



Docs



Simulate IoT Projects in Your Browser

[Discord Community](#)

[Facebook Group](#)

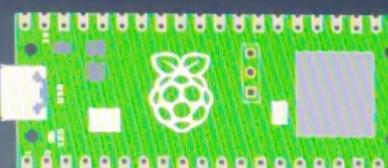
Simulate with Wokwi Online



Arduino



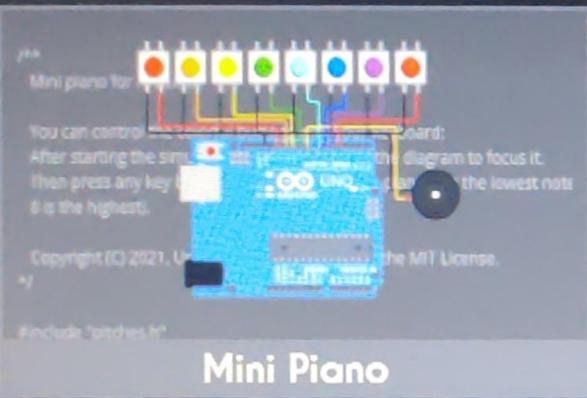
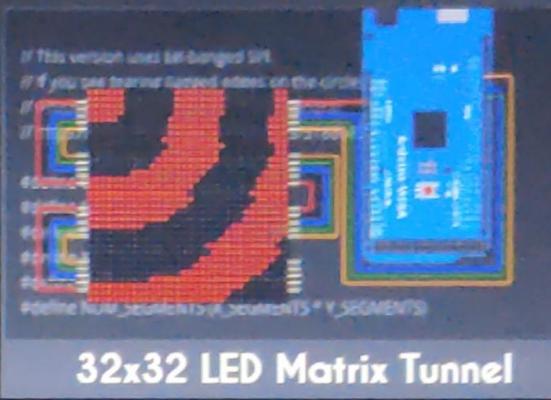
ESP32



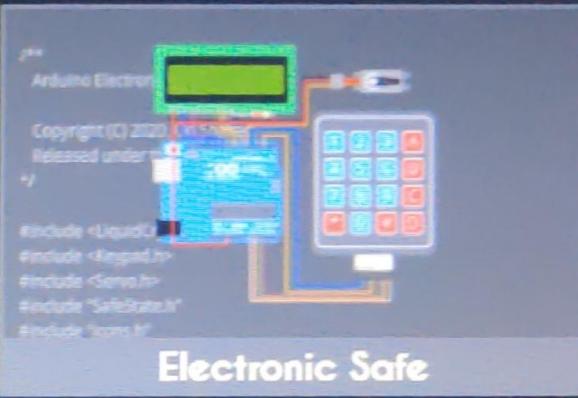
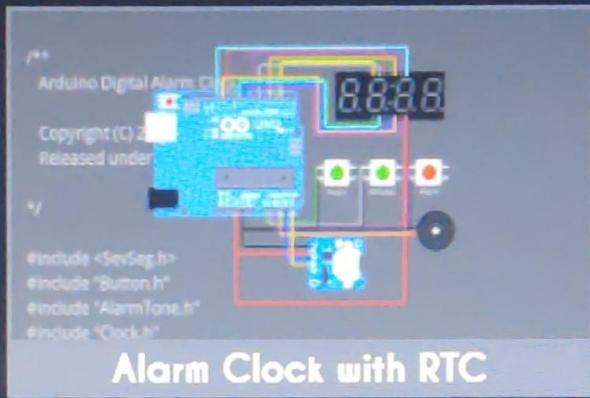
Type here to search



