CPE301 – SPRING 2024

Design Assignment 5

Student Name: Johnathan Widney
Student #: 5000354900
Student Email: widney@unlv.nevada.edu
Primary Github address: https://github.com/JackOfSpades-7/UNLV-Embeded-Systems
Directory: https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/tree/main
Video Playlist:
https://www.youtube.com/playlist?list=PLoASw0sToF2WMGm5XDNz_vnDBR4LoKq7H

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).

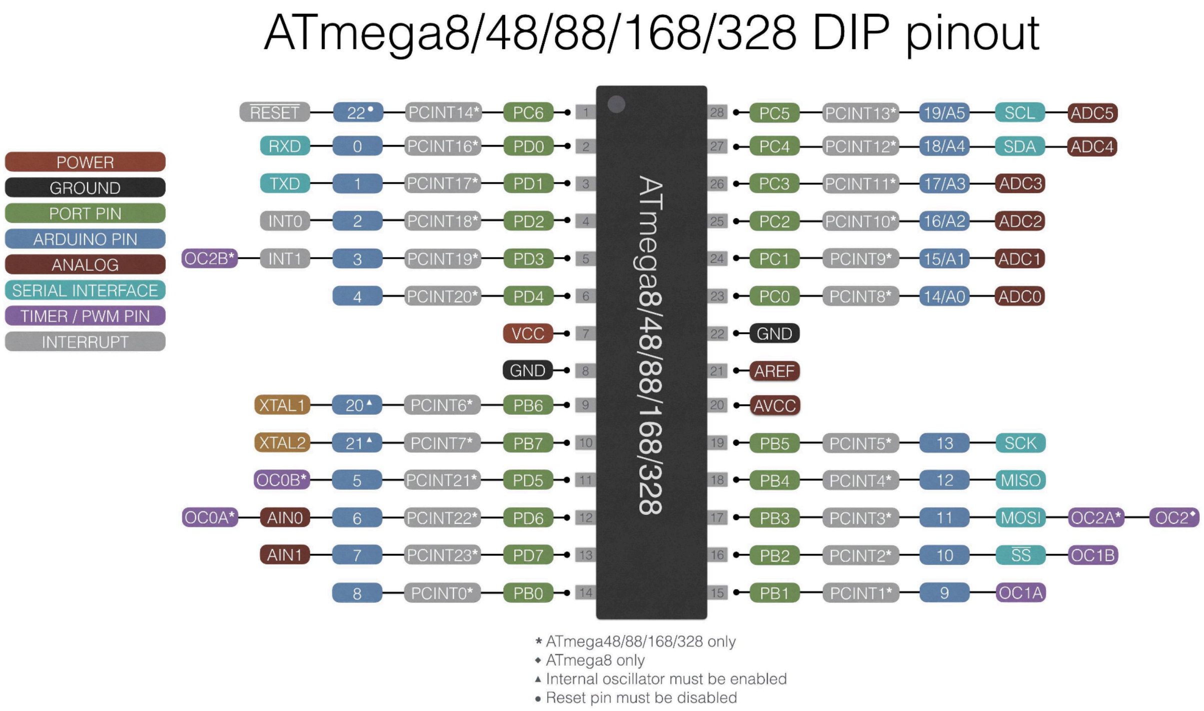
1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

List of Components used
Block diagram with pins used in the Atmega3PB (only)

