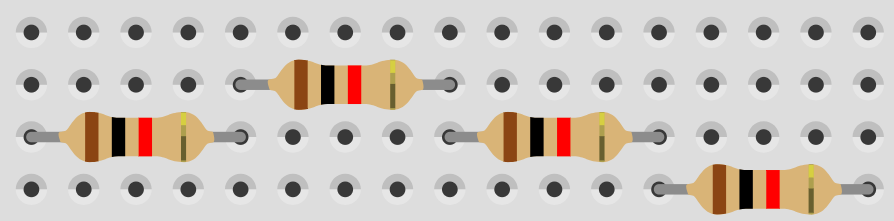
12/01/2024

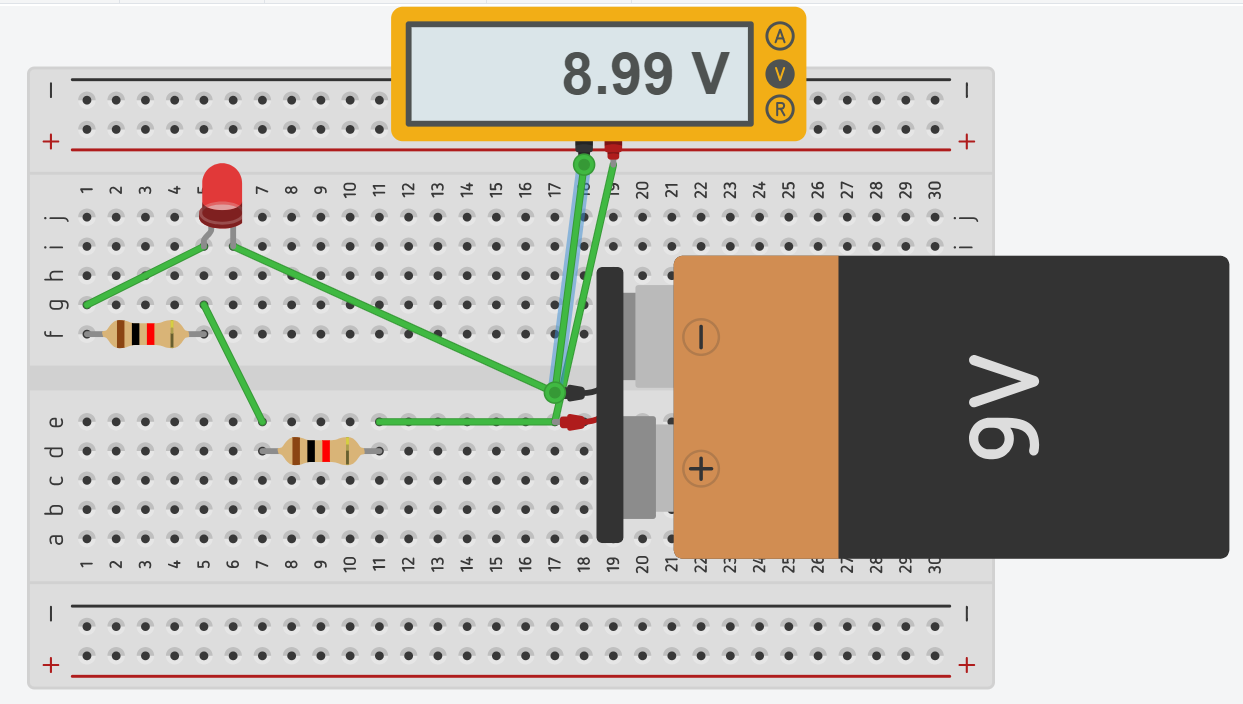
Resistors in Series:



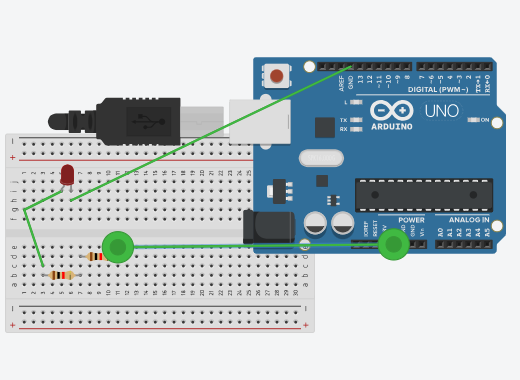
In a series ckt, check its voltage and current across every resistor

In a paralell ckt, check its voltage and current across every resistor

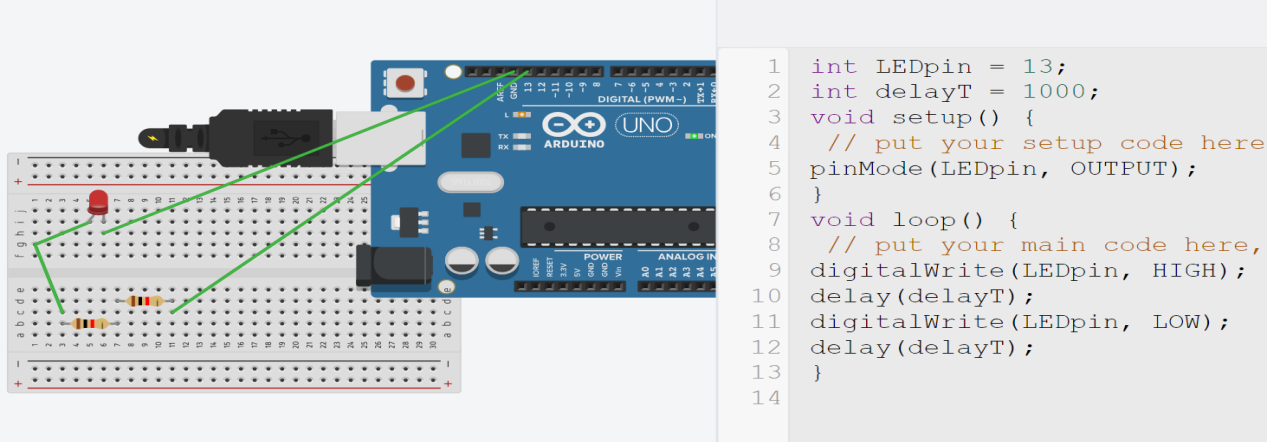
Check voltage using multimeter:



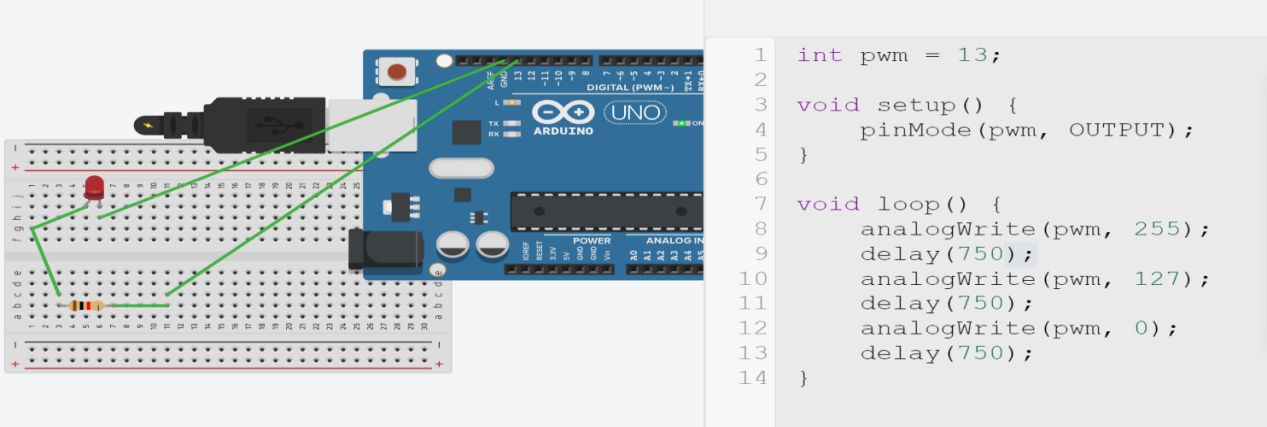
Using arduino instead:



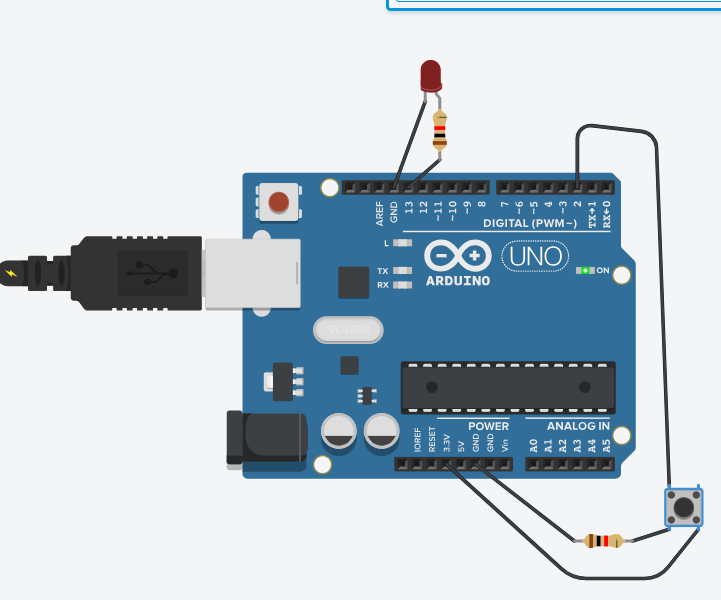
Using code:



