# IoT Workshop-2024

### ETCE LAB



# Welcoming

Welcome to the Internet of Things (IoT) Workshop!

# Introduction

Throughout this workshop,we'll embark on a journey through the fundamentals of IoT, starting with an overview of what IoT is and how it's being use in different industries especially in mining.

# what you will learn

- Experience with popular microcontrollers such as the Arduino Nano 33 and Arduino Uno
- Being able to work with various sensors
- Foundational coding skills, including essential functions and techniques,necessary for IoT projects.

## Attention

Please remember this for all experiments.

- Treat the equipment very gently.
- Look at the diagram first to see how the wires and sensors connect to the Arduino.
- Always ask questions.
- We will begin coding after double-checking that the wires and sensors are correctly connected to the Arduino.
- Try to learn the codes and keep questioning the tutor.
- When you completely understand the codes, start typing them in the Arduino-related software.
- Look optimistically. If you did not have any errors.
- Then connect the Arduino hardware to the computer with the related cable.
- Finally, you upload the codes to Arduino.

*Thank you very much**

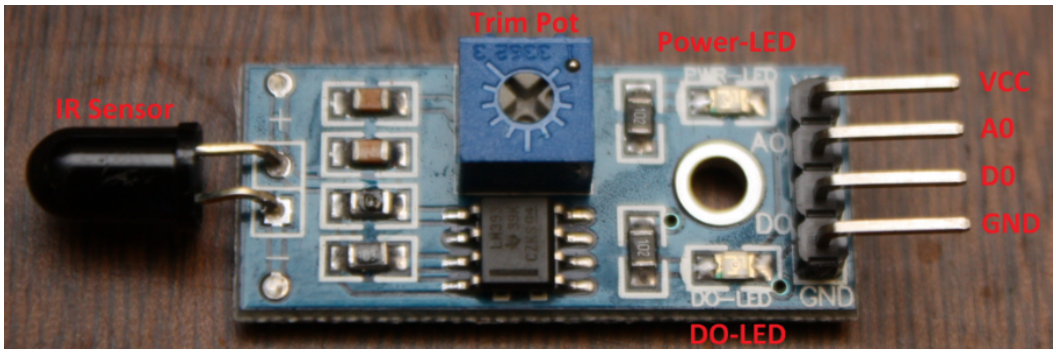
## let's start :)

## 1-Detecting fire

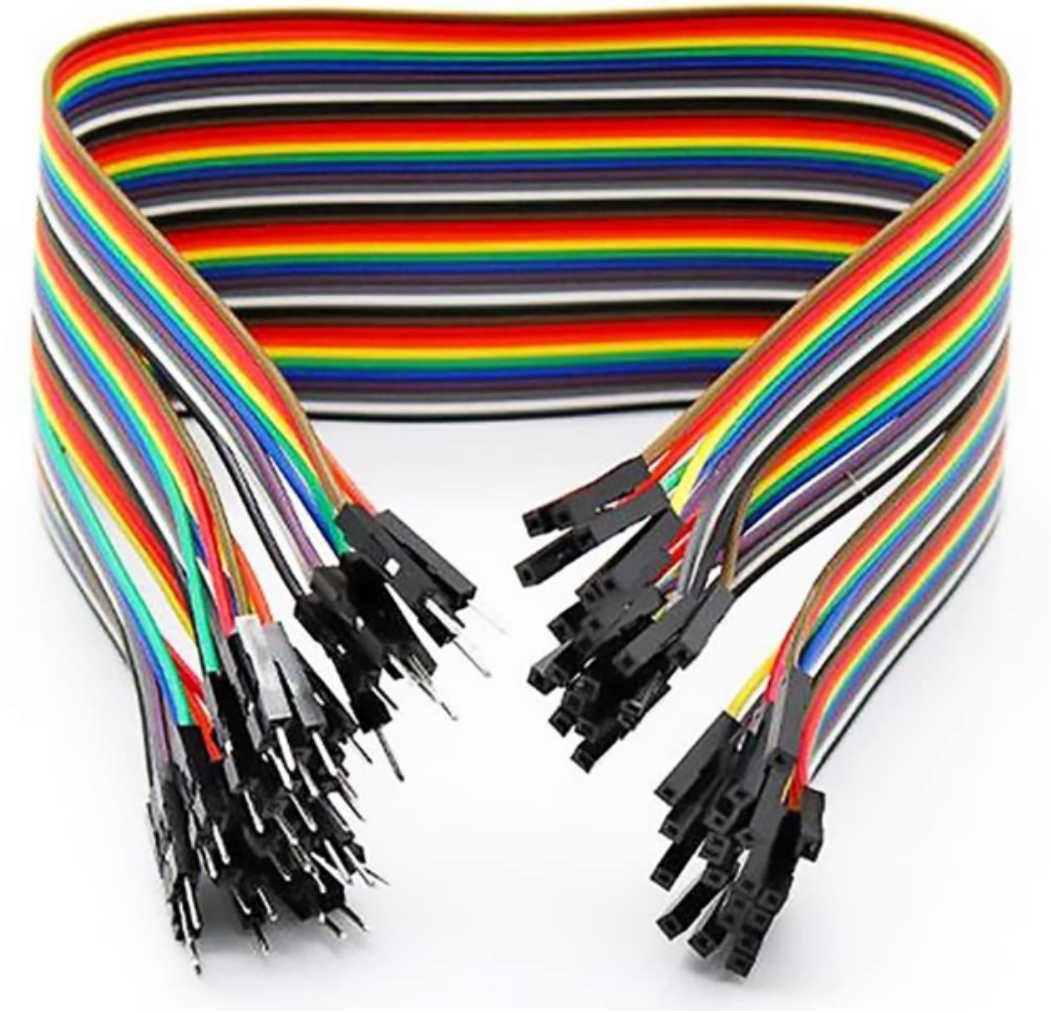
we will learn how to detect fire using a flame sensor.