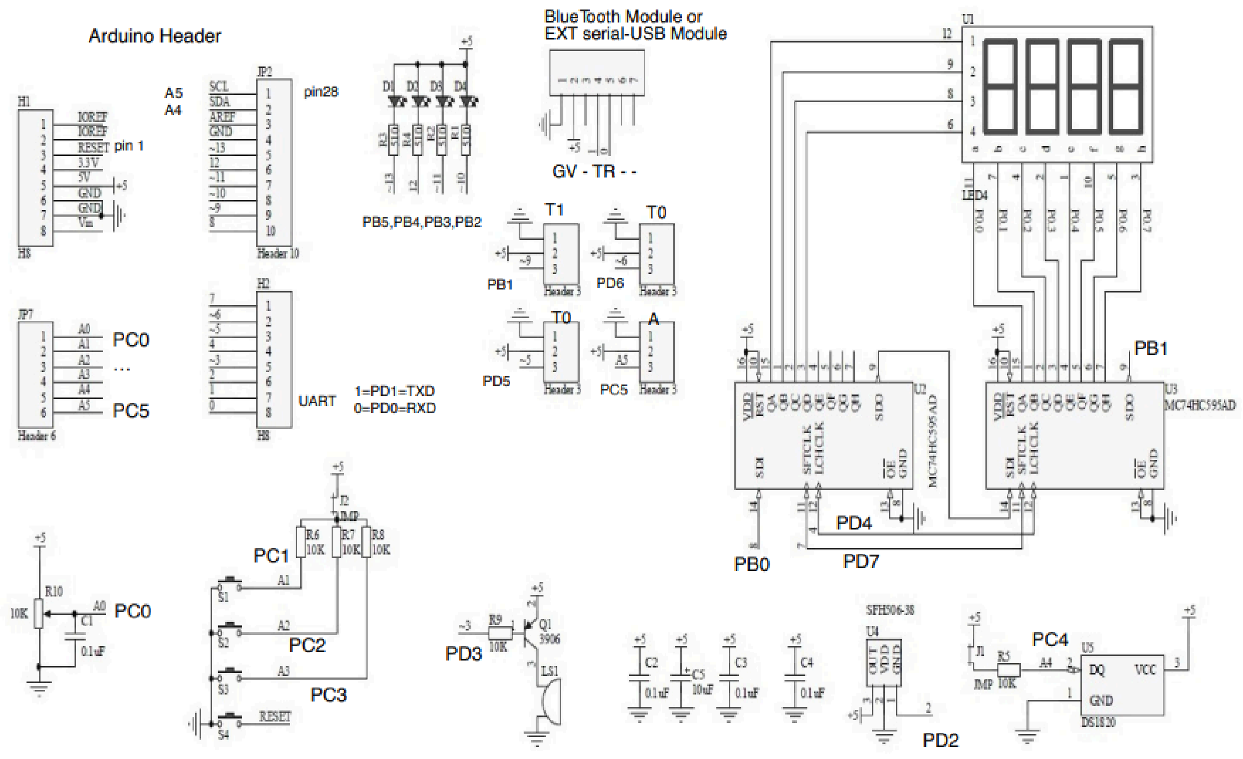
- Atmega328PB Xplained mini microcontroller board
- Arduino compatible external multifunction development shield
- Male-to-male jumper cables
- Mini breadboard
- Ultrasonic sensor
- 180 degree servo motor
- Female-to-female ribbon cable
- PC

Block diagrams and pins:

Atmega328PB Micro controller:



Arduino compatible multifunction development shield:



For assignments dealing with LED - use the pins PB5,PB4,PB3,PB2. For assignments dealing with switches, pin interrupts use pins PC1,PC2,PC3
For assignments in PWM use ~5/PD5 (T0B), ~6/PD6 (T0A), ~9/PB1 (T1A), ~10/PB2 (T1B-LED), ~11/PB3(T2A-LED),~3/PD3(T2B/Buzzer)
For assignments with analog input use A0/PC0-Potentiometer, A4/PC4/LM3X, or EXT @ PC5. PD2 is INT0 pin (external interrupt)

