Design and Implementation of a Scientific Calculator Using AVR-GCC

S.Kavya Anvitha
EE24BTECH11056

March 24, 2025

Abstract

This report details the design and development of a scientific calculator utilizing an Arduino Uno, a JHD162A LCD display, and 23 push buttons. The system is programmed using AVR-GCC on a Debian-based Termux environment. The calculator supports arithmetic, trigonometric, logarithmic, and exponential functions while ensuring efficient input handling and display management. This document outlines the circuit design, software implementation, and experimental results.

1 Introduction

Scientific calculators are essential tools in engineering and academic fields, providing advanced mathematical functionalities. This project aims to construct a scientific calculator based on an embedded system using an Arduino Uno and programming it with AVR-GCC. The focus is on achieving efficient performance while maintaining a user-friendly interface.

2 Hardware Components

The primary components used in this project are:

- **Microcontroller:** Arduino Uno
- **Display:** JHD162A 16x2 LCD module
- **Input:** 23 push buttons

- **Resistors:** 1kΩ, 2kΩ, 1.5kΩ, 15kΩ
- **Connections:** Jumper wires, conducting wires, and a breadboard

3 Circuit Design

The circuit consists of the Arduino as the processing unit, push buttons for input, and an LCD for displaying outputs. The buttons are arranged in two rows:

- **Row 1:** 10 buttons for digits (0-9)
- **Row 2:** 13 buttons for functions (arithmetic, trigonometric, logarithmic, etc.)

The connections for the LCD and push buttons are detailed in Table 1.

Table 1: Connections

First end of jumper wire	Second end of jumper wire
Arduino digital pin 0	Push button no. 16
pin 1	Push button no.17
pin 2	LCD pin 4
pin 3	LCD pin 6
pin 4	LCD pin 11
pin 5	LCD pin 12
pin 6	LCD pin 13
pin 7	LCD pin 14
pin 8	Push button no.18
pin 9	Push button no. 19
pin 10	Push button no. 20
pin 11	Push button no. 21
pin 12	Push button no. 22
pin 13	Push button no. 23
Arduino analog pin A1	Push button no. 15
pin A2	Push button no. 14
pin A3	Push button no. 13
pin A4	Push button no. 12
pin A5	Push button no. 11
pin A0	Push buttons no. 1-10 (digit buttons)
LCD pin 1	Ground
LCD pin 2	5V
LCD pin 15	5V via 1kΩ resistor
LCD pin 16	Ground
LCD pin 3	Ground via 1.5kΩ resistor
LCD pin 5	Ground
LCD pin 5	All push buttons

Note: Circuit designed with contributions from Akshara EE24BTECH11003, Akshita EE24BTECH11054.

4 Software Implementation

The software is written in embedded C using AVR-GCC and compiled in a Termux (Debian) environment. The main features include:

- Basic arithmetic operations (addition, subtraction, multiplication, division)

- Trigonometric functions (`sin`, `cos`, `tan`)
- Logarithmic and exponential functions
- Factorial and power functions
- Efficient keypad scanning and input processing

4.1 Implementation Details

The AVR code is structured as follows:

- **Initialization:** Configuring LCD and keypad
- **Interrupt Handling:** Managing button presses
- **Computation:** Executing mathematical functions
- **Display Update:** Showing results on the LCD

The source code can be accessed at Codes of this folder

Acknowledgment: Code contributions from Akshara EE24BTECH11003, Rasagna EE24BTECH11023, Shannu TejVardan EE24BTECH11034

4.2 Push Button Function Mapping

The key functions associated with each button are listed in Table 2.

Table 2: Push Button Designations

Button Number	Function
1 - 10	Digits 0 - 9
11	Clear
12	$\ln(x)$ and $\log(x)$
13	Right Parenthesis
14	$\sin(x)$, $\cos(x)$, and $\tan(x)$
15	e and π
16	Backspace
17	Decimal Point
18	Equal To
19	Left Parenthesis
20	Division
21	Multiplications
22	Subtraction
23	Addition

5 Experimental Results

The calculator was tested for accuracy and response time. The output was compared against standard scientific calculators, showing minimal error. The system successfully executed all planned operations within the given hardware limitations.

6 Conclusion

This project successfully demonstrates the design and implementation of a scientific calculator using an Arduino Uno and AVR-GCC. The system performs mathematical operations efficiently while mintaining simplicity in user interaction. Future improvements can include an expanded function set and memory storage for calculations.