# Components and supplies:

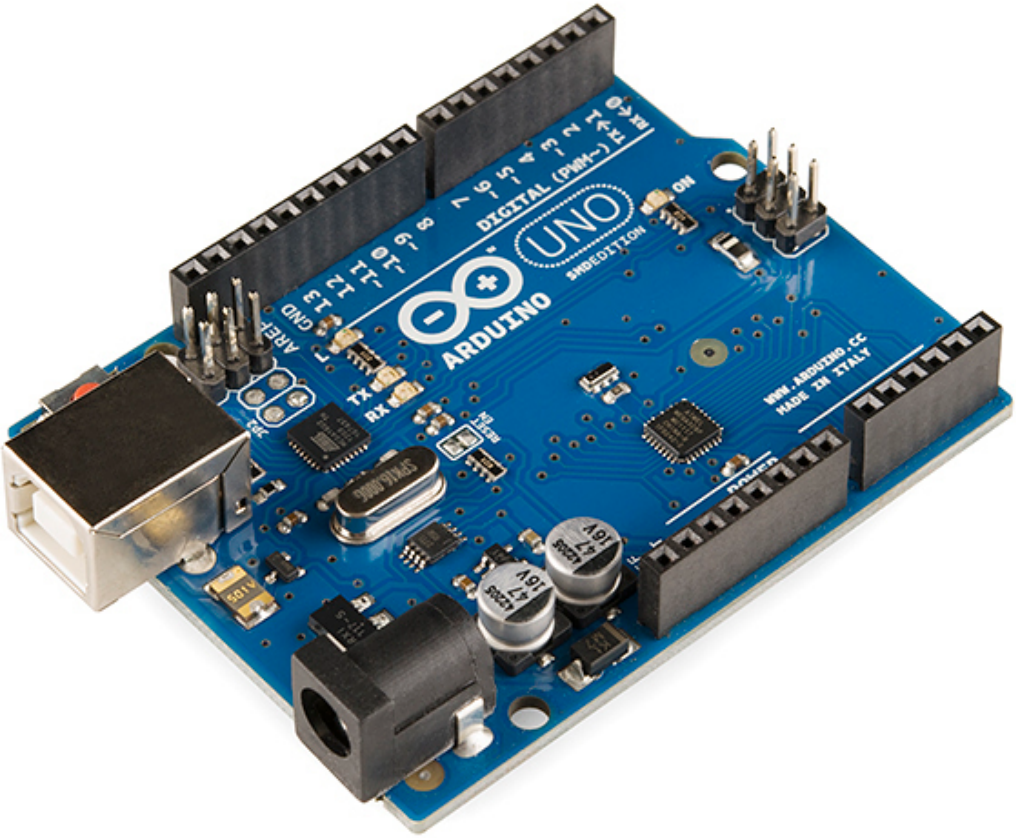
- Flame Sensor (model with an analog out) -> These types of sensors are used for short range fire detection(up to about 3 feet)

