PROJECT OVERVIEW:

This project provides a detailed study of the Calculator, an important electronic device used for performing mathematical calculations. The project explains the meaning, history, types, working principle, applications, advantages, and limitations of calculators. It highlights how calculators help students, professionals, and individuals in solving mathematical problems quickly and accurately.

The aim of this project is to understand the importance of calculators in education and daily life, and to learn how they simplify complex calculations. Through this project, students can gain knowledge about different types of calculators and their practical uses in various fields such as education, banking, engineering, and science.

1.OBJECTIVES

OBJECTIVES OF THE PROJECT

The main objectives of this project are:

1. ***To understand what a calculator is and its importance.***
2. ***To study the history and development of calculators.***
3. ***To identify different types of calculators and their uses.***
4. ***To learn the basic working principle of a calculator.***
5. ***To understand the applications of calculators in education and daily life.***
6. ***To know the advantages and limitations of using calculators.***
7. ***To improve knowledge about mathematical calculations using calculators.***

2.INTRODUCTION

calculator is an electronic device used to perform mathematical calculations easily and quickly. It is designed to solve basic arithmetic operations such as addition, subtraction, multiplication, and division, as well as complex mathematical problems. Calculators are widely used by students, teachers, engineers, scientists, and business professionals.With the advancement of technology, calculators have become more efficient, compact, and accurate. They help reduce human effort and minimize errors in

calculations. Today, calculators play an important role in education and daily life by saving time and improving accuracy in mathematical

3.BLOCK DIAGRAM

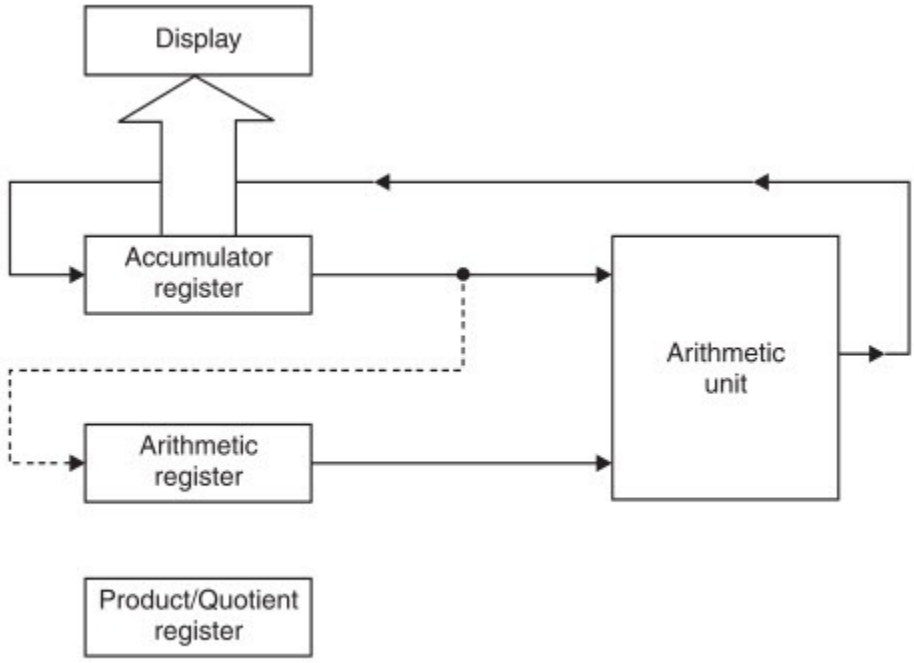


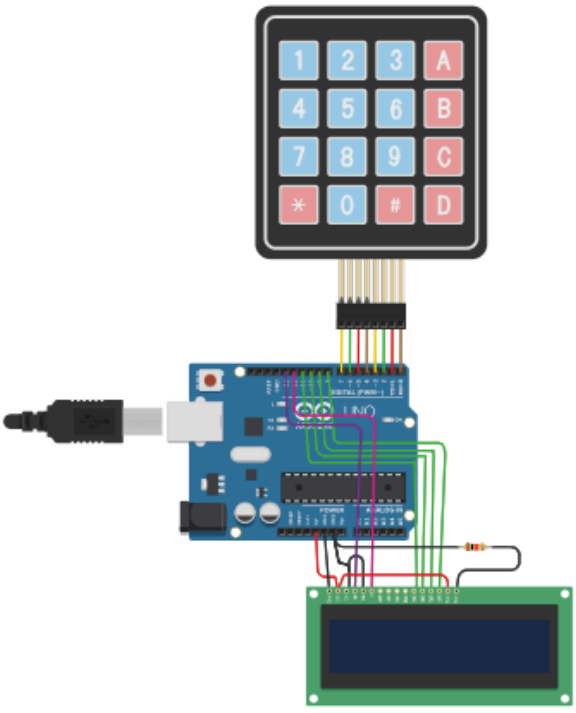
Fig. 46.2 *Block diagram showing register structure of a calculator*

