Scientific Calculator Using AVR-GCC

EE24BTECH11001

March 24, 2025

1 Overview

1.1 Features

The calculator supports:

- Basic operations: addition (+), subtraction (−), multiplication (∗), division (/), and exponentiation (ˆ), factorial (!).
- Trigonometric functions: sin, cos, tan, arcsin, arccos, arctan.
- Logarithmic functions: natural log (ln), log base 10 ($\log_{10}$).
- Constants: π, e.
- Memory recall functionality.
- Autocompleting bracketis, movable cursor (to traverse code).

Components Used

The hardware consists of:

Quantity	Component
25	Pushbuttons
1	LCD 16 x 2
1	Arduino Uno
-	Wires
1	Potentiometer

Table 1: Materials Required

- A button matrix for user input.
- An Arduino Uno microcontroller to process inputs and execute calculations.
- Connecting a 16x2 LCD to arduino for displaying results.
- Connections between the button matrix, LCD, and Arduino Uno as shown in Figure 1.

The schematic for connections is as shown below,

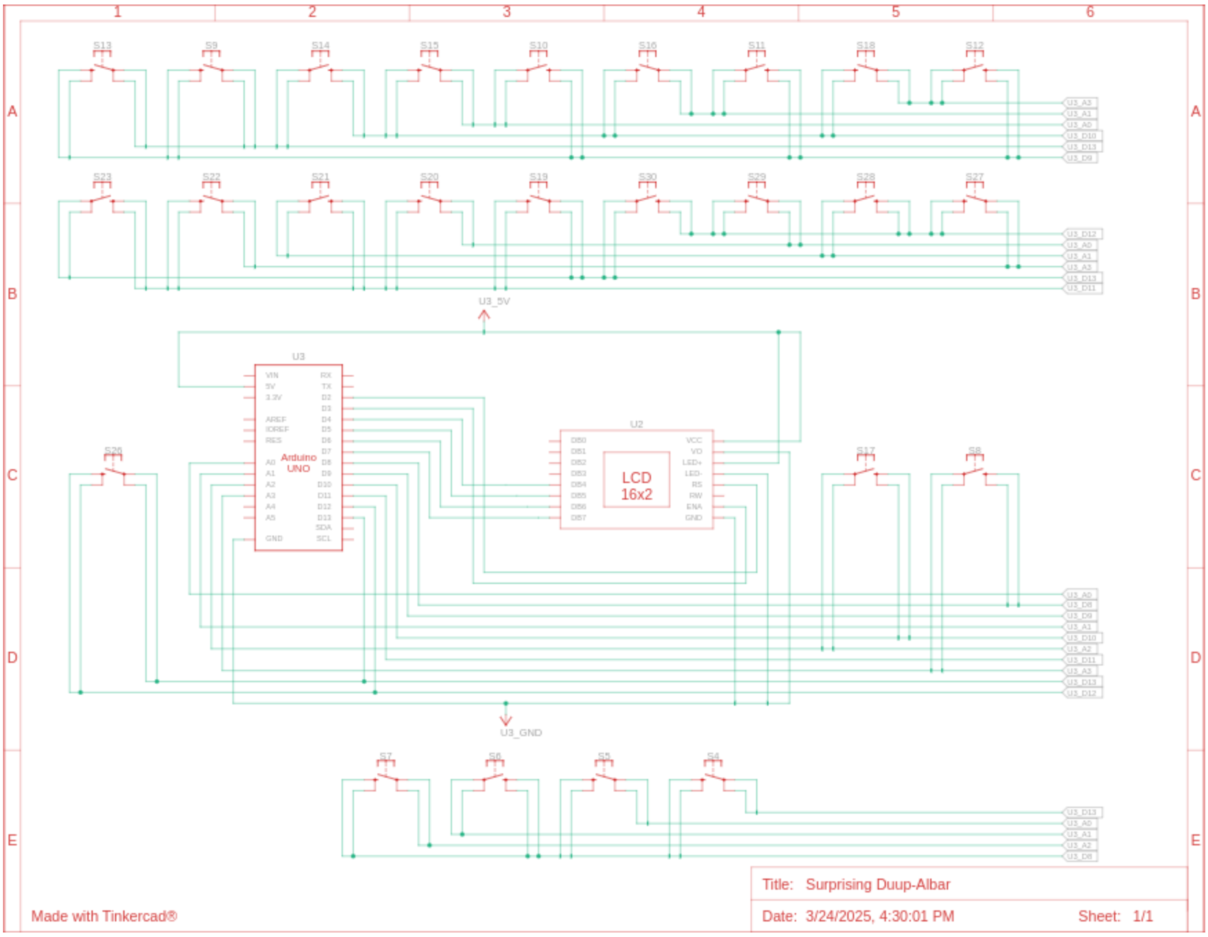