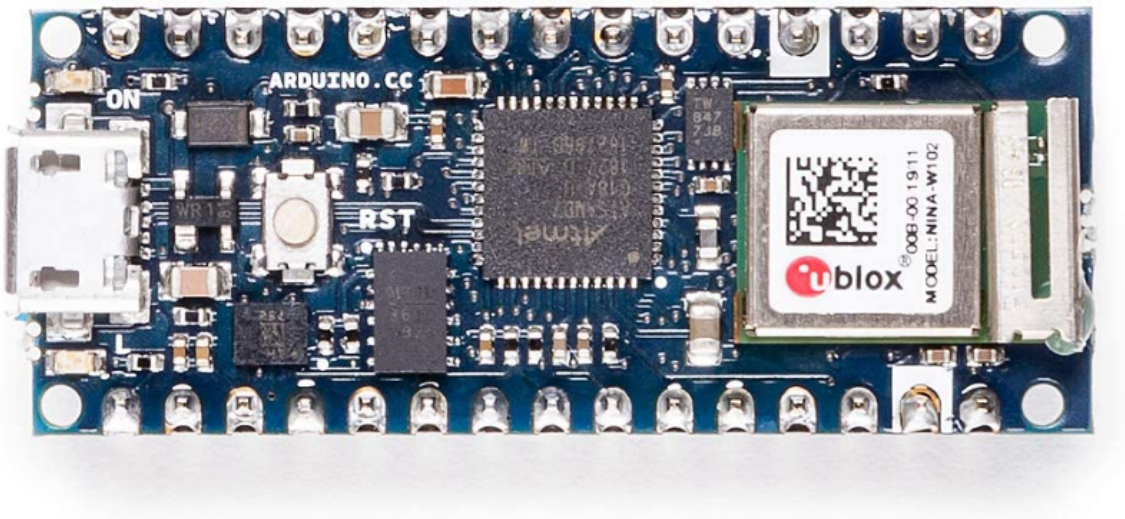


- Male to female jumper wires



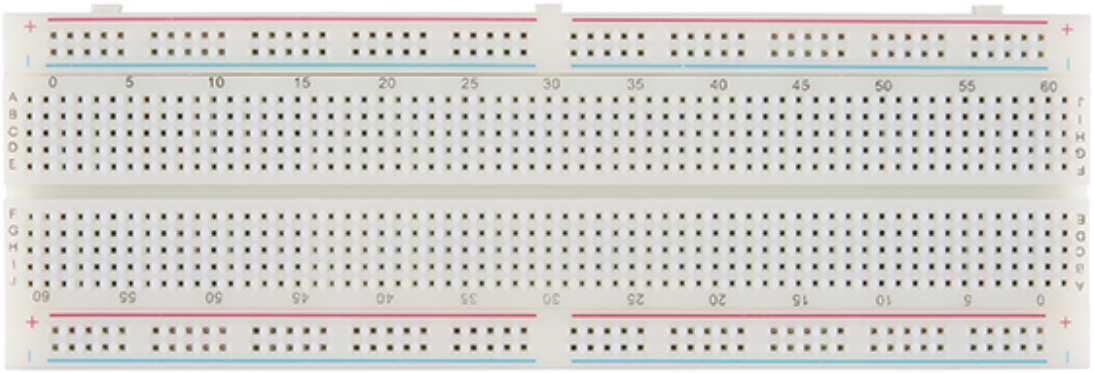
- An arduino, any flavor




- Lighter or another flame source for testing

NOTE : If you are using arduino nano 33, then you need to use " BreadBoard"also.



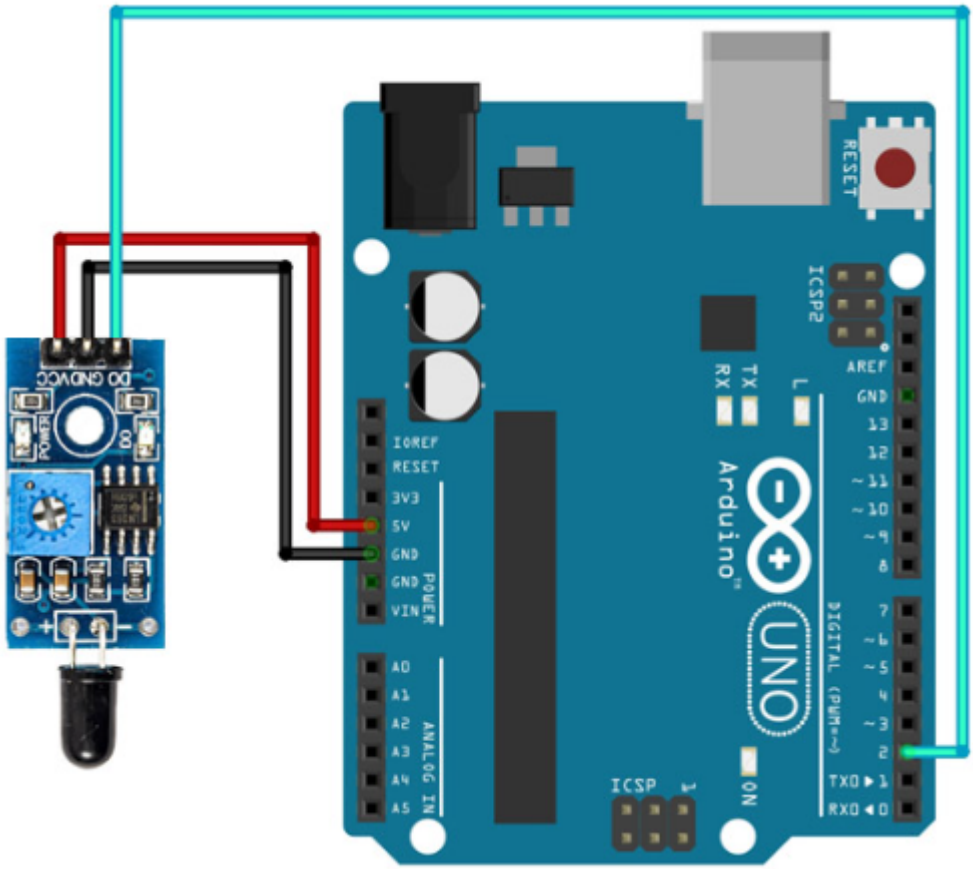
Important pins in EVERY arduino

- VCC --> positive voltage input: 5v for analog and 3.3v for digital
- A0 --> analog output
- D0 --> digital output
- GND --> ground

Wiring to an arduino: To wire the flame sensor to the arduino simply connect the following as shown:

- flame sensor -> arduino

- VCC -> 5v
- GND -> GND
- A0 -> Analog in 0




```
// lowest and highest sensor readings :
const int sensorMin = 0;
// sensor minimum
const int sensorMax = 1024; // sensor maximum
void ---- () {
// initialize serial communication @ 9600 baud :
Serial . begin (9600);
}
void ---- () {
// read the sensor on analog A0 :
int sensorReading = analogRead ( A0 );
// map the sensor range ( four options ):
// ex : ' long int map ( long int , long int ,
// long int , long int , long int ) '
int range = map ( sensorReading , sensorMin , sensorMax ,0 ,3);
// range value :
switch ( range ) {
case 0:
// A fire closer than 1.5 feet away .
Serial . println ( " ** Close Fire ** " );
```

