# Scientific Calculator Implementation Report

EE24BTECH11038-M.B.S Aravind

March 24, 2025

**Abstract**

This report documents the implementation of a scientific calculator using an Arduino microcontroller, LCD display, and push buttons. The calculator supports basic arithmetic operations, trigonometric functions, and multi-step calculations. The implementation demonstrates concepts of embedded systems programming, user interface design, and mathematical function implementation in resource-constrained environments.

## Contents

## 1 Introduction

The scientific calculator project aims to develop a fully functional calculator with numerical input capabilities and scientific functions. Unlike basic calculators, this implementation supports trigonometric functions (sine, cosine, tangent, and their inverse functions) along with standard arithmetic operations.

## 2 Hardware Components

The hardware implementation consists of the following components:

- Arduino board (ATmega328P-based)
- 16x2 LCD display
- Breadboard
- Push buttons (14 total):
  - 10 buttons for digits 0-9
  - 1 button for operator cycling (+, -, *, /)
  - 1 button for trigonometric function cycling (sin, cos, tan, asin, acos, atan)
  - 1 button for decimal point
  - 1 button for equals/enter
- Jumper wires
- Resistors for button pull-ups (if not using internal pull-ups)
- Potentiometer

## 3 LCD

A Liquid Crystal Display (LCD) is an electronic display module used in embedded systems for visual representation of characters, numbers, and symbols. The 16x2 LCD module is one of the most commonly used displays in microcontroller-based projects.

## 4 Features of 16x2 LCD

The 16x2 LCD has the following key features:

- Display: 16 characters per row, 2 rows
- Interface: Parallel (4-bit or 8-bit mode)
- Power Supply: 5V DC
- Backlight: LED backlight for better visibility
- Control Pins:
  - **RS (Register Select)**: Chooses command or data mode
  - **RW (Read/Write)**: Selects read or write operation (often tied to ground for write-only operation)
  - **E (Enable)**: Latches the command/data into the LCD

## 5 Pin Configuration

The 16x2 LCD has 16 pins with the following functions:

| Pin No. | Name | Function |
|---|---|---|
| 1 | VSS | Ground (0V) |
| 2 | VCC | Power Supply (5V) |
| 3 | VEE | Contrast Adjustment |
| 4 | RS | Register Select |
| 5 | RW | Read/Write Control |
| 6 | E | Enable Signal |
| 7-14 | D0-D7 | Data Pins |
| 15 | LED+ | Backlight Positive |
| 16 | LED- | Backlight Negative |

Table 1: 16x2 LCD Pin Configuration

## 6 Circuit Diagram

Below is the circuit diagram for interfacing a 16x2 LCD with an Arduino:

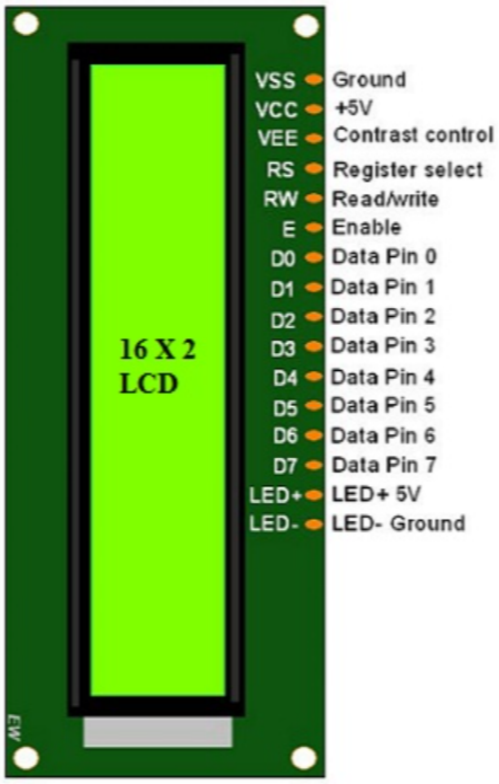


Figure 1: Interfacing 16x2 LCD with Arduino

## 7 Working Principle

The LCD operates using liquid crystal molecules that align in response to an electric field to control the passage of light. It requires initialization commands before displaying data. It supports both **4-bit and 8-bit communication**, with **4-bit mode** reducing the number of GPIO pins used.

