

Florida International University

Project #4
Functions & Loops
EML4804 Introduction to Mechatronics

Submitted by:
Ela Nunez
ID #: 6296607
Date of submission:
06/02/24

Summer 2024

Introduction

The goal of this project was to conduct mathematical calculations, create functions, utilize loops, and timing functions. This was achieved with an LCD circuit that calculated and displayed values of a mathematical expression with given inputs and step size. The specific inputs I used were from option #7:

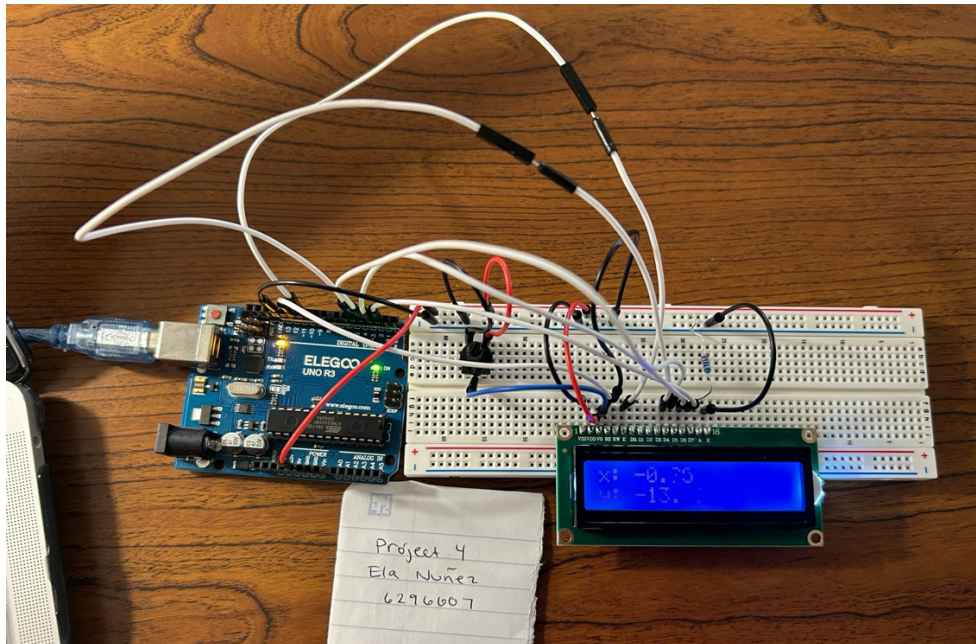
Expression	Range of “x”	Step size
$\cos\left(\frac{1}{x}\right) \cdot x^3 - 15 \frac{\sin(x)}{x}$	-1.2 to 1.2	0.05

Functions in an Arduino sketch are typically only written in one place and called from that location every time they’re used. There are only 2 necessary functions for an Arduino sketch: setup() and loop(). Other functions must be created and named outside the brackets of those functions. Functions are called by name within the setup() or loop() by inputting parameters in the code.

Materials used:

