Hardware Assignment
Scientific Calculator

EE1003

Homa Harshitha Vuddanti
(EE24BTECH11062)

Problem Statement: Design and implement a scientific calculator using AVR-GCC and an LCD display, capable of performing basic arithmetic operations and scientific functions (preferably using differential equations) without relying on external computational libraries.

I. Introduction

This project focuses on developing a scientific calculator using the ATmega328P microcontroller, interfaced with a 16x2 LCD display and input push-buttons. The calculator supports basic arithmetic operations (addition, subtraction, multiplication, division) and scientific computations such as sine, cosine, logarithm, square root, and arctangent calculations using numerical methods like Runge-Kutta approximation. The system also includes a mode-switching feature between standard arithmetic and scientific functions.

II. Materials Required

- ATmega328P (Arduino Uno Board)
- LCD Display
- Breadboard
- Potentiometer (for controlling contrast)
- Resistors (for button input stabilization)
- Push Buttons (for numeric, operator, and mode control inputs)
- Connecting Wires

III. Procedure

A. Hardware Setup

- The 16x2 LCD display is connected to PORTD (pins PD2 to PD7) of the ATmega328P for RS, EN, and data lines.
- Push buttons are connected to PORTB and PORTC pins, each designated for numbers (0–9), arithmetic operators (+, -, *, /), and scientific functions (sin, cos, ln, sqrt, atan).
- Pull-up resistors are configured to ensure stable button readings.
- The mode switch button toggles between normal and scientific calculation modes.

TABLE I
HARDWARE CONNECTIONS

Component	Pin(s)	Description
LCD RS	PD2	Register Select
LCD EN	PD3	Enable
LCD Data Lines (D4–D7)	PD4–PD7	LCD 4-bit Data Interface
Button 0	PB0	Number 0 / Addition (+) in sci mode
Button 1	PB1	Number 1 / Subtraction (-) in sci mode
Button 2	PB2	Number 2 / Multiplication (×) in sci mode
Button 3	PB3	Number 3 / Division (÷) in sci mode
Button 4	PB4	Number 4 / $\sin(x)$ in sci mode
Button 5	PB5	Number 5 / $\cos(x)$ in sci mode
Button 6	PC0	Number 6 / $\ln(x)$ in sci mode
Button 7	PC1	Number 7 / $\sqrt{x}$ in sci mode
Button 8	PC2	Number 8 / $\arctan(x)$ in sci mode
Button 9	PC3	Number 9
Equal button	PC4	Compute Result
Mode toggle button (2nd)	PC5	Switch between Normal and Scientific mode

B. Software Implementation

The software is structured with modular functions and interrupt-driven button handling. Key components include:

- **LCD Driver Functions**: Functions for initializing the LCD, sending commands, displaying characters, and printing strings.
- **Numerical Methods**: Implementation of Runge-Kutta 4th order methods for calculating sine, cosine, logarithm, square root, and arctangent values with high precision for smaller values.
- **Button Reading and Debouncing**: Detection of button presses with debounce logic to prevent false triggers.
- **Main Loop**: Manages input state, detects operator entry, accumulates numbers, switches modes, and triggers calculations upon confirmation.
- **Output Display**: Results are displayed on the LCD with four-decimal-place precision for scientific computations.

IV. BUTTON DESCRIPTION

1) Buttons mapped to PB0–PB5 and PC0–PC5 handle number inputs, operator selection, and scientific functions.
2) A dedicated button toggles between standard and scientific modes.
3) In scientific mode, number keys correspond to functions: sin(x), cos(x), ln(x), sqrt(x), and atan(x).
4) The equal button (PC4) executes the computation and displays the result.
5) so basically,

V. SPECIFIC FEATURES AND USAGE:

6) The lcd displays ”Ready!” followed by the mode it is in when we turn it on by uploading the code.

TABLE II
BUTTON MAPPING IN NORMAL AND SCIENTIFIC MODES

Button (Pin)	Normal Mode Function	Scientific Mode Function
PB0	0	Addition (+)
PB1	1	Subtraction (-)
PB2	2	Multiplication (×)
PB3	3	Division (÷)
PB4	4	$\sin(x)$
PB5	5	$\cos(x)$
PC0	6	$\ln(x)$
PC1	7	$\sqrt{x}$
PC2	8	$\arctan(x)$
PC3	9	none
PC4	Equal (=)	Equal (=)
PC5	Mode toggle (2nd)	Mode toggle (2nd)

7) When we press on the toggle button, the mode gets changed and that change is shown on lcd constantly throughout.
8) The calculator has 0-9 buttons when in normal mode, when we press on toggle button, it's button mappings change as the table above. This has been done because there aren't enough Arduino pins to allot for every button.
9) We can press our required operation and it is coded such that after selecting an operation, it automatically switches back to normal mode so that we can press our second number without repeatedly pressing on toggle button.
10) We can view the output when we press on equal button.
11) The functions implemented are : $\sin$, $\cos$, $\ln$, square root, $\tan$ inverse, using RK4 method for solving differential equations.

VI. IMPROVEMENTS

The $\ln$, $sqrt$, $\tan$ inverse functions are accurate for smaller values but not for larger values. It can be made better by improving the RK4 method.

VII. CODE

The complete code is available in the following GitHub permalink:
https://github.com/HomaHarshitha/Scientific-calculator/tree/0bf909d96f977807a1d9feecd3566ab3c2201ca3/Scientific-calculator-ee062

VIII. RESULT

The scientific calculator successfully performs both standard arithmetic and advanced scientific functions using user input provided through push-buttons. The results are clearly displayed on a 16x2 LCD screen with an intuitive interface for mode switching and calculation. The implemented numerical methods ensure accurate approximations and the calculator operates reliably in real time with efficient resource usage.