Pwm:



USE BTN:



const int buttonPin = 2; // Pin for the button

const int ledPin = 13; // Pin for the LED

void setup() {

pinMode(buttonPin, INPUT\_PULLUP); // Enable internal pull-up resistor for the button

pinMode(ledPin, OUTPUT);

}

void loop() {

// Read the state of the button

int buttonState = digitalRead(buttonPin);

// Check if the button is pressed (buttonState is LOW)

if (buttonState == LOW) {

digitalWrite(ledPin, LOW); // Turn on the LED

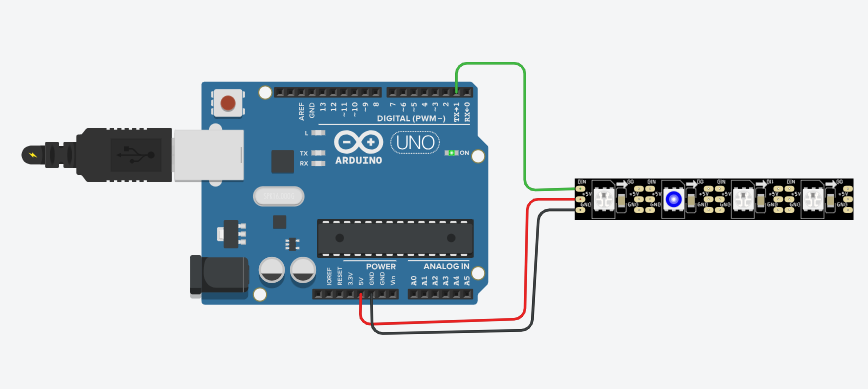
} else {

digitalWrite(ledPin, HIGH); // Turn off the LED if the button is not pressed

}

}

Neopixel



#include <Adafruit\_NeoPixel.h>

#define PIN 1 // Pin where the NeoPixel data line is connected

#define NUMPIXELS 4 // Number of NeoPixels in the strip

Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(NUMPIXELS, PIN, NEO\_GRB + NEO\_KHZ800);

void setup() {

strip.begin(); // Initialize the NeoPixel strip

strip.show(); // Initialize all pixels to 'off'

}

void loop() {

// Set the color of the first NeoPixel to red

strip.setPixelColor(0, strip.Color(255, 0, 0));

strip.show();

delay(1000); // Wait for 1000 milliseconds (1 second)

// Turn off the first NeoPixel

strip.setPixelColor(0, strip.Color(0, 0, 0));

strip.show();

delay(500); // Wait for 500 milliseconds (0.5 second)

// Set the color of the second NeoPixel to blue

strip.setPixelColor(1, strip.Color(0, 0, 255));

strip.show();

delay(1000); // Wait for 1000 milliseconds (1 second)

// Turn off the second NeoPixel

strip.setPixelColor(1, strip.Color(0, 0, 0));

strip.show();

delay(500); // Wait for 500 milliseconds (0.5 second)

// Set the color of the third NeoPixel to green

strip.setPixelColor(2, strip.Color(0, 255, 0));

strip.show();

delay(1000); // Wait for 1000 milliseconds (1 second)

// Turn off the third NeoPixel

strip.setPixelColor(2, strip.Color(0, 0, 0));

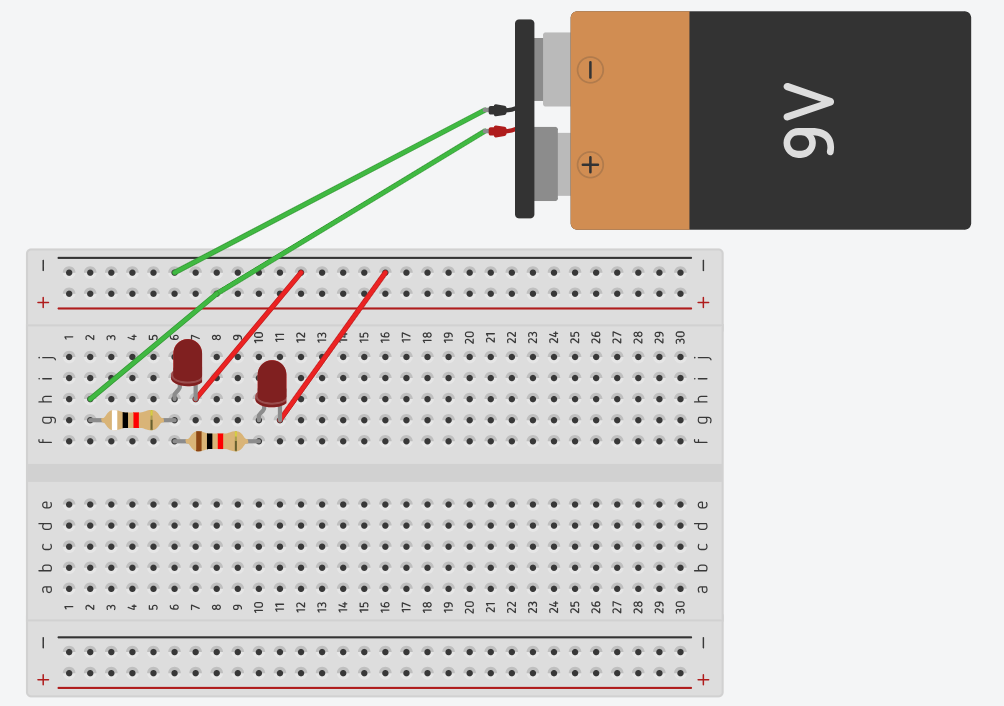
strip.show();

delay(500); // Wait for 500 milliseconds (0.5 second)

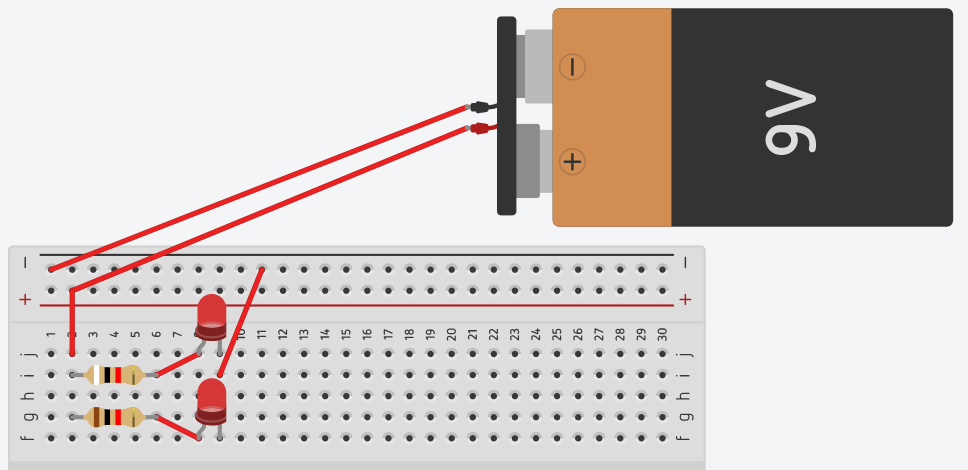
}

13/01/2024

Led’s in series:

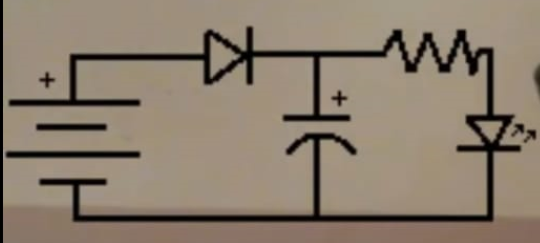


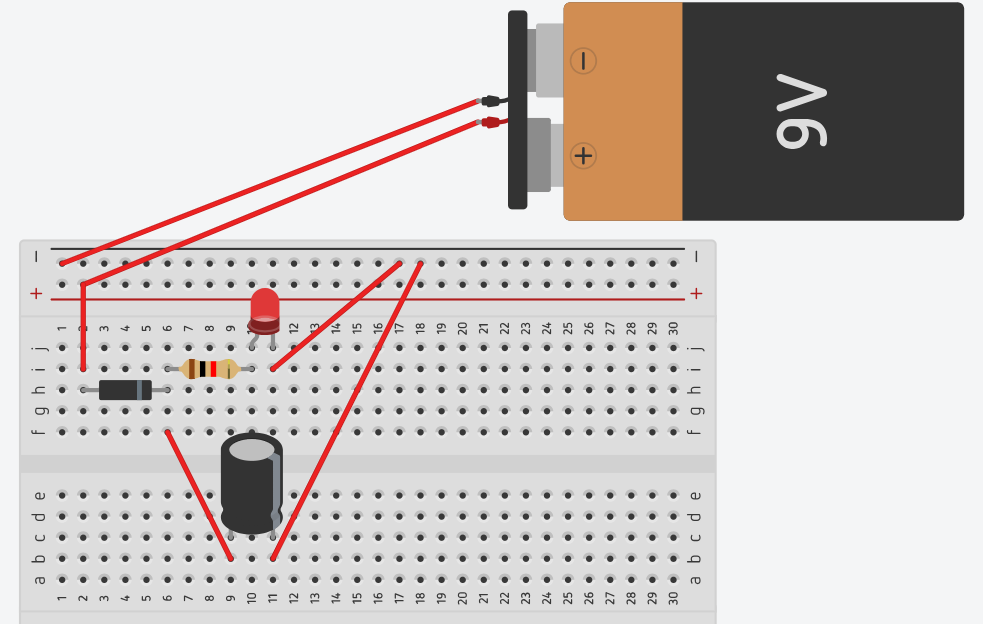
Led’s in parallel: (the resistors are in parallel with each other the leds are in series with the resistors



