

## EE 2305 – Introduction to C Programming Hardware Project 05

Select one of the following projects and document your work with the following sections. Provide a brief description of the system and how you are designing it to operate.

### A. Hardware Diagram:

Provide a hardware diagram of the components.

### B. Program Flowchart:

Draw a flowchart of the program.

### C. Arduino Source Code

Insert the Arduino Source Code into the document.

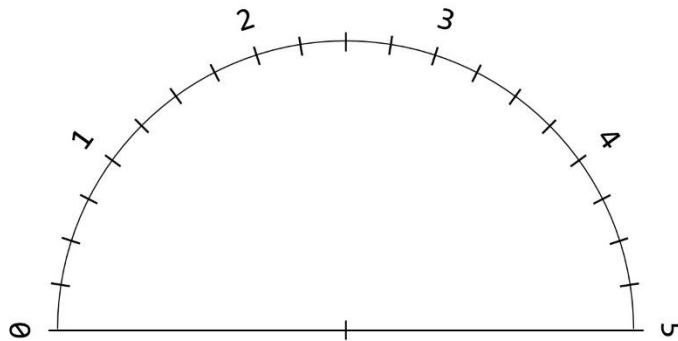
### D. Demonstration Video

Record and upload a video demonstrating the operation of the circuit.

Save the document as a *PDF* file and submit the *PDF* document to *Blackboard*.

## 1. Servo Voltmeter

Program an Arduino board to accept an analog input voltage and display the magnitude of the voltage using a Hobby Servo to display the voltage. Use the servo horn to direct an arrow to the appropriate voltage. Provide as much precision in the display as possible.



## 2. Joystick motor and Servo Control.

Program an Arduino to accept the input of a two-axis joystick. One axis of the joystick shall control the direction and speed of a DC motor. The second axis of the joystick shall control the position of a servo.

### **3. Stepper Motor clock**

Program an Arduino to control a stepper motor to position the second hand of an analog clock.

### **4. Temperature Controlled Fan**

Program an Arduino to accept the input of a temperature sensor. The temperature shall control the speed of a fan motor.

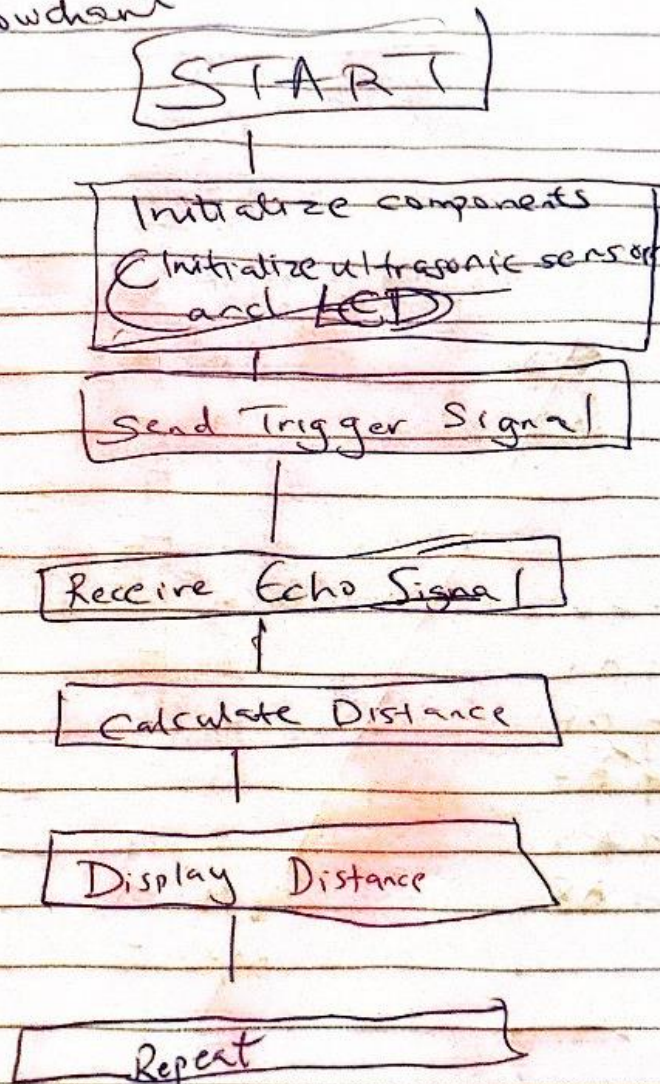
### **5. Ultrasonic Ruler**

Program an Arduino to accept the input from an ultrasonic distance sensor. Display the measured distance of an alphanumeric display.