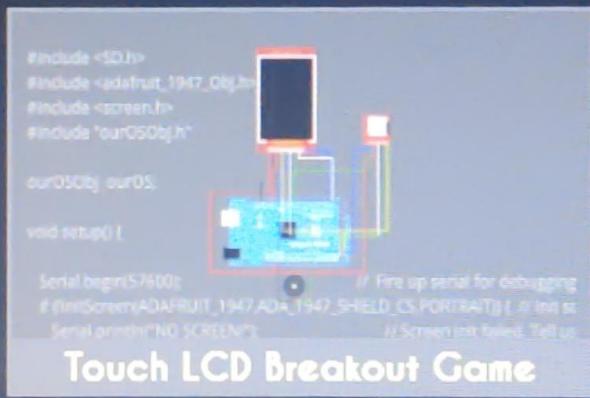
Simon Says Game

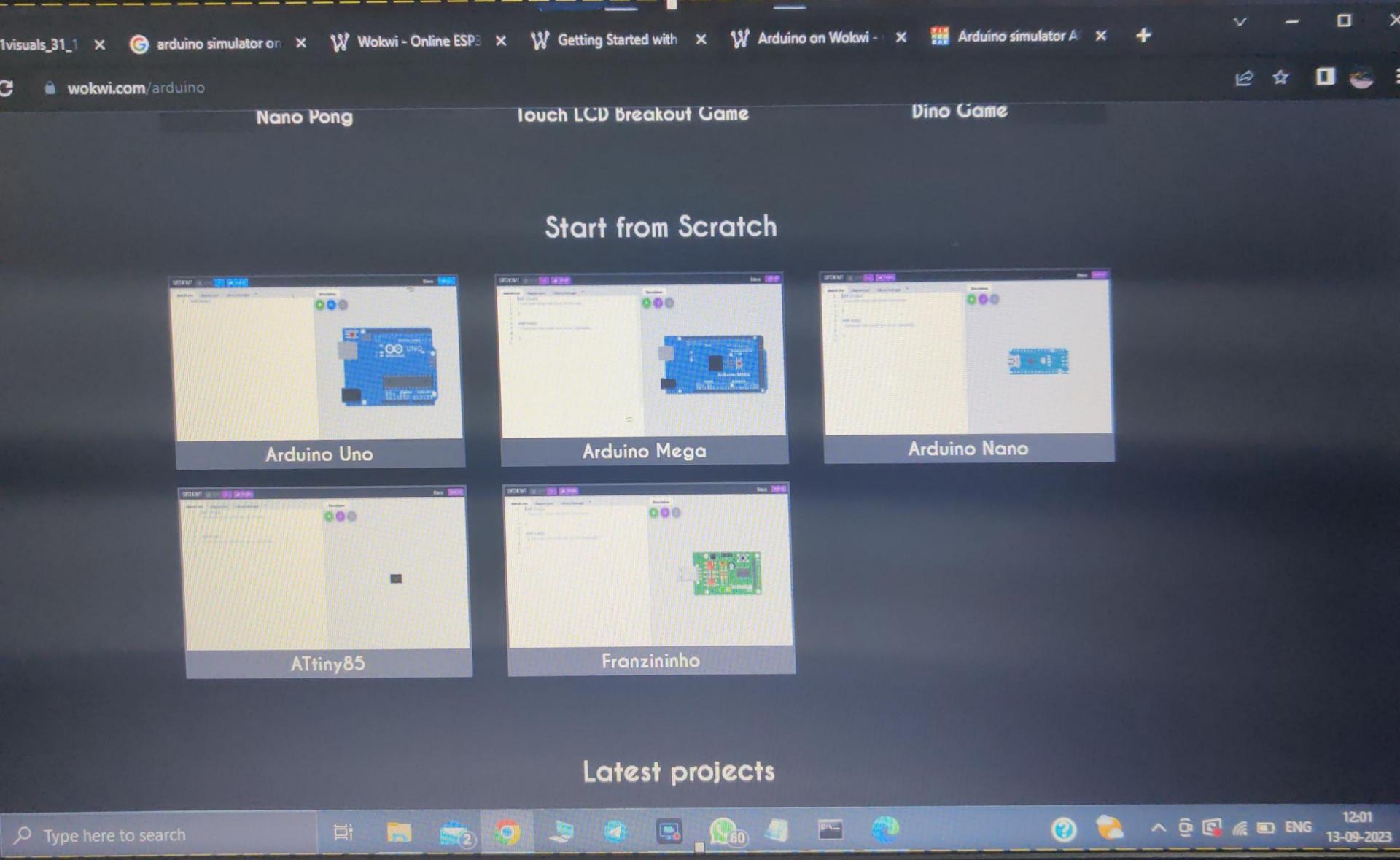
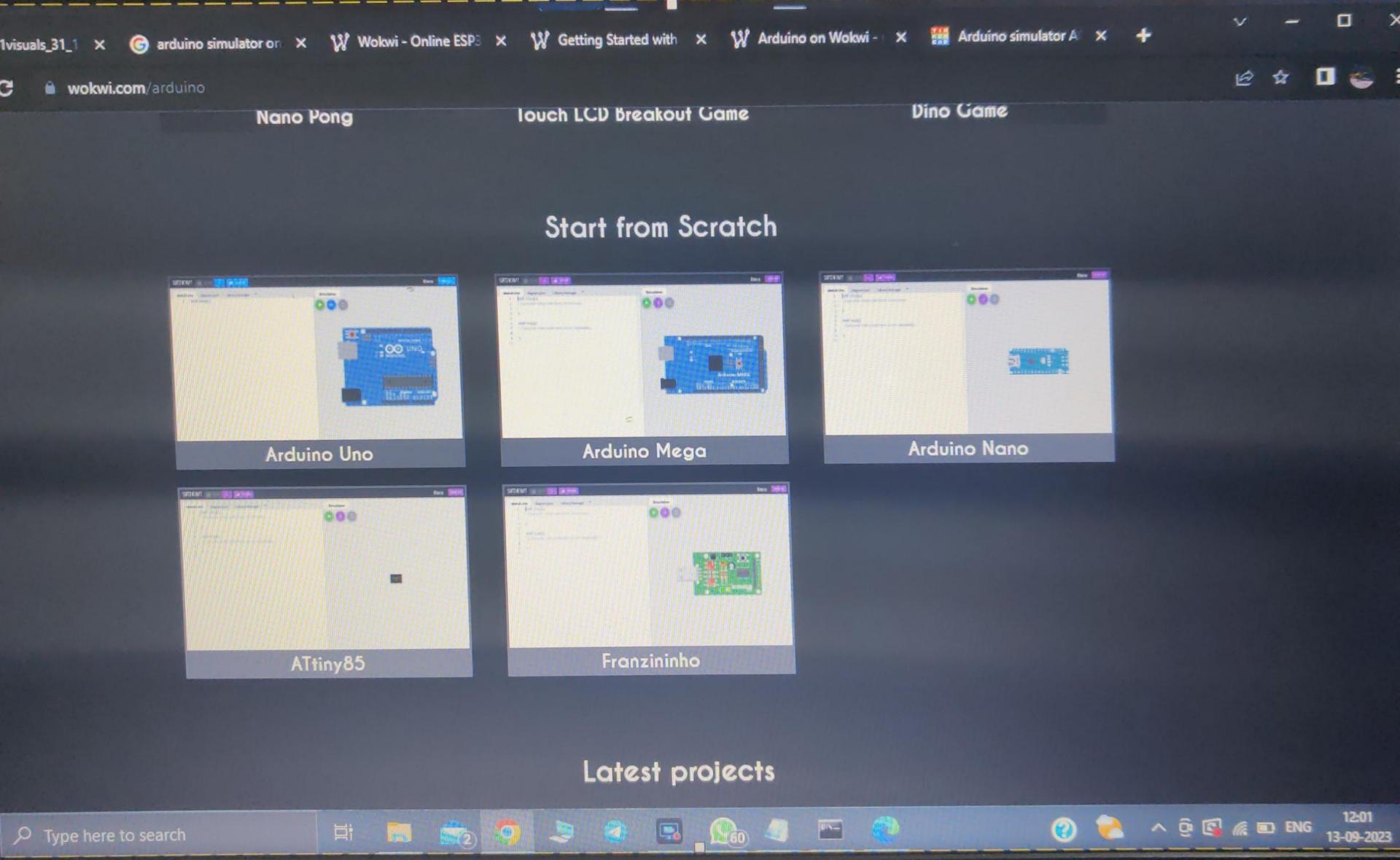
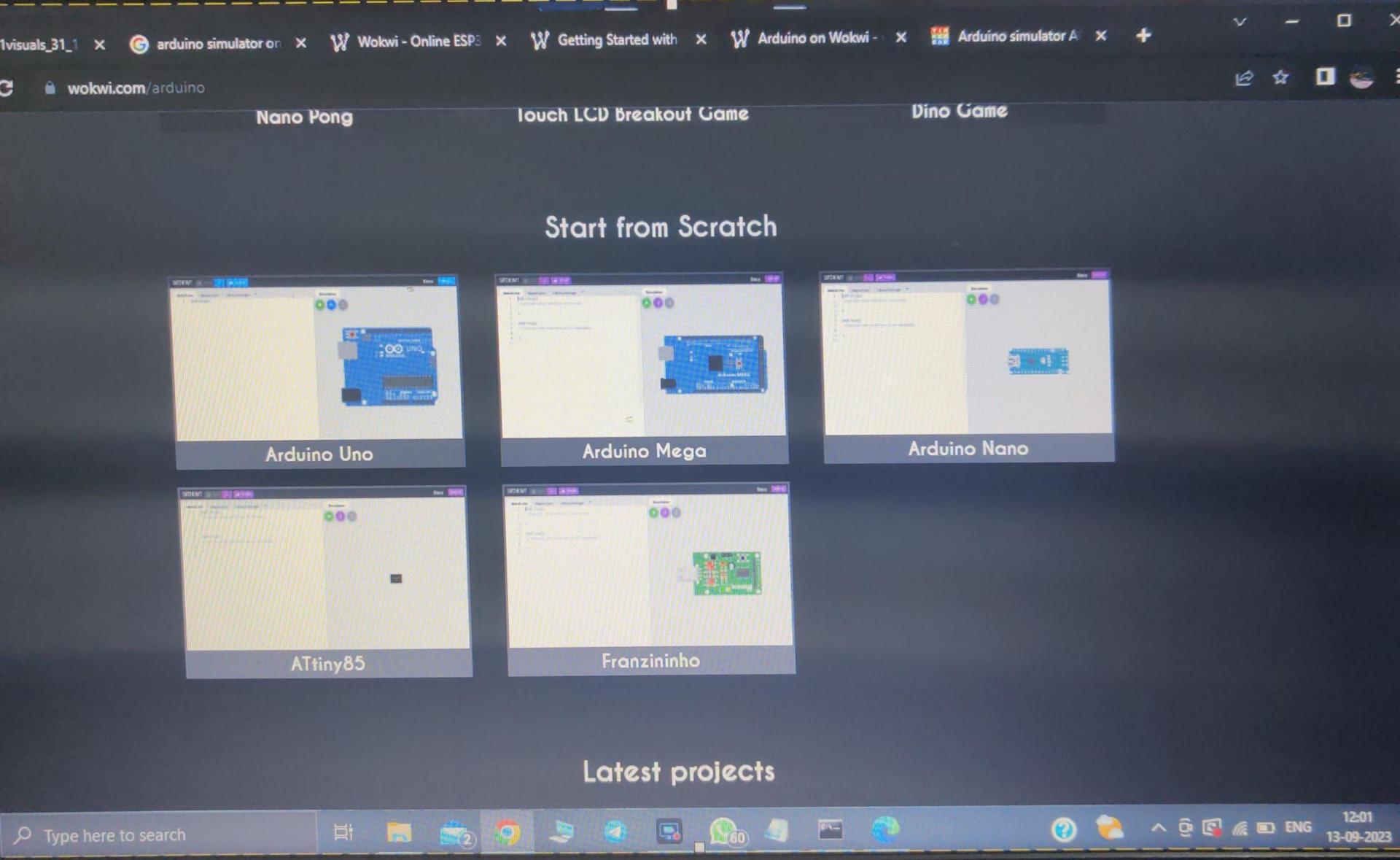
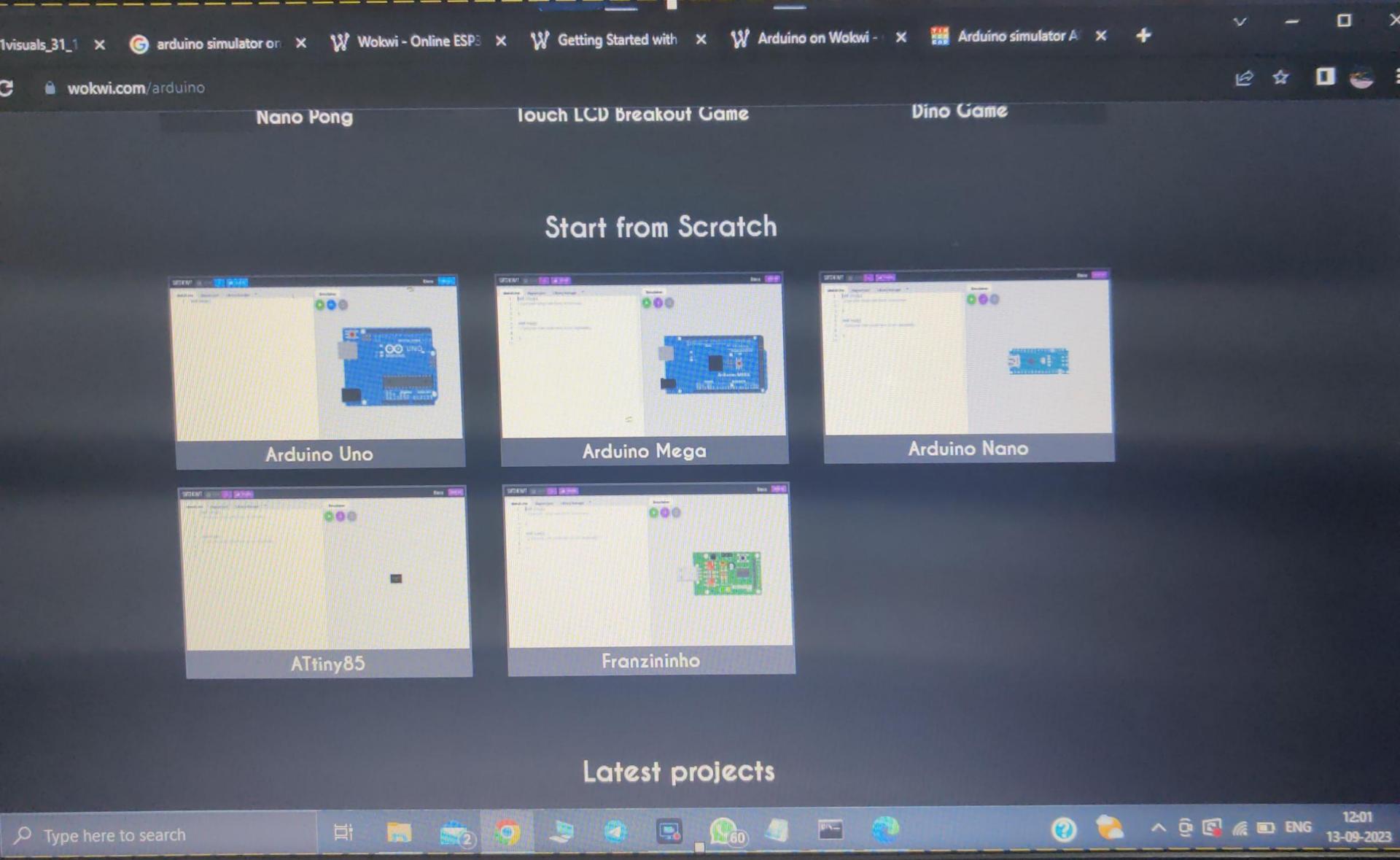
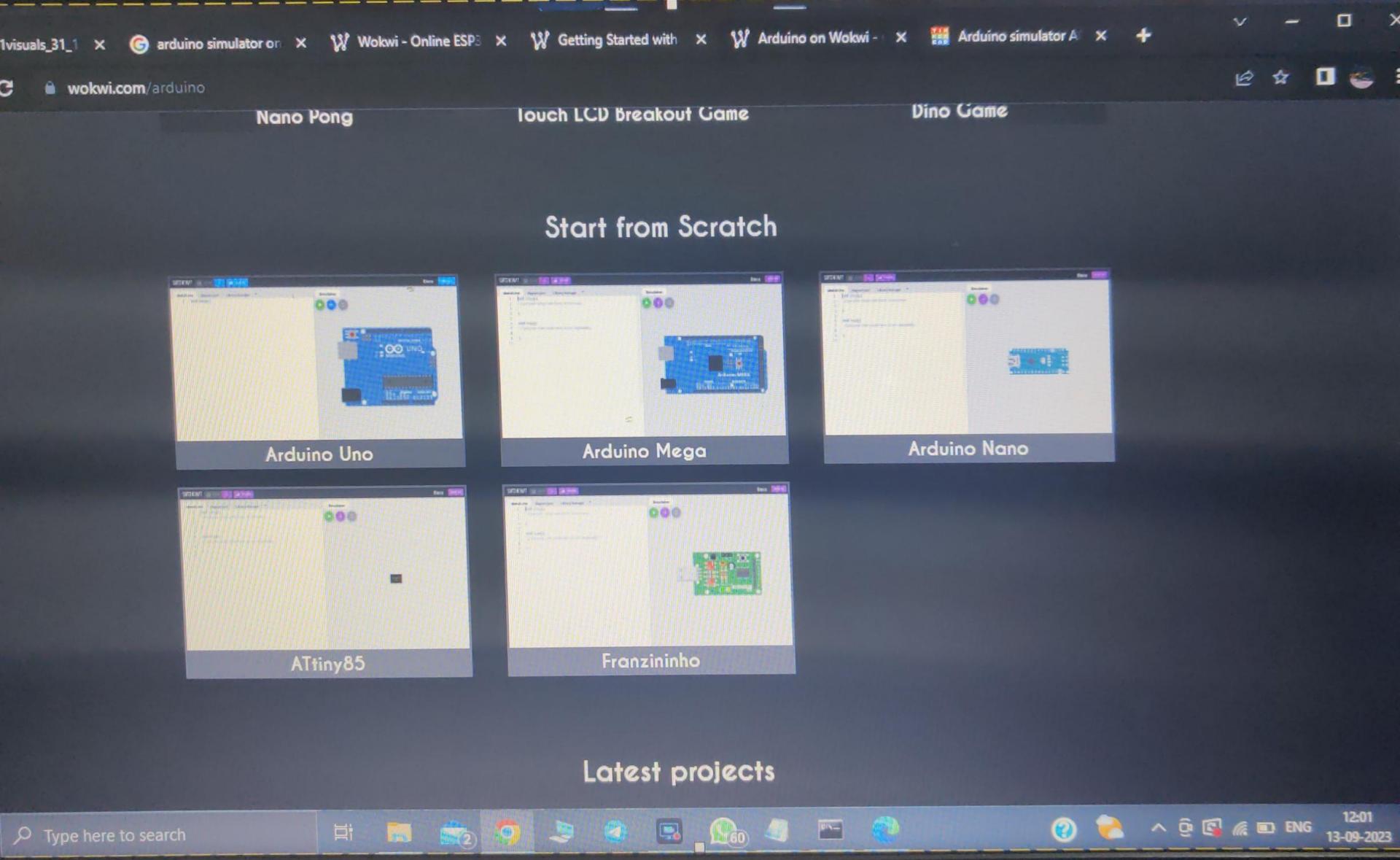