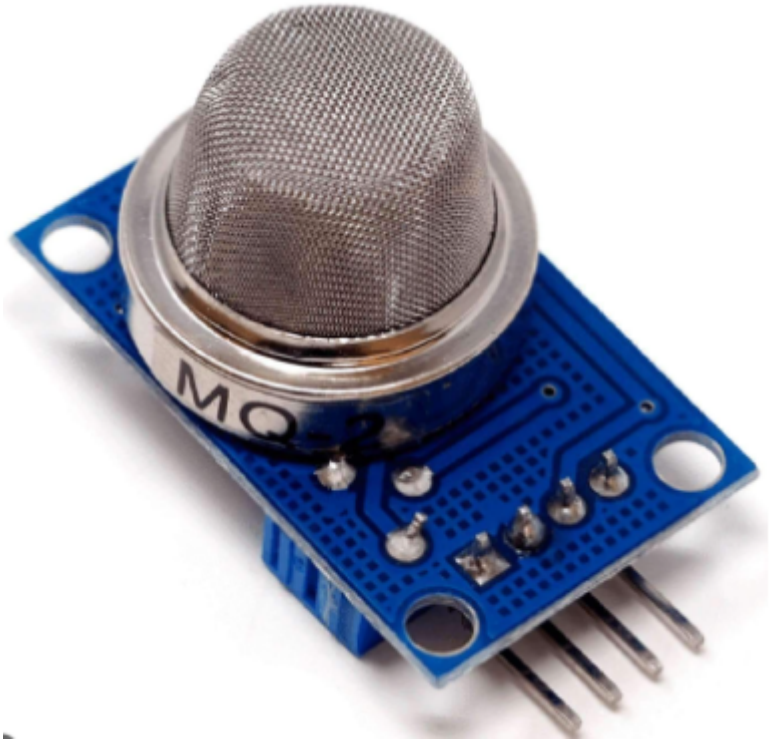
```
break ;
case 1:
// A fire between 1 -3 feet away .
---------------( " ** Distant Fire ** " );
break ;
case 2:
// No fire detected .
---------------( " No Fire " );
break ;
}
delay (1); // delay between reads
}
```

### 2- Smoke Detector

we will learn how to create a smoke detector using gas sensor.

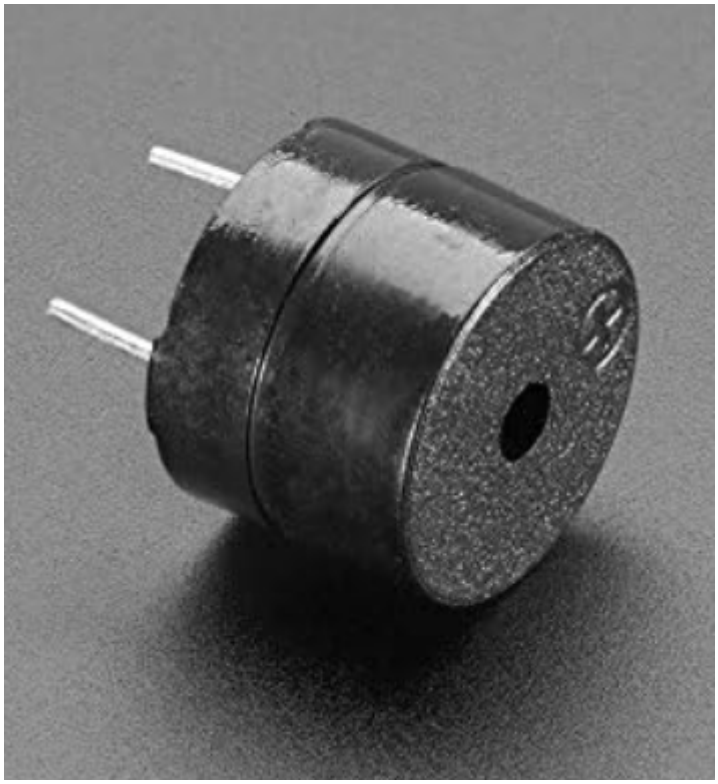
# Components and supplies:

- Resistor 220 ohm




- Gas sensor
- An arduino, any flavor
- Buzzer A buzzer is an electronic device that emits sound when it is powered bya 5-volt electrical current. These buzzers are commonly used in alarms, timers,

confirmation of user input.

