Hardware project - Scientific Calculator

EE24BTECH11012 - Bhavanisankar G S

March 24, 2025

Abstract

This report presents the design and implementation of a scientific calculator using an Arduino UNO microcontroller. The calculator features a comprehensive set of mathematical functions including trigonometric, logarithmic, exponential operations, and numerical methods for solving differential equations. The implementation uses the Runge-Kutta 4th order method to compute various mathematical functions with high precision.

Contents

1 Introduction

The scientific calculator project aims to implement a wide range of mathematical functions on an Arduino UNO microcontroller with limited computational resources. Instead of relying on built-in math libraries, numerical methods have been implemented to compute functions like sine, cosine, logarithms, exponentials and inverse functions.

2 Hardware Components

- Arduino UNO
- 16x2 LCD Display
- 20 push buttons to make 4x5 keypad
- Potentiometer
- Jumper wires

2.1 LCD Interface

The calculator uses a 16x2 LCD display in 4-bit mode with the following connections:

LCD Pin	**AVR Port/Pin**	**Arduino Pin**
Register Select (RS)	PORTD2	12
Enable (EN)	PORTD3	11
Data 4 (D4)	PORTD4	5
Data 5 (D5)	PORTD5	4
Data 6 (D6)	PORTD6	3
Data 7 (D7)	PORTB4	2

Table 1: LCD Pin Connections

2.2 Keypad Interface

A 4x5 matrix keypad is implemented with the following connections:

Keypad	**AVR Port/Pin**	**Arduino Pin**
Row 1	PORTC0	A0
Row 2	PORTC1	A1
Row 3	PORTC2	A2
Row 4	PORTC3	A3
Column 1	PORTB0	8
Column 2	PORTB1	9
Column 3	PORTB2	10
Column 4	PORTD6	6
Column 5	PORTD7	7

Table 2: Keypad Matrix Connections

2.3 Keypad Layout

The 4x5 keypad matrix has the following layout in normal mode:

7	8	9	+	Del
4	5	6	-	Shift
1	2	3	*	Alpha
0	.	=	/	Clr

Table 3: Keypad Layout (Normal Mode)

3 User Interface

3.1 Input Modes

The calculator supports three input modes:

1. **Normal Mode**: Basic arithmetic and number entry
2. **Shift Mode**: Access to inverse functions and special operations
3. **Alpha Mode**: Access to scientific functions and constants

The matrix design also supports these three different input modes without requiring additional hardware. This multiplexing technique allows a single button to have up to three different functions, effectively tripling the functionality without adding more hardware.

3.2 Alpha Mode Functions

In Alpha mode, the keypad provides access to scientific functions:

sin	cos	tan	^	BS
log	ln	e^	$\sqrt{x}$	(
π	x^2	x^3	1/x	)
EXP	ANS	M+	M-	MR

Table 4: Keypad Layout (Alpha Mode)

3.3 Shift Mode Functions

In Shift mode, the keypad provides access to additional functions:

asin	acos	atan	y^x	CLR
10^	e	abs	cbrt	[
deg	rad	mod	fact	]
HEX	DEC	BIN	OCT	MC

Table 5: Keypad Layout (Shift Mode)

4 Mathematical Methods

4.1 Runge-Kutta 4th Order Method

The Runge-Kutta 4th order method (RK4) is used to solve ordinary differential equations (ODEs). It provides a good balance between accuracy and computational complexity. For a first-order ODE of the form:

$$\frac{dy}{dx} = f(x, y) \tag{1}$$

The RK4 method approximates the solution using:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2}$$
$$k_1 = h \cdot f(x_n, y_n) \tag{3}$$
$$k_2 = h \cdot f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \tag{4}$$
$$k_3 = h \cdot f(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \tag{5}$$
$$k_4 = h \cdot f(x_n + h, y_n + k_3) \tag{6}$$

where h is the step size.

4.2 Computing Trigonometric Functions

Sine and cosine functions are computed by solving the second-order ODE:

$$\frac{d^2y}{dx^2} = -y \tag{7}$$

This is converted to a system of first-order ODEs:

$$\frac{dy_1}{dx} = y_2 \tag{8}$$
$$\frac{dy_2}{dx} = -y_1 \tag{9}$$

With initial conditions:

- For sine: $y_1(0) = 0, y_2(0) = 1$
- For cosine: $y_1(0) = 1, y_2(0) = 0$

4.3 Computing Logarithmic Functions

The natural logarithm is computed using the integral definition:

$$\ln(x) = \int_1^x \frac{1}{t} dt \tag{10}$$

This is solved using the RK4 method with the differential equation:

$$\frac{dy}{dx} = \frac{1}{x} \tag{11}$$

4.4 Computing Exponential Functions

The exponential function e^x is computed by solving the ODE:

$$\frac{dy}{dx} = y \tag{12}$$

With the initial condition $y(0) = 1$.