Mini Piano



Nano Pong





```
void setup() {  
    // put your setup code here, to run once:  
}  
  
void loop() {  
    // put your main code here  
}  
  
int temperatureHeater; //use Ra=90  
void setup()  
{  
    Serial.begin(9600);  
    // put your setup code here, to run once:  
}  
  
void loop()  
{  
    // put your main code here  
    int result = analogRead(A0);  
    float voltage = result / 1024.0 * 5.0;  
    float temp = (voltage - 0.5) * 100.0;  
    float resultTemp = temp / 1.8 + 32.0;  
    Serial.print("Result: ");  
    Serial.print(result);  
    Serial.print(" Voltage: ");  
    Serial.print(voltage);  
    Serial.print(" Temp: ");  
    Serial.print(temp);  
    Serial.print(" FTemp: ");  
    Serial.println(resultTemp);  
}  
  
int temperatureHeater; //use Ra=90  
void setup()  
{  
    // put your setup code here, to run once:  
}  
  
void loop()  
{  
    // put your main code here  
    int result = analogRead(A0);  
    float voltage = result / 1024.0 * 5.0;  
    float temp = (voltage - 0.5) * 100.0;  
    float resultTemp = temp / 1.8 + 32.0;  
    Serial.print("Result: ");  
    Serial.print(result);  
    Serial.print(" Voltage: ");  
    Serial.print(voltage);  
    Serial.print(" Temp: ");  
    Serial.print(temp);  
    Serial.print(" FTemp: ");  
    Serial.println(resultTemp);  
}
```