2. INITIAL/MODIFIED/DEVELOPED CODE OF TASK 1

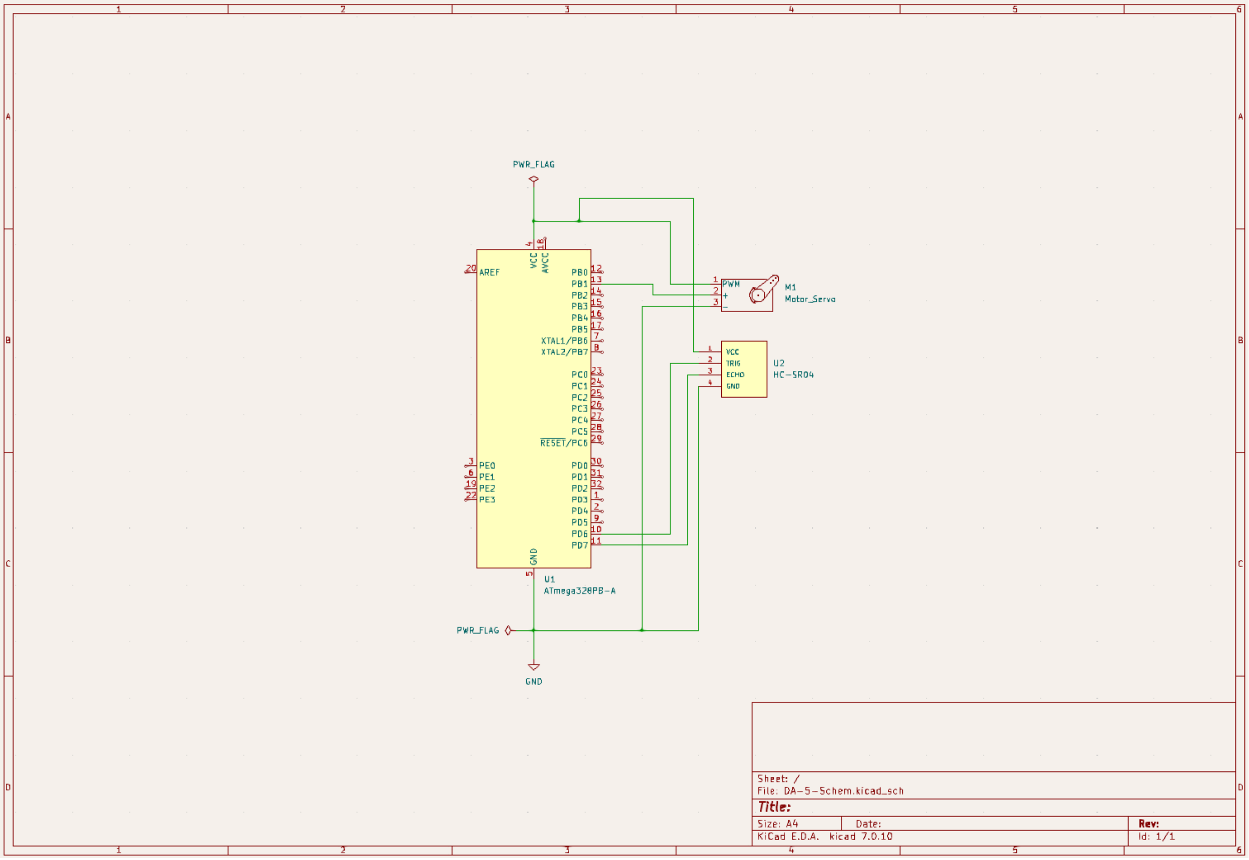
```
/* This code contains all of task 1 */

// Initializing clock, constants, libraries, variables, etc.
#define F_CPU 16000000UL
#define MAX 535
#define MIN 115
#define STEPS (MAX-MIN)/105
#include <avr/io.h>
#include <util/delay.h>

// Defines Trig and Echo pins of the Ultrasonic Sensor
#define trigPin 10
#define echoPin 11

// Define constants
#define WIDTH 1200
#define HEIGHT 680
#define PI 3.14159265358979323846
```

```
// Global Variable
int delayTime = 125;

// Function prototypes
void drawRadar();
void drawObject(int angle, int distance);
void drawLine(int angle);
void drawText(int angle, int distance, char* noObject);

int main()
{
  // Servo motor //
	// Configuring TIMER1: non-inv, prescale 64, fast PWM
	TCCR1A|=(1<<COM1A1)|(1<<COM1B1)|(1<<WGM11);
	TCCR1B|=(1<<WGM13)|(1<<WGM12)|(1<<CS11)|(1<<CS10);

	ICR1=4999;  // set period to 20ms

	DDRB|=(1<<PINB1);   // setting out pin

	while(1)
	{
		for(int i = MIN; i <= MAX; i += STEPS) {
			OCR1A = i;
			_delay_ms(150);
		}
		for(int i = MAX; i >= MIN; i -= STEPS) {
			OCR1A = i;
			_delay_ms(150);
		}
	}

  // Ultrasonic sensor //
  // Initialize variables
	char angle[4] = "";
	char distance[4] = "";
	char data[10] = "";
	char noObject[8] = "";
	int iAngle, iDistance;
	int index1 = 0;

	// Simulate serial input
	strcpy(data, "30,15.");
	index1 = strchr(data, ',') - data;
	strncpy(angle, data, index1);
	strcpy(distance, data + index1 + 1);
	iAngle = atoi(angle);
	iDistance = atoi(distance);

	// Check if object is detected
	if (iDistance > 40) {
        strcpy(noObject, "Nothing");
        } else {
        strcpy(noObject, "Detect");
    }

    // Call drawing functions
    drawRadar();
    drawObject(iAngle, iDistance);
    drawLine(iAngle);
    drawText(iAngle, iDistance, noObject);

    return 0;
}

void drawRadar() {
    int i;
    float radius1 = (WIDTH - WIDTH * 0.0625) / 2.0;
    float radius2 = (WIDTH - WIDTH * 0.27) / 2.0;
    float radius3 = (WIDTH - WIDTH * 0.479) / 2.0;
    float radius4 = (WIDTH - WIDTH * 0.687) / 2.0;
    float x, y;