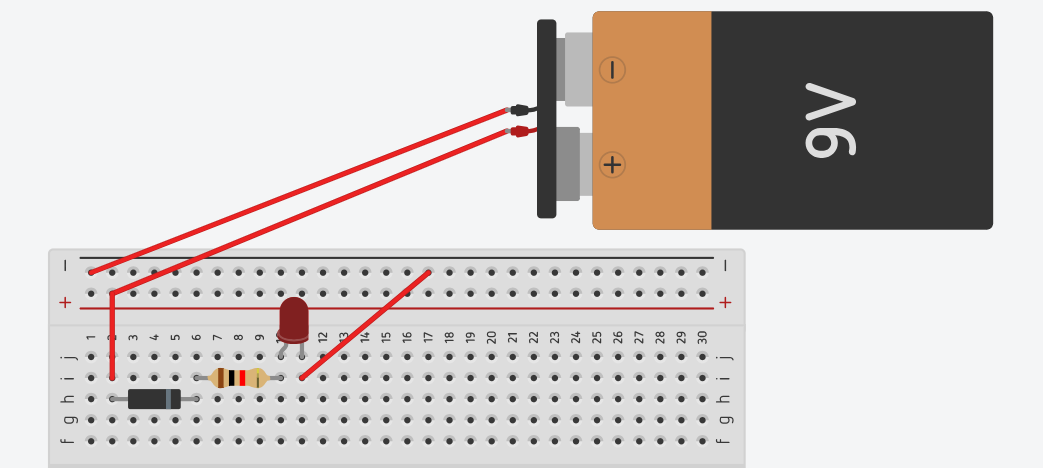
Capacitor

ckt:

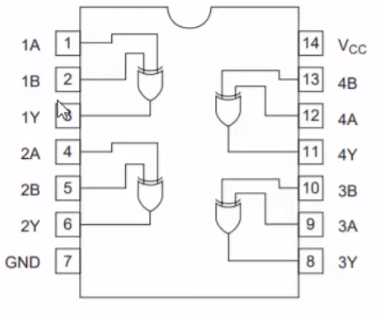
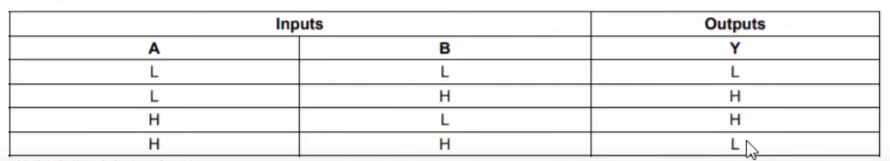


Connections:

Diode

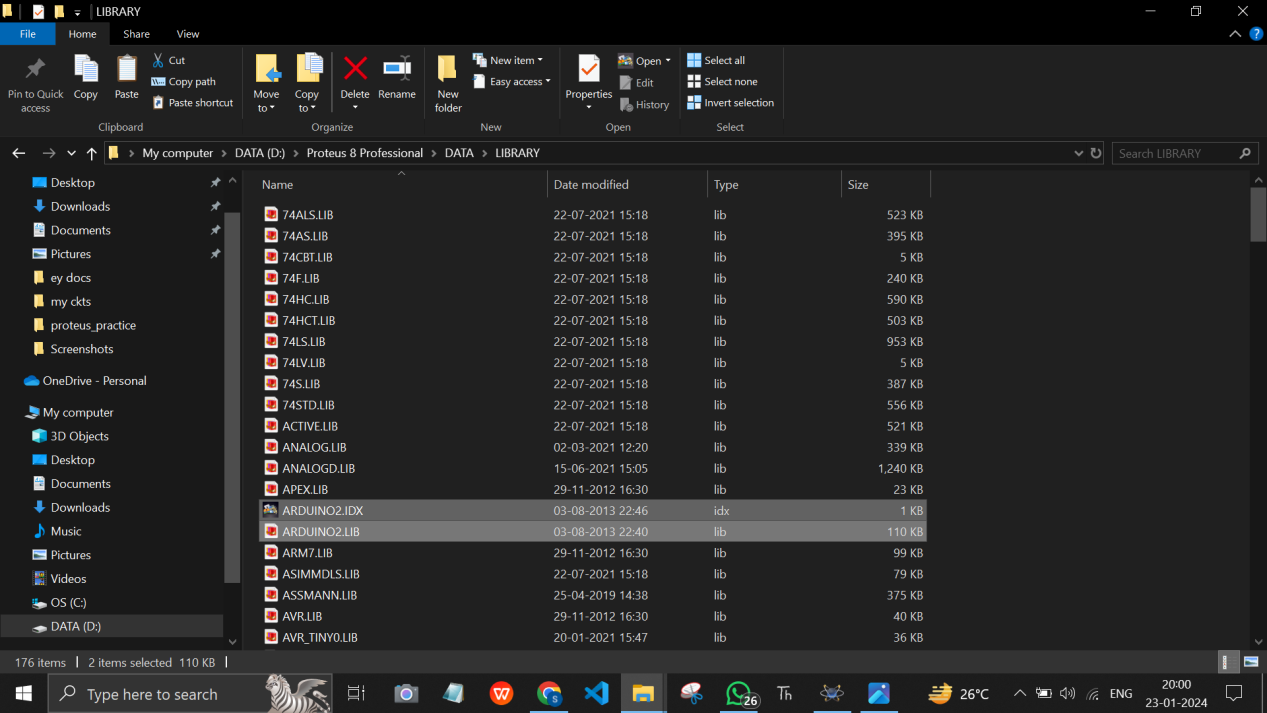


Logic gates:

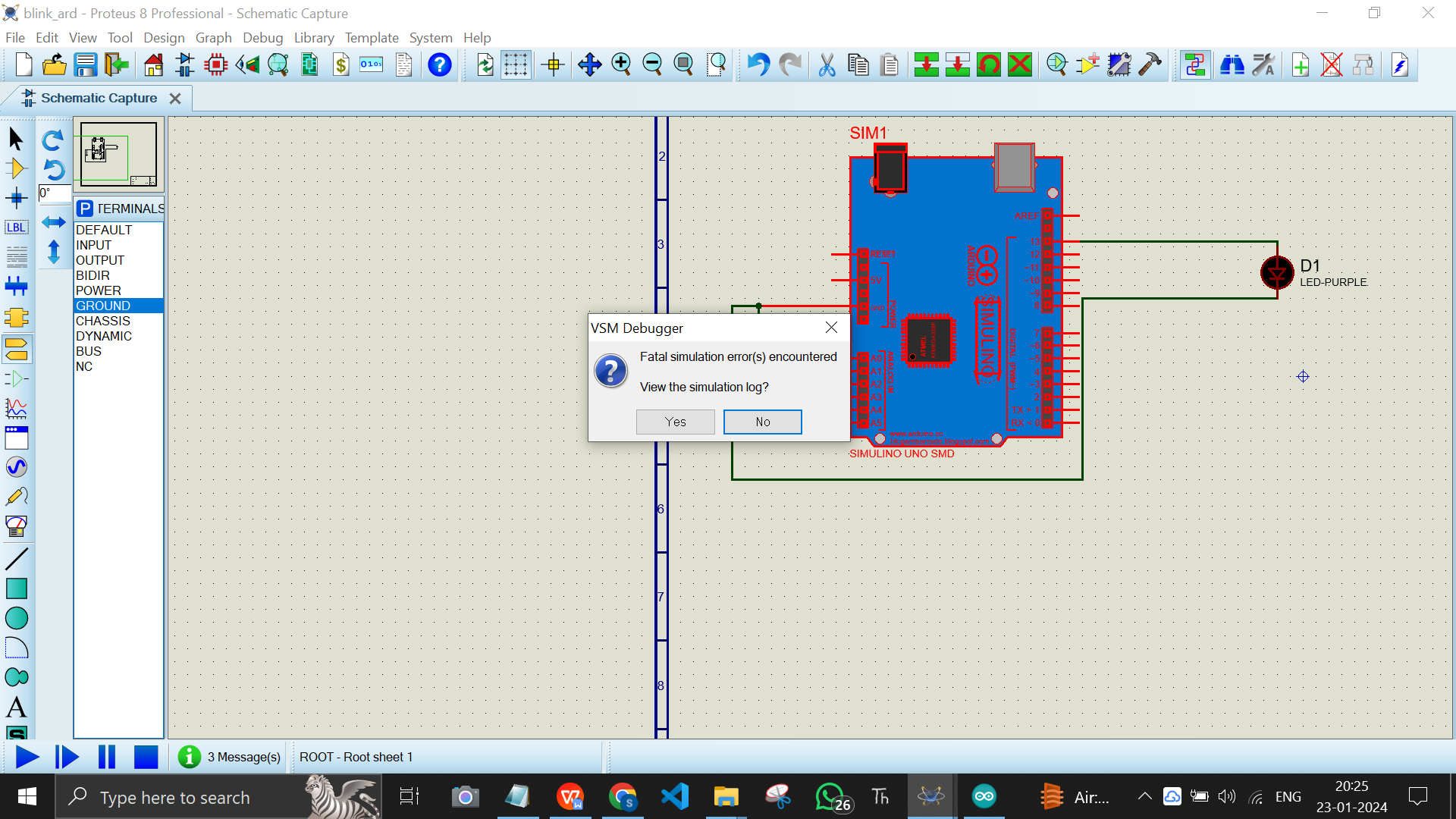
 

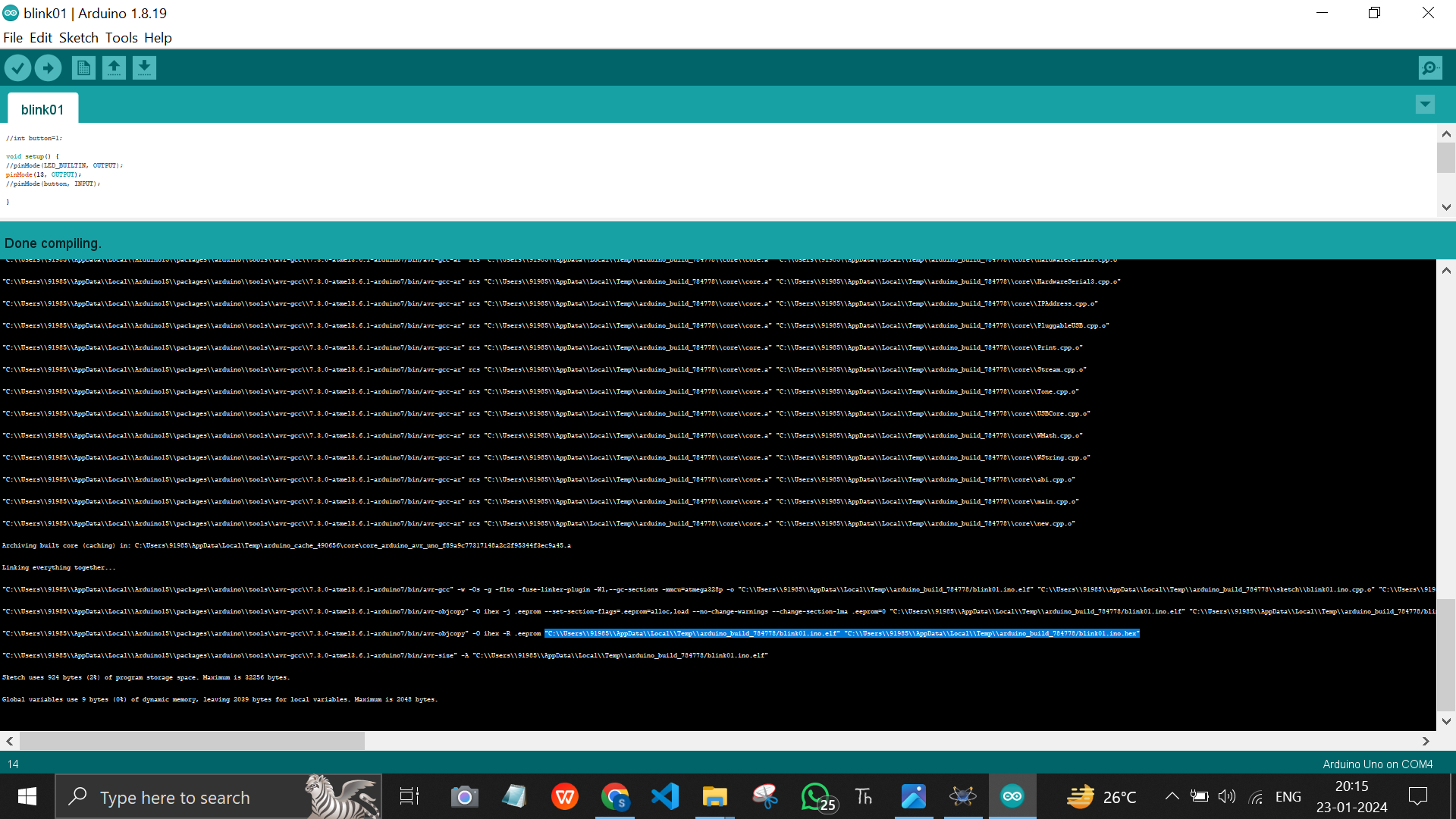
PROTEUS: pls put new ckt in new project

Add arduino lib , download from, https://drive.google.com/file/d/1yV5pCpzhq\_Td8zuksWEae6Dqg-0E7l5v/view

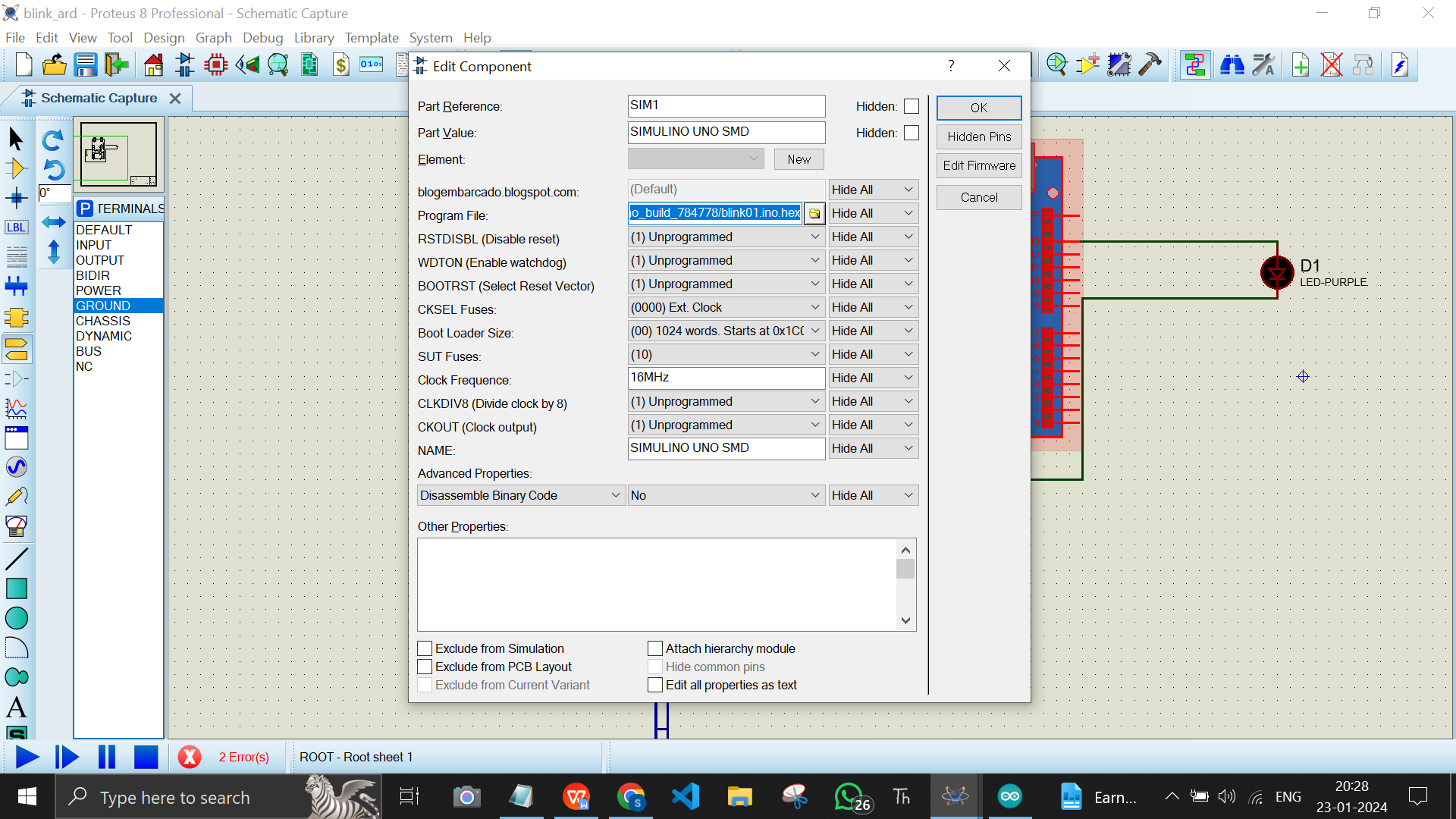


if arduino/simulino isnt programmed yet, it will give this error:



Just right click on arduinos properties and u get the pgm file field, run the code of arduino in ide and get the .hex location 

Like this:



Esp32 lib from: <https://goalmdcat.com/esp32-library-for-proteus>

Rpi library: <https://www.theengineeringprojects.com/document/raspberry-pi-4-library-for-proteus/1>

Bluettoth lib from <https://www.theengineeringprojects.com/document/bluetooth-library-for-proteus/1>

