SCIENTIFIC CALCULATOR

JADHAV RAJESH - EE24BTECH11028

March 24, 2025

Objective

To design and implement a scientific calculator using an Arduino Uno board programmed with AVR-GCC, capable of performing basic and advanced mathematical operations.

Materials Required

0.1 Hardware Components

- Arduino Uno (ATmega328P microcontroller)
- 16×2 LCD Display (for output)
- 4×4 Matrix Keypad (for input)
- Breadboard & Jumper Wires
- Potentiometer (for LCD contrast adjustment)
- USB Cable (for programming)

0.2 Software Tools

- Arduino IDE / AVR-GCC Toolchain
- AVRDUDE (for flashing the program)

Circuit design

Table 1: 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

Experimental Procedure

0.3 Circuit Setup

0.3.1 LCD Interfacing with Arduino

Connect the **16×2 LCD** to Arduino Uno as follows:

- **VSS (GND)** → Arduino GND
- **VDD (5V)** → Arduino 5V
- **VO (Contrast)** → Potentiometer middle pin
- **RS (Register Select)** → Arduino Pin 12
- **RW (Read/Write)** → GND (for write mode)
- **EN (Enable)** → Arduino Pin 11
- **D4-D7 (Data Pins)** → Arduino Pins 5, 4, 3, 2
- **Backlight (A & K)** → 5V & GND (with resistor if needed)

Component	**Arduino Pin**
Button Matrix	
Row 1	2
Row 2	3
Row 3	4
Row 4	5
Column 1	6
Column 2	7
Column 3	8
Column 4	9
Column 5	10
Shift Button	
Shift Button	13
GND	GND
LCD Display (16x2, Non-I2C)	
LCD RS	A0
LCD EN	A1
LCD D4	A2
LCD D5	A3
LCD D6	A4
LCD D7	A5

Table 2: Circuit Connections of the Scientific Calculator

0.3.2 Keypad Interfacing

Connect the **4×4 Matrix Keypad** to Arduino:

- Rows → Arduino Pins 6, 7, 8, 9
- Columns → Arduino Pins A0, A1, A2, A3

0.4 Software Development (AVR-GCC Programming)

0.4.1 Step 1: Initialize LCD and Keypad

- Use the `avr/io.h` library for GPIO control.
- Implement **LiquidCrystal** library functions (or custom functions) for LCD.
- Write a keypad scanning function to detect button presses.

0.4.2 Step 2: Implement Calculator Logic

- **Basic Operations (+, -, *, /)** - Store operands and operator. - Compute result using arithmetic logic.
- **Scientific Functions (sin, cos, tan, log, sqrt, etc.)** - Use <`math.h`> library for advanced computations.
- **Error Handling** - Check for division by zero, invalid inputs.

Step 3: Display Results

- Print input and output on the LCD.
- Clear display when ”C” (clear) is pressed.

0.5 Testing & Validation

- Test each operation (basic & scientific).
- Verify correct display on LCD.
- Debug if any issues arise (e.g., wrong calculations, unresponsive keypad).

Expected Results

The Arduino-based scientific calculator should:

- Accept inputs via the keypad.
- Perform arithmetic and scientific operations accurately.
- Display results on the LCD.

Conclusion

This experiment demonstrates the development of a **scientific calculator using Arduino Uno with AVR-GCC**, integrating hardware (LCD, keypad) and software (mathematical logic). The system should function reliably for engineering and educational applications.