## 8 Hardware Configuration

### 8.1 LCD Connection

The LCD is connected to the Arduino using a 4-bit interface to save I/O pins:

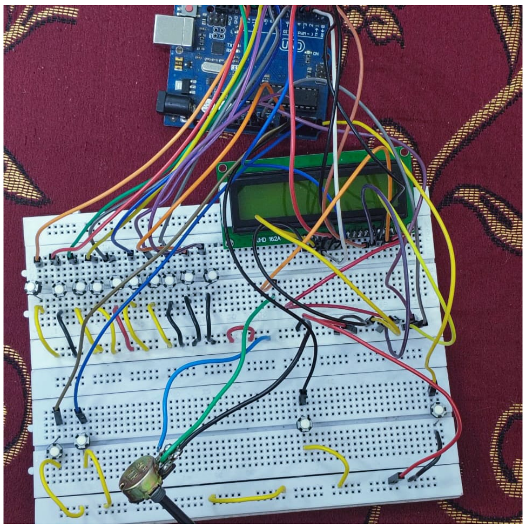
- RS (Register Select): Arduino pin 1 (PD1)
- E (Enable): Arduino pin 0 (PD0)

- D4: Arduino pin 5 (PD5)
- D5: Arduino pin 4 (PD4)
- D6: Arduino pin 3 (PD3)
- D7: Arduino pin 2 (PD2)

## 8.2 Button Connections

The buttons are connected to Arduino pins as follows:

- Digit 0: Arduino pin 13 (PB5)
- Digit 1: Arduino pin 12 (PB4)
- Digit 2: Arduino pin 11 (PB3)
- Digit 3: Arduino pin 10 (PB2)
- Digit 4: Arduino pin 9 (PB1)
- Digit 5: Arduino pin 8 (PB0)
- Digit 6: Arduino pin 7 (PD7)
- Digit 7: Arduino pin 6 (PD6)
- Digit 8: Arduino pin A0 (PC0)
- Digit 9: Arduino pin A1 (PC1)
- Operator: Arduino pin A2 (PC2)
- Trig Function: Arduino pin A3 (PC3)
- Decimal Point: Arduino pin A4 (PC4)
- Enter/Equals: Arduino pin A5 (PC5)



# 9 Software Design

## 9.1 Software Architecture

The calculator software is structured around the following key components:

- LCD Interface: Functions to initialize and communicate with the LCD
- Button Interface: Functions to initialize buttons and detect button presses
- Expression Parser: Functions to build and evaluate mathematical expressions
- Mathematical Functions: Implementation of trigonometric functions

## 9.2 Key Data Structures

- `expression[32]`: Character array to store the current expression
- `exprIndex`: Index for the current position in the expression string
- `operation`: Current selected arithmetic operation
- `trigMode`: Current selected trigonometric function
- `lastResult`: Stores the last calculated result for continued operations

### 9.3 Expression Evaluation

The calculator uses a recursive descent parser to evaluate mathematical expressions. The implementation:

- Handles parentheses and nested expressions
- Maintains correct operator precedence
- Processes expressions from left to right
- Handles trigonometric functions and their arguments

### 9.4 User Interface Design

The UI is designed for simplicity and usability with limited buttons:

- The LCD displays the current expression and results
- The operator button cycles through +, -, *, and /
- The trig button cycles through sin, cos, tan, asin, acos, and atan
- The decimal button adds a decimal point to the current number
- The enter button evaluates the expression and displays the result

## 10 Implementation Challenges

Several challenges were addressed during implementation:

### 10.1 Limited Memory

The Arduino's limited memory required efficient string handling and expression evaluation:

- Optimized string functions to reduce memory usage
- Implemented a custom expression evaluator instead of using library functions
- Limited expression length to 32 characters

### 10.2 Expression Parsing

Implementing a correct expression parser required addressing:

- Order of operations for mathematical expressions
- Handling of nested parentheses
- Proper handling of trigonometric functions
- Error detection and handling (e.g., division by zero)

### 10.3 User Interface Limitations

With limited buttons, the interface required innovative solutions:

- Cycling through operators with a single button
- Cycling through trigonometric functions with a single button
- Clear feedback on the current state through the LCD display
- Handling of continuing calculations from previous results