    printf("Drawing radar...\n");

    // Draw arc lines
    for (i = 0; i <= 180; i++) {
        x = WIDTH / 2 + radius1 * cos(i * PI / 180);
        y = HEIGHT - HEIGHT * 0.074 - radius1 * sin(i * PI / 180);
        printf("Drawing point at (%.2f, %.2f)\n", x, y);
    }

    for (i = 0; i <= 180; i++) {
        x = WIDTH / 2 + radius2 * cos(i * PI / 180);
        y = HEIGHT - HEIGHT * 0.074 - radius2 * sin(i * PI / 180);
        printf("Drawing point at (%.2f, %.2f)\n", x, y);
    }

    // ... (code for drawing other arc lines)
}

void drawObject(int angle, int distance) {
    float pixsDistance = distance * ((HEIGHT - HEIGHT * 0.1666) * 0.025);
    float x1, y1, x2, y2;

    printf("Drawing object...\n");

    if (distance < 40) {
        x1 = WIDTH / 2 + pixsDistance * cos(angle * PI / 180);
        y1 = HEIGHT - HEIGHT * 0.074 - pixsDistance * sin(angle * PI / 180);
        x2 = WIDTH / 2 + (WIDTH - WIDTH * 0.505) * cos(angle * PI / 180);
        y2 = HEIGHT - HEIGHT * 0.074 - (WIDTH - WIDTH * 0.505) * sin(angle * PI / 180);
        printf("Drawing line from (%.2f, %.2f) to (%.2f, %.2f)\n", x1, y1, x2, y2);
```

```
	}
}

void drawLine(int angle) {
	float x2 = (HEIGHT - HEIGHT * 0.12) * cos(angle * PI / 180);
	float y2 = -(HEIGHT - HEIGHT * 0.12) * sin(angle * PI / 180);

	printf("Drawing line...\n");
	printf("Drawing line from (%d, %d) to (%.2f, %.2f)\n", WIDTH / 2, HEIGHT - HEIGHT * 0.074, WIDTH /
2 + x2, HEIGHT - HEIGHT * 0.074 - y2);
}

void drawText(int angle, int distance, char* noObject) {
	printf("Drawing text...\n");
	printf("Object: %s\n", noObject);
	printf("Angle: %d degrees\n", angle);
	printf("Distance: %d cm\n", distance);
	// ... (code for drawing other text elements)
}

// Function for calculating the distance measured by the Ultrasonic sensor
int calculateDistance(){

	digitalWrite(trigPin, LOW);
	delayMicroseconds(2);
	// Sets the trigPin on HIGH state for 10 micro seconds
	digitalWrite(trigPin, HIGH);
	delayMicroseconds(10);
	digitalWrite(trigPin, LOW);
	duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in
microseconds
	distance= duration*0.034/2;
	return distance;
}
```