5 Implementation Details

5.1 Code Structure

The code is organized into several functional modules:

- Numerical methods for mathematical functions
- LCD interface
- Keypad interface
- Expression parsing and evaluation
- Memory management

5.2 User Interface

The calculator provides three modes of operation:

- Normal mode: Basic arithmetic operations
- Alpha mode: Scientific functions (sin, cos, log, etc.)
- Shift mode: Advanced functions (inverse trig, memory operations)

6 Mathematical Functions

The calculator implements the following mathematical functions:

6.1 Trigonometric Functions

- $\sin(x)$: Computed using RK4 for the ODE $\frac{d^2y}{dx^2} = -y$ with initial conditions $y(0) = 0, y'(0) = 1$
- $\cos(x)$: Computed using RK4 for the ODE $\frac{d^2y}{dx^2} = -y$ with initial conditions $y(0) = 1, y'(0) = 0$
- $\tan(x)$: Computed as $\frac{\sin(x)}{\cos(x)}$
- $\sin^{-1}(x)$: Computed using numerical integration of $\frac{1}{\sqrt{1-x^2}}$
- $\cos^{-1}(x)$: Computed as $\frac{\pi}{2} - \sin^{-1}(x)$
- $\tan^{-1}(x)$: Computed using numerical integration of $\frac{1}{1+x^2}$

6.2 Logarithmic and Exponential Functions

- $\ln(x)$: Computed using numerical integration of $\frac{1}{x}$
- $\log_{10}(x)$: Computed as $\frac{\ln(x)}{\ln(10)}$
- e^x: Computed using RK4 for the ODE $\frac{dy}{dx} = y$ with initial condition $y(0) = 1$
- 10^x: Computed using the power function

6.3 Power and Root Functions

- $\sqrt{x}$: Computed using RK4 for the ODE $\frac{dy}{dx} = \frac{1}{2y}$ with initial condition $y(1) = 1$
- x^y: Computed using iterative multiplication
- $\sqrt[3]{x}$: Computed using Newton's method

6.4 Other Functions

- $|x|$: Absolute value
- $n!$: Factorial
- Degree-to-radian and radian-to-degree conversions

6.5 Memory Functions

The calculator includes memory operations:

- Memory Recall (MR): Retrieve stored value
- Memory Add (M+): Add current value to memory
- Memory Subtract (M-): Subtract current value from memory
- Memory Clear (MC): Reset memory to zero

7 Expression Evaluation

The calculator parses and evaluates mathematical expressions following the BODMAS rule (Brackets, Orders, Division, Multiplication, Addition, Subtraction). The implementation handles:

- Basic arithmetic operations: +, -, *, /, $^{\prime} Functioncalls : sin(), cos(), log(), etc.$

- Constants: , e
- Memory operations: M+, M-, MR, MC

8 Conclusion

The scientific calculator implementation demonstrates how complex mathematical functions can be computed on resource-constrained microcontrollers using numerical methods. The Runge-Kutta method provides accurate approximations for differential equations, enabling the computation of transcendental functions without relying on built-in libraries.
The corresponding codes can be seen here -
https://github.com/EE24BTECH11012/EE1003/Hardware/Calculator

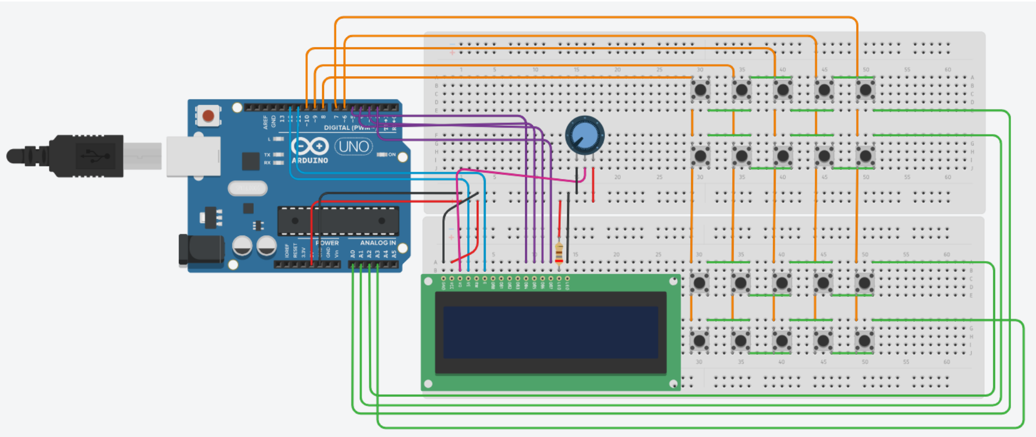

Figure 1: Schematic sketch