Arduino Based Keypad and LCD Calculator

1. Introduction

This project demonstrates how to build a basic calculator using a 4x4 matrix keypad and a 16x2 LCD display with an Arduino. It allows users to perform addition, subtraction, multiplication, and division directly through the keypad.

2. Key Components

- Arduino Uno board

- 16x2 LCD Display

- 4x4 Matrix Keypad

- 10kΩ Potentiometer (for LCD contrast adjustment)

- Breadboard

- Jumper wires

- USB Cable (for programming)

3. Working Principle

- **Step 1:** The user inputs numbers and operations via the 4x4 keypad.

- **Step 2:** The Arduino detects key presses and differentiates between numeric inputs and operations (+, -, *, /).

- **Step 3:** The first number is collected and displayed on the LCD.

- **Step 4:** Upon selecting an operator, the Arduino stores the first number and waits for the second number input.

- **Step 5:** When the user presses '=', the Arduino performs the selected arithmetic operation.

- **Step 6:** The result is displayed on the LCD, and the calculator is ready for a new operation.

- **Step 7:** Pressing 'C' clears the LCD and resets the operation.

4. Circuit Overview

- **LCD Pins:**

    - RS → Arduino Pin 13

- o   E → Arduino Pin 12

- o   D4 → Arduino Pin 11

- o   D5 → Arduino Pin 10

- o   D6 → Arduino Pin 9

- o   D7 → Arduino Pin 8

- **Keypad Connections:**

  - o   Rows connected to Arduino Pins 7, 6, 5, 4

  - o   Columns connected to Arduino Pins 3, 2, 1, 0

- **Power Supply:**

  - o   LCD VCC and Keypad VCC to Arduino 5V

  - o   Ground connections to Arduino GND

- **Additional:**

  - o   10kΩ Potentiometer connected to LCD for contrast adjustment.

5. Code

```
/*Code Written by-

Fuad Hasan

BME, KUET

*/


#include <Keypad.h>

#include <Wire.h>

#include <LiquidCrystal.h>


LiquidCrystal lcd(13, 12, 11, 10, 9, 8);


long first = 0;
```

```cpp
long second = 0;

double total = 0;


char customKey;

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] = {7,6,5,4}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {3,2,1,0}; //connect to the column pinouts of the keypad


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


void setup()

{

 lcd.begin(16,2);   //start lcd

 lcd.setCursor(0,0);

 lcd.print("Calculator");

 lcd.setCursor(0,1);

 lcd.print("Enter Numbers");
```

```arduino
  delay(4000);

  lcd.clear();

  lcd.setCursor(0,0);

}


void loop()

{

  customKey = customKeypad.getKey();

  switch(customKey)

  {

    case '0' ... '9':

    lcd.setCursor(0,0);

    first = first * 10 + (customKey - '0');

    lcd.print(first);

    break;


    case '+':

    case '-':

    case '*':

    case '/':

    first = (total != 0 ? total : first);

    lcd.print(customKey);

    second = secondNumber();

    performOperation(customKey);

    break;
```

```
    case 'C':

    total = 0;

    first = 0;

    second = 0;

    lcd.clear();

    break;

  }

}


long secondNumber()

{

 long secondNum = 0;

 while(1)

 {

  customKey = customKeypad.getKey();

  if(customKey >= '0' && customKey <= '9')

  {

   secondNum = secondNum * 10 + (customKey - '0');

   lcd.setCursor(7,0);

   lcd.print(secondNum);

  }

  if(customKey == '=')

   break;

 }

 return secondNum;

}
```

```
void performOperation(char op)

{

 lcd.setCursor(0,3);

 if(op == '+')

  total = first + second;

 else if(op == '-')

  total = first - second;

 else if(op == '*')

  total = first * second;

 else if(op == '/')

 {

  if(second == 0)

   lcd.print("Invalid");

  else

   total = (float)first / (float)second;

 }

 lcd.print(total);

 first = 0;

 second = 0;

}
```

6. Code Explanation

- **Library Inclusion:**
  - #include <Keypad.h>, #include <Wire.h>, #include <LiquidCrystal.h> to handle keypad and LCD.
- **LCD and Keypad Setup:**

- LCD initialized on pins 13, 12, 11, 10, 9, and 8.
  - Keypad connected to pins 7-0 of Arduino.

- **Setup Function:**
  - Displays "Calculator" and "Enter Numbers" messages for 4 seconds.

- **Loop Function:**
  - Continuously reads key inputs.
  - Numbers are concatenated to form multi-digit numbers.
  - Operator buttons trigger capturing of the second number and performing calculations.

- **secondNumber() Function:**
  - Collects digits until '=' is pressed.

- **performOperation() Function:**
  - Performs the arithmetic operation based on the pressed operator.
  - Handles division-by-zero error.

- **Clear ('C') Button:**
  - Resets all variables and clears the LCD screen.

7. Conclusion

This Arduino-based calculator project effectively demonstrates the integration of a keypad, LCD display, and basic programming concepts. It serves as an ideal project to understand how embedded systems can handle user inputs and display outputs, opening the door for building more advanced electronic applications.