1. Arduino board
2. Breadboard
3. LCD screen
4. 10k ohm potentiometer
5. 220-ohm resistor
6. Wires

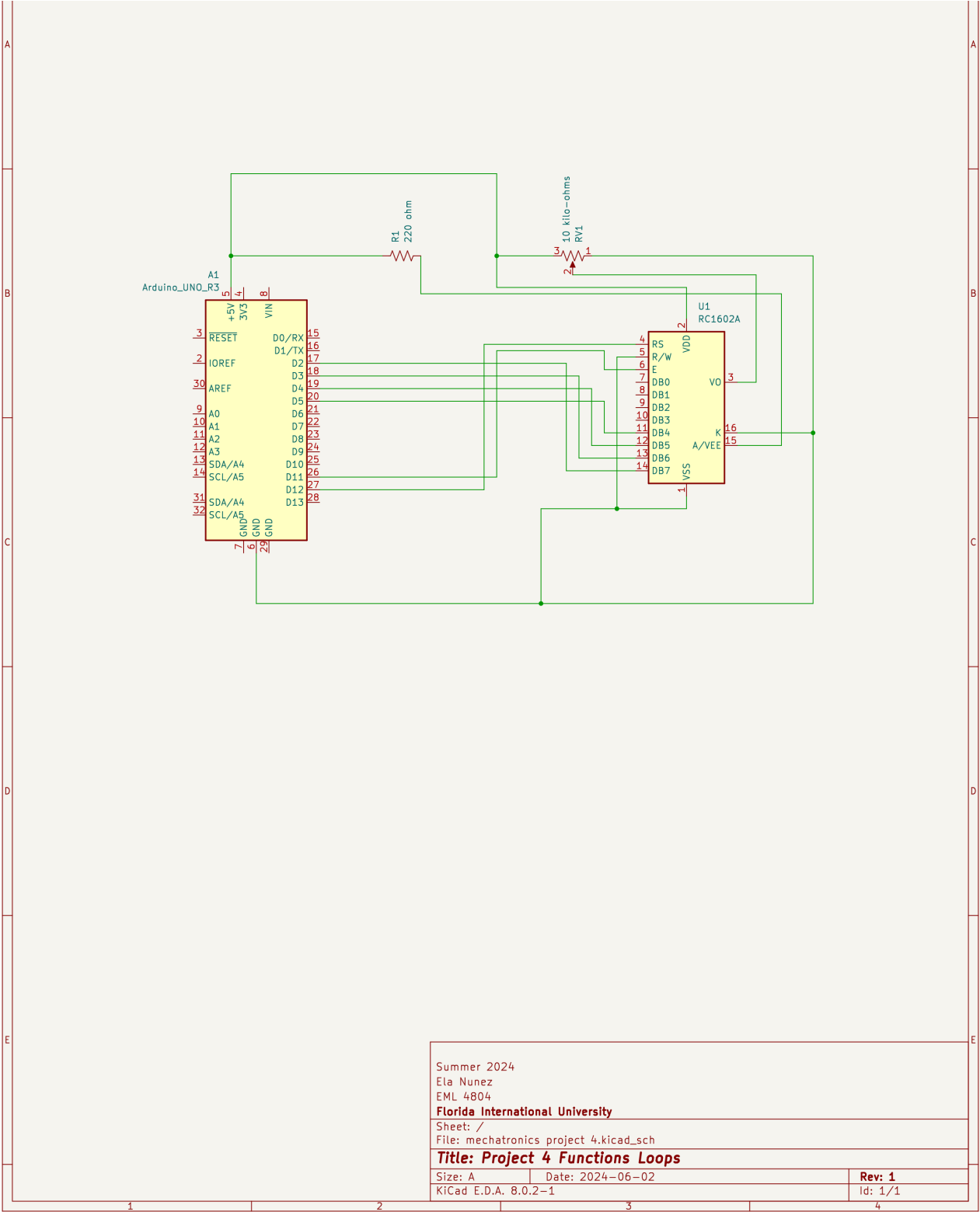
Picture



Video link

<https://www.youtube.com/shorts/axuM-Gt8E4g>

Diagram



Code

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
}

// Set up function in its own loop
// use "float" to input decimal variables
float myFunction(float x){
  float y;
  y = cos(1/x) * pow(x,3) - 15 * (sin(x)/x); // define function
  return y; // return the solution
}

void loop() {
  // (note: line 0 is the first row, since counting begins with 0):
  lcd.setCursor(0, 0); // set the cursor to column 0, line 0
  lcd.print("y=cos(1/x)*x^3-15*sin(x)/x"); // print a message to the LCD
  lcd.setCursor(0, 1); // set the cursor to column 0, line 1
  lcd.print("x:-1.2..1.2,s:.05"); // print message
  delay(2000); // 1 second delay
  // scroll 25 positions (string length) to the left to move it offscreen left:
  for (int positionCounter = 0; positionCounter < 25; positionCounter++) {
    // scroll one position left:
    lcd.scrollDisplayLeft();
    // wait a bit (0.35 seconds):
    delay(350);
  }
  lcd.clear(); // clear the display for next loop

  // define starting input value; ending input value; step size
  for (float i = -1.2; i <= 1.2; i += 0.05){
    lcd.setCursor(0, 0);
    lcd.print("x: ");
    lcd.print(i);
    lcd.setCursor(0, 1);
    lcd.print("y: ");
    lcd.print(myFunction(i));
    delay(150); // wait 0.15 seconds to print the next i/o set
```

```

    lcd.clear(); // clear the display for the next i/o set
}
delay(500); // wait 0.5 seconds to display the next message

// Define variables to store the maximum and minimum values
float maxValue = myFunction(-1.2); // Initialize to the smallest possible value
float minValue = myFunction(-1.2); // Initialize to the largest possible value

// Iterate over the range of values for i
for (float i = -1.2 + 0.05; i <= 1.2; i+= 0.05) {
    // Calculate the function value
    float functionValue = myFunction(i);

    // Update the maximum and minimum values
    maxValue = max(maxValue, functionValue);
    minValue = min(minValue, functionValue);
}

// Print the maximum and minimum values on the LCD display
lcd.setCursor(0, 0);
lcd.print("max: ");
lcd.print(maxValue);

lcd.setCursor(0, 1);
lcd.print("min: ");
lcd.print(minValue);
delay(2000); // add a delay to ensure the results are printed
}

```

Conclusions

This project was the most challenging one so far, since it required us to create some of the code on our own. It reinforced that using established code and adjusting it to my needs is better than writing a script from scratch. I learned that it is important to define decimal variables with “float”, since “int” is designed for integer variables. I also learned the pow() command, how to set the input bounds and step size of a for loop, and the importance of using delay() within the loop() to display variables.

My process for this project was to separate it into 3 parts: display the function and parameters, display each input and output set, and display the minimum and maximum output values. Since my function and parameters took up more than 16 characters, I implemented the “scroll display left” code provided by Arduino’s library. To display each input and output of my function, I defined the function, initiated, and returned variables to be used in said function outside of any other function. Then I created a for loop within the given range, with the given set size, and printed the result. Finally, I initialized maximum and minimum values and compared them to output values of the function using

max() and min() commands within a for loop, then printed the resulting maximum and minimum. For this step, I used the help of AI to get started.

The largest problem I encountered had an easy fix. After displaying all i/o values, I wrote code to display output maximum and minimum. Instead, the LCD looped back to displaying the function and parameters, skipping over the maximum and minimum display. At first, I thought that the script was skipping this section of my code entirely, but after testing it on its own in another script I concluded that couldn't be true. Still, something was missing, so I reached out to a friend in the class who explained that I had forgotten to add a delay() after lcd.print() – essentially the LCD wasn't given enough time to display my command.

References

1. LCD circuit & reference code for scroll display left:
<https://docs.arduino.cc/learn/electronics/lcd-displays/>
2. How to use functions in a sketch:
<https://docs.arduino.cc/learn/programming/functions/>
3. Integer vs decimal errors – use “float”:
<https://www.baldengineer.com/arduino-math-fixes.html>
4. ChatGPT to check other errors in code: <https://chatgpt.com/?oai-dm=1>