

ARDUINO BASED SCIENTIFIC CALCULATOR



A calculator is typically a portable electronic device that is used to perform mathematical operations which can be basic (like addition, subtraction, etc.) or slightly advanced (such as exponentials, cosine, tangent, etc).

always provides the correct answer, whereas there's a scope of human error, when the same calculations are done by hand.

Our program produces a working model of a calculator using an Arduino UNO board. It can evaluate basic operations as well as a few advanced math functions. This provides a simple, portable recreated version of the machine we've all known for almost a century.

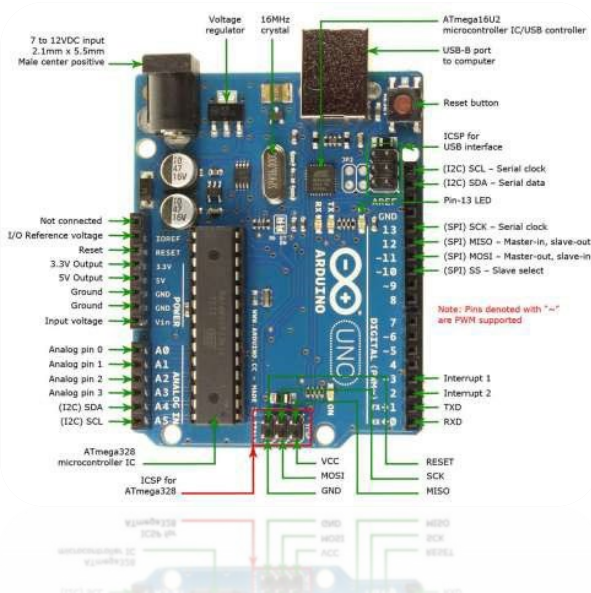
Serial No.	Components	Amount
1	Arduino UNO	1
2	4×4 Keypad	1
3	Jumper Wire (Male to Male)	8
4	LCD (16×2) with I2C module	1
5	Jumper Wire (Male to Female)	4
6	Power bank (or power source)	1
7	USB A-B cable	1

❑ ARDUINO:

“Arduino is an open-source electronics

Simply put, the Arduino is a tiny computer system that can be programmed with our instructions to interact with various forms of input and outputs.

Among many other types - differing primarily in the micro controller, Arduino UNO is a small sized board (using ATmega328).



❖ Hardware Parts

- ☐ Universal Serial Bus (USB) connectors
 - ✓ power supply
 - ✓ upload instructions
 - ✓ send / receive data from computer
 - ✓ **Microcontroller**
 - ✓ processes / executes instructions
 - ✓ memory (EPROM) to hold program
 - ✓ input / output pins
 - ✓ **Power & Analog Sockets**
- ☐ Digital input / output pins
- ☐ RESET button

❑ KEYPAD

- ✓ ACCEPT ALPHA-NUMERIC INPUT FROM USERS
- ✓ USES 8 PINS FOR 16 BUTTONS

WHEN A KEY IS PRESSED, IT SHORTS THE CORRESPONDING ROW & COLUMN. TO CHECK, EACH ROW PIN IS CONNECTED TO DIGITAL OUTPUT AND EACH COLUMN TO DIGITAL INPUT.

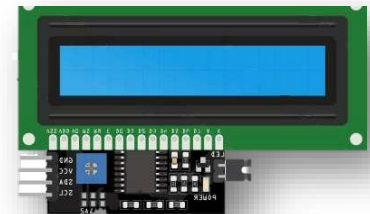
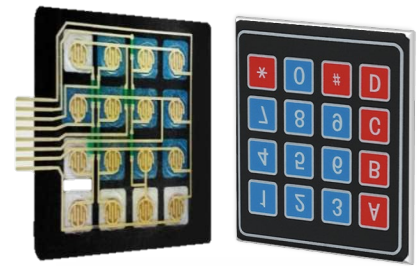
THEN TURNING ALL OUTPUT HIGH, THE SHORTED INPUT MAY BE DETECTED EASILY.