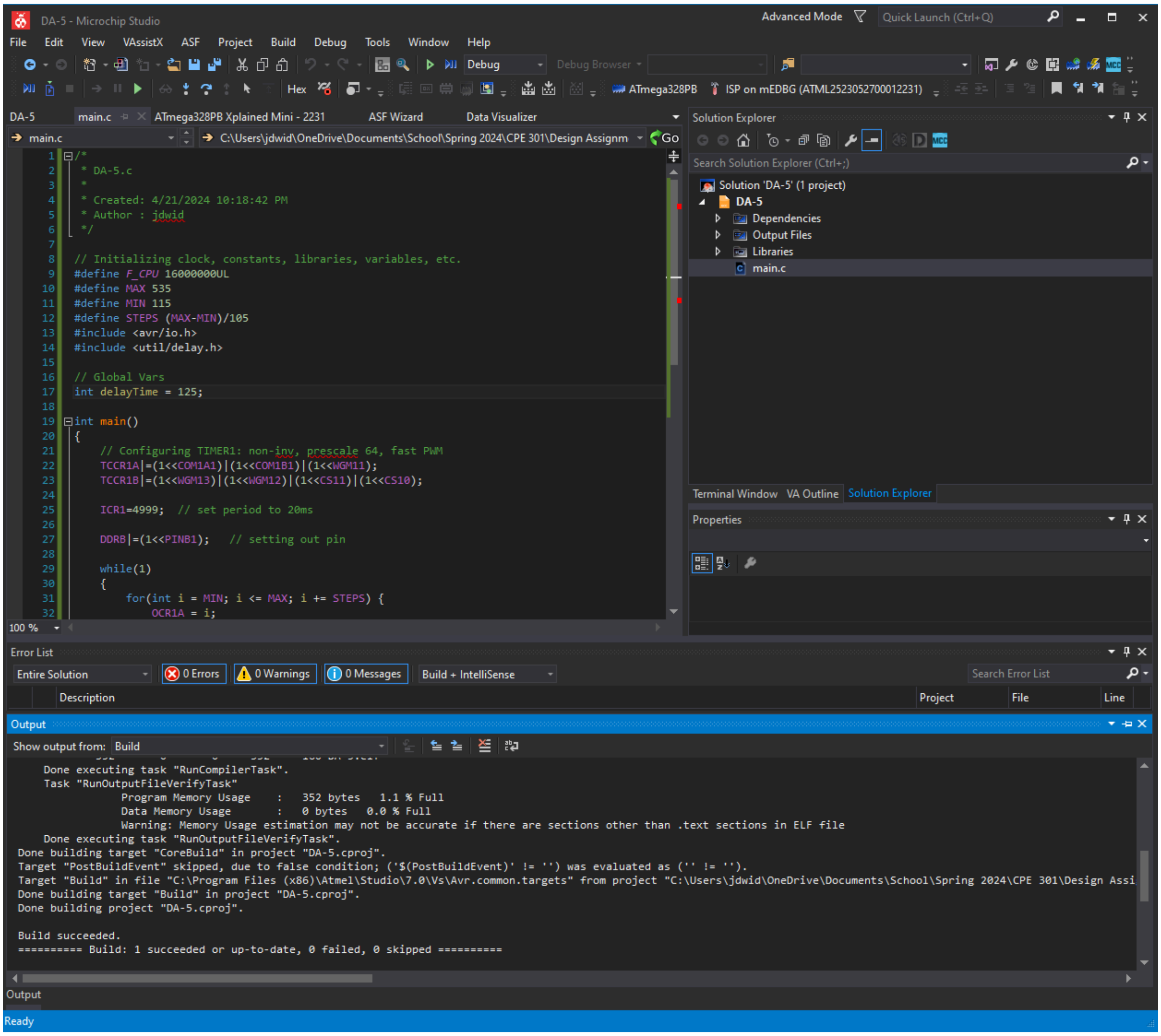
3. SCHEMATICS

Use KICAD schematics only (not required for DA1 simulation)



