Arduino Based Calculator Using Proteous (2026)

Suvarna L. Ghogare
Sir Visvesvaraya Institute of Technology Nashik, Maharashtra, India.

***Abstract*—A calculator is a basic digital device that perform arithmetic calculation like addition, multiplication, subtraction and division. With the advancement of microcontroller technology, calculators can be designed using embedded systems to make them more compact, efficient, and cost-effective. The Arduino-based calculator is a simple embedded system that uses an Arduino UNO as the main controller. A 4×4 keypad provides input from user such as entering numbers and arithmetic operators. The calculated results are displayed on a 16×2 LCD display. The Arduino reads the input from the keypad, processes the arithmetic operation based on the operator selected, and then displays the result on the LCD. This project helps in understanding the basic concepts of embedded system design, keypad interfacing, LCD interfacing, and programming using Arduino IDE. It is useful for educational purposes and underlying basis for more advanced digital systems.**

***Index Terms*—Arduino UNO,4×4 Keypad,16×2 LCD Display, Digital Calculator, proteous simulation.**

I. INTRODUCTION

This Arduino platform is used by beginners and enthusiasts in the field of electronics due to its less complexity, easy to implement and open-source nature. Arduino is an open-source circuit and software developed by a global community that provides microcontrollers on single-board and different tools for building digital and embedded applications. This includes the Arduino Integrated Development Environment (IDE) with a programmable microcontroller board, which is used to develop and upload programs to the Arduino kit. This project presents the design and implementation of an Arduino-based digital calculator using a 4×4 matrix keypad for input and a 16×2 LCD display for output. The calculator performs basic arithmetic operations including addition, subtraction, multiplication, and division. The system is built around the Arduino UNO microcontroller, which scans the keypad for user input, processes the entered operands and operator, and computes the result using embedded programming in Arduino IDE. The calculated result is displayed on the LCD in real time. The system also handles errors such as division by zero by displaying an appropriate message. The project demonstrates the basic concepts of embedded system design, keypad interfacing, LCD interfacing, and arithmetic processing. It serves as a simple, low-cost, and educational solution for performing basic calculations.

II.RELATED WORK

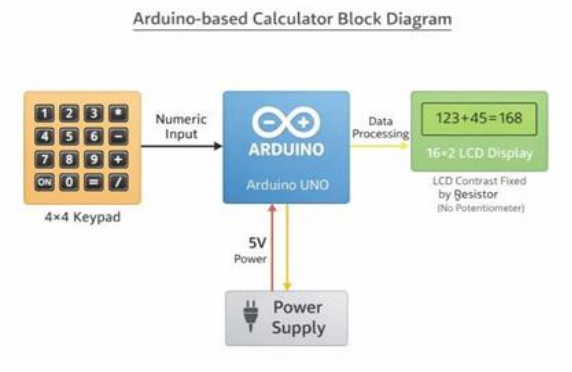
The Arduino platform is popular among software and hardware developer the reason behind this is simplicity, flexibility, open-source nature and low cost.[1] Arduino Based Scientific Calculator, MA Hnin Yu Myaing1, Ma Naing2 1Lecturer, Department of Electronic Engineering, Technological University, Mandalay, Myanmar.2Lecturer, Department of Electronic Engineering, Technological University, Magway, Myanmar. In this electronic scientific calculator develop using Arduino. Using this calculator mathematical operation like square, square root, cosine,sine can be performed. For that purpose, special algorithm and handmade keypad used.[2] Develop a Basic Calculator, Aryash Sengar, Har Ashleen Kaur Department of Computer Science and Engineering Apex Institute of Technology, Chandigarh University, Mohali, Punjab, India. This calculator performs Addition, subtraction, multiplication, and division operations also mean, median, and mode, unit conversions, Deciles, Quartiles, Percentiles, and Kurtosis.[3] Thati, Shubham. (2021). Arduino Based Calculator. SSRN Electronic Journal. 10.2139/ssrn. 3918722.In this in this calculator only some basic calculations are done like addition, subtraction, division and multiplication.

III. PROBLEM DEFINITION

The project aims to design an Arduino-based calculator using Proteus simulation to perform basic arithmetic operations and help understand embedded system design without physical hardware.