❑ I2C LIQUID CRYSTAL DISPLAY (LCD)

A 16×2 CHARACTER LCD (EACH CHARACTER = 5×8 PIXEL RECTANGLE), HAS A BACKLIGHT AND DISPLAYS 32 ASCII CHARACTERS WITH 16 ON EACH ROW OF TWO. CONTRAST MAY BE ADJUSTED BY

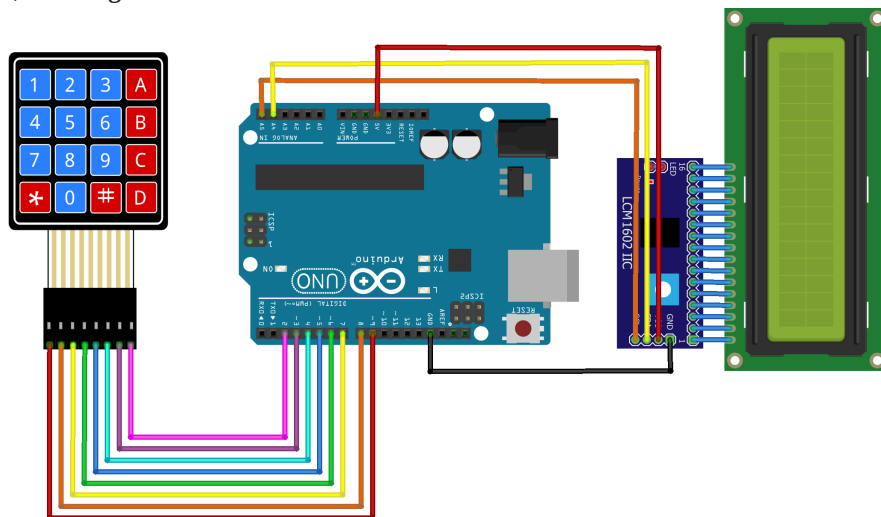
TURNING A POTENTIOMETER ON THE I2C MODULE.

- ✓ TWI (TWO-WIRE INTERFACE) - CONTROL & COMMUNICATE
- ✓ SERIAL DATA PIN - USED FOR BOTH TRANSMIT & RECEIVE
- ✓ SERIAL CLOCK PIN - TIMING SIGNAL GIVEN BY BUS MASTER DEVICE



SCHEMATIC DIAGRAM:

To draw the diagram, *Fritzing* software has been used.



LIBRARY INCLUDE:

To run the program, we must include a library function from Library Manager.

- **Keypad by Community** <https://github.com/Chris-A/Keypad> (Version 3.1.1): Keypad is a library for using matrix style keypads with the Arduino. As of version 3.0 it now supports multiple keypresses. This library is based upon the Keypad Tutorial. It was created to promote Hardware Abstraction. It improves readability of the code by hiding the pinMode and digitalRead calls for the user.

To use the library: `#include<Keypad.h>`

- **LiquidCrystal I2C by Marco Schwartz** (Version 1.1.2): The *LiquidCrystal* library allows an Arduino board to control LiquidCrystal displays (LCDs) based on the Hitachi HD44780 (or a compatible) chipset, which is found on most text-based LCDs. The library works with in either 4- or 8-bit mode (i.e. using 4 or 8 data lines in addition to the rs, enable, and, optionally, the rw control lines). The *LiquidCrystal I2C* library allows to control I2C displays with functions extremely similar to LiquidCrystal library.

