Calculator Project Using Tinkercad

This project involves creating a simple calculator using an Arduino Uno R3, a 4x4 keypad, and a 16x2 LCD display. The calculator performs basic arithmetic operations such as addition, subtraction, multiplication, and division.

Components Used

1. Arduino Uno R3
2. 4x4 Keypad
3. 16x2 LCD Display
4. Breadboard
5. Jumper Wires
6. Potentiometer (for adjusting LCD contrast)

Connections

LCD Display:

- Power and Ground**:**
  - VCC (pin 2) to 5V on the Arduino.
  - GND (pin 1) to GND on the Arduino.
  - LED+ (pin 15, anode) to 5V.
  - LED- (pin 16, cathode) to one end of a 1kΩ resistor.
  - The other end of the 1kΩ resistor to GND on the Arduino.
- Data and Control Lines**:**
  - RS (pin 4) to digital pin 13 on the Arduino.
  - RW (pin 5) to GND.
  - E (pin 6) to digital pin 12 on the Arduino.
  - D4 (pin 11) to digital pin 11 on the Arduino.
  - D5 (pin 12) to digital pin 10 on the Arduino.
  - D6 (pin 13) to digital pin 9 on the Arduino.
  - D7 (pin 14) to digital pin 8 on the Arduino.
- Contrast Control**:**
  - V0 (pin 3) to the middle pin of the potentiometer.
  - One end of the potentiometer to 5V.
  - The other end of the potentiometer to GND.

Keypad*:*

- Rows:
  - R1 to digital pin 7 on the Arduino.
  - R2 to digital pin 6 on the Arduino.
  - R3 to digital pin 5 on the Arduino.
  - R4 to digital pin 4 on the Arduino.
- Columns:
  - C1 to digital pin 3 on the Arduino.
  - C2 to digital pin 2 on the Arduino.
  - C3 to digital pin 1 on the Arduino.

- C4 to digital pin 0 on the Arduino.

Code

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

// Initialize the LCD library with the numbers of the interface pins
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

long first = 0;
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

// Initialize an instance of class Keypad
Keypad customKeypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
```

```
void setup() {
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("Calculator");
  lcd.setCursor(0, 1);
  lcd.print("Enter numbers");
  delay(4000);
  lcd.clear();
  lcd.setCursor(0, 0);
}

void loop() {
  customKey = customKeypad.getKey();
  switch (customKey) {
    case '0' ... '9': // This keeps collecting the first value until an operator is pressed
      lcd.setCursor(0, 0);
      first = first * 10 + (customKey - '0');
      lcd.print(first);
      break;

    case '+':
      first = (total != 0 ? total : first);
      lcd.print("+");
      second = SecondNumber(); // get the collected the second number
      total = first + second;
      lcd.setCursor(0, 1);
      lcd.print(total);
      first = 0;
      second = 0; // reset values back to zero for next calculation
```

```
  break;

case '-':
  first = (total != 0 ? total : first);
  lcd.print("-");
  second = SecondNumber();
  total = first - second;
  lcd.setCursor(0, 1);
  lcd.print(total);
  first = 0;
  second = 0;
  break;

case '*':
  first = (total != 0 ? total : first);
  lcd.print("*");
  second = SecondNumber();
  total = first * second;
  lcd.setCursor(0, 1);
  lcd.print(total);
  first = 0;
  second = 0;
  break;

case '/':
  first = (total != 0 ? total : first);
  lcd.print("/");
  second = SecondNumber();
  lcd.setCursor(0, 1);
```

```
      if (second == 0) {
        lcd.print("Invalid");
      } else {
        total = (float) first / (float) second;
        lcd.print(total);
      }
      first = 0;
      second = 0;
      break;

    case 'C':
      total = 0;
      lcd.clear();
      break;
  }
}

long SecondNumber() {
  while (1) {
    customKey = customKeypad.getKey();
    if (customKey >= '0' && customKey <= '9') {
      second = second * 10 + (customKey - '0');
      lcd.setCursor(7, 0);
      lcd.print(second);
    }

    if (customKey == '=') break;
  }
  return second;
```

}

How is it helpful to an electronics and communication student?

From my perspective as an electronics student, working on the calculator project using Arduino and Tinkercad has been incredibly insightful. It's given me a hands-on opportunity to delve into microcontroller interfacing, which is crucial for understanding how different electronic components communicate and work together.

Through this project, I've strengthened my Arduino programming skills by learning how to read input from a keypad, execute arithmetic operations, and display results on an LCD screen. This practical application has deepened my understanding of programming logic and control structures, which are foundational in electronics and communication engineering.

Designing the circuit on Tinkercad allowed me to visualize and simulate the connections between components such as the LCD display, keypad, and Arduino. This experience has been invaluable for understanding circuit behavior and troubleshooting potential issues before implementing the physical circuit.

Moreover, completing this project has reinforced my knowledge of basic electronics concepts, including digital logic and circuit design principles. It's bridged the gap between theoretical knowledge from textbooks and practical application in real-world projects.

As I progress in my studies, projects like these not only enhance my technical skills but also cultivate creativity and problem-solving abilities. They serve as a testament to my capability to apply classroom learning to practical scenarios, preparing me for future challenges in the field of electronics and communication engineering.

Link for output on tinkercad

https://www.tinkercad.com/things/hiMJecbcNsd-daring-jaagub-hillar/editel?sharecode=_n3hUOxe37VMbEguaBrPsqwj_35jBhOFR9kTvgihUfw