

Scientific Calculator Using AVR-GCC

Mandara Hosur
EE24BTECH11040

March 24, 2025

Abstract

This report presents the design and implementation of a scientific calculator using an Arduino Uno, a JHD162A LCD display, 23 input buttons, and other supporting components. The calculator is programmed using AVR-GCC on Termux (Debian) and supports various mathematical functions within the constraints of 23 push buttons. The report covers the circuit design, software implementation, and testing results.

1 Introduction

A scientific calculator is an essential tool for engineers and students, capable of performing complex mathematical operations. This project aims to develop a scientific calculator using an embedded system, specifically the Arduino Uno, programmed with AVR-GCC.

2 Hardware Components

- Arduino Uno
- JHD162A LCD display
- 23 push buttons
- $1k\Omega$, $2k\Omega$, $1.5k\Omega$, $15k\Omega$, resistors
- Jumper wires and conducting wires
- Breadboard

3 Circuit Design

1. Connect 5V and Ground from the Arduino onto the breadboard.
2. Connect the push buttons in 2 rows (each from grid to power lines not connected to Ground or 5V). The first row must have 10 buttons (for digits), and the second row must have 13 buttons (for functions). Connect one terminal of each button to Ground.
3. Make the following connections to establish the required circuit:

Table 1: Connections

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

Remark: Connections designed with the help of Akshara EE24BTECH11003, Akshita EE24BTECH11054

4 Software Implementation

The software is written in embedded C using AVR-GCC and compiled in Termux. The main features include:

- Basic arithmetic operations (addition, subtraction, multiplication, division)
- Trigonometric functions (sin, cos, tan)
- Logarithmic and exponential functions
- Factorial and power functions
- Keypad scanning and input processing

4.1 AVR Code

The code follows a structured approach:

- Initialization of LCD and keypad
- Interrupt-based button handling
- Mathematical function execution
- Displaying results on the LCD

Code can be found at the following

Remark: Code sourced from Akshara EE24BTECH11003, Akhila EE24BTECH11055, Rasagna EE24BTECH11023, Spoorthi EE24BTECH11065, Akshita EE24BTECH11054

4.2 Push Button Designations

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 Results

The calculator was tested for accuracy and response time. The output was verified against standard scientific calculators, and the results showed a minimal error margin. The system successfully handled all planned operations within the given hardware constraints.

6 Conclusion

This project successfully demonstrates the implementation of a scientific calculator using AVR-GCC and embedded C. It provides accurate calculations while maintaining a simple user interface.