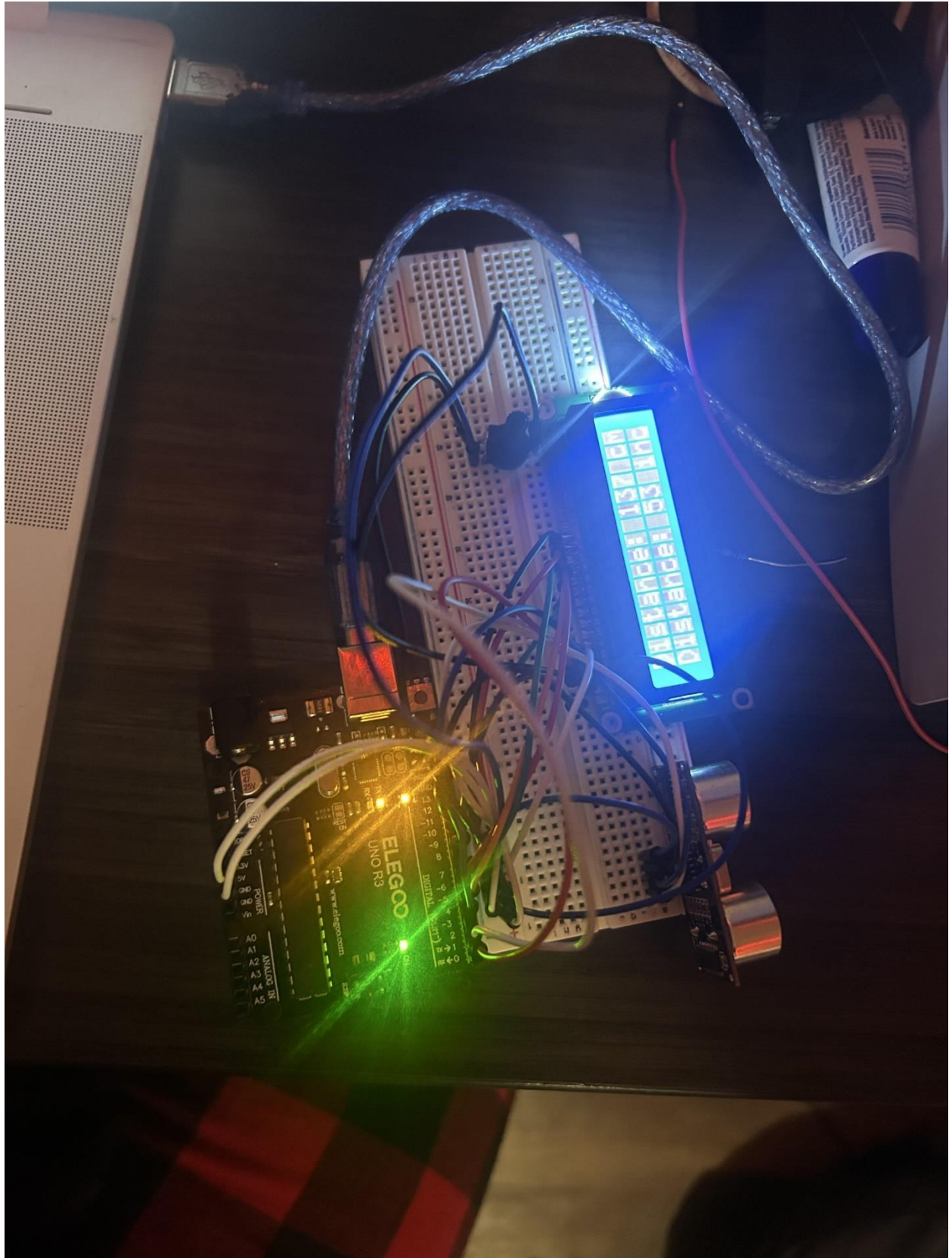
Ultrasonic Ruler

Flowchart:

## ~~Hardware~~ Flowchart



Hardware Diagram:



Code:

```
#include <LiquidCrystal.h> // Use angle brackets for including libraries

LiquidCrystal lcd(1, 2, 4, 5, 6, 7); // Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)

const int trigPin = 9;
const int echoPin = 10;
long duration;
int distanceCm, distanceInch;

void setup() {
  lcd.begin(16, 2); // Initializes the interface to the LCD screen, and specifies the dimensions
  (width and height) of the display
  pinMode(trigPin, OUTPUT); // Set TRIG pin as output
  pinMode(echoPin, INPUT); // Set ECHO pin as input
}

void loop() {
  // Send a pulse from the TRIG pin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the ECHO pin and calculate the duration
  duration = pulseIn(echoPin, HIGH);

  // Convert duration to distances
  distanceCm = duration * 0.034 / 2; // Convert to centimeters
  distanceInch = duration * 0.0133 / 2; // Convert to inches

  // Display the distance in centimeters on the first row
  lcd.setCursor(0, 0);
  lcd.print("Distance: ");
  lcd.print(distanceCm);
  lcd.print(" cm");

  // Display the distance in inches on the second row
  lcd.setCursor(0, 1);
  lcd.print("Distance: ");
  lcd.print(distanceInch);
  lcd.print(" inch");

  delay(500); // Wait for 0.5 seconds before next measurement
```

}