = (unsigned char)(ubrr >> 8); //setting high byte of UART baud rate

UBRR0L = (unsigned char)ubrr; //setting low byte of UART baud rate

UCSR0B = (1 << TXEN0); //turning on the UART transmitter

UCSR0C = (1 << UCSZ01) | (1 << UCSZ00); //8 data bits, 1 stop bit for communication

}

void uart\_transmit(char data) { //character sending

while (!(UCSR0A & (1 << UDRE0))); //wait for empty transmit buffer

UDR0 = data; //putting data into buffer to send

}

void uart\_print(const char\* str) { //string sending

while (\*str) { //looping through each character

uart\_transmit(\*str++); //sending each character

}

}

//ADC configuration with timer auto trigger

void adc\_init(void) {

//configuring ADC

ADMUX = (1 << REFS0); //reference voltage (AVcc)

ADCSRB = (1 << ADTS1) | (1 << ADTS0); //timer0 Compare Match A trigger source

ADCSRA = (1 << ADEN) | (1 << ADATE) | (1 << ADIE) | (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);

//enable ADC, auto-trigger, interrupt, prescaler = 1024

DIDR0 = (1 << ADC0D); //disable digital input on ADC0

}

//timer0 configuration for 10ms interval

void timer0\_init(void) {

TCCR0A = (1 << WGM01); //CTC mode

TCCR0B = (1 << CS02) | (1 << CS00); // prescaler 1024

OCR0A = 155; //10ms

TIMSK0 = (1 << OCIE0A); //enable compare match interrupt

}

volatile *uint16\_t* adc\_value = 0; //stores latest ADC result

volatile *uint8\_t* adc\_ready = 0; //flag indicating new ADC value is ready

//interrupt service routines

//timer0 Compare Match A interrupt

ISR(TIMER0\_COMPA\_vect) {

//ADC triggered automatically by hardware

}

//interrupt for ADC conversion completion

ISR(ADC\_vect) {

adc\_value = ADC; //storing ADC value

adc\_ready = 1; //setting flag for notifying new data is ready to be displayed

}

int main(void) {

uart\_init(103); //9600 baud @ 16MHz

adc\_init(); //initialize ADC with auto-trigger

timer0\_init(); //initialize timer for auto-trigger

*sei*(); //enable global interrupts

char buffer[20]; //holding the formatted voltage value

float voltage; //store the formatted voltage

while (1) { //loop to display data

if (adc\_ready) {

adc\_ready = 0; //clear flag

//convert to voltage (0-5V)

voltage = (adc\_value / 1023.0) \* 5.0;

//format with 0.1V resolution

*snprintf*(buffer, sizeof(buffer), "%.1fV\n", voltage);

uart\_print(buffer);

}

}

}

**A close up of a screen

AI-generated content may be incorrect.**

1. **SCHEMATIC**

**A computer screen shot of a diagram

AI-generated content may be incorrect.**

1. **SCREENSHOTS OF EACH TASK OUTPUT (**

**SERIAL MONITOR**

**A screenshot of a computer

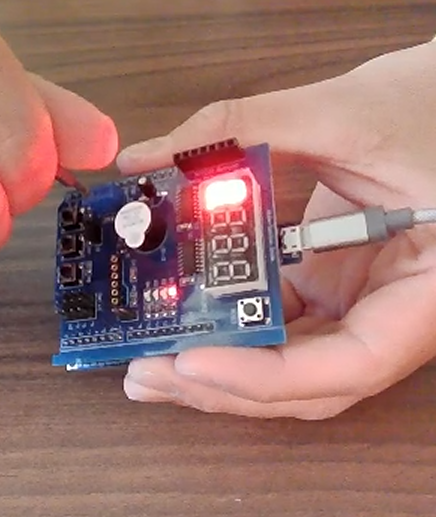
AI-generated content may be incorrect.**

**SERIAL PLOTTER**

**A screen shot of a graph

AI-generated content may be incorrect.**

1. **SCREENSHOT OF DEMO**

****

1. **VIDEO LINK FOR DEMO**

Task:

<https://www.youtube.com/watch?v=y5oWs7Rvzzk&ab_channel=CarlosGordfgg>

1. **GITHUB LINK OF THIS DA**

Tasks c code: <https://github.com/CaFu0320/submission_da/tree/main/DesignAssignments/DA4/DA_4_again>

**Student Academic Misconduct Policy**

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

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

Carlos Funes