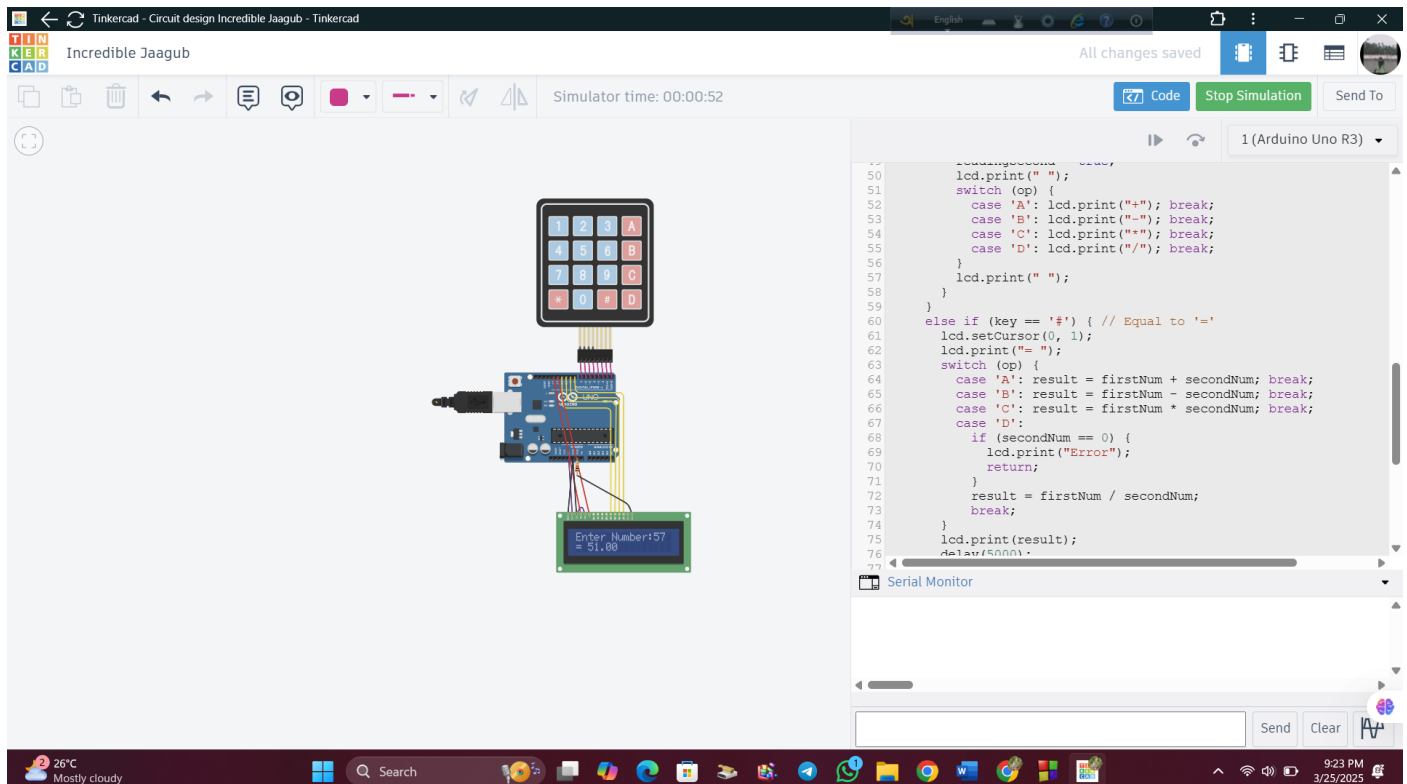


Summary of the *DIY Calculator Using Arduino* Video

The video explains how to create a **DIY calculator** using an **Arduino Uno**, a **4x4 keypad**, and a **16x2 LCD display**. The creator **Harshita** walks through the hardware setup, circuit connections, and the Arduino code for the calculator.

1. Hardware Setup & Circuit Explanation

- **Components Used:**
 - **4x4 keypad** (for number input)
 - **16x2 LCD display** (to show calculations)
 - **Arduino Uno** (to process inputs and display results)
- **Keypad Connections:**
 - Rows (1-4) connected to digital pins 7, 6, 5, 4
 - Columns (1-4) connected to digital pins 3, 2, 1, 0
- **LCD Connections:**
 - Data pins (DB4-DB7) → Arduino 11, 10, 9, 8
 - Enable (E) → 12
 - Register Select (RS) → 13
 - Power (VCC, LED) connected to 5V supply
 - Ground (GND, RW, V0) properly grounded



Code:

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

```

#include <Wire.h>

#include <LiquidCrystal.h>

// LCD Pins: RS, EN, D4, D5, D6, D7
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

// Keypad setup
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}};
byte rowPins[ROWS] = {7, 6, 5, 4};
byte colPins[COLS] = {3, 2, 1, 0};
Keypad customKeypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// Variables
float firstNum = 0, secondNum = 0, result = 0;
char op = '\0';
bool readingSecond = false;

void setup() {
  lcd.begin(16, 2);
  lcd.print("Calculator Ready");
  delay(2000);
  lcd.clear();
  lcd.print("Enter Number:");
}

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

  if (key) {
    if (key >= '0' && key <= '9') {
      lcd.print(key);

      if (!readingSecond) {
        firstNum = firstNum * 10 + (key - '0');
      }
    }
  }
}

```

```

else {
    secondNum = secondNum * 10 + (key - '0');} }
else if (key == 'A' || key == 'B' || key == 'C' || key == 'D') {
    if (!readingSecond) {
        op = key;
        readingSecond = true;
        lcd.print(" ");
        switch (op) {
            case 'A': lcd.print("+"); break;
            case 'B': lcd.print("-"); break;
            case 'C': lcd.print("*"); break;
            case 'D': lcd.print("/"); break; }
        lcd.print(" ");    } }
else if (key == '#') { // Equal to '='
    lcd.setCursor(0, 1);
    lcd.print("=");
    switch (op) {
        case 'A': result = firstNum + secondNum; break;
        case 'B': result = firstNum - secondNum; break;
        case 'C': result = firstNum * secondNum; break;
        case 'D':
            if (secondNum == 0) {
                lcd.print("Error");
                return;    }
            result = firstNum / secondNum;
            break;    }
    lcd.print(result);
    delay(5000);
    resetCalculator(); }
else if (key == '*') { // Clear
    resetCalculator(); }}

```

```
void resetCalculator() {  
    lcd.clear();  
    lcd.print("Enter Number:");  
    firstNum = 0;  
    secondNum = 0;  
    result = 0;  
    op = '\0';  
    readingSecond = false;}  

```

Arduino Code Overview

- Libraries Used:
 - Keypad.h for keypad functionality
 - Wire.h for Arduino-LCD communication
 - LiquidCrystal.h for LCD control
- Key Features of Code:
 - Initializes keypad and LCD
 - Stores numeric inputs and operations
 - Uses **switch cases** for addition, subtraction, multiplication, and division
 - Handles multi-digit input and displays results
 - Special keys:
 - C (clear display)
 - = (calculate result)
 - +, -, *, / for operations
 - Includes a function to read the **second operand**
 - **Error Handling:** If the user tries to divide by zero, the display shows "Invalid"

Simulation & Testing on TinkerCAD

- The creator demonstrates the circuit working in **TinkerCAD**, an online simulator.
- Example calculations:
 - $98 - 100 = -2.00$
 - $9 * 11 = 99.00$
- The LCD correctly displays results with decimal values.
- The C button successfully clears the display for a new calculation.