Implementation of a Calculator Using Arduino: Circuit Design and Software Analysis

EE24BTECH11024 - Abhimanyu Koushik

March 24, 2025

Abstract

This report details the implementation of a calculator using an Arduino microcontroller, a 16x2 LCD display, and a 5x5 matrix keypad. The system allows users to input expressions via the keypad, processes them using the Arduino, and displays the result on the LCD. The report focuses on the hardware connections and the software functions responsible for interfacing with the LCD, scanning the keypad, and processing user input.

1 Introduction

The calculator project uses an Arduino microcontroller as its core, interfacing with a 16x2 LCD display for output and a 5x5 matrix keypad for input. The implementation provides both basic arithmetic functions and advanced scientific operations. This report examines the hardware connections, the keypad matrix operation, and the software functions that power the system.
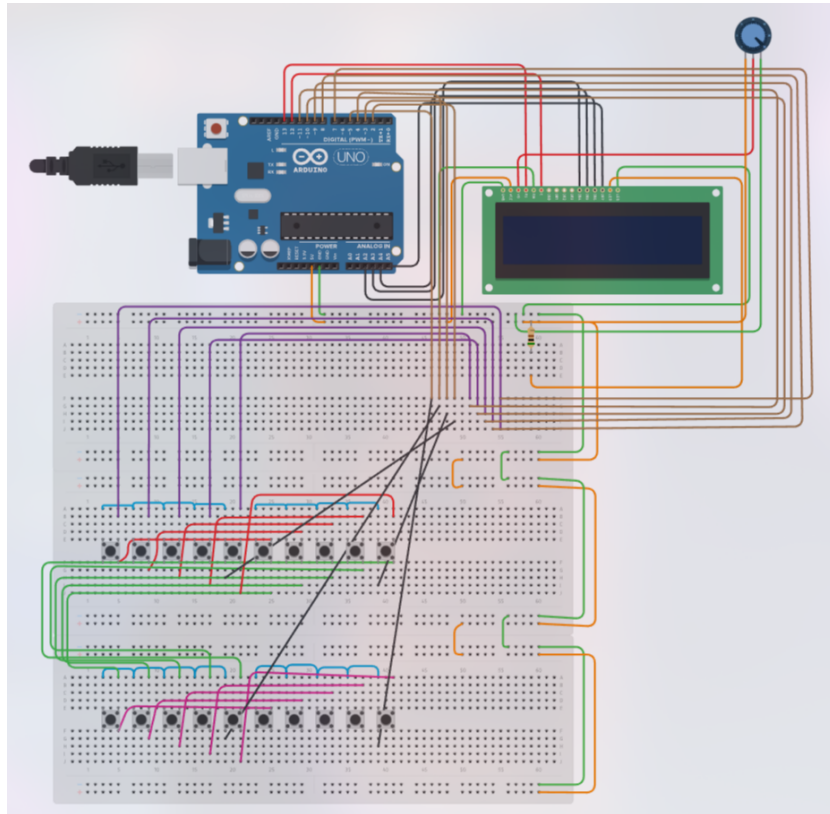
2 Hardware Components and Circuit Design

Components

- Arduino Uno microcontroller
- 16x2 LCD display
- 5x5 matrix keypad

- Connecting wires
- Breadboard (for prototyping)

Circuit Diagram

The circuit diagram of the calculator is shown below:

Explanation of Circuit Components and Connections

The circuit consists of three main components: an Arduino Uno microcontroller, a 16x2 LCD display, and a 5x5 matrix keypad. Each component is connected to perform its specific function in the calculator system.

1. Arduino Uno:
- Acts as the central processing unit for handling inputs from the keypad, performing calculations, and controlling outputs to the LCD.

- Provides power to all components through its 5V and GND pins.

    2. 16x2 LCD Display:

- Used to display user inputs (expressions) and results.
- Operates in 4-bit mode to reduce pin usage.
- Connections:
- RS (Register Select): Connected to pin D13 on Arduino.
- EN (Enable): Connected to pin D12 on Arduino.
- Data pins D4-D7: Connected to pins A2-A5 on Arduino.
- VCC and GND: Connected to Arduino's 5V and GND pins.
- A potentiometer is connected to adjust contrast via the Vo pin.
3. 4x5 Matrix Keypad:
- Serves as the input interface for entering numbers, operators, and functions.
- Organized in a grid of rows (R1-R4) and columns (C1-C5).
- Connections:
- Row pins (R1-R4): Connected to digital pins D2-D5 on Arduino.
- Column pins (C1-C5): Connected to digital pins D7-D11 on Arduino.
- Each button press connects one row pin to one column pin, allowing detection of which key was pressed by scanning rows and columns.
4. Power Supply:
- The Arduino is powered via USB or an external power adapter. It provides power to all other components through its onboard voltage regulator.