4.COMPONENTS

Component	*Function*
Input Device (Keyboard/Keypad)	*Allows the user to enter numbers and operations.*
Display	*Shows the numbers entered and the results of calculations.*
Central Processing Unit (CPU) / Processor	*Performs the arithmetic and logical operations.*

Memory / Storage	***Stores intermediate results, constants, and instructions.***
Power Supply	***Provides the necessary electrical power for the calculator to function.***
Control Unit	***Directs the flow of data between input, processor, memory, and output.***
Arithmetic Logic Unit (ALU)	***Handles all mathematical calculations and logical operations.***
Output Device	***Delivers the final result (usually integrated with the display).***

5.WIRINGAND



PIN DIAGRAM

6.BUILDING CIRCUITS IN TINKERCAD

Step 1: Place Components

1. **Drag an Arduino Uno onto the workspace.**
2. **Drag a breadboard.**
3. **Place the 16x4 keypad and 16x2 LCD.**

Step 2: Connect the Keypad to Arduino

A 4x4 keypad has 8 pins (4 rows, 4 columns). Connect them to Arduino digital pins, for example:

Keypad Pin	Arduino Pin
R1	9
R2	8
R3	7
R4	6
C1	5
C2	4

C3	*3*
C4	*2*

Step 3: Connect the LCD

For a 16x2 LCD (without I2C module):

- *RS → Arduino 12*
- *EN → Arduino 11*
- *D4-D7 → Arduino 5,4,3,2 (or any free pins)*
- *VSS → GND*
- *VDD → 5V*
- *V0 → Potentiometer middle pin (for contrast)*
- *RW → GND*

If using I2C LCD, just connect:

- *SDA → A4*
- *SCL → A5*

Step 4: Arduino Code

You need Keypad and LiquidCrystal libraries. Here's a basic example for addition:

```
#include <Keypad.h>
```

```
#include <LiquidCrystal.h>


const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

  {'1','2','3','+'},

  {'4','5','6','-'},

  {'7','8','9','*'},

  {'C','0','=','/'}

};


byte rowPins[ROWS] = {9, 8, 7, 6};

byte colPins[COLS] = {5, 4, 3, 2};


Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS,
COLS);

LiquidCrystal lcd(12, 11, 10, 9, 8, 7, 6); // Adjust if not using
I2C


String input = "";

float result = 0;
```

```
void setup(){
  lcd.begin(16,2);
  lcd.print("Calculator");
}

void loop(){
  char key = keypad.getKey();

  if(key){
    if(key == 'C'){
      input = "";
      lcd.clear();
    } else if(key == '='){
      result = eval(input);
      lcd.clear();
      lcd.print(result);
      input = "";
    } else {
      input += key;
      lcd.setCursor(0,1);
      lcd.print(input);
```

```
    }
  }
}

// Simple eval function (supports +, -, *, /)
float eval(String exp){
  char op;
  int index = -1;
  if(exp.indexOf('+')!=-1){op='+'; index=exp.indexOf('+');}
  else if(exp.indexOf('-')!=-1){op='-'; index=exp.indexOf('-');}
  else if(exp.indexOf('*')!=-1){op='*'; index=exp.indexOf('*');}
  else if(exp.indexOf('/')!=-1){op='/'; index=exp.indexOf('/');}
  else return exp.toFloat();

  float a = exp.substring(0,index).toFloat();
  float b = exp.substring(index+1).toFloat();

  switch(op){
    case '+': return a+b;
    case '-': return a-b;
    case '*': return a*b;
```

```
    case '/': return a/b;
  }
  return 0;
}
```

8.CODE EXPLANATION

1. Libraries

The code begins by including two libraries: Keypad and LiquidCrystal.

