JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Simulation of scientific calculator using Arduino uno with Tinkercad

[1]Naga Swetha R, [2] B.Pavitra, [3] D.Narendar Singh, [4] A.Bhavitha, [5] A.Anirudh Goud, [6] A.Rajesh , [7] A.Likhitha Priya, [8] A.Maheshwari, [9] K.Sai Eshwar.

[1,2] Asst.Prof, Dept. of ECE, Anurag University,
[3] Assoc.Prof, Dept. of ECE, Anurag University,
[4,5,6,7,8] B.Tech III year Students, Dept. of ECE, Anurag University

Abstract : The purpose of this project is to utilize Tinker cad, an online platform for designing and modeling electronic circuits, to create and model a scientific calculator. Basic arithmetic operations including power, square, logarithmic, and root functions will be available on the calculator. The accuracy, efficiency, and utility of the calculator will be assessed through a series of tests and evaluations.

IndexTerms - 16x2 LCD display, a 1kohm resistor, a 4x4 keypad, and jumper wires are all components of an Arduino Uno.

I. INTRODUCTION

A wide range of disciplines, from the academic to the professional, rely heavily on scientific calculators in today's digital environment. To create and test a flexible scientific calculator, this project makes use of Tinkercad, a user-friendly web-based platform for electrical prototyping. The goal of this project is to provide a calculator interface that is easy to use and understand, and that can do fundamental arithmetic operations, by combining electrical components with computer algorithms. Using Tinkercad's extensive component library and simulation capabilities, this project demonstrates the practicality and efficiency of building a digital scientific calculator. In the end, this project provides a user-friendly and engaging platform for digital mathematical computation and analysis.