## 11 Software Functional Blocks

### 11.1 LCD Control Functions

These functions handle all LCD communication:

- `lcd_init()`: Initializes the LCD
- `lcd_send_command()`: Sends a command to the LCD
- `lcd_send_data()`: Sends data to the LCD
- `lcd_clear()`: Clears the LCD display
- `lcd_set_cursor()`: Sets the cursor position
- `lcd_print()`: Displays a string on the LCD

### 11.2 Button Interface Functions

These functions manage button input:

- `init_buttons()`: Sets up button pins
- `check_button()`: Checks if a specific button is pressed

### 11.3 Expression Handling Functions

These functions manage the mathematical expression:

- `append_char()`: Adds a character to the expression
- `append_string()`: Adds a string to the expression
- `append_operator()`: Handles operator input
- `append_trig_function()`: Handles trigonometric function input

### 11.4 Calculation Functions

These functions perform the mathematical calculations:

- `evaluate_expression()`: Evaluates a mathematical expression
- `process_enter()`: Processes the enter button press
- `balance_parentheses()`: Ensures all parentheses are balanced

## 12 Testing Results

### 12.1 Arithmetic Operations

Tests were conducted to verify the correct operation of basic arithmetic:

| **Expression** | **Expected Result** | **Actual Result** |
|---|---|---|
| 2 + 3 | 5 | 5 |
| 5 - 2 | 3 | 3 |
| 4 * 3 | 12 | 12 |
| 10 / 2 | 5 | 5 |
| 2 + 3 * 4 | 14 | 14 |

Table 2: Arithmetic operation test results

### 12.2 Trigonometric Functions

Tests were conducted to verify the correct operation of trigonometric functions:

| **Expression** | **Expected Result** | **Actual Result** |
|---|---|---|
| sin(30) | 0.5 | 0.5 |
| cos(60) | 0.5 | 0.5 |
| tan(45) | 1.0 | 1.0 |
| asin(0.5) | 30.0 | 30.0 |
| acos(0.5) | 60.0 | 60.0 |
| atan(1.0) | 45.0 | 45.0 |

Table 3: Trigonometric function test results

## 13 Limitations and Future Improvements

### 13.1 Current Limitations

- Expression length limited to 32 characters
- Limited input method requires cycling through operators and functions
- No memory functions (store/recall)
- Limited error handling and reporting
- No parentheses input button (parentheses only added by trig functions)

### 13.2 Potential Improvements

- Add a backspace/delete function
- Implement memory functions (M+, MR, MC)
- Add more scientific functions (logarithmic, exponential)
- Implement a better error reporting system
- Add dedicated parentheses buttons
- Use a larger LCD display for better visibility
- Implement a more efficient parsing algorithm

## 14 Conclusion

The scientific calculator implementation successfully demonstrates a functional calculator with both basic arithmetic and trigonometric capabilities. Despite the limitations of the Arduino platform and the minimal input interface, the calculator provides accurate calculations and a usable interface.

The project showcases embedded systems programming techniques, mathematical expression parsing, and user interface design with limited resources. The modular code structure allows for future extensions and improvements.

## A Code Highlights

### A.1 Expression Evaluation Function

This is the core function that evaluates mathematical expressions:

## B Pin Assignment Table

| **Component** | **Arduino Pin** | **AVR Port/Pin** |
|---|---|---|
| LCD RS | 1 | PD1 |
| LCD E | 0 | PD0 |
| LCD D4 | 5 | PD5 |
| LCD D5 | 4 | PD4 |
| LCD D6 | 3 | PD3 |
| LCD D7 | 2 | PD2 |
| Button 0 | 13 | PB5 |
| Button 1 | 12 | PB4 |
| Button 2 | 11 | PB3 |
| Button 3 | 10 | PB2 |
| Button 4 | 9 | PB1 |
| Button 5 | 8 | PB0 |
| Button 6 | 7 | PD7 |
| Button 7 | 6 | PD6 |
| Button 8 | A0 | PC0 |
| Button 9 | A1 | PC1 |
| Operator Button | A2 | PC2 |
| Trig Button | A3 | PC3 |
| Decimal Button | A4 | PC4 |
| Enter Button | A5 | PC5 |

Table 4: Pin assignment table