To use the library: `#include<LiquidCrystal_I2C.h>`

PROGRAM CODE:

```
#include<Keypad.h>
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C sujaylcd(0x27,16,2);
const byte row =4;
const byte col = 4;
char key[row][col]= {
    {'1','2','3','D'}, // at last of this row there is null "\0" character
    {'4','5','6','*'}, // automatically assigned to tell computer that its
    {'7','8','9','C'}, // the end of this row.
    {'#','0','A','B'}
};
char key1[row][col]= {
    {'s','c','t','e'},
    {'q','b','T','('},
    {'l','^','o','')'},
    {'m','p','n','r'}
};
byte rowpin[row]={9,8,7,6};
byte colpin[col]={5,4,3,2};

int angle=0,i=0,ans=0,k=0,s=0,j,l,b=0;
float answer,num[10];
int inv_angle;
double sina;
double cosa;
double tana;
float asina,acosa,atana,expa,loga,root;
double pi = 3.1415926535 ;
int sinp=10, cosp=11, tanp=12, logp=13;
char action[10];
String que;
/*****Two keypads for more keys to access*****/

Keypad keysujay = Keypad(makeKeymap(key), rowpin , colpin , row , col);
Keypad keysujay1 = Keypad(makeKeymap(key1), rowpin , colpin , row , col);

void setup() {
    Serial.begin(9600);
    sujaylcd.clear();
    sujaylcd.init();
    sujaylcd.backlight();
}

void loop() {
/*****retrieving answer by calling bracket_answer function*****/

    answer=bracket_answer(0);
    sujaylcd.setCursor(0,1);
    sujaylcd.print("=");
    sujaylcd.print(answer);
    sujaylcd.setCursor(0,0);
    int r=0;
    i=0;
    s=0;
    while(r==0){
        char val=getval();
/*****resetting the calculator*****/

        if(val and val=='r'){
            r=1;
            s=0;
        }
    }
}
```

```

        sujaylcd.clear();
        que='\u0000';
    }
/*****shifting answer display to left of right *****/

    else if (val and val=='m'){
        s=0;
        if((que.length())>16 and i>=0){
            i++;
            sujaylcd.setCursor(0,0);
            sujaylcd.print("                                ");
            sujaylcd.setCursor(0,0);
            sujaylcd.print(que.substring(i,i+16));
        }
    }
    else if (val and val=='n'){
        s=0;
        if((que.length())>16 and i>0){
            i--;
            sujaylcd.setCursor(0,0);
            sujaylcd.print("                                ");
            sujaylcd.setCursor(0,0);
            sujaylcd.print(que.substring(i,i+16));
        }
    }
    else if (val and val=='#'){
        s=1;
    }
    else;
}
}

```

/Initialization of function for taking input of numbers and actions , then stroing into arrays called num[] and action[] then calculatin g answer by calling other supporting functions defined below, returning the answer as float number/

```

float bracket_answer(int i){
int b=0;
int l=i;
char val;
sujaylcd.print(que);
while(b==0){
    while(ans!=1){
        val=getval();
        Serial.println(val);
        if (val and val=='s'){
            i++;
            num[i]=getsin();
        }
        else if (val and val=='c'){
            i++;
            num[i]=getcos();
        }
        else if (val and val=='t'){
            i++;
            num[i]=gettan();
        }
        if (val and val=='q'){
            i++;
            num[i]=getsinInv();
        }
        else if (val and val=='b'){
            i++;

```

```

        num[i]=getcosInv();
    }
    else if (val and val=='b'){
        i++;
        num[i]=getcosInv();
    }
    else if (val and val=='l'){
        i++;
        num[i]=getlog();
    }
    else if (val and val=='o'){
        i++;
        num[i]=getroot();
    }
    else if (val and val=='e'){
        i++;
        num[i]=getexp();
    }
    else if(val and val=='('){
        printall(val);
        sujaylcd.clear();
        s=0;
        i++;
        j=i;
        Serial.print(s);
        delay(200);
    }
    /*****Recursion relation used to operate bracket action first*****/
    /e.g 4-(3+2) //her bracket_answer function will be entered into bracket to return 3+2=5
    and 2nd bracke-answer will be called to calculate 4-5=-1./

        num[i+1]=bracket_answer(j-1);
        i=j;
    /*****/

    }

    /*****/get numbers and actions*****/

    else if(val and val=='0' || val=='1' || val=='2' || val=='3' || val=='4' ||
        val=='5' || val=='6' || val=='7' || val=='8' || val=='9' ||
        val=='p'){
        int j=0;
        i++;
        num[i]=0;
        if(val=='p'){
            num[i]=(num[i]*10)+ pi;
        }
        else{
            num[i]=(num[i]*10)+ (val-'0');
        }
        printall(val);
        while(j<10){
            char val = getval();
            if(val and val=='0' || val=='1' || val=='2' || val=='3' ||
            val=='4' || val=='5' || val=='6' || val=='7' || val=='8' ||
            val=='9' || val=='p' ){
                printall(val);
                if(val=='p'){
                    num[i]=(num[i]*10)+ pi;
                }
                else{
                    num[i]=(num[i]*10)+ (val-'0');
                }
            }
            j++;