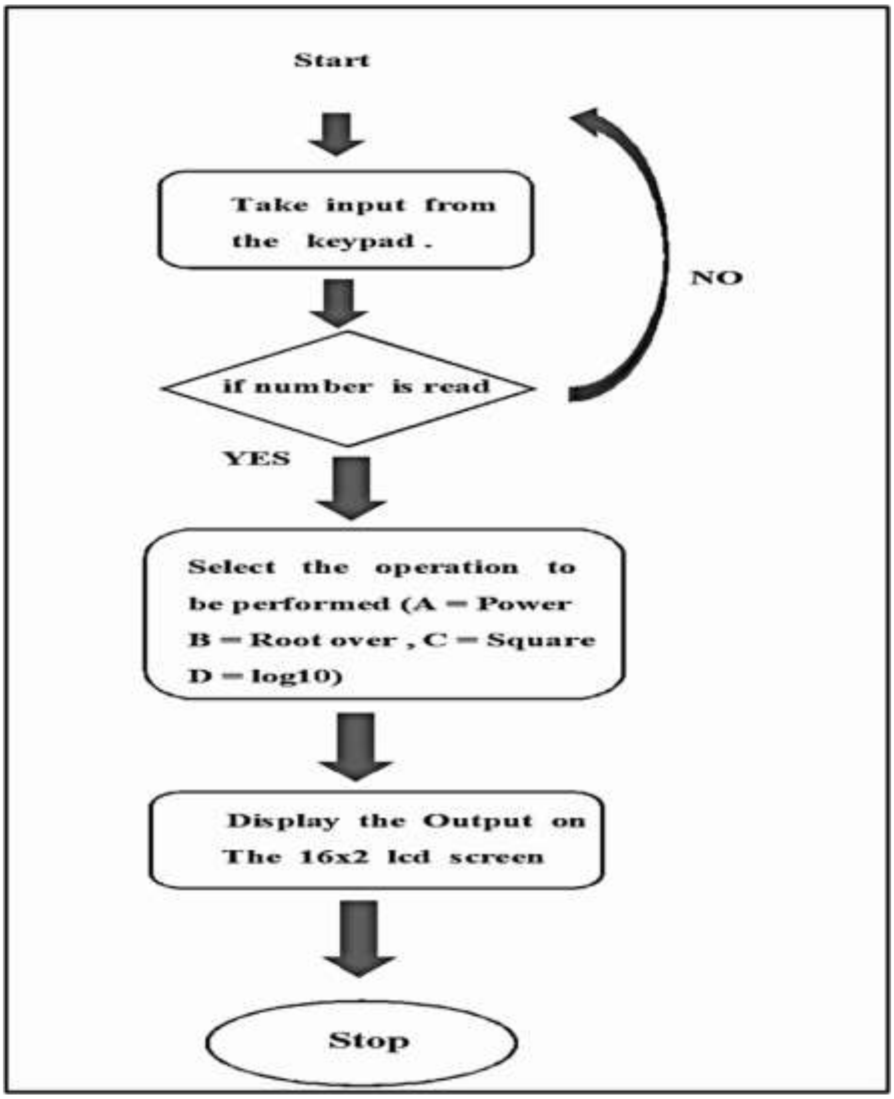
II. SYSTEM DESIGN

The parts that make up the suggested system are inexpensive and easily accessible:

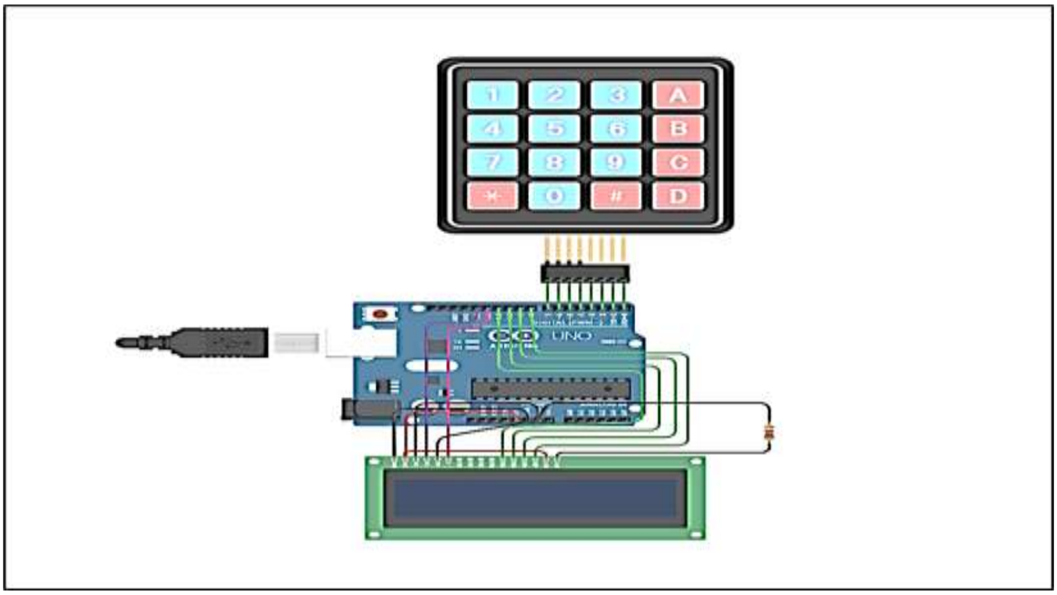
The scientific calculator project's microcontroller unit is the Arduino Uno. It takes in data from the keypad, runs the numbers, and then turns on the LCD screen to show the results. The 16x2 LCD (Liquid Crystal Display) is the calculator's user interface; it shows the input numbers, the results of mathematical operations, and the calculations themselves. Users are able to engage with the calculator through its clear and legible display.

A 1k ohm resistor is utilized to control the amount of current that may pass through the LCD display's backlight. By avoiding an excessive current draw, it safeguards the LCD and guarantees steady functioning. The 4x4 keypad is the primary means by which users input numerical values, mathematical operations, and function keys. It has a grid of sixteen buttons that users may use to enter numbers from zero to nine, as well as additional function keys (#,*,A, B, C, D) for computations. First step in operating the system is to press the 4x4 keypad to enter user input. The user's keypad presses are read and interpreted by the Arduino Uno microcontroller. Third, based on the user's input sequence, the system determines the mathematical expression or function. The internal computing algorithms in the Arduino Uno process the mathematical expression or function once the input sequence is recognized. The system determines the operation's outcome after processing the mathematical phrase. After that, the accurately computed result is shown on the 16x2 LCD screen.