Figure 1: Schematic of Circuit.

Code Implementation

The code is implemented using avr-gcc. It includes:

- Implementing button matrix to check which key is pressed using software multiplexing.
- Storing input as a string (infix)
- Converting infix to postfix using Shunting Yard algorithm.
- Evaluating postfix and returning answer.
- Support for functions like trigonometric functions (normal and inverse) logarithms, factorials, etc. All done without using **math.h** library, solving differential equations by RK4 method.

Now let us explore some of the features mentioned above in greater depth.

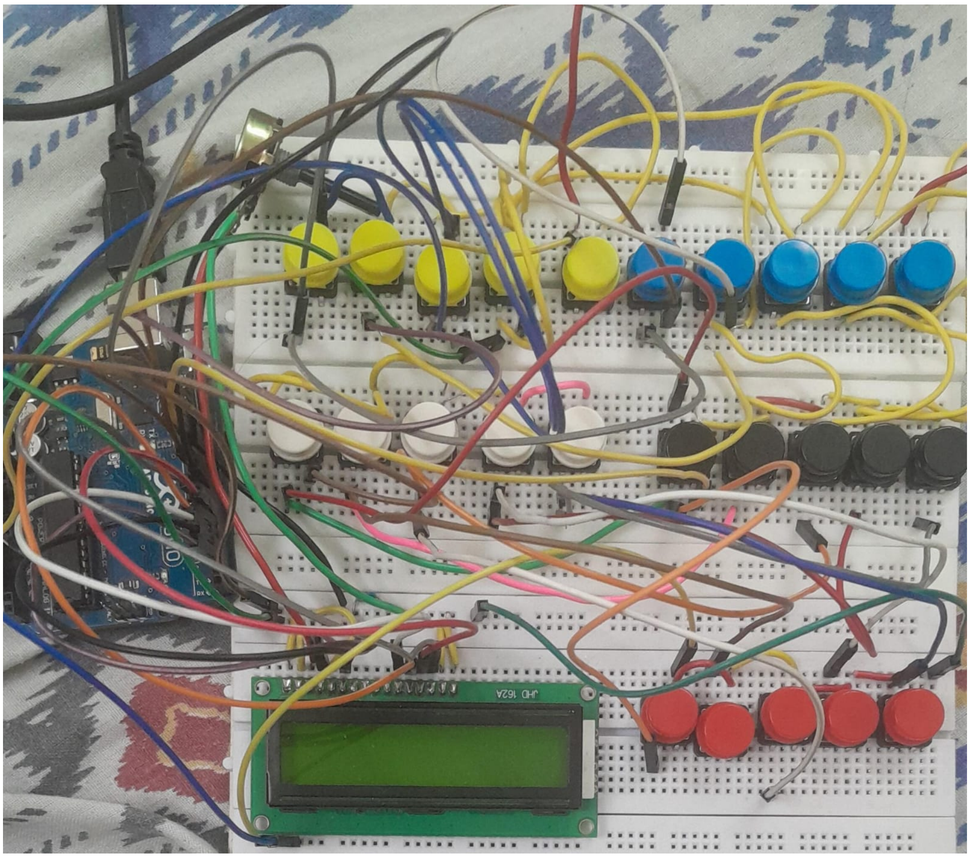


Figure 2: Circuit Image

Arduino Pins	LCD Pins	LCD Pin Label	LCD Pin Description
GND	1	GND	
5V	2	Vcc	
GND	3	Vee	Contrast
D2	4	RS	Register Select
GND	5	R/W	Read/Write
D3	6	EN	Enable
D4	11	DB4	Serial Connection
D5	12	DB5	Serial Connection
D6	13	DB6	Serial Connection
D7	14	DB7	Serial Connection
5V	15	LED+	Backlight
GND	16	LED-	Backlight