4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Task 1:

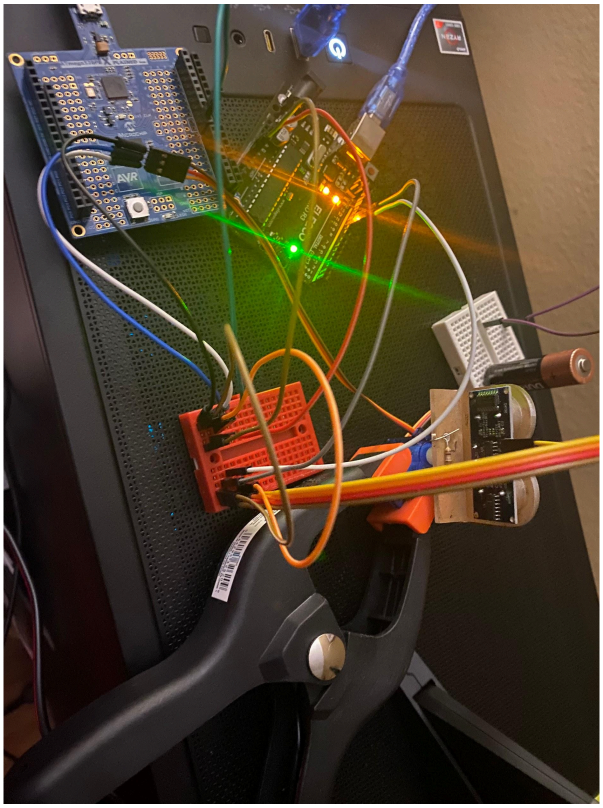


```c
/*
 * DA-5.c
 *
 * Created: 4/21/2024 10:18:42 PM
 * Author : jdwid
 */

// Initializing clock, constants, libraries, variables, etc.
#define F_CPU 16000000UL
#define MAX 535
#define MIN 115
#define STEPS (MAX-MIN)/105
#include <avr/io.h>
#include <util/delay.h>

// Global Vars
int delayTime = 125;

int main()
{
    // Configuring TIMER1: non-inv, prescale 64, fast PWM
    TCCR1A|=(1<<COM1A1)|(1<<COM1B1)|(1<<WGM11);
    TCCR1B|=(1<<WGM13)|(1<<WGM12)|(1<<CS11)|(1<<CS10);

    ICR1=4999;  // set period to 20ms

    DDRB|=(1<<PINB1);   // setting out pin

    while(1)
    {
        for(int i = MIN; i <= MAX; i += STEPS) {
            OCR1A = i;
```

```
    Done executing task "RunCompilerTask".
    Task "RunOutputFileVerifyTask"
                Program Memory Usage    :   352 bytes   1.1 % Full
                Data Memory Usage       :   0 bytes   0.0 % Full
                Warning: Memory Usage estimation may not be accurate if there are sections other than .text sections in ELF file
    Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "DA-5.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Users\jdwid\OneDrive\Documents\School\Spring 2024\CPE 301\Design Assi
Done building target "Build" in project "DA-5.cproj".
Done building project "DA-5.cproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Task 1:



6. VIDEO LINKS OF EACH DEMO

Task 1: https://youtu.be/oTQY63WrAl0

7. GITHUB LINK OF THIS DA

Task 1:
https://github.com/JackOfSpades-7/UNLV-Embeded-Systems/tree/main/Design%20Assignment%205

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"This assignment submission is my own, original work".
Johnathan Widney