2.1 System Block Diagram


Start
Take input from the keypad .
NO
if number is read
YES
Select the operation to be performed (A = Power B = Root over , C = Square D = log10)
Display the Output on The 16x2 lcd screen
Stop

Choose the operation that has to be executed (A = Power, B = Root over, C = Square, and D = log10).Present the Results on the 16x2 LCD Monitor



2.2 System Implementation

1. Prepare Your Computer for Working with Arduino: - Download and install the Arduino IDE. - Join your computer to your Arduino board and set it up.

2. Create the User Interface: - Create the user interface on a display that is suitable, like an LCD screen. Put in buttons for things like numbers, powers, squares, roots, logarithms, and so on.
3. Following the circuit schematic, connect the display and input buttons to the Arduino board.
   This completes step three of connecting the components.
4. Initialize Library Files: - Make sure that your Arduino sketch includes all of the required libraries, such Liquid Crystal. h, which is used to operate the LCD display.
5. Declare variables to contain input values, operator selections, and result values. This will initialize the variables.
6. configuration of the Display: - Initiate the LCD display during the configuration process. Create the layout of the screen, including the labels for the various buttons and their functions.
7. Handling Input: - Write code to process button presses and other user input. You may detect button pushes and change the input data accordingly using switch-case statements or if-else situations.
8. Square Root Function: - Create a function that takes an integer and returns its square root, either by utilizing the built-in sqrt() function or by developing a unique approach.
9. Logarithm Function: - Use the log() function or a user-specified base to compute logarithms, or the natural logarithm (base e).

10. Result Display (Section 10): - Revise the display to reflect the latest input values, chosen operations, and computation outcomes.
11. Error managing: - Add support for managing errors caused by incorrect operations, such as division by zero or taking the square root of negative values.
    In case of errors, show them on the LCD panel.
12. Run in Continuous Mode: — Let the calculator run several computations in parallel without having to reset between each one.
13. Testing: Try out different input values and operations on the calculator to make sure it works properly and produces correct results.
14. Make Final Adjustments and Optimizations: - Improve code performance and readability by refactoring and optimizing it.
    - Incorporate any suggestions for new features or enhancements made by test subjects.

2.3 Step-By-Step System Testing:

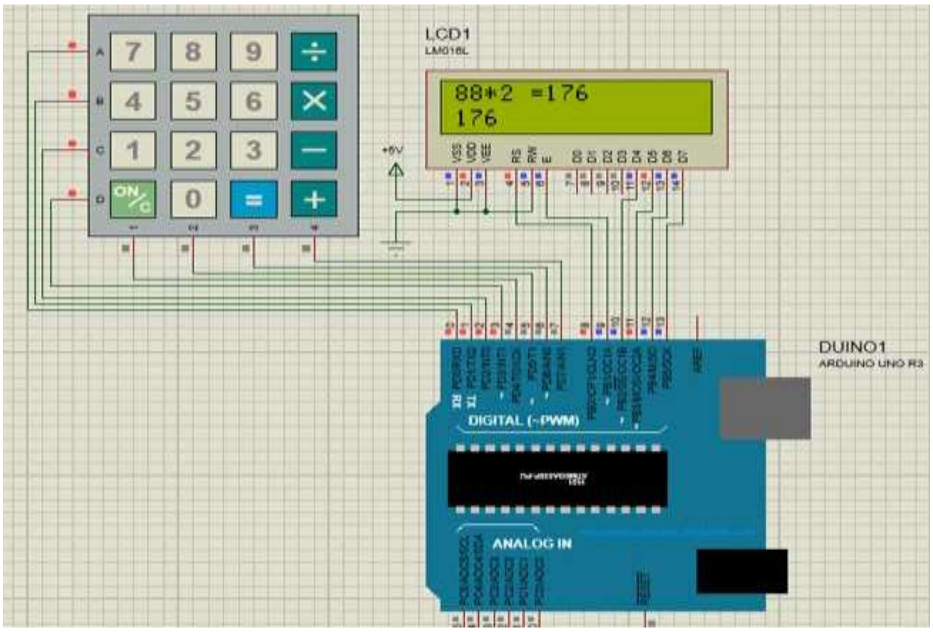
1. First, Gather All Necessary Components: Obtain an Arduino board, an LCD screen, input buttons, resistors, and cables, among other things.
2. Get the Arduino Setup Ready: - Follow the circuit schematic to connect the LCD display and input buttons to the Arduino.
3. Input Software: - Create the scientific calculator project's Arduino sketch, which should contain code for the user interface, input processing, arithmetic operations, square root, logarithms, and so on.
4. Upload and Compile: - Use the Arduino IDE to compile the sketch.
   - Transfer the assembled code to the Arduino boards.
5. Turn on the Power Source: - Be sure the Arduino board's LCD screen is lit up before turning it on.
6. User Interface Verification: - Press buttons to test the user interface and make sure the input is shown accurately on the LCD.
7. Test Your Square Root: - Type a number and find its square root. Make sure the square root you got is accurate.