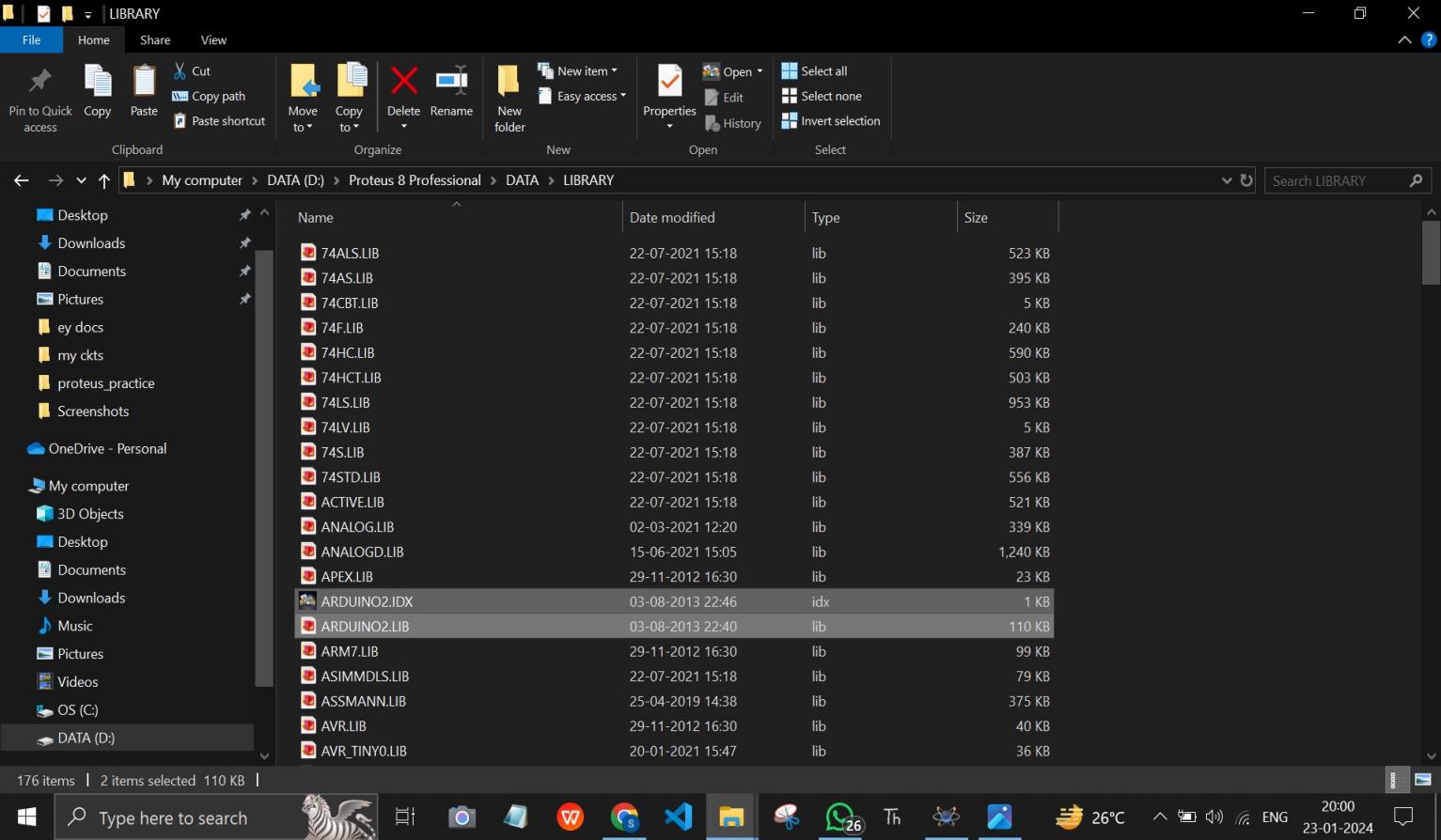
Wait for 30sec counter then link appears for rar file

Ir sensor

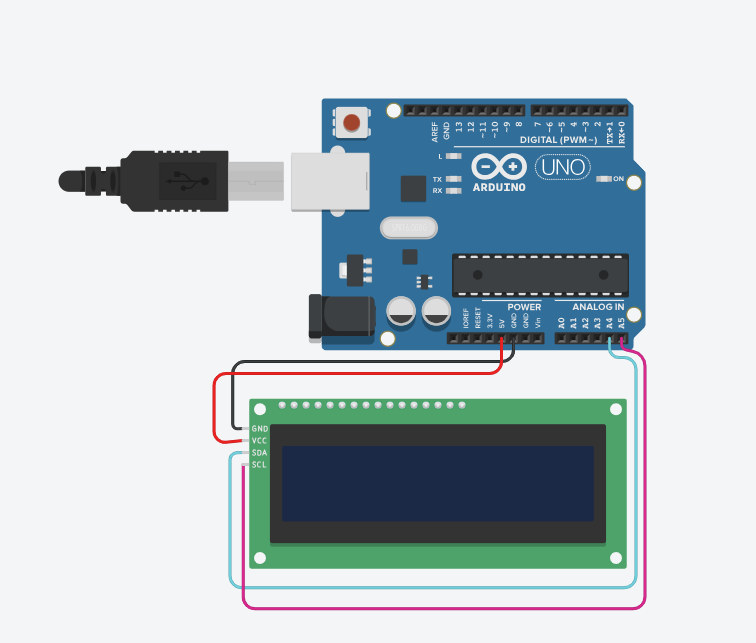
<https://goalmdcat.com/infrared-sensor-library/#google_vignette>

Scroll neeche tak click on download

Paste the extracted “LIB” and “IDX” files inside the lib folder of proteus:



Using lcd in tinkercad:



Code:

// C++ code

//

#include <Adafruit\_LiquidCrystal.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd\_1(32, 16, 2);

void setup()

{

lcd\_1.init();

lcd\_1.setCursor(0, 0);

lcd\_1.setBacklight(1);

lcd\_1.display();

}

void loop()

{

lcd\_1.setCursor(0, 0);

lcd\_1.print("sanjana");

delay(2000); // Wait for 2000 millisecond(s)

lcd\_1.setCursor(0, 10);

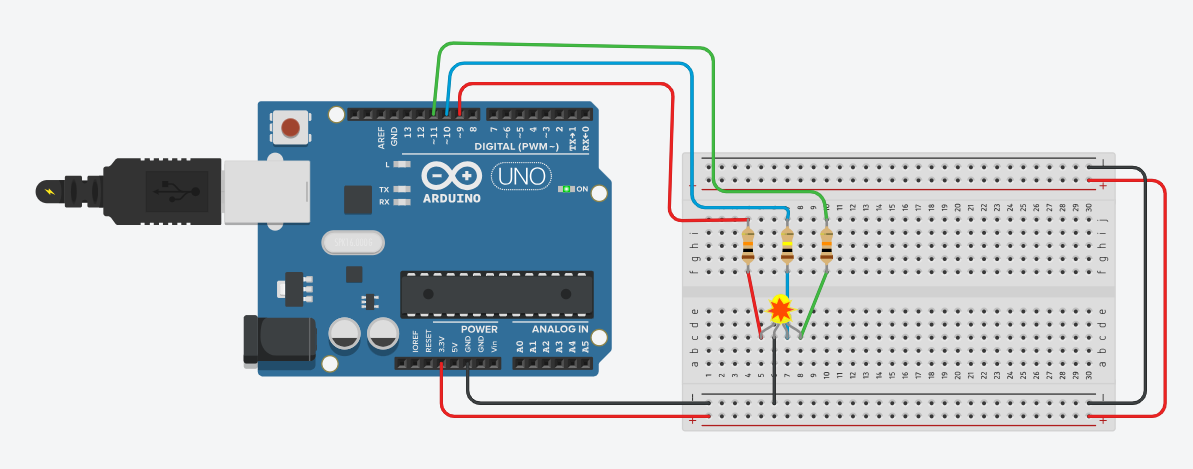
lcd\_1.print("asrani");

delay(2000); // Wait for 2000 millisecond(s)

lcd\_1.clear();

}

Rgb led



const int redPin = 9; // Pin for the red component of the RGB LED

const int bluePin = 10; // Pin for the green component

const int greenPin = 11; // Pin for the blue component

void setup() {

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

}

void loop() {

// Call the function with different colors to see the RGB LED transition smoothly

fadeRGB(255, 0, 0); // Red

delay(1000);

fadeRGB(0, 255, 0); // Green

delay(1000);

fadeRGB(0, 0, 255); // Blue

delay(1000);

fadeRGB(255, 255, 0); // Yellow

delay(1000);

}

// Function to smoothly transition between colors

void fadeRGB(int redValue, int greenValue, int blueValue) {

// Smoothly fade the red component

for (int i = 0; i <= redValue; i++) {

analogWrite(redPin, i);

delay(5);

}

// Smoothly fade the green component

for (int i = 0; i <= greenValue; i++) {

analogWrite(greenPin, i);

delay(5);

}

// Smoothly fade the blue component

for (int i = 0; i <= blueValue; i++) {

analogWrite(bluePin, i);

delay(5);