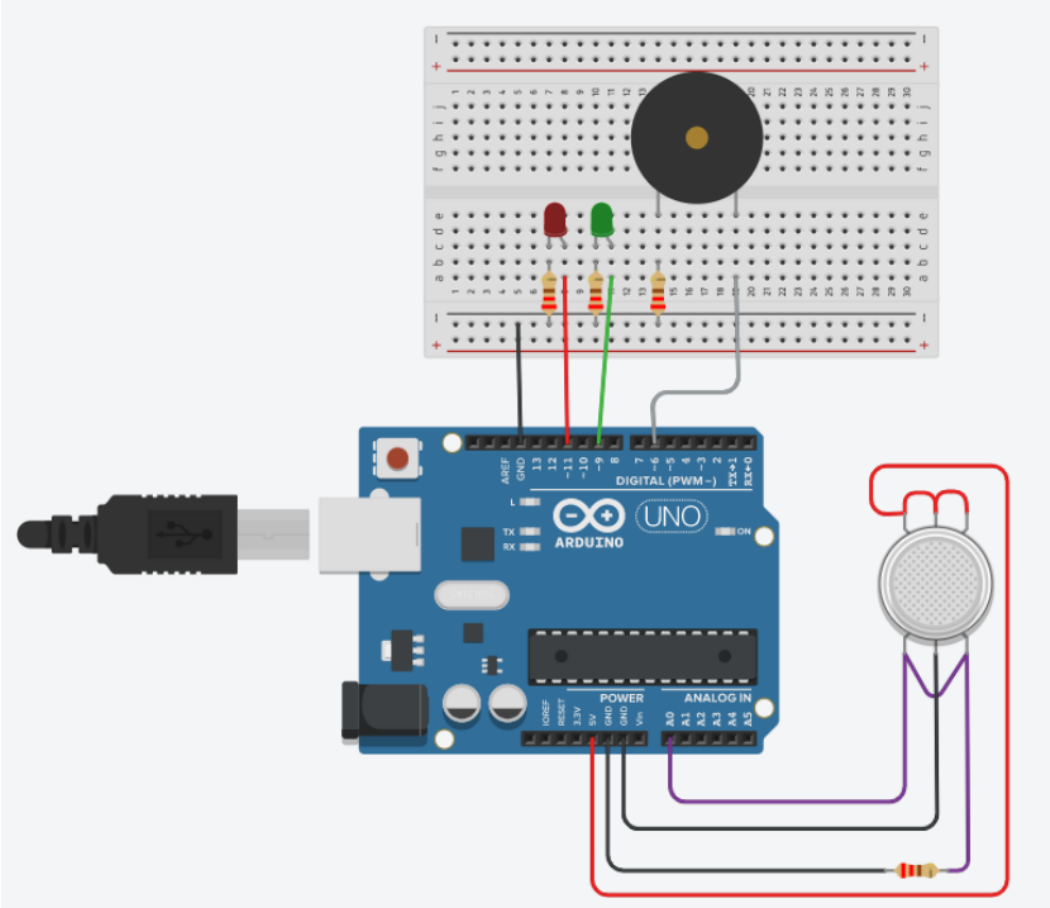


- Breadborad
- Jumper wires

- LED lights

## Wiring to an Arduino:

To wire the smoke sensor to the Arduino simply connect the following as shown:




```
int red_LED_PIN = 11;
int green_LED_PIN = 9;
int blue_LED_PIN = 10;
int buzzer = 6;
int smoke_detector = A0 ;
safety_lim = 60; // Sets smoke density safe limit
void setup () {
pinMode ( red_LED_PIN , ----);
pinMode ( green_LED_PIN , ---- );
pinMode ( blue_LED_PIN , ----);
pinMode ( buzzer , ---- );
pinMode ( smoke_detector , ----);
Serial . begin (9600); // baud rate
}
void loop () {
int sensor_read = analogRead ( smoke_detector );
// reads and stores the reading from the detector in sensor_read
Serial . print ( " Smoke Density : " );
Serial . println ( sensor_read );
if ( sensor_read > safety_lim )
// Checks if reading is beyond safety limit
{
```

```
analogWrite ( red_LED_PIN ,255);
analogWrite ( green_LED_PIN , 0);
tone ( buzzer ,500 , 100); // piezo rings
}
else
{
analogWrite ( green_LED_PIN , 255);
analogWrite ( red_LED_PIN ,0);
noTone ( buzzer ); // peizo wont ring
}
delay (50);
}
```

### Vibration Detection

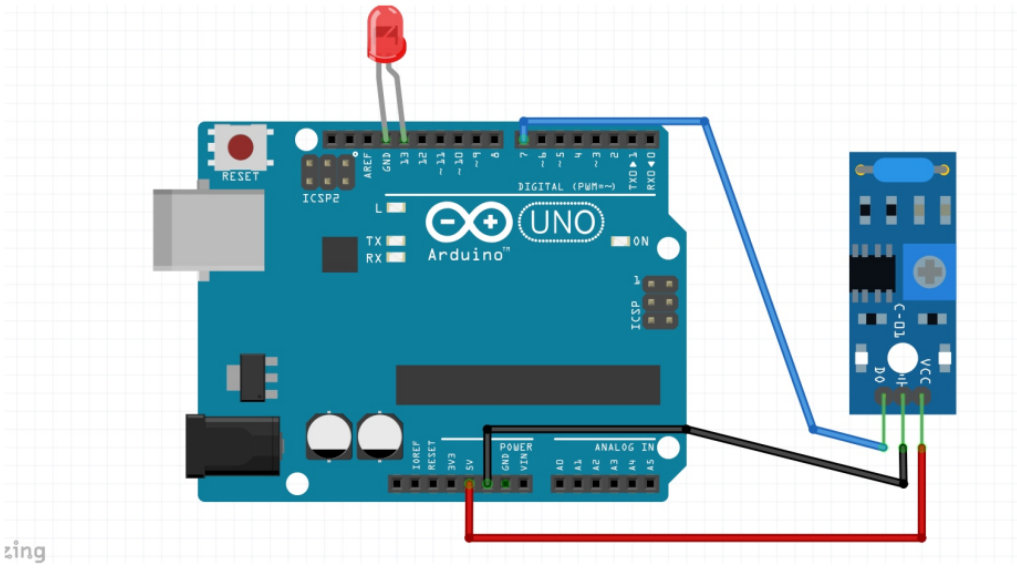
We will learn to be alert by LED lights if there is a vibration.