Table 2: Arduino to LCD Pin Connections

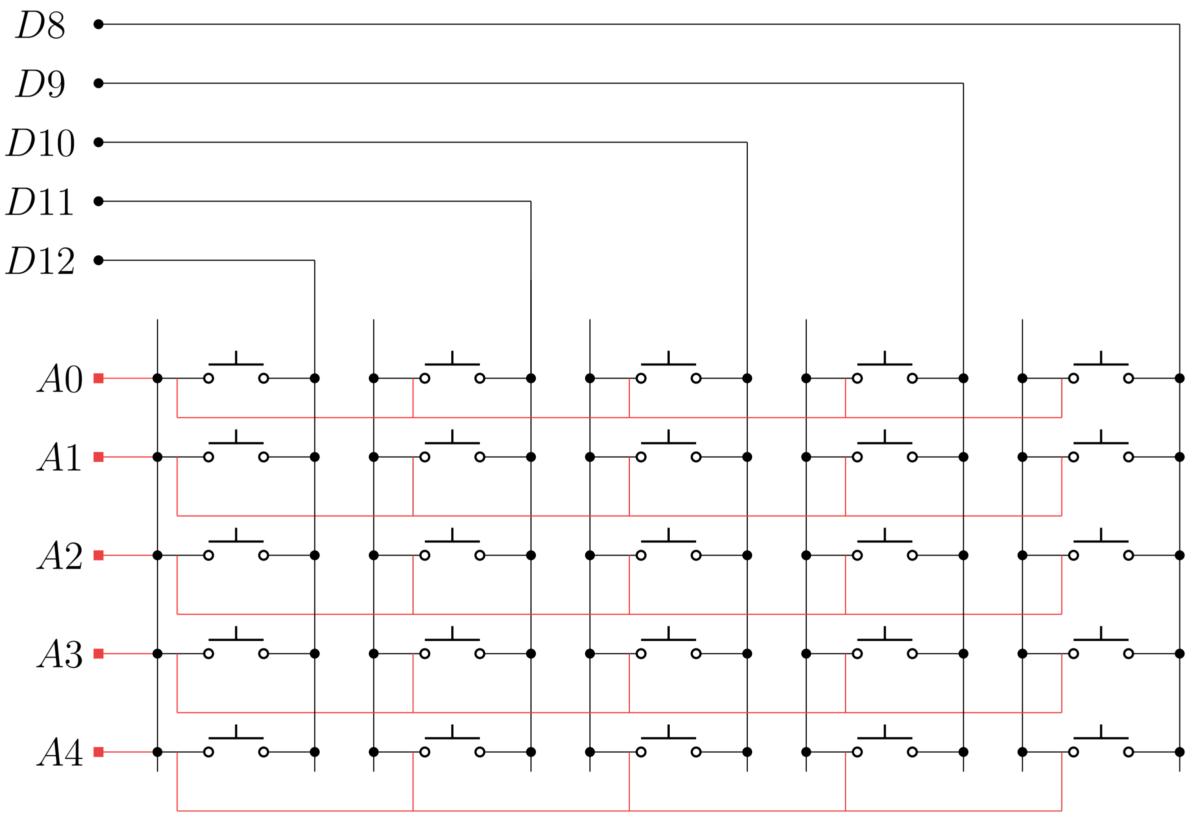
Button Matrix

The button matrix is a grid of push-button switches arranged in rows and columns. It allows multiple buttons to be connected to the microcontroller using fewer pins. In my circuit, the rows are connected to analog pins (A0 to A4), while the columns are connected to digital pins (D2 to D6) of the Arduino.

Working Principle:

- The arduino scans the matrix by activating each row (setting it HIGH) one at a time while reading the columns for signals.
- When a button is pressed, it completes the circuit between its corresponding row and column.
- By identifying which row and column are active, the arduino determines which button was pressed.
- Suppose the first column is set to HIGH, and a button (unknown to the circuit) is pressed. When reading from the rows, the microcontroller detects a signal from the second row. This confirms that the button pressed is the second button in the first column.