IV. PROPOSED SYSTEM

Block diagram:



Respective working of each module is given as below:
Block Description (For Record)
1. Keypad Block
- Used to enter numbers and arithmetic operators
- Sends key signals to Arduino

2. Arduino UNO (Processing Block)
- Central control unit
- Reads keypad input
- Performs arithmetic operations (+, −, ×, ÷)
- Sends output data to LCD

3. LCD Display Block
- Displays entered values and final result
- 16×2 alphanumeric display

4. Power Supply Block
- Provides power to Arduino and peripherals
- Can be USB or external DC adapter

IV. METHODOLOGY

1. System Design Overview
The system is designed as an embedded calculator using an Arduino UNO as the main controller. The design includes:
- Input unit → 4×4 Keypad
- Processing unit → Arduino UNO
- Output unit → 16×2 LCD

2. Hardware/Simulation Setup
- Create the circuit in Proteus simulation software
- Connect:
- Keypad to Arduino digital pins
- LCD to Arduino using appropriate interfacing
- Provide power supply in simulation

3. Input Handling (Keypad Interfacing)
- The keypad is used to enter:
- Numbers (0–9)
- Operators (+, −, ×, ÷)
- Arduino scans keypad rows and columns
- Detects which key is pressed

4. Processing Logic (Arduino Programming)
- Program is written in Arduino IDE
- Steps:
1. Read input from keypad
2. Store operands and operator
3. Perform selected arithmetic operation
4. Handle errors (e.g., division by zero)

5. Output Display (LCD Interfacing)
- The result is displayed on 16×2 LCD
- LCD shows:
- Entered numbers
- Selected operation
- Final result

6. Simulation and Testing
- Run the circuit in Proteus
- Test different cases:
- Addition, subtraction, multiplication, division
- Verify correct output on LCD

7. Result Verification
- Compare calculated output with expected values
- Ensure accuracy and proper functioning

VI. ALGORITHM

Step 1: Start
Begin the program execution.

Step 2: Initialize System
The calculator system is mathematically modeled as a function that maps user inputs to a computed output using predefined arithmetic operations.

- Initialize Arduino pins.
- Initialize the 16×2 LCD display.
- Initialize the 4×4 keypad.
- Display a welcome message on the LCD.

Step 3: Read Keypad Input
- Continuously scan the keypad to detect a key press.
- If no key is pressed, continue scanning.

Step 4: Check Key Type
- If the key is a numeric digit (0–9), store it as part of the current operand.
- If the key is an operator (+, −, ×, ÷), store the operator and switch to second operand input.
- If the key is Clear (C), reset all variables and clear the LCD.

Step 5: Perform Calculation
- If the Equal (=) key is pressed:
  - Perform the arithmetic operation based on the selected operator.
  - Check for division by zero in case of division operation.

Step 6: Display Result
- Display the calculated result or error message on the LCD.

Step 7: Reset for Next Operation
- Clear stored operands and operator.
- Prepare the system for the next calculation.

Step 8: Stop / Repeat
- Repeat Steps 3 to 7 until power is turned off.Working of the Algorithm

1. System
2. Start and Initialization When the power supply is turned ON, the Arduino initializes the LCD display and keypad. A welcome message is shown on the LCD, indicating that the system is ready for operation.
3. Keypad Scanning

The Arduino continuously scans the rows and columns of the 4×4 keypad to detect any key press. This allows the system to identify user input in real time.

4. Number Entry

When a numeric key (0–9) is pressed, the Arduino stores the digit and displays it on the LCD. Multiple digits are combined to form complete numbers (operands).

5. Operator Selection

When an operator key (+, −, ×, ÷) is pressed, the Arduino stores the selected operator and waits for the second number input.

6. Calculation Execution

After entering the second number, pressing the equal (=) key triggers the Arduino to perform the selected arithmetic operation using the stored operands.

7. Error Handling

If a division by zero is detected, the Arduino stops the calculation and displays an error message on the LCD to prevent invalid results.

8. Result Display

The computed result is sent to the 16×2 LCD and displayed clearly to the user.

1. Reset and Repeat Pressing the clear (C) key resets all stored values and clears the display, allowing a new calculation to begin.

Step No.	Step Name	Description	Purpose
1	System Start & Initialization	Arduino initializes LCD and keypad; displays welcome message	To prepare system and ensure all components are ready
2	Keypad Scanning	Arduino scans keypad rows and columns to detect key press	To capture user input in real time
3	Number Entry	User enters digits (0–9); digits are stored and displayed on LCD	To input operands for calculation
4	Operator Selection	User selects operator (+, −, ×, ÷); Arduino stores operator	To define arithmetic operation
5	Calculation Execution	After second number, pressing (=) performs calculation	To process input and generate result
6	Error Handling	Detects division by zero and shows error message	To prevent invalid operations and ensure reliability
7	Result Display	Displays calculated result on 16×2 LCD	To provide output to user
8	Reset & repeat	Pressing (C) clears values and resets system	To allow new calculations

VII. MATHEMATICAL MODEL

1.System Definition
The calculator system is mathematically modeled as a function that maps user inputs to a computed output using predefined arithmetic operations.