# Components and supplies:


- vibration sensor
- Arduino
- Green and red led lights
- Jumper wires

# Wiring to an Arduino:

To wire the vibration sensor to the Arduino simply connect the following as shown:




```
int vib_pin =7;
int led_pin =---;
void --- () {
pinMode ( vib_pin , --- );
pinMode ( led_pin , --- );
}
void --- () {
int val ;
val = digital--- ( vib_pin );
if ( val ==1)
{
digital--- ( led_pin , HIGH );
delay (1000);
digital---( led_pin , LOW );
delay (1000);
}
else
digitalWrite ( led_pin , LOW );
}
```

### 4- Make a siren **

We will learn how to use button controlled siren with different LED transi- tions.

# Components and supplies:

- Push button switch 12mm

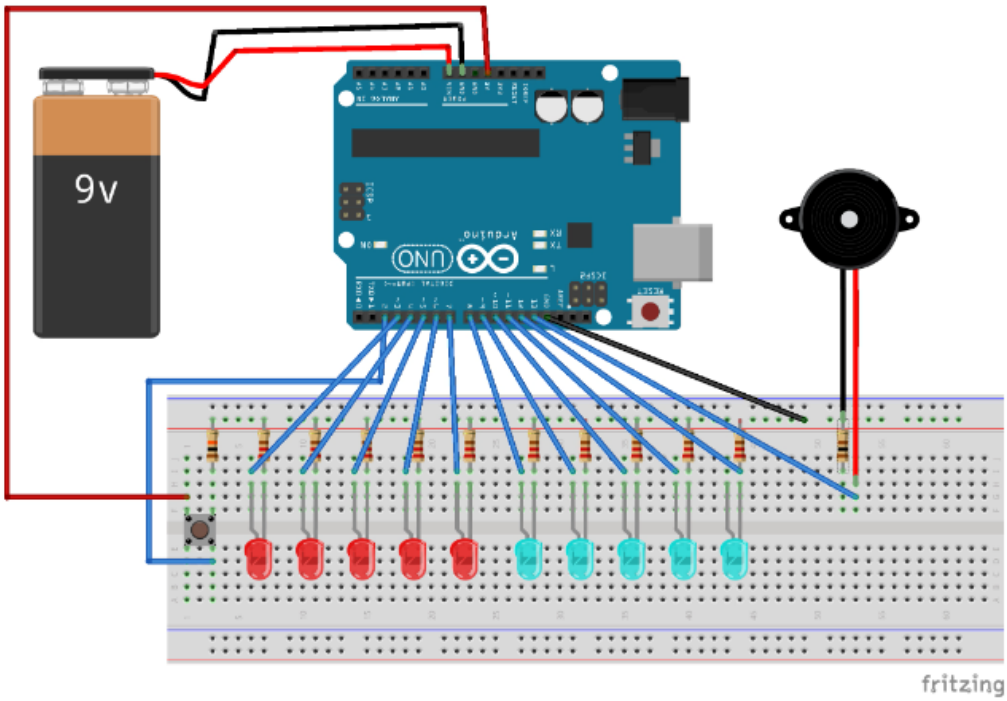

- 9v battery
    - Arduino
    - Buzzer
    - Resistor 100 ohm
    - Resistor 221 ohm
    - Resistor 10k ohm
    - Jumper wires
    - Breadboard
    - Led light (green and red)

# Wiring to an Arduino:

To wire the LED light and resistors to the Arduino simply connect the following as shown:

- negative terminal of all LED’s → 220 ohm resistor
- positive terminal of all red LED’s → pin 3 to pin 7
- positive terminal of all blue LED’s → pin 8 to pin 12
- negative rail of buzzer → 100 ohm resistor
- positive rail of buzzer → pin 13
- one out of four pins of pushButton → pin 2
- down of pushButton → GND rail using a pull down resistor of 10k ohm