8. Calculate Logarithm: - Press the corresponding button to test the logarithm function. Enter a number and the function will return its logarithm. The correctness of the outcome must be confirmed.
9. Test for Error Handling: - Evaluate the system's ability to handle common types of erroneous inputs, such as negative

integers, square roots of non-positive values, and division by zero. Make sure the calculator smoothly handles these instances and provides suitable error messages.

III. ADVANTAGES:

1. Affordable: Unlike specialized scientific calculators, Arduino boards and components are more affordable, making them a great choice for personal or school applications
2. Personalization: The calculator's UI, features, and operations are all up for grabs when it comes to personalization.
3. Adaptability: Arduino's programmability makes it easy to update or add new functionalities, as well as change the user interface.
4. Fourthly, it's a great chance to master some new skills in electronics, programming, and mathematics by constructing a scientific calculator using Arduino.
5. Connectivity: You may extend the calculator's capabilities—for example, by adding temperature sensors or wireless connectivity—by integrating it with a variety of sensors and modules.
6. Thanks to the Arduino open-source community, there are a plethora of online resources, tutorials, and libraries that may help you fix problems and improve your project.
7. Portability: A scientific calculator based on Arduino can be portable and suited for usage in diverse situations depending on the form factor and components employed.
8. The calculator may be upgraded to include new features or enhance performance by modifying its hardware or software components.
9. Application-Specific Functions: In addition to the usual mathematical functions, you have the option to incorporate application-specific functions or algorithms. This allows you to do calculations in engineering or scientific analyses with ease.
10. Using Arduino to build and program your scientific calculator is a great way to unleash your creativity while also honing your technical abilities and learning about new developments in computers and electronics.



IV. LIMITATIONS:

1. Processing Power Limitations: A comparison with dedicated computing devices reveals that Arduino boards have significantly less processing power and memory. As a result, the complexity and speed of computations may be severely constrained.
2. Small Screen: The quantity of information and accuracy provided on the calculator's interface could be affected by

the small screen size of most LCD panels that are compatible with Arduino.

3. Input/Output Limitations: The amount of input buttons or sensors that may be directly attached to an Arduino board may be limited due to the board's limited number of digital and analog pins.
4. Accuracy and accuracy: Calculations based on Arduino may display modest errors, particularly in complex mathematical processes, because to restrictions in numerical accuracy and floating-point arithmetic.
5. Battery Life: The amount of time an Arduino-based calculator may be run on a single charge depends on the power consumption of its individual components and how often it performs calculations.
6. The Difficulty of Applying Advanced Functions: Applying algorithms or complicated mathematical functions on Arduino could be difficult since it might necessitate more processing resources or optimizations.
7. Graphical Capabilities: Unlike graphing calculators, Arduino-based calculators might not be able to display all kinds of graphs or mathematical operations.
8. Extra Points of Failure Due to Dependence on External Components: The calculator's performance and functionality could be affected by factors outside the calculator's control, such power sources, display modules, or sensors.
9. You Need to Know Your Ways Around Code: If you're not technically savvy or just starting out, you could find that building and programming an Arduino-based calculator is too much for you.
10. Updates and Maintenance: Just like any other electrical equipment, calculators built on the Arduino platform may need regular updates, troubleshooting to fix software or hardware issues, or maintenance.