Weather Control Heater

```
void setup() {  
    // put your setup code here, to run once:  
}  
  
void loop() {  
    // put your main code here  
}  
  
const int ledCount = 10;  
int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11};  
int WaktuDelay = 100;  
  
void setup(){  
    for(int thisLed = 0; thisLed < ledCount; thisLed++)  
        pinMode(ledPins[thisLed], OUTPUT);  
}  
  
void loop(){  
    for(int thisLed = 0; thisLed < ledCount; thisLed++)  
        digitalWrite(ledPins[thisLed], HIGH);  
    delay(WaktuDelay);  
    for(int thisLed = 0; thisLed < ledCount; thisLed++)  
        digitalWrite(ledPins[thisLed], LOW);  
    delay(WaktuDelay);  
}
```

7_segment_Common_cathode

```
int a=0;  
void setup()  
{  
    Serial.begin(115200);  
    pinMode(4, INPUT);  
    pinMode(3, OUTPUT);  
}  
  
void loop()  
{  
    int a=analogRead(A0);  
    Serial.println(a);  
}
```

pir sensor

```
void setup() {  
    // put your setup code here, to run once:  
}  
  
void loop() {  
    // put your main code here  
}
```

Abitha V 3

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(2, OUTPUT);  
    pinMode(3, OUTPUT);  
    pinMode(4, OUTPUT);  
    pinMode(5, OUTPUT);  
    pinMode(6, OUTPUT);  
    pinMode(7, OUTPUT);  
    pinMode(8, OUTPUT);  
    pinMode(9, OUTPUT);  
    pinMode(10, OUTPUT);  
    pinMode(11, OUTPUT);  
}  
  
void loop() {  
    // put your main code here  
}
```

Ibnu7

```
int a=0;  
void setup()  
{  
    Serial.begin(9600);  
    pinMode(13, OUTPUT);  
}  
  
void loop()  
{  
    // put your main code here  
    int a=digitalRead(4);  
    Serial.print("EE");  
    Serial.println(a);  
}
```

9 LED GLOW

```
void setup() {  
    // put your setup code here, to run once:  
}  
  
void loop() {  
    // put your main code here  
}
```

johndy

```
const int ledCount = 10;  
int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11};  
int WaktuDelay = 100;  
  
void setup(){  
    for(int thisLed = 0; thisLed < ledCount; thisLed++)  
        pinMode(ledPins[thisLed], OUTPUT);  
}  
  
void loop(){  
    for(int thisLed = 0; thisLed < ledCount; thisLed++)  
        digitalWrite(ledPins[thisLed], HIGH);  
    delay(WaktuDelay);  
    for(int thisLed = 0; thisLed < ledCount; thisLed++)  
        digitalWrite(ledPins[thisLed], LOW);  
    delay(WaktuDelay);  
}
```

arsikom pratikum

```
int a=0;  
void setup()  
{  
    Serial.begin(9600);  
    pinMode(3, OUTPUT);  
}  
  
void loop()  
{  
    // put your main code here  
    int a=digitalRead(4);  
    Serial.print("EE");  
    Serial.println(a);  
}
```

PIR MOTION SENSOR



Sensor



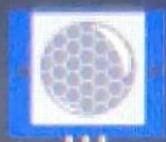
HC-SR04 Ultrasonic Distance Sensor



DHT22



DS18B20 Temperature Sensor (beta)



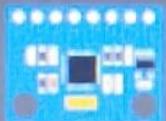
PIR Motion Sensor



Analog Temperature Sensor (NTC)



Photoresistor (LDR) Sensor



MPU6050 Accelerometer + Gyroscope



DS1307 RTC

Output



Input



Potentiometer



Slide Potentiometer



16 Channel Analog Multiplexer (CD74HC4067)

Sensor



HC-SR04 Ultrasonic Distance Sensor



DHT22



DS18B20 Temperature Sensor (beta)



PIR Motion Sensor



Analog Temperature Sensor (NTC)



sketch.ino diagram.json Library Manager

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }  
10
```

Simulation



Type here to search



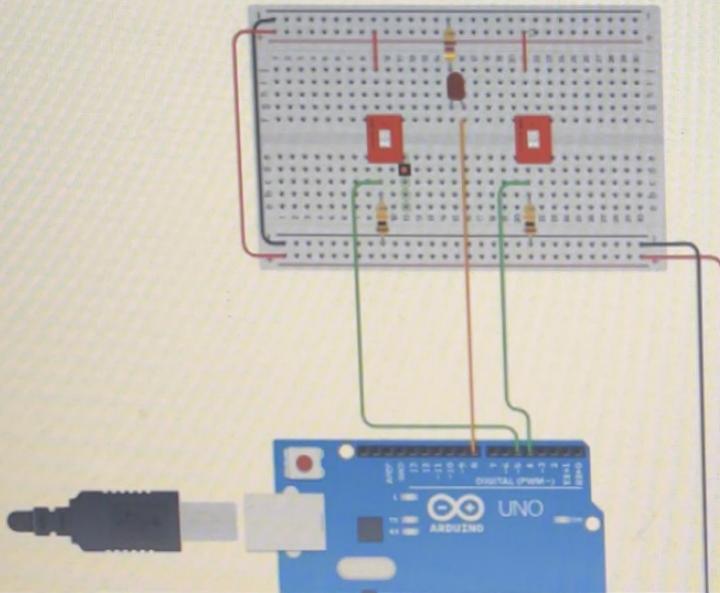
12:04

13-09-2023

Public

Arduino simulator AND

Simulate

 Remixed Circuit by
SoftLogicSoftLogLogic simulation by
Arduino. AND.

プログラムで学ぶロジック回路

show more

Remixed 104,608 times

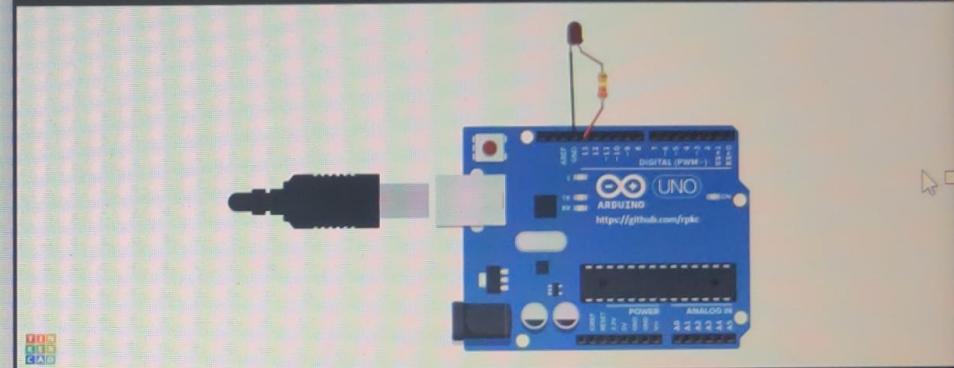
Sign up to copy

```
// the setup function runs once when you press reset or
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

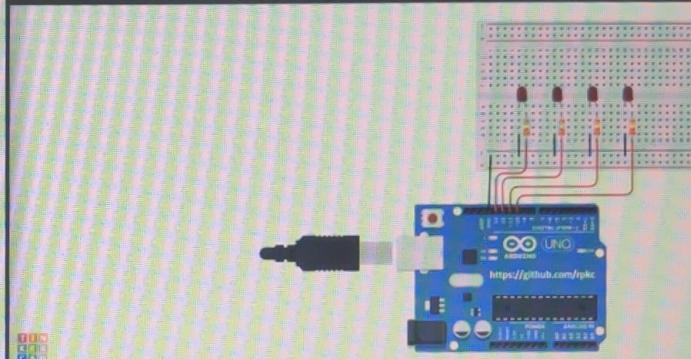
// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on
    delay(2000);                      // wait for a second
    digitalWrite(LED_BUILTIN, LOW);       // turn the LED off
    delay(2000);                      // wait for a second
}
```

Download our new video editor X
Create videos in minutes using customizable templates.

