Scientific Calculator using Arduino

Arnav Yadnopavit
EE24BTECH11007

March 24, 2025

Contents

1 Introduction

A scientific calculator is an essential tool for performing complex mathematical operations such as trigonometry, logarithms, exponentiation, and numerical methods. This project implements a scientific calculator using an Arduino board, a 16x2 LCD display, and a button matrix. The calculator is designed to evaluate expressions efficiently and accurately using numerical methods like the **CORDIC algorithm** for trigonometric functions and the **Runge-Kutta 4th order method (RK4)** for logarithms and exponentiation.

2 Components

This section briefly describes the components used in the project.

2.1 Arduino Board

The Arduino acts as the central processing unit, handling input from the button matrix, performing calculations, and displaying results on the LCD.

2.2 16x2 LCD Display

The 16x2 LCD display is used to show the input expression and the computed result. It operates in 4-bit mode to save I/O pins.

2.3 Button Matrix

A **4x5 button matrix** is used for input. It operates in two modes:

- **Normal Mode**: Directly enters numbers and basic operations.
- **Shift Mode**: Activates advanced functions like trigonometry and logarithms.

2.4 Push Button for Shift Mode

A dedicated shift button enables alternate functions for each key.

2.5 Resistors and Wires

Resistors ensure proper signal transmission, while jumper wires connect the components.

Component	Arduino Pin
Button Matrix	
Row 1	2
Row 2	3
Row 3	4
Row 4	5
Column 1	6
Column 2	7
Column 3	8
Column 4	9
Column 5	10
Shift Button	
Shift Button	13
GND	GND
LCD Display (16x2, Non-I2C)	
LCD RS	A0
LCD EN	A1
LCD D4	A2
LCD D5	A3
LCD D6	A4
LCD D7	A5

Table 1: Circuit Connections of the Scientific Calculator

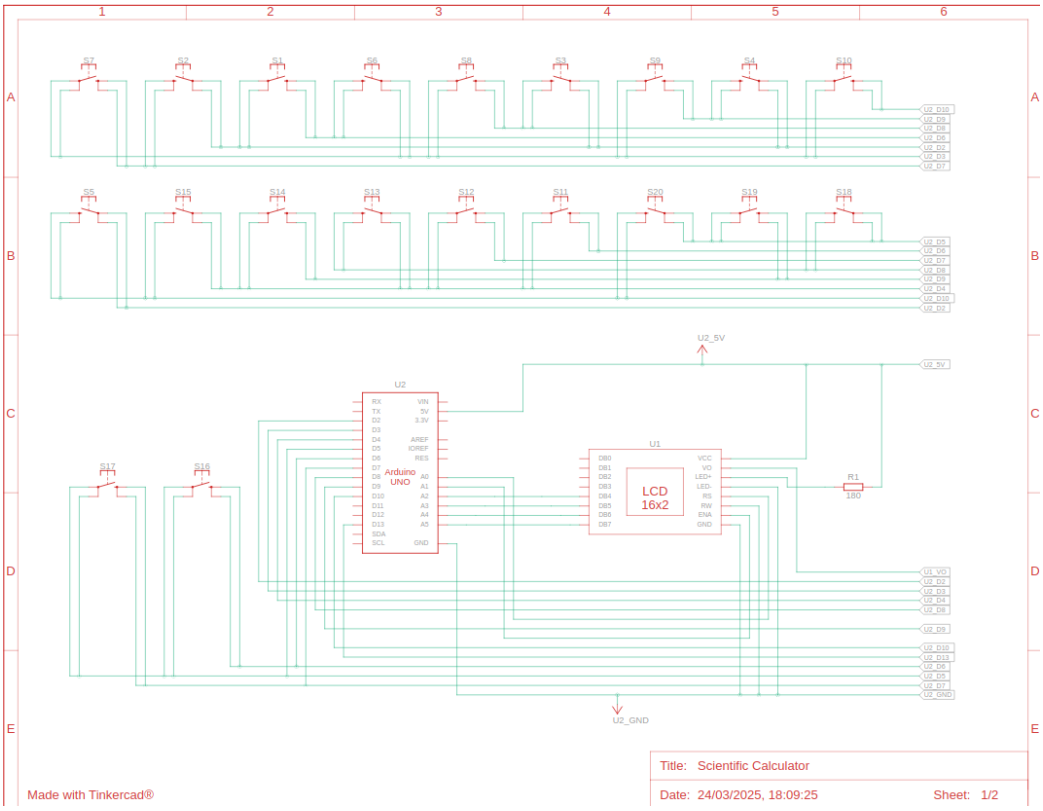

Figure 1: Circuit Diagram of the Scientific Calculator (Sheet 1)