This idea helped us reduce the number of pins required to implement a calculator with 25 buttons. Figure **??**.



1.2 Numerical Methods

All mathematical functions were computed by numerically solving certain differential equations.

Runge-Kutta 4th Order (RK4) Method

The RK4 method is given by the following iterative steps to solve the differential equation $\frac{dy}{dx} = f(x, y)$:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where h is the step size, y_n is the value of the solution at time x_n, and $f(x, y)$ represents the differential equation.

1.2.1 Trigonometric Functions

For trigonometric functions, we solve the following differential equation whose solution is $y(x) = \sin(x)$:

$$\frac{d^2y}{dx} + y = 0;$$

Once we have the function to calculate sine, we can use the following facts to calculate cosine and tan.

$$\cos(x) = \sin\left(\frac{\pi}{2} - x\right)$$

$$\tan(x) = \frac{\sin(x)}{\cos(x)}$$

1.2.2 Inverse Trigonometric Functions

For inverse trigonometric functions, we solve the differential equation whose solution is $y(x) = \arcsin(x)$:

$$\frac{dy}{dx} = \frac{1}{\sqrt{1 - x^2}}$$

Once we obtain the method to calculate inverse sine, cosine inverse can be implemented by using the property,

$$\arccos(x) + \arcsin(x) = \frac{\pi}{2}$$

to obtain tan inverse we need to solve a seperate differential equation,

$$\frac{dy}{dx} = \frac{1}{1 + x^2}$$

1.2.3 Logarithmic Function

For the logarithmic function, we solve the differential equation whose solution is $y(t) = \ln(t)$:

$$\frac{dy}{dx} = \frac{1}{x}$$

1.2.4 Power (Exponent) Function

For the power function, we solve the differential equation whose solution is $y = x^a$:

$$\frac{dy}{dx} = \frac{ay}{x}$$

2 Showcase of functions implemented

2.1 BODMAS

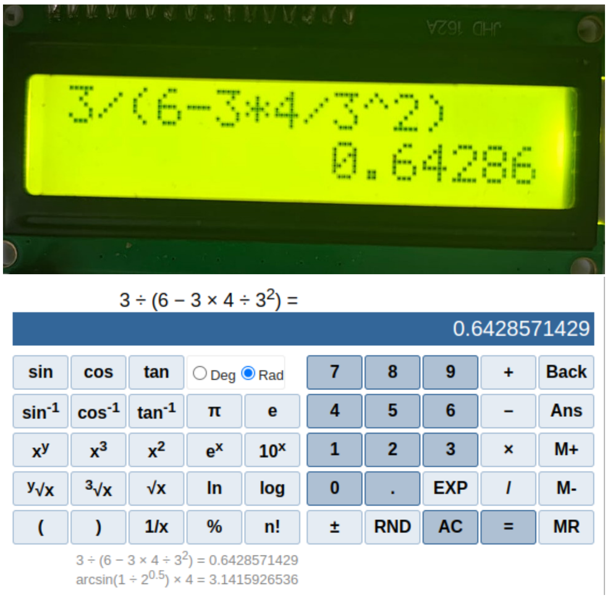


Figure 3: BODMAS

2.2 Trigonometric and inverse trigonometric functions

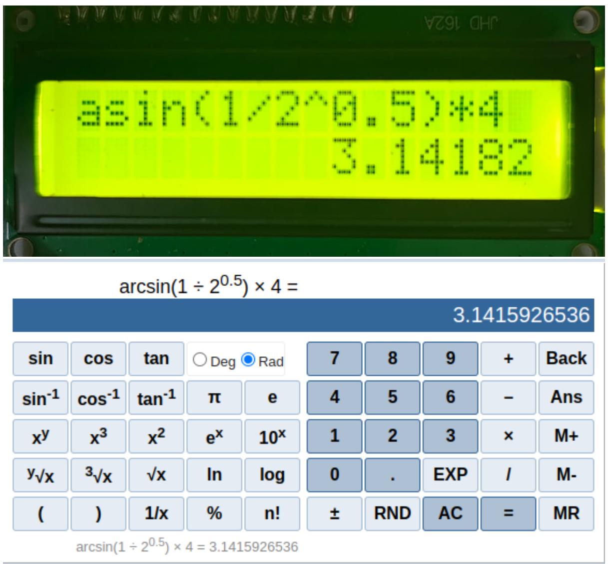


Figure 4: Trigonometric and inverse trigonometric functions

2.3 Power and Exponent

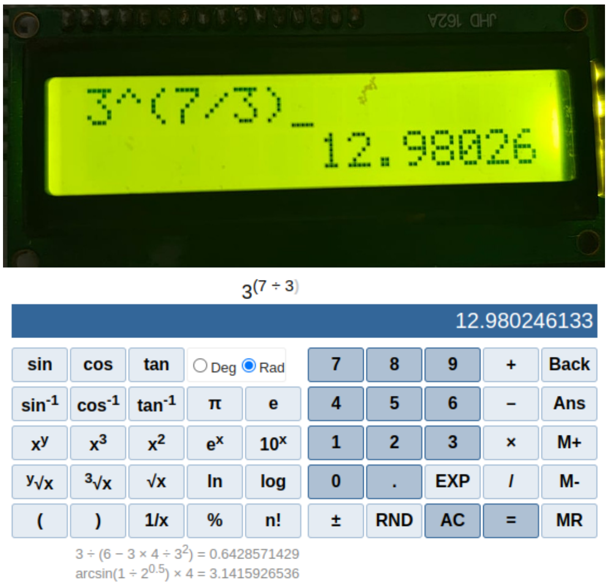


Figure 5: Power

2.4 Natural Log and base-10 log

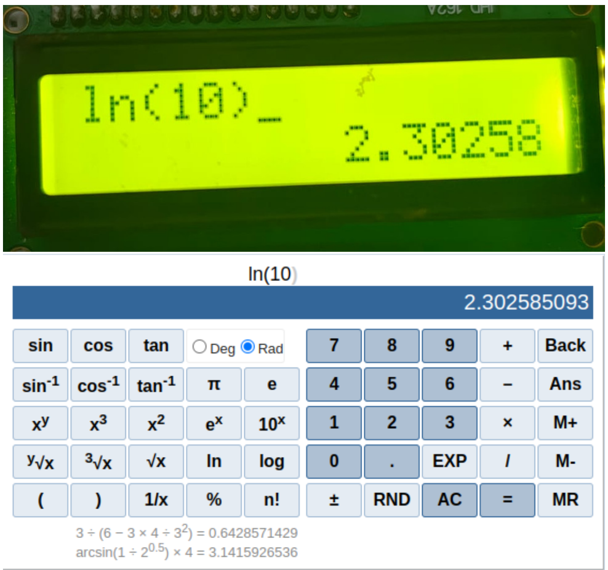


Figure 6: Natural Log

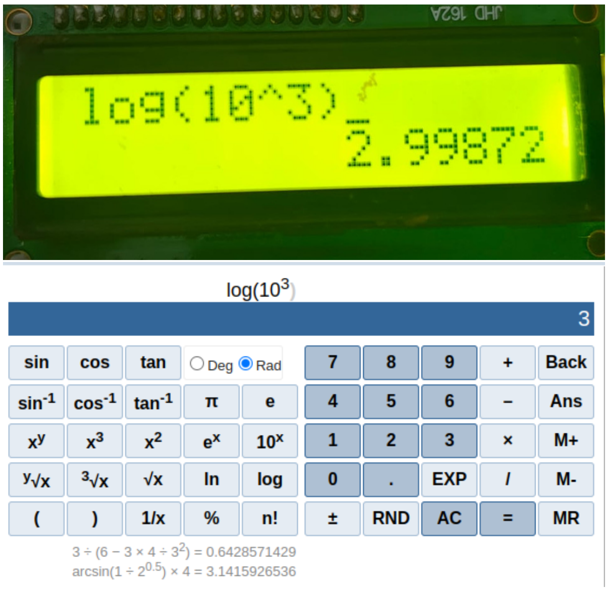


Figure 7: Log base-10

3 Conclusion

We have implemented a calculator with many features found on any standard scientific calculator.