Install Microsoft Clipchamp



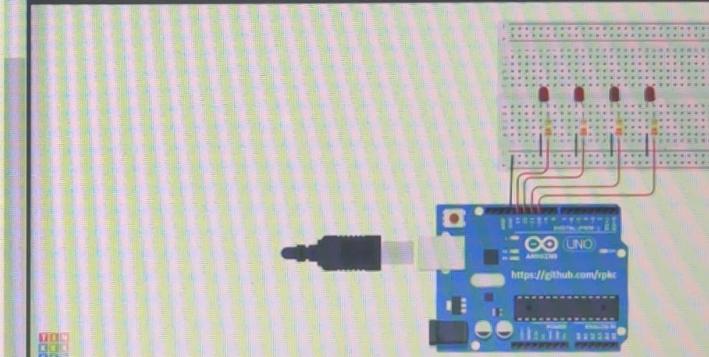
```
void setup()
{
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
}
void loop()
{
    digitalWrite(10, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(10, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(11, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(11, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(12, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(12, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(13, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(13, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
}
```



```
void setup()
{
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
}
void loop()
{
    digitalWrite(10, HIGH);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    digitalWrite(13, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    digitalWrite(13, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
    digitalWrite(13, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
```

```
digitalWrite(13, LOW);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(12, HIGH);
digitalWrite(13, LOW);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(12, LOW);
digitalWrite(13, HIGH);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(10, HIGH);
digitalWrite(11, HIGH);
digitalWrite(12, LOW);
digitalWrite(13, LOW);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(10, LOW);
digitalWrite(11, HIGH);
digitalWrite(12, HIGH);
digitalWrite(13, LOW);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(10, LOW);
digitalWrite(11, LOW);
digitalWrite(12, HIGH);
digitalWrite(13, HIGH);
delay(1000); // Wait for 1000 millisecond(s)
```

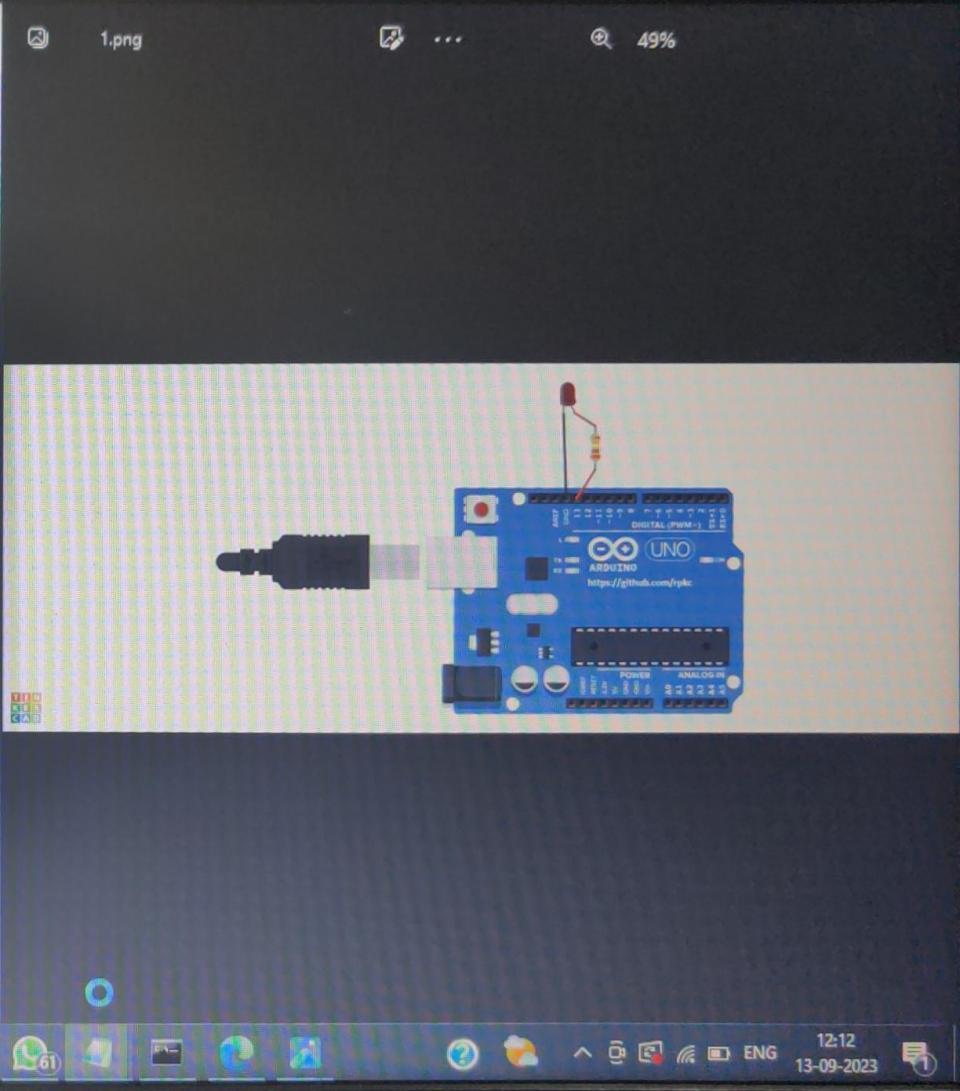
}



```
// the setup function runs once when you press reset or
int buzzerPin=13;

void setup() {
  // initialize digital pin buzzerPin as an output.
  pinMode(buzzerPin, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(buzzerPin, HIGH);    // turn the Buzzer on
  delay(1000);                    // wait for a second
  digitalWrite(buzzerPin, LOW);     // turn the Buzzer off
  delay(1000);                    // wait for a second
}
```

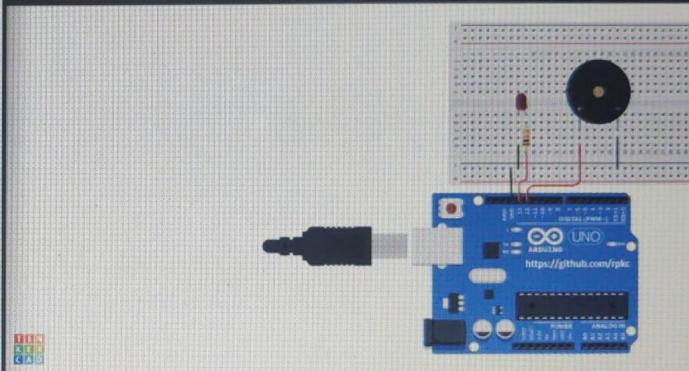


Type here to search



```
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  digitalWrite(12, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  digitalWrite(12, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```



*6 - Notepad

File Edit Format View Help

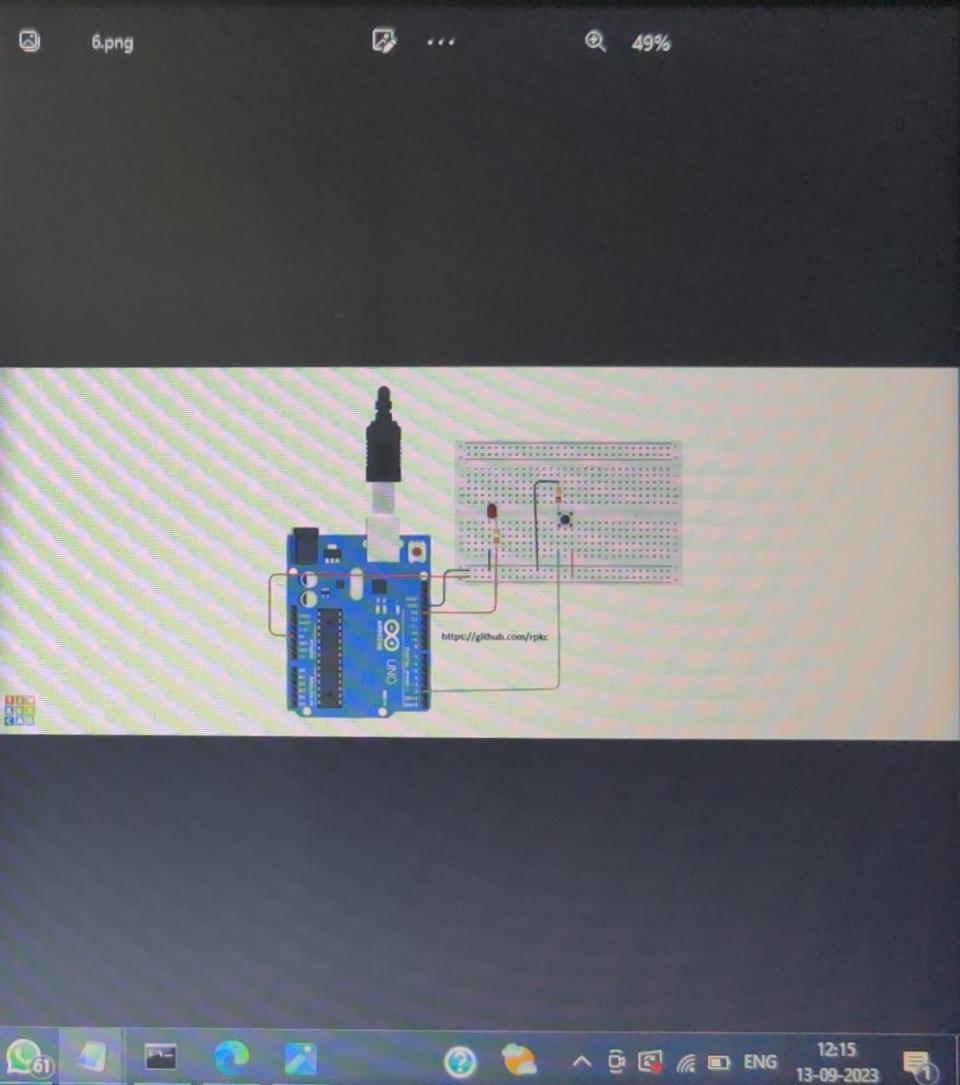
```
// constants won't change. They're used here to set pins
const int buttonPin = 2; // the number of the pushbutton pin
const int ledPin = 13; // the number of the LED pin