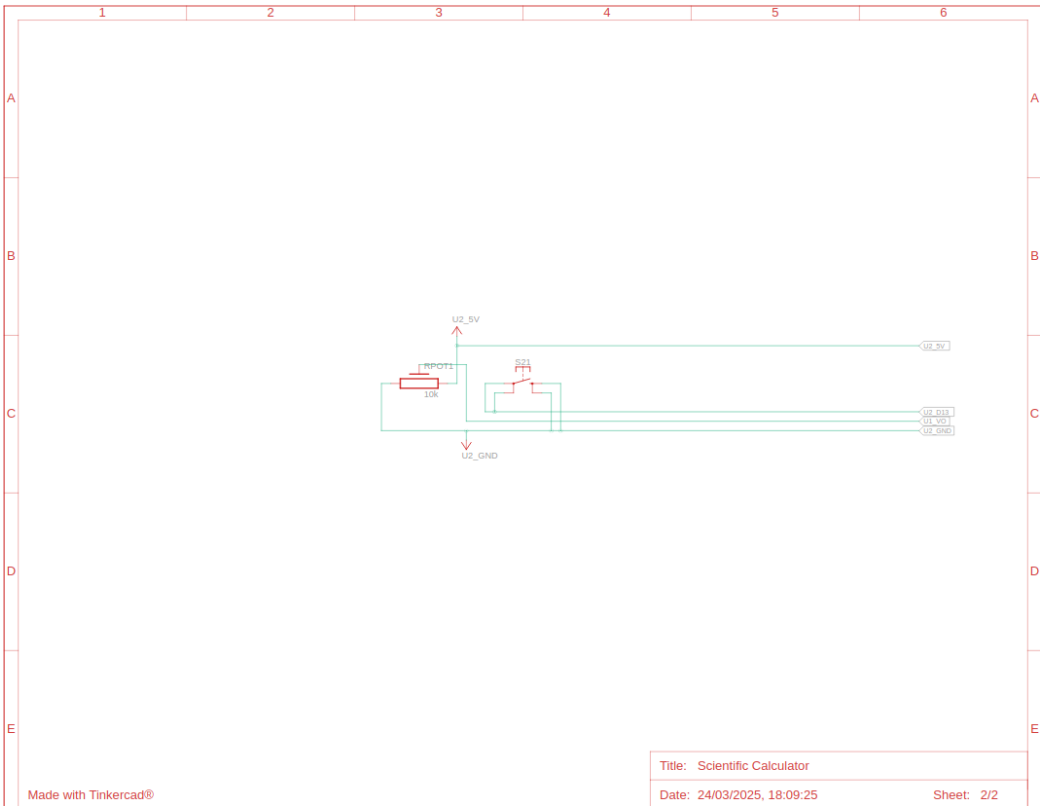

Figure 2: Circuit Diagram of the Scientific Calculator (Sheet 2)

3 Functions Available

The calculator supports a variety of mathematical operations.

3.1 Normal Mode Button Layout

1	2	3	/	C
4	5	6	*	D
7	8	9	-	(
.	0	=	+	)

Table 2: Normal mode button layout

3.2 Shift Mode Button Layout

sin	cos	tan	x^y	C
!	π	e	$\|x\|$	D
log	ln	sqrt	cbrt	r
sin^{-1}	cos^{-1}	tan^{-1}	x^2	x^3

Table 3: Shift mode button layout

4 Numerical Methods

To ensure accuracy in mathematical computations, this calculator uses two powerful numerical methods: CORDIC for trigonometric functions and RK4 for logarithmic and exponential functions.

4.1 CORDIC Algorithm for Trigonometric Functions

CORDIC (COordinate Rotation DIgital Computer) is an iterative algorithm used for computing trigonometric functions efficiently without floating-point operations.

4.1.1 CORDIC Equations

$$x_{i+1} = x_i - d_i \cdot y_i \cdot 2^{-i}$$
$$y_{i+1} = y_i + d_i \cdot x_i \cdot 2^{-i}$$
$$z_{i+1} = z_i - d_i \cdot \text{atan}(2^{-i})$$

where:

- x, y represent the coordinates of the rotated vector.
- z is the angle being processed.
- d_i is the sign of z.

4.1.2 Applications in the Calculator

- $\sin(x), \cos(x), \tan(x)$

4.2 Runge-Kutta 4th Order Method (RK4)

The RK4 method is a numerical approach for solving differential equations and is used in this calculator for logarithmic and power functions.

4.2.1 RK4 Equations

$$k_1 = hf(x_n, y_n)$$
$$k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$$
$$k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$$

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

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

4.2.2 Applications in the Calculator

- **Logarithms**: Computing $\ln(x)$ and $\log_{10}(x)$.
- **Exponentiation**: Evaluating x^n.
- **Square and Cube Roots**: Computing $\sqrt{x}$ and $\sqrt[3]{x}$.
- **Inverse Trigonometric Functions**: $\sin^{-1}(x)$, $\cos^{-1}(x)$, and $\tan^{-1}(x)$.

5 Expression Evaluation Logic

To handle complex expressions, the calculator uses:

- **Stack-based computation**: Uses two stacks for values and operators.
- **Operator precedence rules**: Implements precedence to ensure correct order of operations.
- **String parsing**: Extracts numbers, operators, and function names.
- **Error handling**: Handles division by zero and invalid inputs.

6 Implementation Challenges and Solutions

- **Handling Button Multiplexing**: Since the number of input pins is limited, a multiplexing technique was used to read button presses efficiently.
- **Non-I2C LCD Handling**: The LCD was operated in 4-bit mode to optimize pin usage.
- **Efficient Mathematical Computation**: Using CORDIC and RK4 improved accuracy while reducing computation time.

7 Conclusion

This scientific calculator successfully implements a variety of mathematical functions using efficient numerical methods. The combination of CORDIC and RK4 ensures accurate and fast computations. The button matrix provides an intuitive interface, making it a practical and functional scientific calculator.
For codes refer
`https://github.com/ArnavYadnopavit/EE1003/tree/main/Calculator`

Thank you