Working Principle of Circuit

The circuit operates as follows:
1. Keypad Input:
- The keypad uses a matrix layout where each button connects a specific row to a column when pressed.
- The Arduino sequentially activates each row by setting it LOW while keeping others HIGH.
- It then scans all columns to detect if any column reads LOW (indicating a pressed key).
- By identifying which row is active and which column reads LOW, the specific pressed key is determined.
2. Processing Input:
- Once a key press is detected, its corresponding value (e.g., number, operator) is sent to the Arduino for processing.

- The Arduino stores user inputs as part of an expression or processes them immediately if they correspond to operations like "CLR" or "=".
3. LCD Output:
- The processed data or results are sent to the LCD for display.
- The LCD operates in 4-bit mode, where data is sent in two parts (higher nibble first, then lower nibble) along with control signals like RS (Register Select) and EN (Enable).
4. Power Distribution:
- All components are powered by the Arduino's 5V output pin.
- A potentiometer connected to the LCD adjusts its contrast for better visibility.

LCD Connection

The LCD operates in 4-bit mode, requiring 6 pins from the Arduino:

- RS (Register Select): Pin 13 (PB5) - Selects between command (0) and data (1) modes

- EN (Enable): Pin 12 (PB4) - Activates LCD instructions when pulsed

- D4: Pin A2 (PC2) - Data bit 4

- D5: Pin A3 (PC3) - Data bit 5

- D6: Pin A4 (PC4) - Data bit 6

- D7: Pin A5 (PC5) - Data bit 7

Keypad Connection

The 4x5 matrix keypad requires 9 pins from the Arduino:

- Rows (1-4): Pins D2-D5 (PD2-PD5)

- Columns (1-5): Pins D7-D11 (PD7, PB0-PB3)

3 Keypad Matrix Operation

Matrix Layout

The calculator uses a 4x5 matrix keypad with the following layout:

sin	cos	tan	ˆ	BS
log	ln	e^x	$\sqrt{}$	(
π	x^2	x^3	1/x	)
asin	acos	atan	y^x	CLR

And for the basic numeric and operator keys:

S	A	C	B	=
1	2	3	4	5
6	7	8	9	0
+	-	*	/	ˆ

Working Principle

The keypad matrix operates on the principle of row-column scanning:

1. All rows are initially set HIGH

2. Each row is sequentially set LOW (activated) while others remain HIGH

3. When a key is pressed, it connects the active row to a specific column

4. The column input (normally pulled HIGH) will read LOW when a key is pressed

5. By detecting which row is active and which column reads LOW, the specific pressed key can be identified

4 Software Implementation

LCD Interface Functions

lcd_init()

Initializes the LCD display in 4-bit mode, configuring the pins and sending initialization commands.

lcd_command()

Sends commands to the LCD in 4-bit mode.

lcd_char()

Sends character data to the LCD in 4-bit mode.

lcd_string()

Displays a string on the LCD.

lcd_clear()

Clears the LCD display.

5 Conclusion

This calculator demonstrates efficient use of hardware resources by using a 4-bit LCD interface and a matrix keypad. It covers direct port manipulation, matrix scanning techniques, and peripheral communication protocols.

6 System Algorithm Flow

The calculator's operation follows this general flow:

1. Initialize hardware (LCD and keypad)

2. Display welcome message

3. Enter main loop:

   (a) Scan keypad for input

   (b) If a key is pressed:

      i. If it's a numeric key or operator, add to expression
      ii. If it's a function key, process accordingly
      iii. If it's "=" key, evaluate the expression
      iv. If it's "CLR" key, clear the display and expression

(c) Update LCD display

4. Return to start of main loop

7 Conclusion

This calculator implementation demonstrates efficient use of hardware resources by using a 4-bit LCD interface and a matrix keypad scanning technique. The code provides robust functions for interacting with the LCD and detecting keypad inputs, which form the foundation for the calculator's user interface.

The design intelligently maps complex calculator functions to a limited number of physical buttons by using a combination of special characters and function keys. The implementation balances hardware limitations with software capabilities to create a fully-functional calculator with both basic arithmetic and advanced scientific functions.

This project showcases fundamental embedded system concepts including:

- Direct port manipulation for efficient I/O operations

- Matrix scanning techniques for input detection

- User interface design with limited display resources