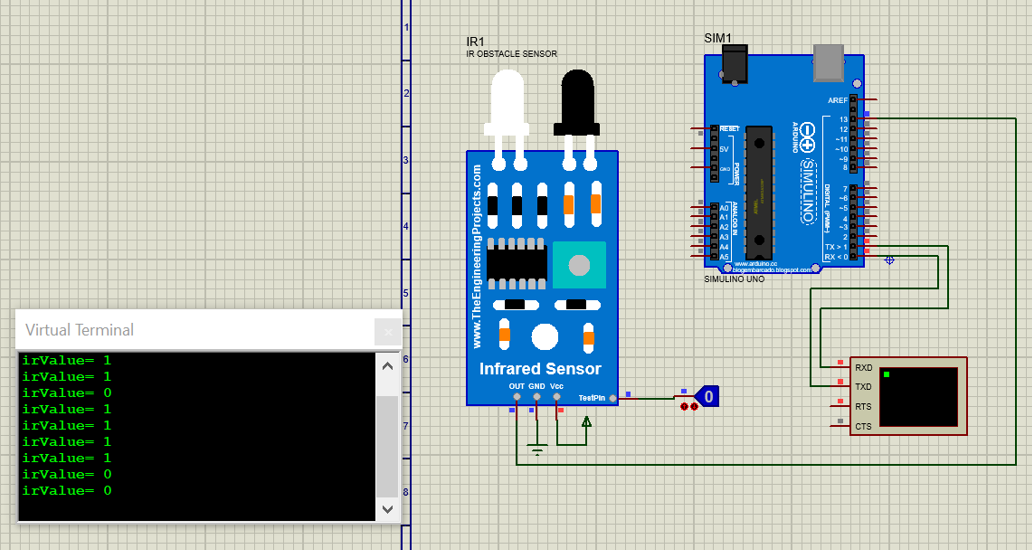
}

}

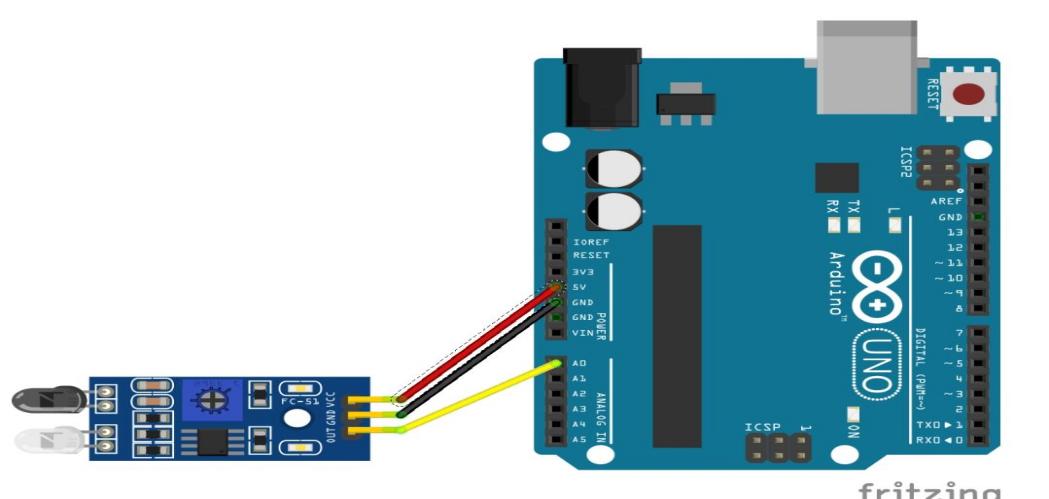
37 Sensors of Arduino:

1. Infrared sensor

(digital read)



(analog read)



Code:

int irPin=A0;

//int irPin=13;

void setup() {

pinMode(irPin, INPUT);

Serial.begin(9600);

}

void loop() {

int irValue=analogRead(irPin);

//int irValue=digitalRead(irPin);

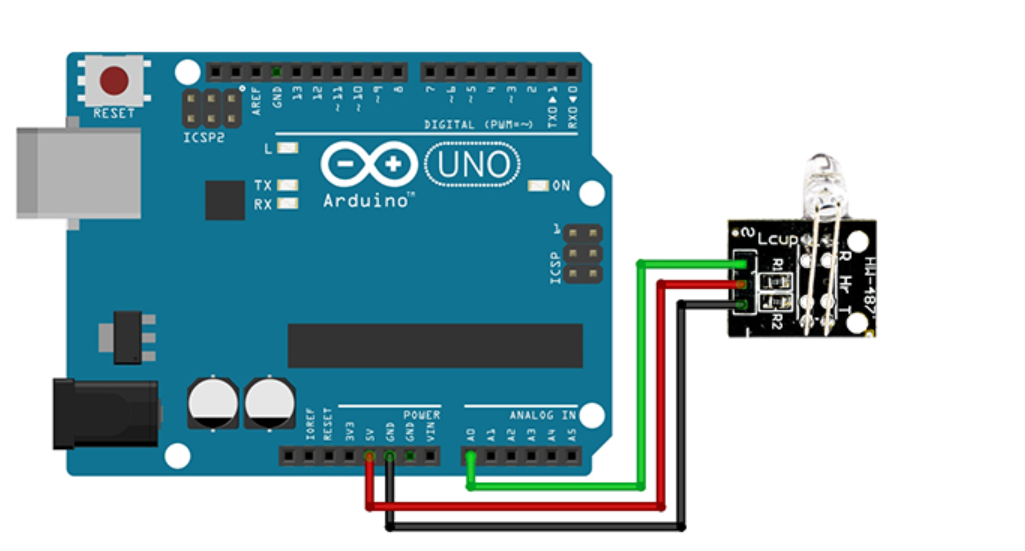
Serial.print("irValue= ");

Serial.println(irValue);

delay(500);

}

1. Heartbeat Sensor (Lcup) KY-039



Code:

/\*

Made on Jan 16, 2021

By MehranMaleki @ Electropeak

Home

\*/

void setup() {

pinMode(A0, INPUT);

Serial.begin(9600);

}

void loop() {

float pulse;

int sum = 0;

for (int i = 0; i < 20; i++)

sum += analogRead(A0);

pulse = sum / 20.00;

Serial.println(pulse);

delay(100);

}

When u install the max30100 library, itll be added into examples

Make sure when u upload the code, the baud rate in serial monitor mathches the baud rate in code given as serial.begin(xyz)

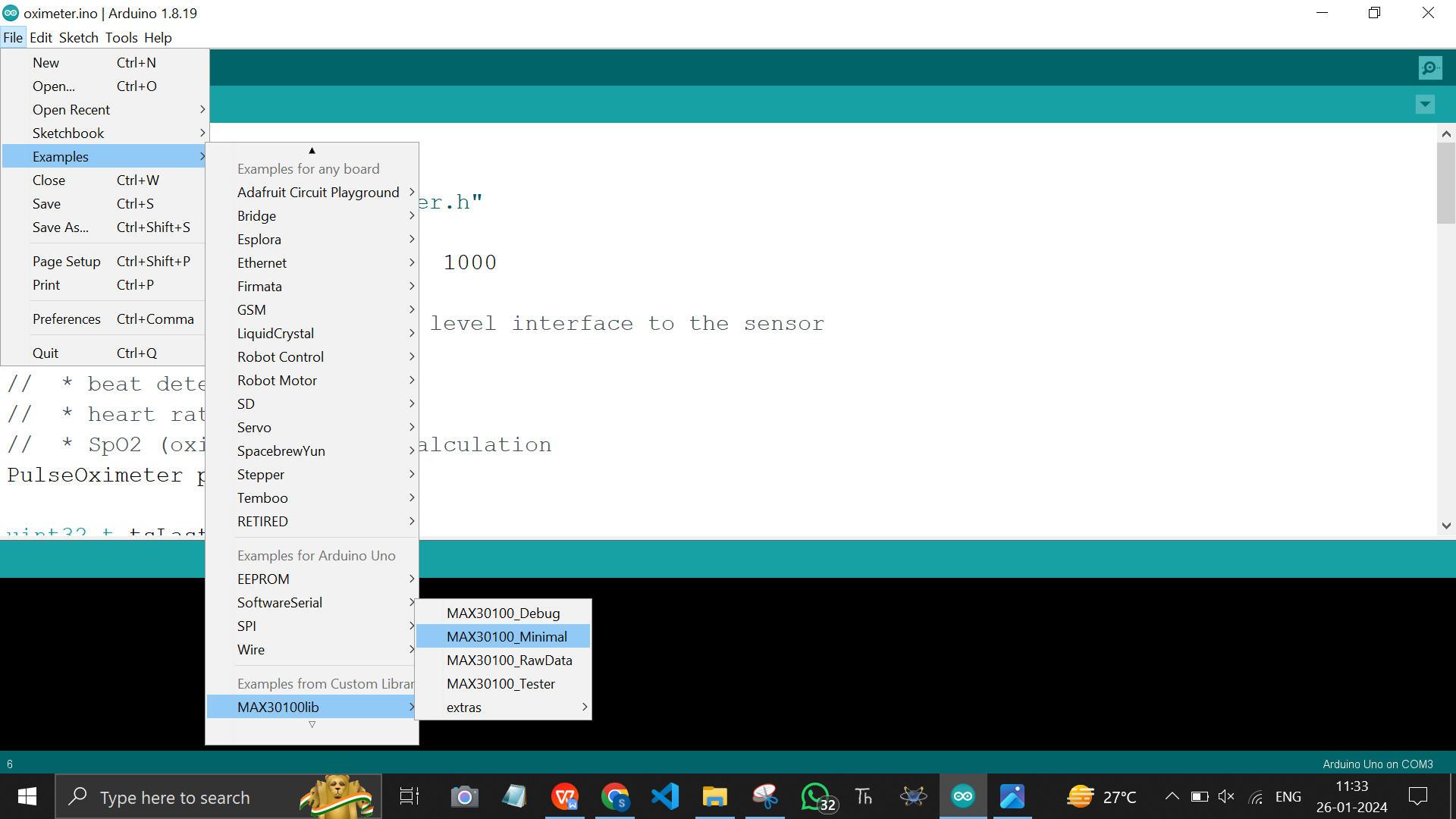
….……….