- The Keypad library allows the Arduino to detect which button you press on a keypad.
- The LiquidCrystal library lets the Arduino display numbers and text on an LCD screen.

---

2. Keypad Setup

- The keypad has 4 rows and 4 columns.
- Each key is assigned a value, like numbers 0–9, the operators (+, -, ×, ÷), = for calculate, and C for clear.
- The Arduino pins connected to the rows and columns of the keypad are defined.
- A keypad object is created so Arduino can read key presses easily.

---

3. LCD Setup

- *The LCD is connected to specific Arduino pins.*
- *The LCD will be used to show the numbers you type and the results of calculations.*

---

4. Variables

- *`input` stores the sequence of keys you press as a text string.*
- *`result` stores the outcome of a calculation.*

---

5. Setup Function

- *Runs once when the Arduino starts.*
- *Initializes the LCD to a 16-column, 2-row display.*
- *Displays the word "Calculator" as a welcome message.*

---

6. Main Loop

- *The Arduino constantly checks if a key is pressed on the keypad.*
- *If a key is pressed:*
    - *If it is `C`, the input and LCD are cleared.*
    - *If it is `=`, the input string is sent to the evaluation function, the calculation result is returned, and the LCD shows the result. Input is then reset for a new calculation.*

- *Otherwise (number or operator), the key is added to the input string, and the LCD displays the current input.*

---

7. Evaluation Function

- *The function checks which operator is in the input string (`+`, `-`, `*`, `/`).*
- *The input string is split into two numbers: the part before the operator and the part after.*
- *The numbers are converted from text to numeric values.*
- *Based on the operator, the corresponding calculation is performed: addition, subtraction, multiplication, or division.*
- *The result is returned to the main loop and displayed on the LCD.*

---

8. How the Program Works Together

1. *The user types numbers and an operator on the keypad.*
2. *The input is stored as a string.*
3. *Pressing `=` calculates the result using the evaluation function.*
4. *The LCD displays both the input as it is typed and the final result after calculation.*
5. *Pressing `C` clears everything so you can start a new calculation.*

9.TESTING AND DEBUGGING IN TINKERCAD

Testing

1. *Check connections – Ensure all wires, keypad, and display pins are correct.*
2. *Test components individually – e.g., blink LED, press keypad keys and see output in Serial Monitor.*
3. *Run simulation – Click Start Simulation in Tinkercad.*
4. *Check inputs – Press keys and see if Arduino registers them.*
5. *Check outputs – Verify numbers and results appear correctly on the display.*

Debugging

1. *Verify wiring – Wrong pin connections cause most issues.*
2. *Use Serial Monitor – Print key presses and calculations to check logic.*
3. *Test operations separately – Add, subtract, multiply, divide one by one.*
4. *Check code logic – Ensure formulas and variable updates are correct.*
5. *Restart simulation if glitchy – Sometimes Tinkercad needs a reset.*

10. WHAT-IF TESTING AND BEHAVIOR ANALYSIS

Testing

1. *Check if all keypad buttons work.*
2. *Test each operation (add, subtract, multiply, divide).*
3. *Verify the display shows correct results.*
4. *Test edge cases (e.g., divide by zero, large numbers).*

Behaviour Analysis

1. *Check if results are correct.*
2. *Check if the calculator responds quickly to inputs.*
3. *See how it handles errors (like invalid inputs).*
4. *Check stability (no crashes or freezes).*
5. *Check user-friendliness (easy to use).*

11.OBSERVATIONS

Observation

1. *Input – Keypad buttons register correctly.*
2. *Operations – Addition, subtraction, multiplication, division give correct results.*
3. *Display – Numbers and results appear properly.*
4. *Error Handling – Dividing by zero or invalid input shows appropriate response.*
5. *Responsiveness – Calculator reacts immediately to key presses.*
6. *Stability – No crashes or freezes during use.*
7. *User Interaction – Easy and clear to operate*

https://www.tinkercad.com/things/3igaiRb3Kro-tremendous-kieran?sharecode=xmqlvjBLiQFcoeFmeeKvVDKXFHN_G6TMjEx0__nKv_E