```

```
    }
    else if(val and val=='*'){
        action[i]='*';
        printall(val);
        break;
    }
    else if(val and val=='B'){
        action[i]='B';
        printall('+');
        break;
    }
    else if(val and val=='C'){
        action[i]='C';
        printall('-');
        break;
    }
    else if(val and val=='D'){
        action[i]='D';
        printall('/');
        break;
    }
    else if(val and val=='^'){
        action[i]='^';
        printall(val);
        s=0;
        break;
    }
    else if(val and val=='A' ){
        ans=1;
        break;
    }
    else if(val and val==')'){
        ans=1;
        printall(val);
        s=0;
        break;
    }
    else if(val and val=='#'){
        delay(200);
        s++;
    }
}
val=' ';
}

else if(val and val=='*' || val=='D' || val=='B' || val=='C' || val=='^'){
    if(val and val=='*'){
        action[i]='*';
        printall('x');
    }
    else if(val and val=='B'){
        action[i]='B';
        printall('+');
    }
    else if(val and val=='C'){
        action[i]='C';
        printall('-');
    }
    else if(val and val=='D'){
        action[i]='D';
        printall('/');
    }
    else if(val and val=='^'){
        action[i]='^';
    }
}
```

```

        printall(val);
        s=0;
    }
}
else if(val and val=='A' ){
    ans=1;
    break;
}
else if(val and val=='#'){
    s++;
    delay(200);
    val=' ';
}
else if(val and val==' '){
    ans=1;
    s=0;
    break;
}
else if(val and val=='^'){
    printall(val);
}

}

/***** numbers and actions are stored in arrays*****/
/ If the input was like 5+3-4*6/7+2 , Then the arrays are as follows:
    num[]={5,3,4,6,7,2} and action[]={ '+', '-', '*', '/', '+' }
*****/

    answer=getanswer1(action,l,i);
    b=1;
    ans=0;
}
return answer;
}

int getangle(){
    int i=0;
    int angle=0;
    while(i<10){
        char val = getval();
        if(val and val=='0' || val=='1' || val=='2' || val=='3' ||
            val=='4' || val=='5' || val=='6' || val=='7' || val=='8' ||
            val=='9' || val=='p'){
            printall(val);
            if(val=='p'){
                angle=(angle*10)+pi;
            }
            else{
                angle=(angle*10)+(val-'0');
            }
            i++;
        }
        else if(val and val==' ' and val!='#'){
            sujaylcd.print("");
            que=que+val;
            s=0;
            return angle;
            break;
        }
        else if(val and val=='#'){
            s=1;
        }
    }
}

```



```
    }

    float getanswer1(char action[10],int l,int i){
        int j=i-1;
        while(j>1){
            if(action[j]=='^'){
                num[j]=pow(num[j],num[j+1]);
                action[j]=action[j+1];
                i--;
                for(k=j+1;k<=i;k++){
                    num[k]=num[k+1];
                    action[k]=action[k+1];
                }
                j--;
            }
            else {
                j--;
                continue;
            }
        }
        j=i-1;
        while(j>1){
            if(action[j]=='D'){
                num[j]=(float) num[j]/(float) num[j+1];
                action[j]=action[j+1];
                i--;
                for(k=j+1;k<=i;k++){
                    num[k]=num[k+1];
                    action[k]=action[k+1];
                }
                j--;
            }
            else {
                j--;
                continue;
            }
        }
        j=i-1;
        while(j>1){
            if(action[j]=='*'){
                num[j]=num[j]*num[j+1];
                action[j]=action[j+1];
                i--;
                for(k=j+1;k<=i;k++){
                    num[k]=num[k+1];
                    action[k]=action[k+1];
                }
                j--;
            }
            else {
                j--;
                continue;
            }
        }
        j=i-1;
        while(j>1){
            if(action[j]=='C'){
                num[j]=num[j]-num[j+1];
                action[j]=action[j+1];
                i--;
                for(k=j+1;k<=i;k++){
                    num[k]=num[k+1];
                    action[k]=action[k+1];
                }
            }
        }
    }
}
```

```

        }
        j--;
    }
    else {
        j--;
        continue;
    }
}
j=i-1;
while(j>1){
    if(action[j]=='B'){
        num[j]=num[j]+num[j+1];
        action[j]=action[j+1];
        i--;
        for(k=j+1;k<=i;k++){
            num[k]=num[k+1];
            action[k]=action[k+1];
        }
        j--;
    }
    else{
        j--;
        continue;
    }
}
return num[i]; // Final answer returned.....
}

```

```

float getsin(){
    sujaylcd.print("sin\(");
    que=que+"sin\(";
    s=0;
    angle=getangle();
    sina = sin(angle*DEG_TO_RAD);
    return sina;
}