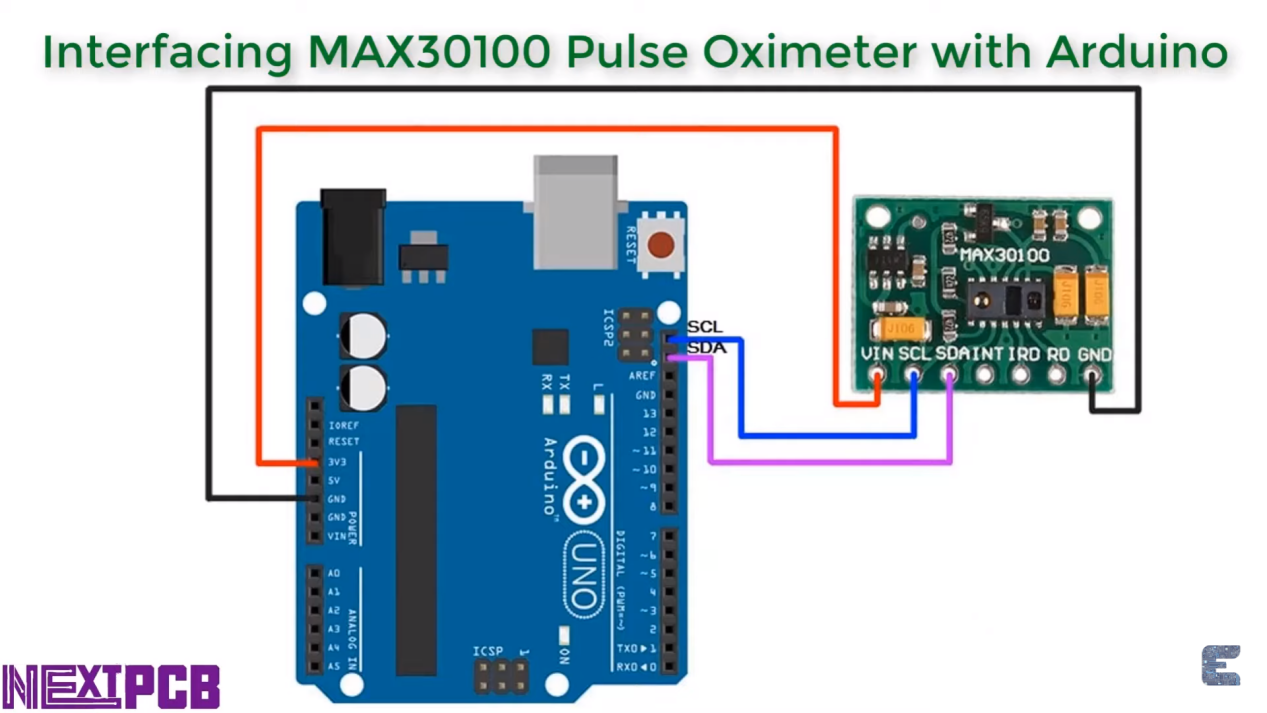
in Arduino, to go Sketch > Include Library > Manage Library...

In the Library Manager: Search for and Select "PulseSensor.com" and max30100

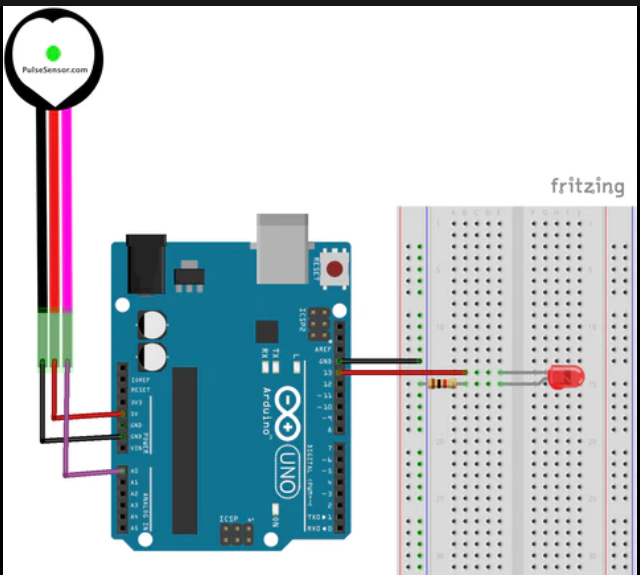
these can be looked for in file > examples > custom

Install or update to the lastest version.





4.



/\* PulseSensor Starter Project and Signal Tester

\* The Best Way to Get Started With, or See the Raw Signal of, your PulseSensor.com™ & Arduino.

\*

\* Here is a link to the tutorial

\* https://pulsesensor.com/pages/code-and-guide

\*

\* WATCH ME (Tutorial Video):

\* https://www.youtube.com/watch?v=RbB8NSRa5X4

\*

\*

-------------------------------------------------------------

1) This shows a live human Heartbeat Pulse.

2) Live visualization in Arduino's Cool "Serial Plotter".

3) Blink an LED on each Heartbeat.

4) This is the direct Pulse Sensor's Signal.

5) A great first-step in troubleshooting your circuit and connections.

6) "Human-readable" code that is newbie friendly."

\*/

// Variables

int PulseSensorPurplePin = 0; // Pulse Sensor PURPLE WIRE connected to ANALOG PIN 0

int LED = 13; // The on-board Arduion LED

int Signal; // holds the incoming raw data. Signal value can range from 0-1024

int Threshold = 580; // Determine which Signal to "count as a beat", and which to ingore.

// The SetUp Function:

void setup() {

pinMode(LED,OUTPUT); // pin that will blink to your heartbeat!

Serial.begin(9600); // Set's up Serial Communication at certain speed.

}

// The Main Loop Function

void loop() {

Signal = analogRead(PulseSensorPurplePin); // Read the PulseSensor's value.

// Assign this value to the "Signal" variable.

Serial.println(Signal); // Send the Signal value to Serial Plotter.

if(Signal > Threshold){ // If the signal is above "550", then "turn-on" Arduino's on-Board LED.

digitalWrite(LED,HIGH);

} else {

digitalWrite(LED,LOW); // Else, the sigal must be below "550", so "turn-off" this LED.

}

delay(10);

}

27/01/24

Embedded ML: done w the edge impulse arduino library creation for custom dataset

Manage librarieS: get fastLED, neopixel

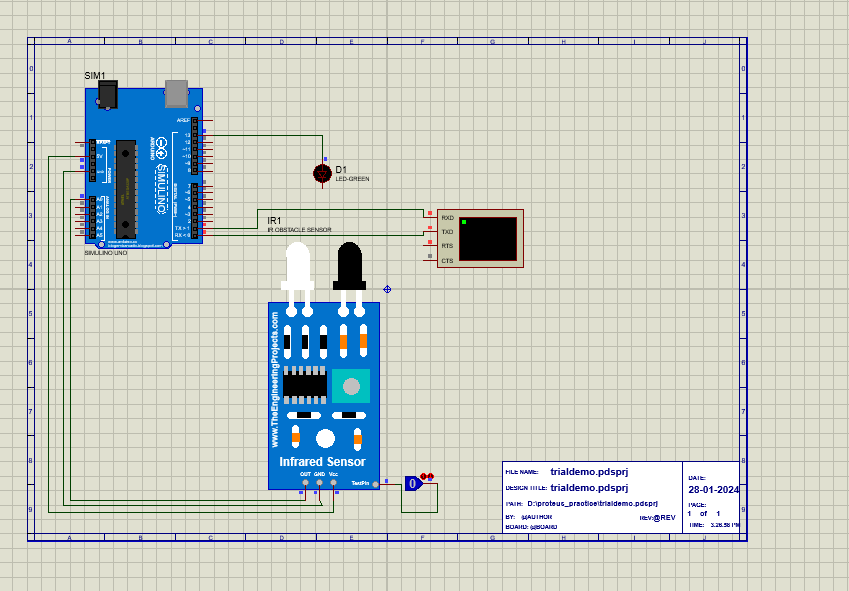
Neopixel in proteus:

Ws2812… right click on it and chose decompose (2pin se 4 pin aajati)

Xymatrix.ino, led display.pdoprj

MY PROJECT NI BGM

1. Ir sensor, virtual terminal, arduino



CODE:

int irPin=A0;

//int irPin=13;

void setup() {

pinMode(irPin, INPUT);

Serial.begin(9600);

}

void loop() {

int irValue=analogRead(irPin);

//int irValue=digitalRead(irPin);