2. Input Set (I)
Let the input set I consist of: I= {N1, N2, OP} I = \ {N_1, N_2, OP \} I= {N1, N2, OP} Where:

- N1N_1N1 = First operand (integer)
- N2N_2N2 = Second operand (integer)
- OP∈ {+, −, ×, ÷} OP \in \ {+, -, \times, \div\} OP∈ {+, −, ×, ÷} = Arithmetic operator

3. Output Set (O)
O={R}O = \ {R \} O={R}
Where: RRR = Result of arithmetic operation displayed on LCD.

4. Functional Model
The system function f is defined as:
f (N1, N2, OP) =Rf (N_1, N_2, OP) = Rf (N1, N2, OP) =R

5. Operation Mapping
R={N1+N2,if OP=+N1−N2,if OP=−N1×N2,if OP=×N1N2,ifOP=÷ and N2≠0Error,if OP=÷ and N2=0R = \begin{cases} N_1 + N_2, & \text{if } OP = + \\ N_1 - N_2, & \text{if } OP = - \\ N_1 \times N_2, & \text{if } OP = \times \\ \frac{N_1}{N_2}, & \text{if } OP = \div \text{ and } N_2 \neq 0 \\ \text{Error}, & \text{if } OP = \div \text{ and } N_2 = 0 \end{cases}R= ⎩⎨⎧N1+N2,N1−N2,N1×N2,N2N1,Error, if OP=+if OP=−if OP=×if OP=÷ and N2 OP=÷ and N2 =0

6. Constraints

- range N1, N2∈ZN_1, N_2 \in \mathbb{Z}N1, N2∈Z (integer values)
- Division by zero is not permitted
- Inputs are limited by keypad

7. Mathematical Representation of System
S={I,O,f} S = \ {I, O, f \} S={I,O,f}
Where:

- I = Input set
- O = Output set

VIII. EXPERIMENTAL SETUP

Hardware (Virtual in Proteus):

- Arduino UNO
- 4×4 Keypad
- 16×2 LCD Display
- Connecting wires
- Power supply

Software:

- Proteus Design Suite
- Arduino IDE

Circuit Setup

- The 4×4 keypad is connected to Arduino digital pins for input.
- The 16×2 LCD is interfaced with Arduino to display input and output.
- Proper pin configuration is done for data and control signals.
- Entire circuit is designed and simulated in Proteus.

Program Setup

- Arduino code is written in Arduino IDE.
- Code includes: □=0if
- Keypad scanning logic
- Arithmetic operation logic
- LCD display functions
- The compiled HEX file is generated and loaded into Arduino in Proteus.

Execution Procedure

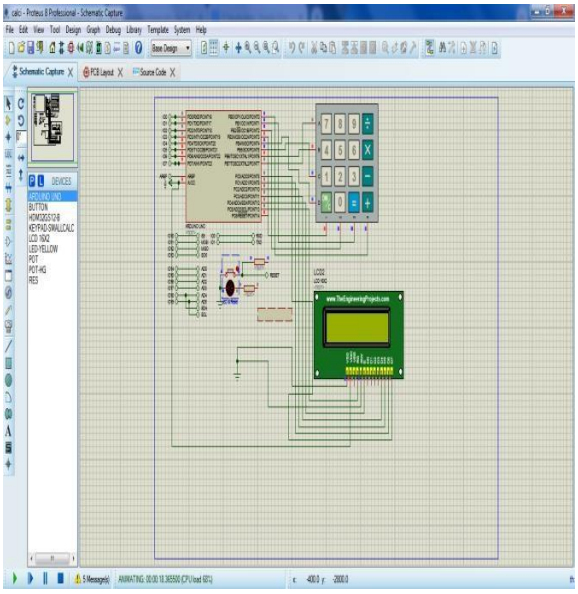
1. Start Proteus simulation
2. LCD displays welcome message
3. Enter numbers using keypad
4. Select operator (+, −, ×, ÷)
5. Press (=) to calculate
6. Result displayed on LCD

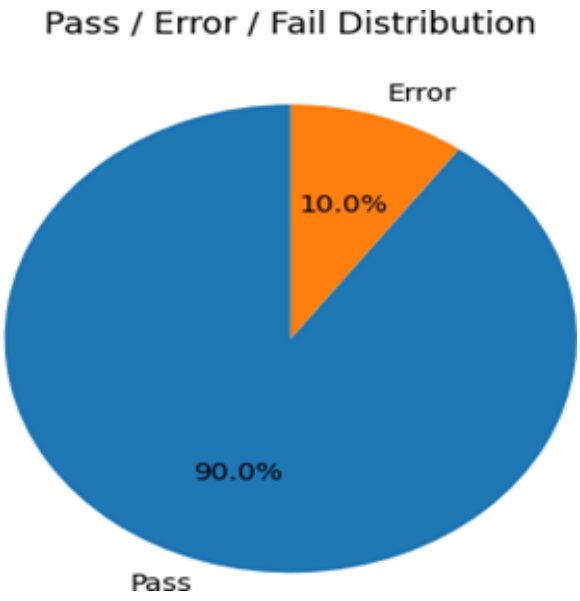
IX. RESULT TABLES AND DISCUSSION

Test Case	Input 1 (N_1)	Operator	Input 2 (N_2)	Expected Output	Actual Output
1	5	+	3	8	8
2	10	−	4	6	6
3	6	×	7	42	42
4	20	÷	5	4	4
5	15	÷	0	Error	Error