// variables will change:
int buttonState = 0; // variable for reading the pushbutton state
void setup() {
    // initialize the LED pin as an output:
    pinMode(ledPin, OUTPUT);
    // initialize the pushbutton pin as an input:
    pinMode(buttonPin, INPUT);
}

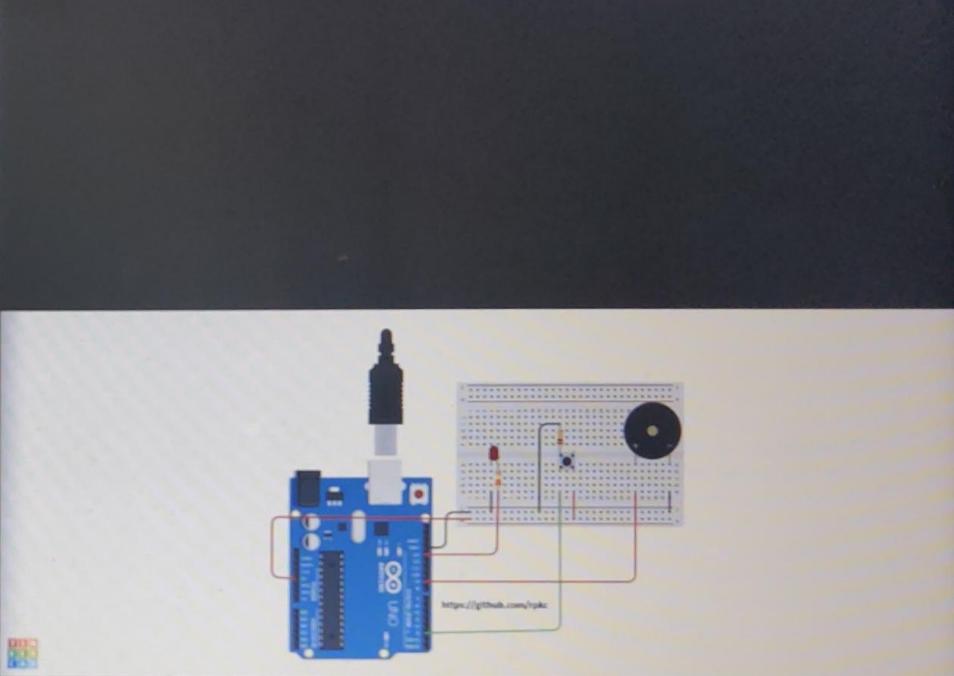
void loop() {
    // read the state of the pushbutton value:
    buttonState = digitalRead(buttonPin);

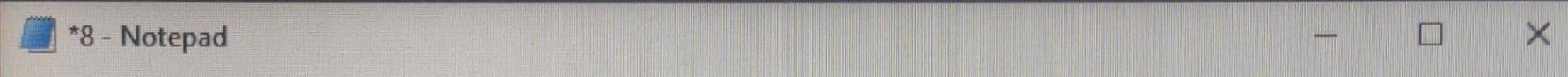
    // check if the pushbutton is pressed. If it is, then turn LED on:
    if (buttonState == HIGH) {
        // turn LED on:
        digitalWrite(ledPin, HIGH);
    } else {
        // turn LED off:
        digitalWrite(ledPin, LOW);
    }
}
```

Ln 10, Col 7 100% Unix (LF) UTF-8



```
const int buttonPin = 2; // the number of the pushbutton
const int ledPin = 13; // the number of the LED pin
const int buzzerPin = 9; // the number of the Buzzer pin
bool ledStatus = true;
// variables will change:
int buttonState = 0; // variable for reading the pushbutton state
void setup() {
    // initialize the LED pin as an output:
    pinMode(ledPin, OUTPUT);
    // initialize the Buzzer pin as an output:
    pinMode(buzzerPin, OUTPUT);
    // initialize the pushbutton pin as an input:
    pinMode(buttonPin, INPUT);
}
void loop() {
    // read the state of the pushbutton value:
    buttonState = digitalRead(buttonPin);
    // check if the pushbutton is pressed. If it is, then:
    if (buttonState == HIGH) {
        // turn LED on:
        digitalWrite(ledPin, ledStatus);
        digitalWrite(buzzerPin, HIGH);
        delay(100);
        digitalWrite(buzzerPin, LOW);
        ledStatus = !ledStatus; // Toggle current LED status
    }
}
```





File Edit Format View Help

```
const int led_1=10;
const int led_2=11;
const int led_3=12;
const int led_4=13;
const int incr_btn=7;
const int decr_btn=8;
int led_counter=0;
void setup()
{
    pinMode(led_1,OUTPUT);
    pinMode(led_2,OUTPUT);
    pinMode(led_3,OUTPUT);
    pinMode(led_4,OUTPUT);
    pinMode(incr_btn,INPUT);
    pinMode(decr_btn,INPUT);
    //Serial.begin(9600);
}
void loop()
{
    if(digitalRead(incr_btn)) // increment counter when F
    {
        led_counter++;
        //Serial.println(led_counter);
    }
    if(digitalRead(decr_btn)) // decrement counter when F
    {
        led_counter--;
    }
}
```



File Edit Format View Help

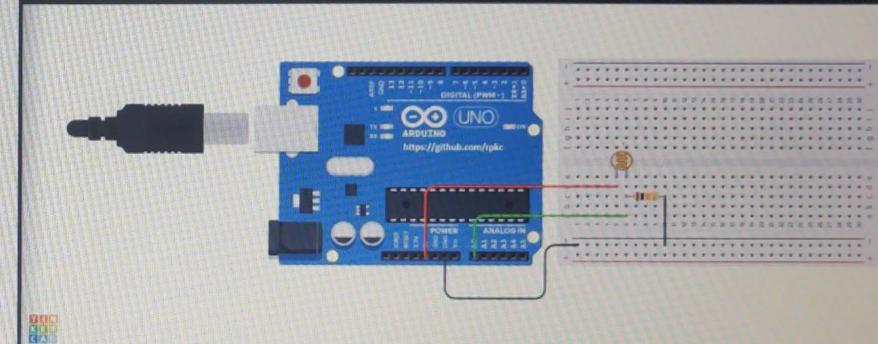
```
if(digitalRead(decr_btn)) // decrement counter when F ^  
{  
    led_counter--;  
    //Serial.println(led_counter);  
}  
  
switch(led_counter)  
{  
  
    case 0:  
        digitalWrite(led_1,LOW);  
        digitalWrite(led_2,LOW);  
        digitalWrite(led_3,LOW);  
        digitalWrite(led_4,LOW);  
        break;  
  
    case 1:  
        digitalWrite(led_1,HIGH);  
        digitalWrite(led_2,LOW);  
        digitalWrite(led_3,LOW);  
        digitalWrite(led_4,LOW);  
        break;  
  
    case 2:  
        digitalWrite(led_1,HIGH);  
        digitalWrite(led_2,HIGH);  
        digitalWrite(led_3,LOW);  
        digitalWrite(led_4,LOW);  
        break;  
}
```