V. CONCLUSION:

I have learned a lot about electronics, programming, and mathematics via this project to build an Arduino-based scientific calculator. Arduino was able to be customized and integrated with many functions and features thanks to its cost-effectiveness and flexibility. The calculator was able to execute standard arithmetic, square roots, logarithms, and user-defined functions despite its limited processing power and screen size. The benefits of open-source communities and the tools accessible for Arduino development were demonstrated in the project. Although the calculator encountered difficulties with complexity and precision, it showed promise for mobility and future upgrades. In conclusion, I have enjoyed and learned a lot from this project, which has demonstrated the usefulness of Arduino in both academic and individual contexts.

VI. REFERENCES:

1. Abhilash Kanakanti and D.Singh, “College Buses And Students Monitoring System With IOT”, International Journal of Advanced Research in Advance Engineering & Technology, 6 (1), 200–204, (2017).D. N. Singh and U. Divya, "An intelligent parking guidance and information system with ARM9 microcontroller", Int. J. Latest Trends Eng. Technol., vol. 2, pp. 234-239, Sep. 2013.
2. D. Narendhar Singh, M. Hema and M. Joseph Stalin, "IoT Based Healthcare Monitoring for Driver's Community", International Journal of Engineering Science and Computing, March 2017.
3. Swetha, R. Naga and Gona, Ashwini and Singh, D. Narendar, IoT Based Smart Garbage Monitoring System with Geo-Tag (February 21, 2020). Proceedings of the 4th International Conference: Innovative Advancement in Engineering & Technology (IAET) 2020, Available at SSRN: https://ssrn.com/abstract=3554257 or http://dx.doi.org/10.2139/ssrn.3554257
4. D.Narendar Singh, P. Ramakrishna, J. Nikhileshwar, G. Akshaya, K. Karthik, & K. Vinitha Lakshmi. (2022). Lorawan-Bassed Satellite Monitoring System Uses IMU Sensor. *RES MILITARIS*, *12*(4),1300–1309.Retrieved from https://resmilitaris.net/index.php/resmilitaris/article/view/1949
5. **Abhilash Kanakanti and D.Singh, “College Buses And Students Monitoring System With IOT”, International Journal of Advanced Research in Advance Engineering & Technology, 6 (1), 200–204, (2017).**
6. Pavitra, B., Singh, D. N., Sharma, S. K., & Hashmi, M. F. (2023). Dementia prediction using novel IOTM (Internet of Things in Medical) architecture framework. *Intelligent Data Analysis*, (Preprint), 1-17.

7. Pavitra, B., Singh, D. N., & Hashmi, M. F. (2021). Voice-Controlled Biped Walking Robot for Industrial Applications. *Innovations in the Industrial Internet of Things (IIoT) and Smart Factory*, 79.

8. D. N. Singh, B. Pavitra, A. Singh, and J. A. Reddy, "Performance of IoT-Enabled Devices in Remote Health Monitoring Applications," in Computational Intelligence in Medical Decision Making and Diagnosis, CRC Press, 2023, pp. 131–140.

9. **Singh, D. N. (2013). Ravi teja ch. v.". Vehicle Speed Limit Alerting and Crash Detection System at Various Zones "International Journal of Latest Trends in Engineering and Technology (IJLTET) IJLTET, 2(1).**

10. D.Narendar Singh and K.Tejaswi (M.Tech), "Real Time Vehicle Theft Identity and Control System Based on ARM 9'',International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol. 2 Issue 1 January 2013.

11. https://projecthub.arduino.cc/123samridhgarg/arduino-calculator-bce0df

12. https://github.com/nikita9604/Scientific-Calculator-with-Keypad

13. https://www.instructables.com/Scientific-Calculator-With-Evive-arduino-Powered-E/

14. https://circuitdigest.com/microcontroller-projects/arduino-calculator-using-4x4-keypad