- last button pin → 5v




```
boolean lastbutton = LOW ;
boolean currentbutton = LOW ;
int input =0;
int buzz =13; // Buzzer Pin
int j =3;
int k =12;
void setup () {
for ( int i =3; i <=12; i ++)
pinMode (i , OUTPUT );
pinMode (2 , INPUT );
}
boolean debounce ( boolean last ){
// Function to solve the problem of button debouncing
boolean current = digitalRead (2);
if ( last != current )
{
delay (5);
current = digitalRead (2);
}
return current ;
}
void loop () {
for ( int i =3; i <=12; i ++)
digitalWrite (i , LOW );
currentbutton = debounce ( lastbutton );
if ( lastbutton == LOW && currentbutton == HIGH )
{
```

```
input ++;
}
lastbutton = currentbutton ;
settone ( input );
}
void settone ( int input )
{
if ( input ==1)
one ();
else if ( input ==2)
oneA ();
else if ( input ==3)
two ();
else if ( input ==4)
twoA ();
else if ( input ==5)
three ();
else if ( input ==6)
threeA ();
else if ( input ==7)
four ();
}
void one () {
// This function produces the 1 st siren sound with ON / OFF led .
// Whoop up
for ( int hz = 440; hz < 1000; hz +=25){
tone ( buzz , hz , 50);
delay (5);
for ( int i =3; i <=7; i ++)
digitalWrite (i , HIGH );
}
// Whoop down
for ( int hz = 1000; hz > 440; hz -=25){
tone ( buzz , hz , 50);
delay (5);
for ( int i =3; i <=7; i ++)
{
digitalWrite (i , LOW );
digitalWrite ( i +5 , HIGH );
}
}
}
void oneA () {
// This function produces differnt transition on 1 st siren .
// Whoop up
for ( int hz = 440; hz < 1000; hz +=25){
tone ( buzz , hz , 50);
delay (5);
}
if (j >=3){
digitalWrite (j , HIGH );
j = j +1;
```

```
digitalWrite (k , HIGH );
k =k -1;
if ( j ==8)
j =3;
if ( k ==7)
k =12;
}
// Whoop down
for ( int hz = 1000; hz > 440; hz -=25){
tone ( buzz , hz , 50);
delay (5);
}
}
void two () {
// This function produces the 2 nd siren sound with progressive led .
// Whoop up
for ( int hz = 440; hz < 1000; hz +=25){
tone ( buzz , hz , 50);
delay (5);
}
loopF (3 ,12 ,20);
loopR (12 ,3 ,20);
// Whoop down
for ( int hz = 1000; hz > 440; hz -=25){
tone ( buzz , hz , 50);
delay (5);
}
}
void twoA () {
// This function produces differnt transition on 2 nd siren .
// Whoop up
for ( int hz = 440; hz < 1000; hz +=25){
tone ( buzz , hz , 50);
delay (5);
}
loopF (3 ,k ,20);
loopR (12 , j ,20);
k - -;
if ( k ==3)
k =12;
j ++;
if ( j ==12)
j =3;
// Whoop down
for ( int hz = 1000; hz > 440; hz -=25){
tone ( buzz , hz , 50);
delay (5);
}
}
void three () {
// This function produces the 3 rd siren ( AMBULANCE ) sound with led .
tone ( buzz ,440 ,200);
```

```
delay (300);
for ( int i =3; i <=6; i ++)
digitalWrite (i , HIGH );
noTone ( buzz );
tone ( buzz ,494 ,500);
delay (300);
for ( int i =3; i <=6; i ++){
digitalWrite (i , LOW );
digitalWrite ( i +6 , HIGH );
}
noTone ( buzz );
tone ( buzz ,523 ,300);
delay (200);
digitalWrite (7 , HIGH );
delay (50);
digitalWrite (8 , HIGH );
delay (50);
noTone ( buzz );
}
void threeA () {
// This function produces differnt transition on 3 rd siren .
tone ( buzz ,440 ,200);
delay (100);
loopF (5 ,10 ,20);
loopR (10 ,5 ,20);
noTone ( buzz );
for ( int i =3; i <=4; i ++){
digitalWrite (i , HIGH );
digitalWrite ( i +8 , HIGH );}
tone ( buzz ,494 ,500);
delay (300);
noTone ( buzz );
for ( int i =3; i <=4; i ++){
digitalWrite (i , LOW );
digitalWrite ( i +8 , LOW );
}
tone ( buzz ,523 ,300);
delay (300);
noTone ( buzz );
}
void four () {
// This function produces the 4 th siren ( POLICE ) sound with led .
for ( int i =3; i <=11; i +=2)
digitalWrite (i , HIGH );
for ( int hz = 440; hz < 1000; hz ++){
tone ( buzz , hz , 50);
delay (5);
}
for ( int i =3; i <=11; i +=2)
digitalWrite (i , LOW );
for ( int i =4; i <=12; i +=2)
digitalWrite (i , HIGH );
```