```

```

float getcos(){
    sujaylcd.print("cos\(");
    que=que+"cos\(";
    s=0;
    angle=getangle();
    cosa = cos(angle*DEG_TO_RAD);
    return cosa;
}

```

```

float gettan(){
    sujaylcd.print("tan\(");
    que=que+"tan\(";
    s=0;
    angle=getangle();
    tana = tan(angle*DEG_TO_RAD);
    return tana;
}

```

```

float getsinInv(){
    sujaylcd.print("sinInv\(");
    que=que+"sinInv\(";
    s=0;
}

```

```

        angle=getangle();
        asina = asin(angle)*RAD_TO_DEG;
        return asina;

    }
    float getcosInv(){
        sujaylcd.print("cosInv\");
        que=que+"cosInv\(";
        s=0;
        angle=getangle();
        acosa = acos(angle)*RAD_TO_DEG;
        return acosa;

    }
    float gettanInv(){
        sujaylcd.print("tanInv\");
        que=que+"tanInv\(";
        s=0;
        angle=getangle();
        atana = atan(angle)*RAD_TO_DEG;
        return atana;

    }
    float getlog(){
        sujaylcd.print("log\");
        que=que+"log\(";
        s=0;
        angle=getangle();
        loga = log(angle);
        return loga;

    }
    float getroot(){
        sujaylcd.print("sqrt\");
        que=que+"sqrt(";
        s=0;
        angle=getangle();
        root = sqrt(angle);
        return root;

    }
    float getexp(){
        sujaylcd.print("exp\");
        que=que+"exp(";
        s=0;
        angle=getangle();
        expa = exp(angle);
        return expa;

    }
}
/*****Defining getval() function to get val from keypad using shift key*****/

char getval(){
    char val;
    if(s==0){
        val=keysujay.getKey();
    }
    else{
        val=keysujay1.getKey();
    }

    return val;
}

```

/******Printing total answer on lcd monitor, if number of character gets overflowed lcd is shifted to increase area of vision******/

```
void printall(char val){
    que=que+val;
    sujaylcd.clear();
    if((que.length())>16 and i>=0){
        i++;
        sujaylcd.setCursor(0,0);
        sujaylcd.print("                                ");
        sujaylcd.setCursor(0,0);
        sujaylcd.print(que.substring(i,i+16));
    }
    else{
        sujaylcd.setCursor(0,0);
        sujaylcd.print(que);
    }
}
```

RESULTS:

At first the setup of all parts needs to be completed. After that, compile and uploading the code to the board using *Arduino IDE* software. Calculator wil strat operating like any other calculator.

e.g.

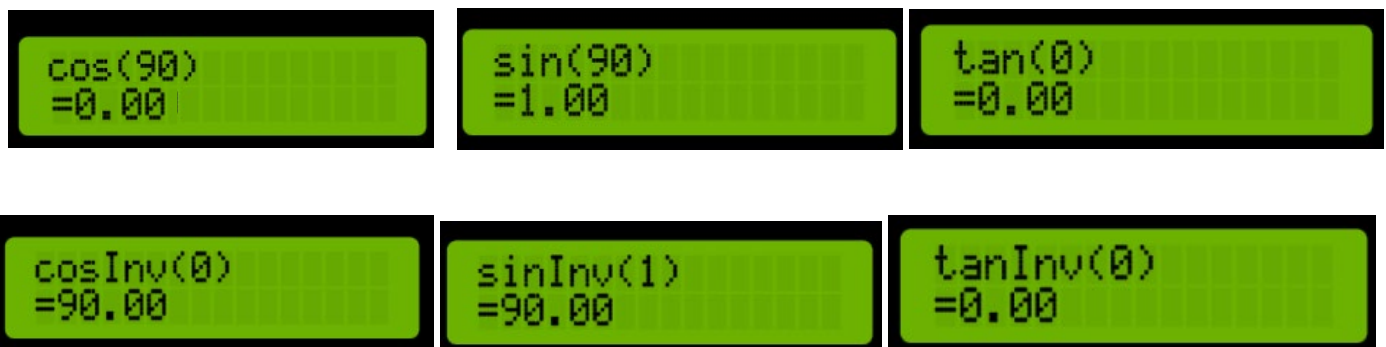
BODMAS RULE:



BRACKET GETS PRIORITY OVER BODMAS RULE:



TRIGONOMETRIC FUNCTIONS:

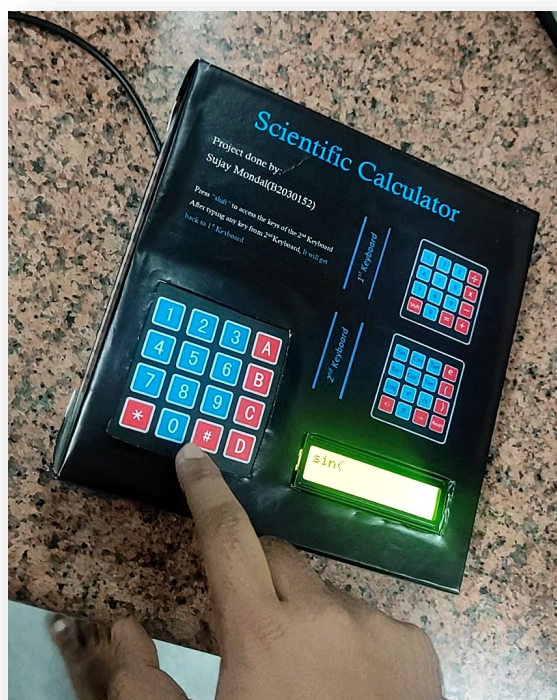


LOGARTHIMIC AND EXPONENTTIAL FUNCTIONS:**ANOTHER FEATURES:**

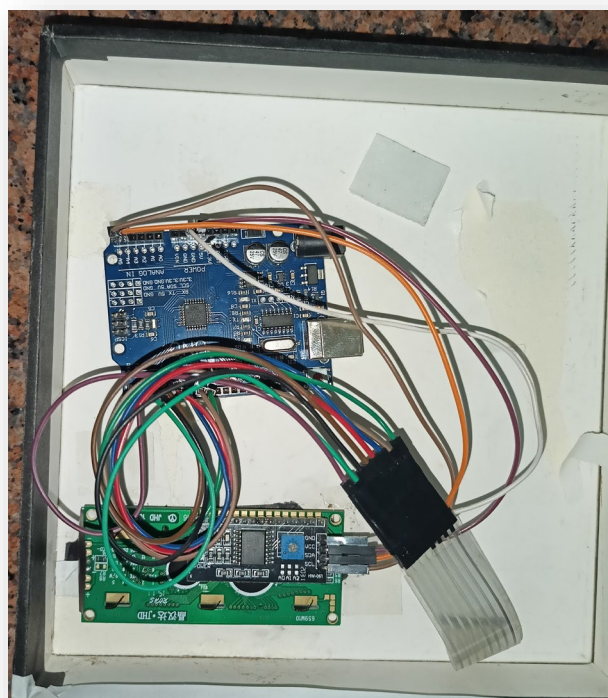
Whenever there is an over flow in the upper row the lcd will shift the words to left, so that viewer can view the next characters they are typing .

After whole calculation viewer can slide the words of upper row by typing 'shift'+'->' or 'shift'+ '<-' . And reset the calculator by typing 'shift'+ 'reset' .

Some pictures of the Final Circuit and results are given below:



Calculator Interface



Total Circuit

DISCUSSION: Considering the small memory I used functions and recursion relations and used same variable multiple times for multiple purposes.

Additions to this features like BODMAS RULE and some extra function we can also add some advanced features like Differentiation, Integration, etc...Thaat's just matter of some extra lines of code.

Possibilities are limitless....



Arduino Based Scientific Calculator
Submitted by SujayMondal
Submission date: 16th December, 2021

ADVANTAGES:

- You can solve very tedious and most difficult arithmetic problems.
- It is very portable and easy to use machine.
- Moreover, the biggest advantage is calculation is always correct.

DISADVANTAGES:

- 1. On the other hand, it is a machine. which can breakdown and make noise.
- You need to take basic training to operate this device.
- Some technical calculators are costly.

REFERENCES:

1. Other calculator projects on Github(<https://github.com>)
2. *Arduino official website* ([arduino.cc](https://www.arduino.cc))
3. *Tinkercad: Create online circuits* (<https://www.tinkercad.com>)
4. *Wokwi: Create online simulation* (<https://wokwi.com>)