6	9	+	8	17	17
7	12	−	15	-3	-3
8	3	×	0	0	0

Table 1: estimate on values for pr opposed System

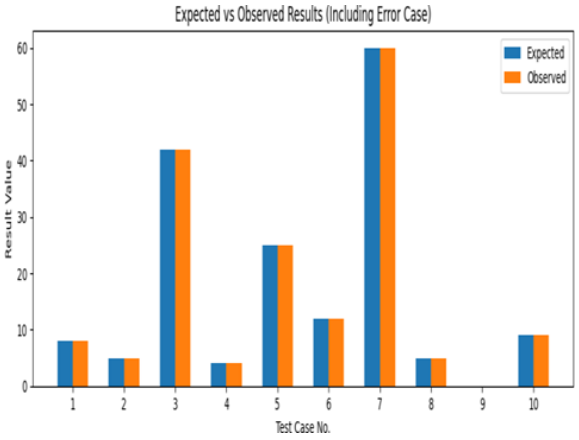




Pie Chart (Pass / Error / Fail)

This shows the percentage of test cases that passed and the error case (division by zero).

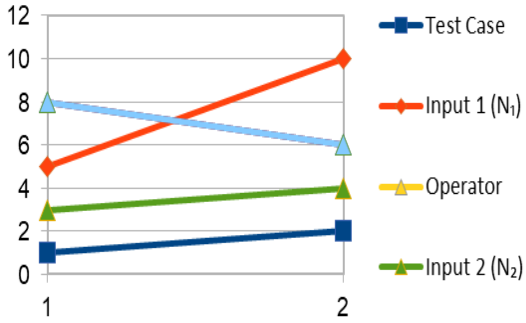
✓□	Pass	=	90%
✓□	Error	=	10%
✓□ Fail = 0%			



Bar Chart (Expected vs Observed)

This includes all test cases, including the error case.

- Test Case 9 is shown as 0 (since it's an error case).
- All other cases show exact match, proving the calculator works correctly.



VII. CONCLUSION

The Arduino-based calculator using Proteus was successfully designed and simulated. The system is capable of performing basic arithmetic operations such as addition, subtraction, multiplication, and division using a 4×4 keypad for input and a 16×2 LCD for output. The project demonstrates the effective use of embedded system design concepts, including keypad interfacing, LCD interfacing, and Arduino programming. The simulation in Proteus helped in testing the system without requiring physical hardware, making it cost-effective and efficient.The calculator produced accurate results for all test cases and handled error conditions like division by zero properly. Overall, the project provides a strong foundation for understanding microcontroller-based

system design and can be further extended to develop more advanced digital systems.

XI. ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported and guided me in the successful completion of this project titled "Arduino Based Calculator Using Proteus (2026)".I am especially thankful to my respected senior staff for their valuable guidance, encouragement, and continuous support throughout the project. Their suggestions and insights helped me to understand the concepts clearly and complete the work effectively.I would also like to thank my institution for providing the necessary facilities and resources required for this project.
Finally, I am grateful to my family and friends for their constant support and motivation.

REFERENCES

[1] M. H. Y. Myaing and M. Naing, "Arduino based scientific calculator," Dept. of Electronic Engineering, Technological University.
[2] Sengar and H. Kaur, "Develop a basic calculator," Dept. of Computer Science and Engineering, Apex Institute of Technology, Chandigarh University, Mohali, Punjab, India.
[3] Anonymous, "Arduino Board Mega 2560," 2018. [Online]. Available: http://www.arduino.cc/en/Main/ArduinoBoardMega2560
[4] Anonymous, "Arduino Pin Mapping 2560," 2018. [Online]. Available: http://www.arduino.cc/en/Hacking/PinMapping2560
[5] T. Agarwal, "Different types of Arduino boards," 2017. [Online]. Available: http://www.elprocus.com/different-types-of-arduino-boards/
[6] "Push-button," *Wikipedia*, 2013. [Online]. Available: https://en.wikipedia.org/wiki/Push-button
[7] Anonymous, "Serial I2C 1602 16x2 character LCD module," 2012. [Online]. Available: http://www.geeetech.com/Serial_I2C_1602_16x2_Character_LCD_Module