Bolck of CIRCUIT:

Click here:calculator.ino - Wokwi Arduino and ESP32 Simulator

Diagram connections:

```
{
  "version": 1,
  "author": "Uri Shaked",
  "editor": "wokwi",
  "parts": [
    { "id": "uno", "type": "wokwi-arduino-uno", "top": 200, "left": 20 },
    {
      "id": "keypad",
      "type": "wokwi-membrane-keypad",
      "left": 360,
      "top": 100,
      "attrs": {
        "keys": [
          "1", "2", "3", "+",
          "4", "5", "6", "-",
          "7", "8", "9", "*",
          ".", "0", "=", "/"
        ]
      }
    },
    { "id": "lcd", "type": "wokwi-lcd1602", "top": 8, "left": 20 },
    {
      "id": "r1",
      "type": "wokwi-resistor",
      "top": 140,
      "left": 220,
      "attrs": { "value": "220" }
    }
  ],
  "connections": [
    ["uno:GND.1", "lcd:VSS", "black", ["v-51", "*", "h0", "v18"]],
    ["uno:GND.1", "lcd:K", "black", ["v-51", "*", "h0", "v18"]],
    ["uno:GND.1", "lcd:RW", "black", ["v-51", "*", "h0", "v18"]],
    ["uno:5V", "lcd:VDD", "red", ["v16", "h-16"]],
    ["uno:5V", "r1:2", "red", ["v16", "h-118", "v-244", "h50"]],
    ["r1:1", "lcd:A", "pink", []],
    ["uno:12", "lcd:RS", "blue", ["v-16", "*", "h0", "v20"]],
    ["uno:11", "lcd:E", "purple", ["v-20", "*", "h0", "v20"]],
    ["uno:10", "lcd:D4", "green", ["v-24", "*", "h0", "v20"]],
    ["uno:9", "lcd:D5", "brown", ["v-28", "*", "h0", "v20"]],
    ["uno:8", "lcd:D6", "gold", ["v-32", "*", "h0", "v20"]],
    ["uno:7", "lcd:D7", "gray", ["v-36", "*", "h0", "v20"]],
    ["uno:A3", "keypad:C1", "brown", ["v76", "*", "h0", "v0"]],
    ["uno:A2", "keypad:C2", "gray", ["v80", "*", "h0", "v0"]],
    ["uno:A1", "keypad:C3", "orange", ["v84", "*", "h0", "v0"]],
    ["uno:A0", "keypad:C4", "pink", ["v88", "*", "h0", "v0"]],
    ["uno:5", "keypad:R1", "blue", ["v-34", "h96", "*", "v12"]],
    ["uno:4", "keypad:R2", "green", ["v-30", "h80", "*", "v16"]],
    ["uno:3", "keypad:R3", "purple", ["v-26", "h64", "*", "v20"]],
    ["uno:2", "keypad:R4", "gold", ["v-22", "h48", "*", "v24"]]
  ]
}
```

Program for Arduino

```
/**
  Arduino Calculator


  Copyright (C) 2020, Uri Shaked.
  Released under the MIT License.
*/


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


/* Display */
LiquidCrystal lcd(12, 11, 10, 9, 8, 7);


/* Keypad setup */
const byte KEYPAD_ROWS = 4;
const byte KEYPAD_COLS = 4;
byte rowPins[KEYPAD_ROWS] = {5, 4, 3, 2};
byte colPins[KEYPAD_COLS] = {A3, A2, A1, A0};
char keys[KEYPAD_ROWS][KEYPAD_COLS] = {
  {'1', '2', '3', '+'},
  {'4', '5', '6', '-'},
  {'8', '8', '9', '*'},
  {'.', '0', '=', '/'}
};
```

```
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, KEYPAD_ROWS, KEYPAD_COLS);

uint64_t value = 0;

void showSpalshScreen() {
  lcd.print("GoodArduinoCode");
  lcd.setCursor(3, 1);
  String message = "Calculator";
  for (byte i = 0; i < message.length(); i++) {
    lcd.print(message[i]);
    delay(50);
  }
  delay(500);
}

void updateCursor() {
  if (millis() / 250 % 2 == 0 ) {
    lcd.cursor();
  } else {
    lcd.noCursor();
  }
}

void setup() {
  Serial.begin(115200);
  lcd.begin(16, 2);

  showSpalshScreen();
  lcd.clear();
  lcd.cursor();


  lcd.setCursor(1, 0);
}


char operation = 0;
String memory = "";
String current = "";
uint64_t currentDecimal;
bool decimalPoint = false;


double calculate(char operation, double left, double right) {
  switch (operation) {
    case '+': return left + right;
    case '-': return left - right;
    case '*': return left * right;
    case '/': return left / right;
  }
}


void processInput(char key) {
  if ('-' == key && current == "") {
    current = "-";
    lcd.print("-");
    return;
  }


  switch (key) {
```

```
  case '+':
  case '-':
  case '*':
  case '/':
    if (!operation) {
      memory = current;
      current = "";
    }
    operation = key;
    lcd.setCursor(0, 1);
    lcd.print(key);
    lcd.setCursor(current.length() + 1, 1);
    return;

  case '=':
    float leftNum = memory.toDouble();
    float rightNum = current.toDouble();
    memory = String(calculate(operation, leftNum, rightNum));
    current = "";
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print(memory);
    lcd.setCursor(0, 1);
    lcd.print(operation);
    return;

}

if ('.' == key && current.indexOf('.') >= 0) {
  return;
  }

  if ('.' != key && current == "0") {
    current = String(key);
  } else if (key) {
    current += String(key);
  }

  lcd.print(key);
}

void loop() {
  updateCursor();

  char key = keypad.getKey();
  if (key) {
    processInput(key);
  }
}
```