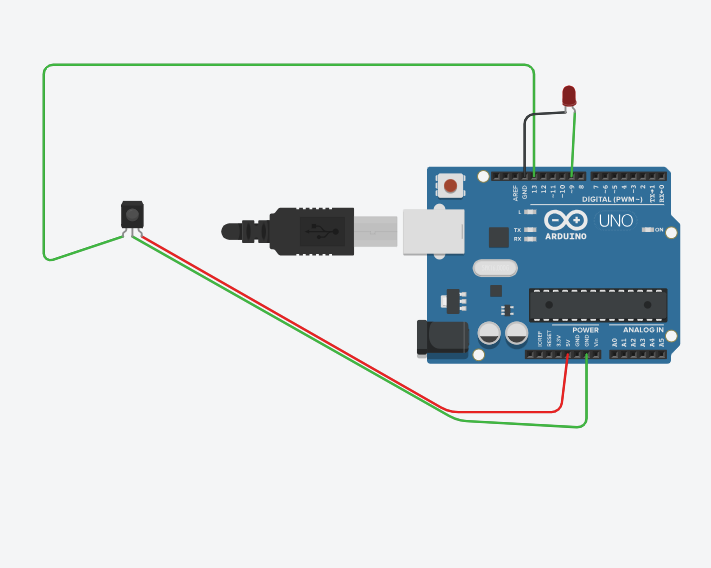
Serial.print("irValue= ");

Serial.println(irValue);

delay(500);

}

1. Ir sensor, virtual terminal, arduino, LED



//int irPin=A0;

int irPin=13;

int ledPin = 9; // Replace with the pin number where your LED is connected

void setup() {

pinMode(irPin, INPUT);

Serial.begin(9600);

}

void loop() {

//int irValue=analogRead(irPin);

int irValue=digitalRead(irPin);

Serial.print("irValue= ");

Serial.println(irValue);

delay(500);

if (irValue == 1) {

digitalWrite(ledPin, HIGH); // Turn on the LED when irValue is 1

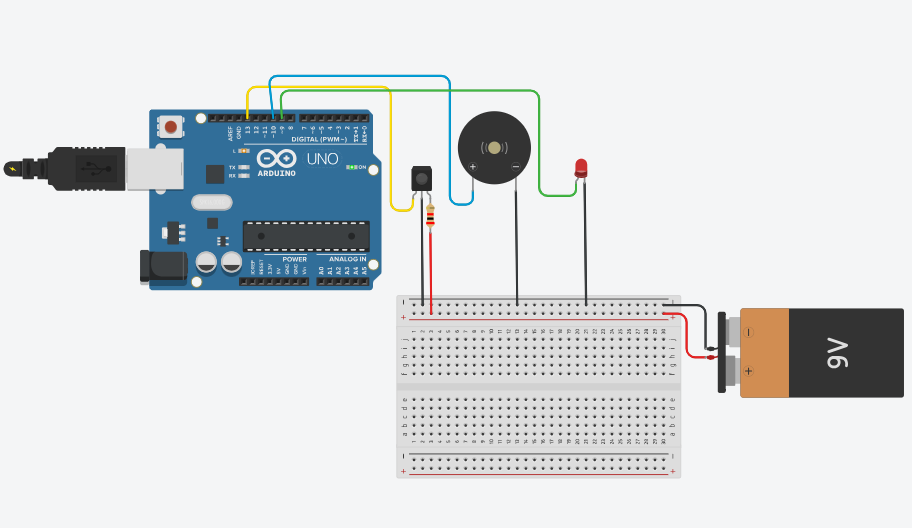
} else {

digitalWrite(ledPin, LOW); // Turn off the LED otherwise

}

}

1. Ir sensor, virtual terminal, arduino, LED, buzzer



//int irPin = A0;

int irPin = 13;

int ledPin = 9; // Replace with the pin number where your LED is connected

int buzzerPin = 10; // Replace with the pin number where your buzzer is connected

void setup() {

pinMode(irPin, INPUT);

pinMode(ledPin, OUTPUT); // Set the LED pin as OUTPUT

pinMode(buzzerPin, OUTPUT); // Set the buzzer pin as OUTPUT

Serial.begin(9600);

}

void loop() {

//int irValue = analogRead(irPin);

int irValue = digitalRead(irPin);

Serial.print("irValue = ");

Serial.println(irValue);

if (irValue == HIGH) {

digitalWrite(ledPin, HIGH); // Turn on the LED when irValue is 1

tone(buzzerPin, 1000); // Turn on the buzzer at a frequency of 1000 Hz

} else {

digitalWrite(ledPin, LOW); // Turn off the LED otherwise

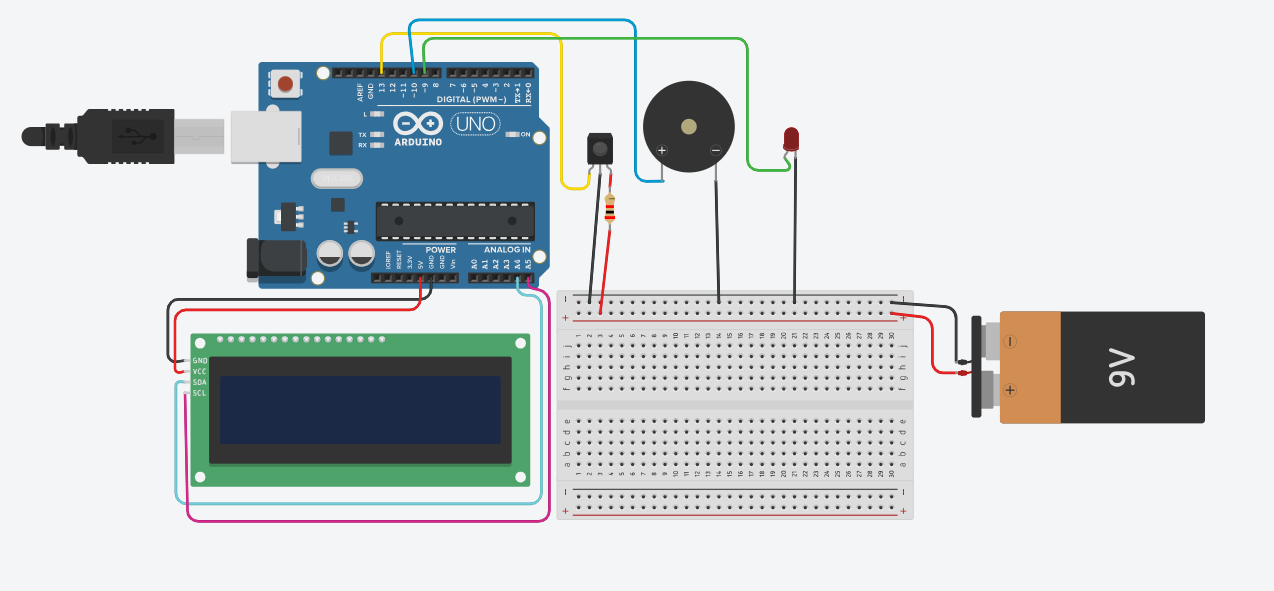
noTone(buzzerPin); // Turn off the buzzer

}

delay(500);

}

1. Ir sensor, virtual terminal, arduino, LED, buzzer, lcd



#include <LiquidCrystal\_I2C.h>

//int irPin = A0;

int irPin = 13;

int ledPin = 9; // Replace with the pin number where your LED is connected

int buzzerPin = 10; // Replace with the pin number where your buzzer is connected

LiquidCrystal\_I2C lcd(32, 16, 2); // I2C address 0x27, 16 column and 2 rows

void setup() {

pinMode(irPin, INPUT);

pinMode(ledPin, OUTPUT); // Set the LED pin as OUTPUT

pinMode(buzzerPin, OUTPUT); // Set the buzzer pin as OUTPUT

lcd.init();

lcd.setCursor(0, 0);

lcd.setBacklight(1);

lcd.display();// Initialize the LCD

lcd.print("started yayy");

delay(2000);

}

void loop() {

//int irValue = analogRead(irPin);

int irValue = digitalRead(irPin);

Serial.print("irValue = ");

Serial.println(irValue);

lcd.setCursor(0, 0); // Set the cursor to the beginning of the second line

if (irValue == HIGH) {

digitalWrite(ledPin, HIGH); // Turn on the LED when irValue is 1

//tone(buzzerPin, 1000); // Turn on the buzzer at a frequency of 1000 Hz

lcd.clear();

lcd.print("IR Detected "); // Display on the LCD when IR is detected

} else {

digitalWrite(ledPin, LOW); // Turn off the LED otherwise

//noTone(buzzerPin); // Turn off the buzzer

lcd.print("No IR Detected"); // Display on the LCD when no IR is detected

}

delay(500);

lcd.clear(); // Clear the LCD for the next iteration

}

1. Ir sensor, virtual terminal, arduino, LED, buzzer, lcd, bluetooth

All biomedical sensors on arduino:

1. Heart Rate Sensor: Measures the heartbeats per minute, commonly used in fitness trackers and medical devices.

2. Blood Pressure Sensor: Measures the pressure of blood against the walls of arteries.

3. Temperature Sensor: Monitors body temperature or ambient temperature.

4. Electrocardiogram (ECG or EKG) Sensor: Records the electrical activity of the heart over a period, used for diagnosing heart conditions.

5. Pulse Oximeter: Measures oxygen saturation in the blood by clipping onto a fingertip.

6. Respiratory Rate Sensor: Monitors the number of breaths per minute.

7. Glucometer: Measures blood glucose levels, commonly used by people with diabetes.

13. Humidity Sensor: Measures the amount of moisture in the air.

These sensors find applications in fields like healthcare, fitness, consumer electronics, industrial automation, environmental monitoring, and more. Advances in technology continue to expand the capabilities and applications of sensors in various domains.