```
digitalWrite(led_1,HIGH);
digitalWrite(led_2,HIGH);
digitalWrite(led_3,LOW);
digitalWrite(led_4,LOW);
break;
case 3: [REDACTED]
digitalWrite(led_1,HIGH);
digitalWrite(led_2,HIGH);
digitalWrite(led_3,HIGH);
digitalWrite(led_4,LOW);
break;
case 4: [REDACTED]
digitalWrite(led_1,HIGH);
digitalWrite(led_2,HIGH);
digitalWrite(led_3,HIGH);
digitalWrite(led_4,HIGH);
break;
default: [REDACTED]
digitalWrite(led_1,LOW);
digitalWrite(led_2,HIGH);
digitalWrite(led_3,HIGH);
digitalWrite(led_4,LOW);
break;
}
delay(100);
```

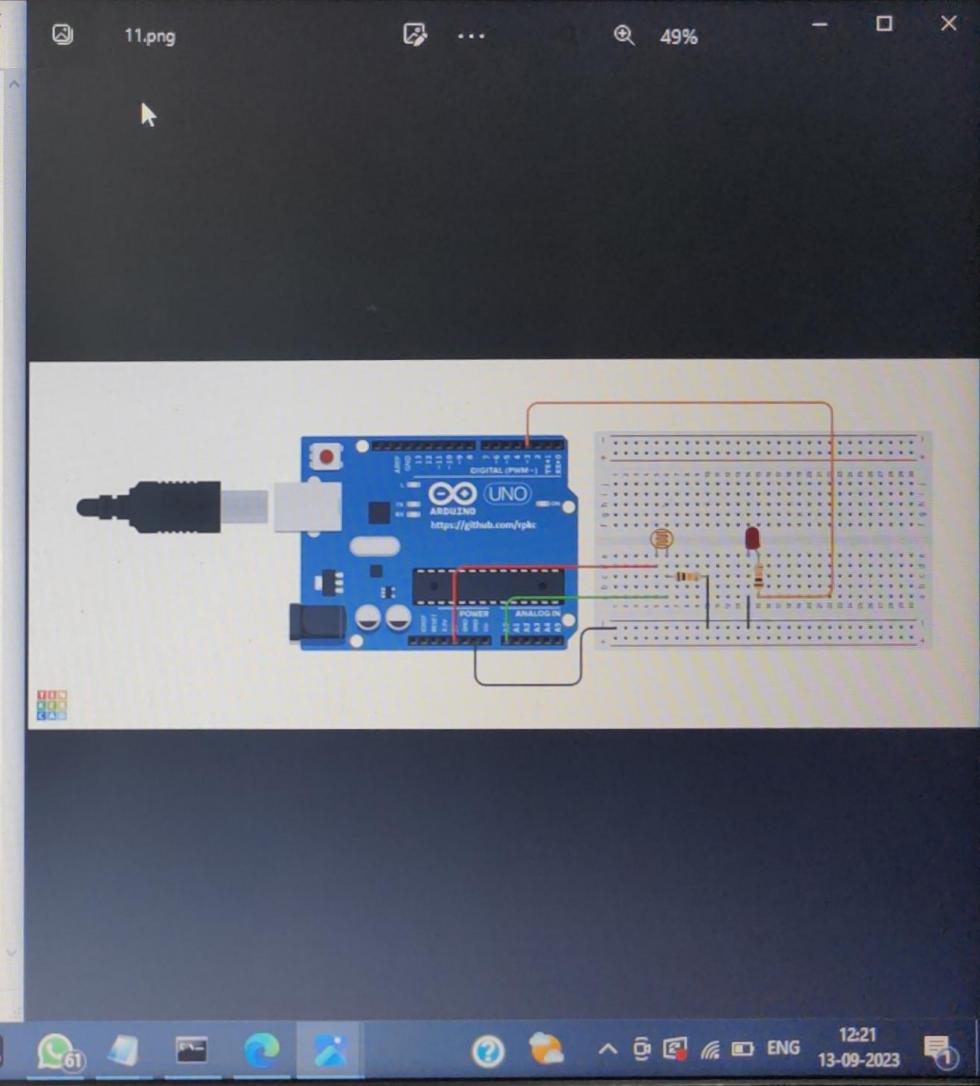
}

```
const int ldrPin = A0; // analog pin 0
void setup() {
Serial.begin(9600); // start Serial at 9600
pinMode(ldrPin, INPUT); // Here LDR sensor is determined
}

void loop() { // Void loop is ran again and again and ...
int ldrStatus = analogRead(ldrPin);
Serial.println(ldrStatus);
delay(2000);
}
```



```
// Interfacing Arduino uno with LDR sensor
const int ledPin = 3; // digital pin 3
const int ldrPin = A0; // analog pin 0
void setup() { // The setup() function will only run once
Serial.begin(9600);
pinMode(ledPin, OUTPUT); // Here LED is determined as an output
pinMode(ldrPin, INPUT); // Here LDR sensor is determined as an input
}
void loop() { // Void loop is ran again and again and checks
int ldrStatus = analogRead(ldrPin);
if (ldrStatus <= 200) {digitalWrite(ledPin, HIGH); // If LDR senses darkness
Serial.print("Darkness over here, turn on the LED :");
Serial.println(ldrStatus);
} else {
digitalWrite(ledPin, LOW); // If LDR senses light led goes off
Serial.print("There is sufficient light , turn off the LED :");
Serial.println(ldrStatus);
}
}
```



12 - Notepad

File Edit Format View Help

```
const int ledPin_1 = 3; // LED connected to digital pin 3
const int ledPin_2 = 5;

void setup() {
    pinMode(ledPin_1,OUTPUT);
    pinMode(ledPin_2,OUTPUT);

}

void loop() {
    // fade in from min to max in increments of 5 points:
    for (int fadeValue = 0; fadeValue <= 255; fadeValue += 5) {
        // sets the value (range from 0 to 255):
        analogWrite(ledPin_1, fadeValue);
        // wait for 30 milliseconds to see the dimming effect
        delay(30);
    }

    // fade out from max to min in increments of 5 points:
    for (int fadeValue = 255; fadeValue >= 0; fadeValue -= 5) {
        // sets the value (range from 0 to 255):
        analogWrite(ledPin_1, fadeValue);
        // wait for 30 milliseconds to see the dimming effect
        delay(30);
    }
}
```

```
}
```

```
// fade out from max to min in increments of 5 points
for (int fadeValue = 255; fadeValue >= 0; fadeValue -
    // sets the value (range from 0 to 255):
    analogWrite(ledPin_1, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
```

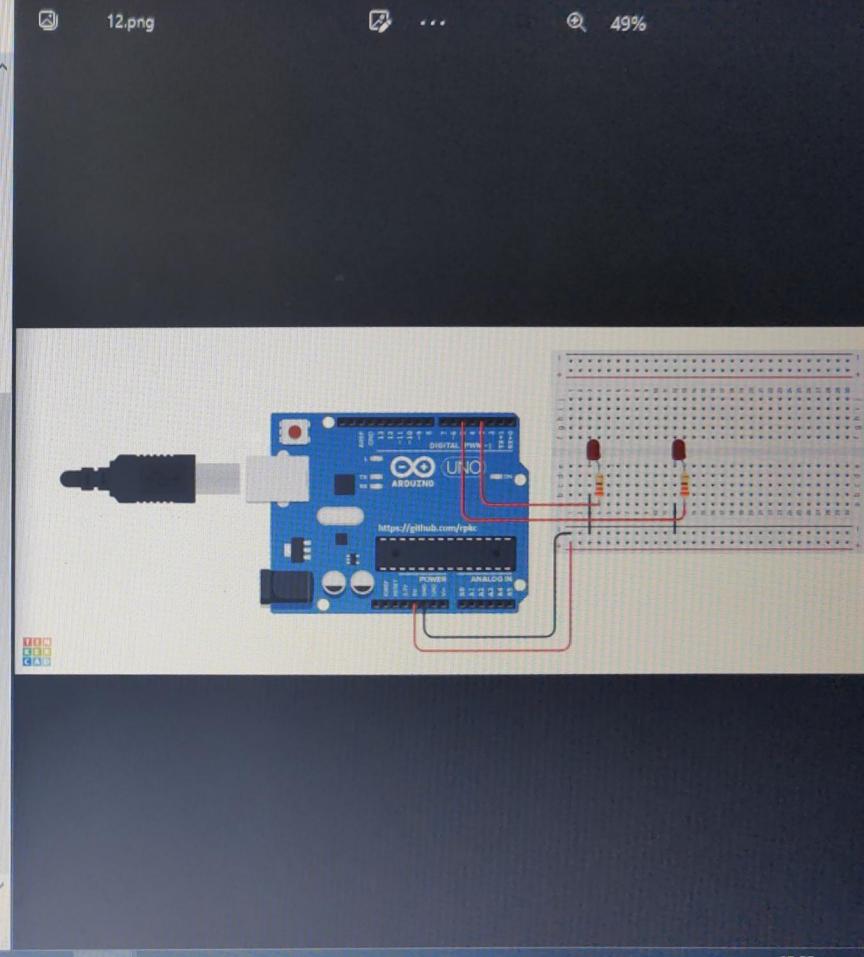
```
}
```

