

Scientific Calculator

Eshan Ray
EE24BTECH11021

March 24, 2025

1 Introduction

This report details the implementation of a scientific calculator that uses the Runge-Kutta 4th order (RK4) method to compute various mathematical functions. The calculator is implemented on an AVR microcontroller and features a 4x5 keypad matrix for input and an LCD display for output.

2 Mathematical Implementation

The core of this project is the implementation of scientific functions using numerical methods instead of standard library functions. The Runge-Kutta 4th order method is used to solve differential equations that represent various mathematical functions.

2.1 RK4 Method

The Runge-Kutta 4th order method (RK4) is a numerical technique used to solve ordinary differential equations (ODEs) of the form:

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0 \quad (1)$$

Given a step size h , the RK4 method approximates the solution at $x_{n+1} = x_n + h$ using the formula:

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

Where:

$$k_1 = hf(x_n, y_n) \quad (3)$$

$$k_2 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \quad (4)$$

$$k_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \quad (5)$$

$$k_4 = hf(x_n + h, y_n + k_3) \quad (6)$$

The RK4 method is a fourth-order method, meaning that the local truncation error is on the order of $O(h^5)$, while the total accumulated error is on the order of $O(h^4)$.

Two implementations of the RK4 method are used:

1. **First-Order Differential Equations:** Used for functions like exponential, logarithm, square root, and inverse trigonometric functions.
2. **Second-Order Differential Equations:** Used for sine and cosine functions.

For second-order differential equations of the form $y'' = f(x, y, y')$, we convert them into a system of two first-order equations:

$$y_1' = y_2 \quad (7)$$

$$y_2' = f(x, y_1, y_2) \quad (8)$$

Then we apply the RK4 method to this system:

$$y_{1,n+1} = y_{1,n} + \frac{1}{6}(k_1 + 2s_1 + 2l_1 + p_1) \quad (9)$$

$$y_{2,n+1} = y_{2,n} + \frac{1}{6}(k_2 + 2s_2 + 2l_2 + p_2) \quad (10)$$

Where:

$$k_1 = hf_1(x_n, y_{1,n}, y_{2,n}) \quad (11)$$

$$k_2 = hf_2(x_n, y_{1,n}, y_{2,n}) \quad (12)$$

$$s_1 = hf_1\left(x_n + \frac{h}{2}, y_{1,n} + \frac{k_1}{2}, y_{2,n} + \frac{k_2}{2}\right) \quad (13)$$

$$s_2 = hf_2\left(x_n + \frac{h}{2}, y_{1,n} + \frac{k_1}{2}, y_{2,n} + \frac{k_2}{2}\right) \quad (14)$$

$$l_1 = hf_1\left(x_n + \frac{h}{2}, y_{1,n} + \frac{s_1}{2}, y_{2,n} + \frac{s_2}{2}\right) \quad (15)$$

$$l_2 = hf_2\left(x_n + \frac{h}{2}, y_{1,n} + \frac{s_1}{2}, y_{2,n} + \frac{s_2}{2}\right) \quad (16)$$

$$p_1 = hf_1(x_n + h, y_{1,n} + l_1, y_{2,n} + l_2) \quad (17)$$

$$p_2 = hf_2(x_n + h, y_{1,n} + l_1, y_{2,n} + l_2) \quad (18)$$

2.2 Scientific Functions

The following scientific functions are implemented using the RK4 method:

Function	Differential Equation	Mode
Exponential (e^x)	$y' = y$ with $y(0) = 1$	1
Sine ($\sin x$)	$y'' + y = 0$ with $y(0) = 0, y'(0) = 1$	2
Cosine ($\cos x$)	$y'' + y = 0$ with $y(0) = 1, y'(0) = 0$	3
Tangent ($\tan x$)	Computed as $\frac{\sin x}{\cos x}$	4
Natural Logarithm ($\ln x$)	$y' = \frac{1}{x}$ with $y(1) = 0$	5
Base-10 Logarithm ($\log_{10} x$)	$y' = \frac{1}{2.3025x}$ with $y(1) = 0$	6
Square Root (\sqrt{x})	$y' = \frac{1}{2y}$ with $y(1) = 1$	7
Arcsine ($\arcsin x$)	$y' = \frac{1}{\sqrt{1-x^2}}$ with $y(0) = 0$	8
Arccosine ($\arccos x$)	Computed as $\frac{\pi}{2} - \arcsin x$	9
Arctangent ($\arctan x$)	$y' = \frac{1}{1+x^2}$ with $y(0) = 0$	10

3 Hardware Interface

3.1 LCD Interface

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

3.2 Keypad Interface

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

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

Table 1: LCD Pin Connections

Keypad	AVR Port/Pin	Arduino Pin
Row 1	PORTB0	8
Row 2	PORTB1	9
Row 3	PORTB2	10
Row 4	PORTB3	11
Column 1	PORTC5	A5
Column 2	PORTC4	A4
Column 3	PORTC3	A3
Column 4	PORTC2	A2
Column 5	PORTC1	A1

Table 2: Keypad Matrix Connections

3.3 Pin-Saving Matrix Design

The 4x5 keypad matrix implementation in this scientific calculator project helped save Arduino pins through efficient multiplexing. Instead of requiring 20 individual pins for 20 buttons, the matrix arrangement needs only 9 pins (4 rows + 5 columns).

This design works by setting one row LOW at a time while keeping others HIGH, then checking each column to see if a button is pressed. When a button is pressed, it connects a row to a column, creating a circuit that can be detected by the microcontroller.

The scanning function implements this approach:

1. It iterates through each row, setting the current row LOW
2. For each row, it checks all five columns
3. When a pressed button is detected, it returns the corresponding character from the keypad layout

3.4 Keypad Layout

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

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

Table 3: Keypad Layout (Normal Mode)

Where:

- D: Delete/Backspace
- S: Shift mode toggle
- A: Alpha mode toggle

- C: Clear

4 User Interface

4.1 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.

4.2 Alpha Mode Functions

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

EXP	cos	tan	\wedge	BS
log	ln	e^{\wedge}	\sqrt{x}	(
π	*	/	1/x)
sin	ANS	cos	M-	MR

Table 4: Keypad Layout (Alpha Mode)

4.3 Shift Mode Functions

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

asin	acos	atan	$y^{\wedge}x$	CLR
10^{\wedge}	e	abs	cbrt	[
deg	rad	mod	fact]
7	8	asin	acos	MC

Table 5: Keypad Layout (Shift Mode)

4.4 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

5 Expression Evaluation

The calculator implements a BODMAS (Brackets, Orders, Division, Multiplication, Addition, Subtraction) evaluation system for arithmetic expressions. The evaluation process includes:

1. Parsing the input expression
2. Identifying special functions (sin, cos, log, etc.)
3. Applying the appropriate mathematical operations
4. Handling operator precedence
5. Displaying the result on the LCD

6 Project Repository

The complete source code and documentation for this project are available on GitHub: Scientific Calculator with RK4 Implementation

7 Conclusion

This scientific calculator project demonstrates the successful implementation of mathematical functions using numerical methods, specifically the RK4 method for solving differential equations. The approach provides accurate approximations of complex functions without relying on standard library implementations.

The modular design allows for easy extension with additional functions, and the user interface provides intuitive access to both basic and advanced mathematical operations. The implementation on AVR microcontrollers makes it suitable for embedded applications where computational resources are limited.

The matrix keypad design efficiently uses microcontroller pins, allowing a full-featured calculator interface with minimal hardware requirements.