```
for (int fadeValue = 0; fadeValue <= 255; fadeValue +
    // sets the value (range from 0 to 255):
    analogWrite(ledPin_2, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
```

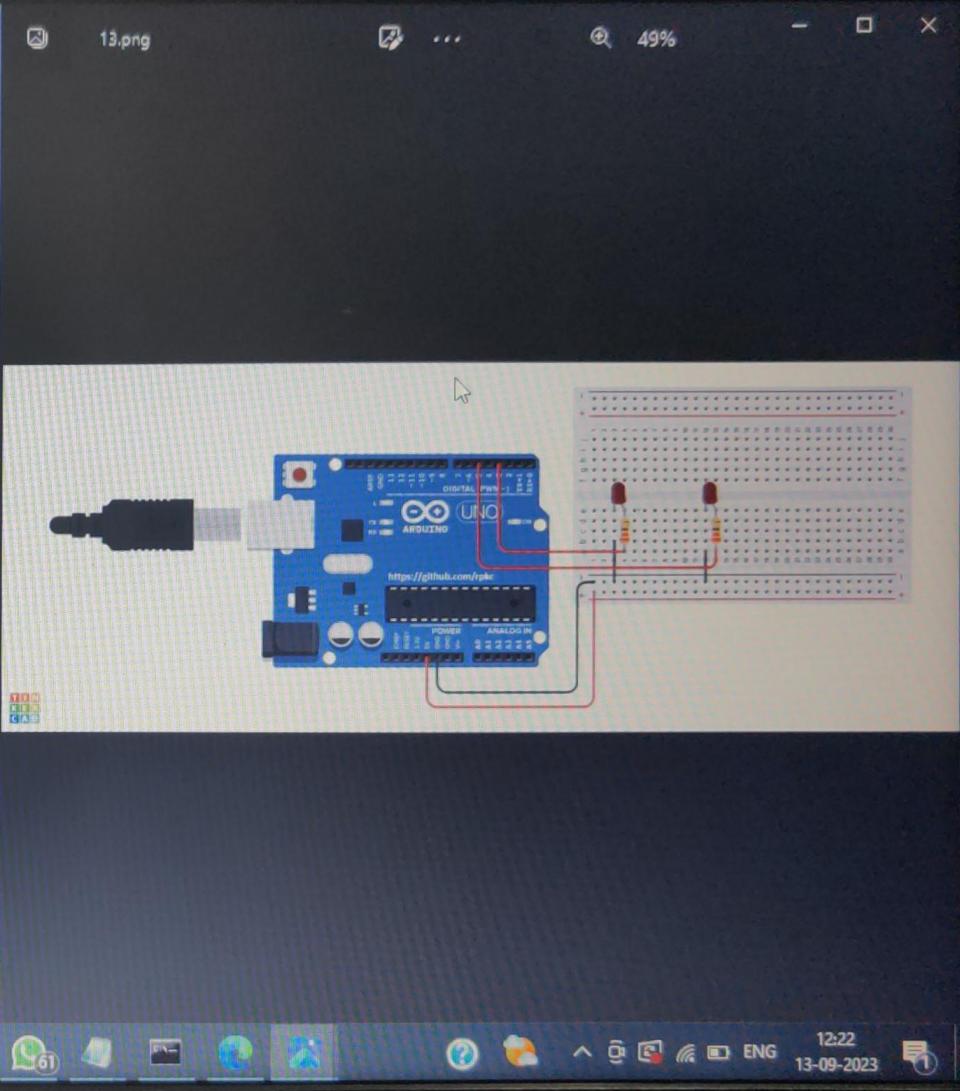
```
}
```

```
// fade out from max to min in increments of 5 points
for (int fadeValue = 255; fadeValue >= 0; fadeValue -
    // sets the value (range from 0 to 255):
    analogWrite(ledPin_2, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
```

```
}
```



```
// Interfacing Arduino uno with LDR sensor
const int ledPin = 3; // digital pin 3
const int ldrPin = A0; // analog pin 0
void setup() { // The setup() function will only run once
Serial.begin(9600);
pinMode(ledPin, OUTPUT); // Here LED is determined as an output
pinMode(ldrPin, INPUT); // Here LDR sensor is determined as an input
}
void loop() { // Void loop is ran again and again and checks
int ldrStatus = analogRead(ldrPin);
if (ldrStatus <= 800) {analogWrite(ledPin, 1024-ldrStatus);
Serial.println(ldrStatus);
} else {
analogWrite(ledPin, 0);
Serial.println(ldrStatus);
}
}
```

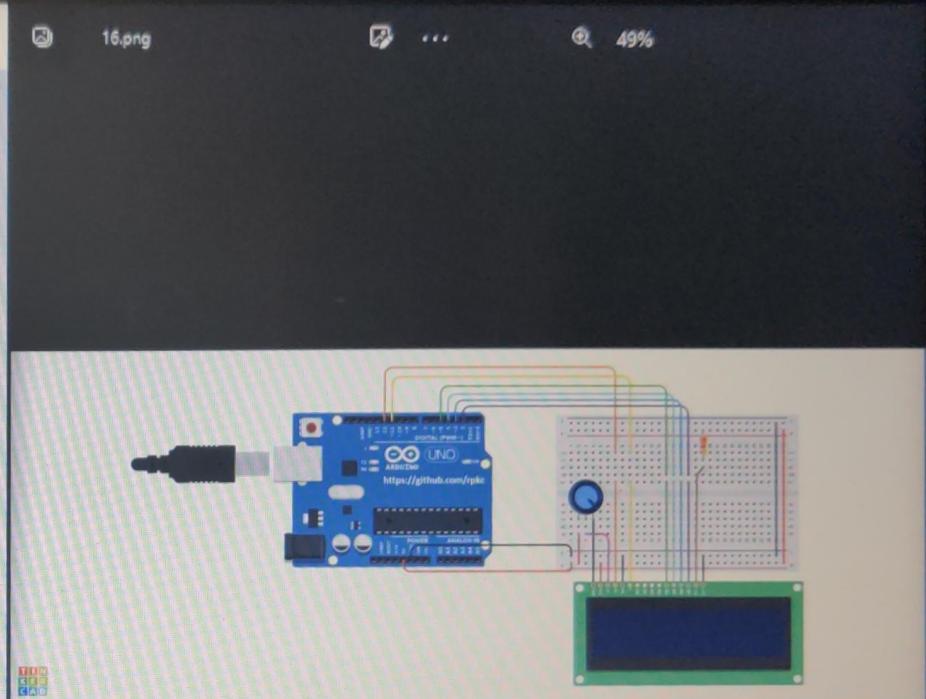


```
#include <LiquidCrystal.h>

LiquidCrystal lcd_1(12, 11, 5, 4, 3, 2);

void setup()
{
    lcd_1.begin(16, 2); // Set up the number of columns &
    // Print a message to the LCD.
    lcd_1.print("hello world!");
}

void loop()
{}
```



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

```
const byte numRows= 4; //number of rows on the keypad
const byte numCols= 4; //number of columns on the keypad
```

```
//keymap defines the key pressed according to the row and columns just as appears on the keypad
char keymap[numRows][numCols]=
```

```
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};
```

```
//Code that shows the keypad connections to the arduino terminals
```

```
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
```

```
byte colPins[numCols]= {5,4,3,2}; //Columns 0 to 3
```

```
//initializes an instance of the Keypad class
```

```
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);
```

```
void setup()
```

```
{
Serial.begin(9600);
}
```

```
//If key is pressed, this key is stored in 'keypressed' variable
```



Type here to search



};

```
//Code that shows the keypad connections to the arduino
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols]= {5,4,3,2}; //Columns 0 to 3

//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins);

void setup()
{
Serial.begin(9600);
}

//If key is pressed, this key is stored in 'keypressed'
//If key is not equal to 'NO_KEY', then this key is printed
//if count=17, then count is reset back to 0 (this means we have 17 keys)
void loop()
{
char keypressed = myKeypad.getKey();
if (keypressed != NO_KEY)
{
Serial.println(keypressed);
}
}
```

Ln 37, Col 2

100% Unix (LF)

UTF-8