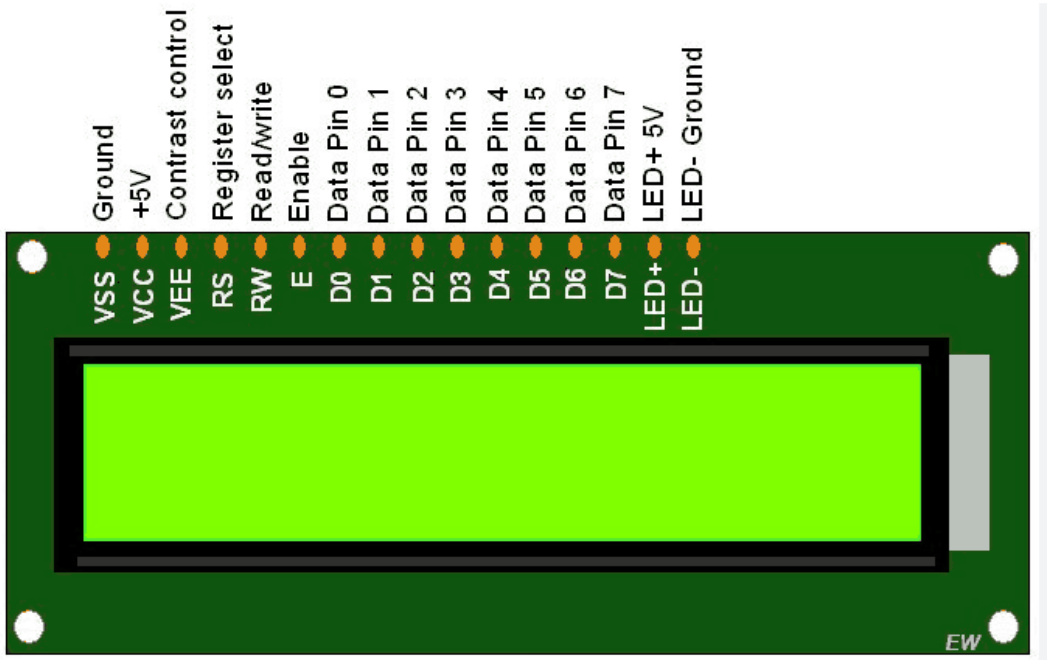
```
for ( int hz = 1000; hz > 440; hz - -){
tone ( buzz , hz , 50);
delay (5);
}
}
// SOME EXTRA FUNCTIONS OTHER THAN THE SIREN TONES
void loopF ( int spin , int epin , int dela ){
// loopF can blink the led in forward direction so spin must
// be lower than epin .
for ( int i = spin ;i <= epin ; i ++){
digitalWrite (i , HIGH );
delay ( dela );
low ();
if ( spin == epin ){
spin =3;
epin =12;}
}}
void loopR ( int epin , int spin , int dela ){
// loopR can blink the led in reverse / backward direction
// so epin must be lower than spin .
for ( int i = epin ;i >= spin ;i - -){
digitalWrite (i , HIGH );
delay ( dela );
low ();
if ( spin == epin ){
spin =3;
epin =12;}
}}
void low (){
// Used by loopF and loopR
for ( int i =3; i <=12; i ++)
digitalWrite (i , LOW );
}
```

### 5- I2C Liquid Crystal Displays using Arduino

we will learn how to work with LCD and printing simple texts using Arduino.

## Components and supplies:

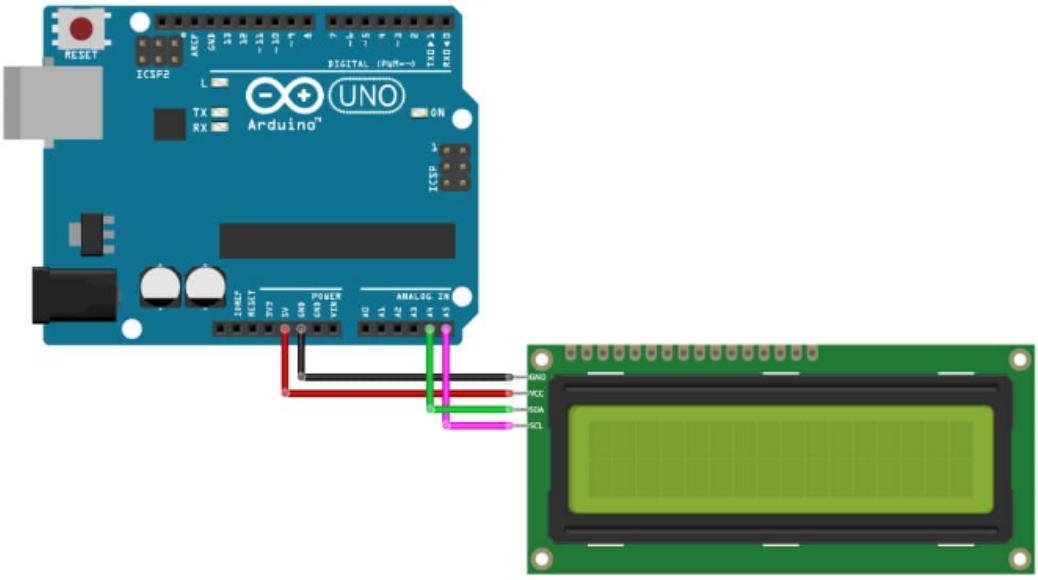
- Arduino UNO
- Male/Female Jumper Wires
- I2C 16x2 Arduino LCD Display Module




- 

# Wiring to an Arduino:

Please follow the diagram




```
# include LiquidCrystal_I2C . h
# include Wire . h
// initialize the liquid crystal library
// the first parameter is the I2C address
// the second parameter is how many rows are on your screen
```

```
// the third parameter is how many columns are on your screen
LiquidCrystal_I2C lcd (0 x27 , 16 , 2);
void setup () {
// initialize lcd screen
lcd . init ();
// turn on the backlight
lcd . backlight ();
}
void loop () {
// wait for a second
delay (1000)
// tell the screen to write on the top row
lcd . setCursor (0 ,0);
// tell the screen to write
hello , f r o m on the top row
lcd . print ( Hello , F r o m );
// tell the screen to write on the bottom row
lcd . setCursor (0 ,1);
// tell the screen to write
Arduino_uno_guy
on the bottom row
// you can change whats in the quotes to be what you want
it to be !
lcd . print (
Arduino_uno_guy
);
}
```

### 6- Blinking LED using Arduino and Push button switch

we will learn how to turn on and turn off the LED light using Push button switch.

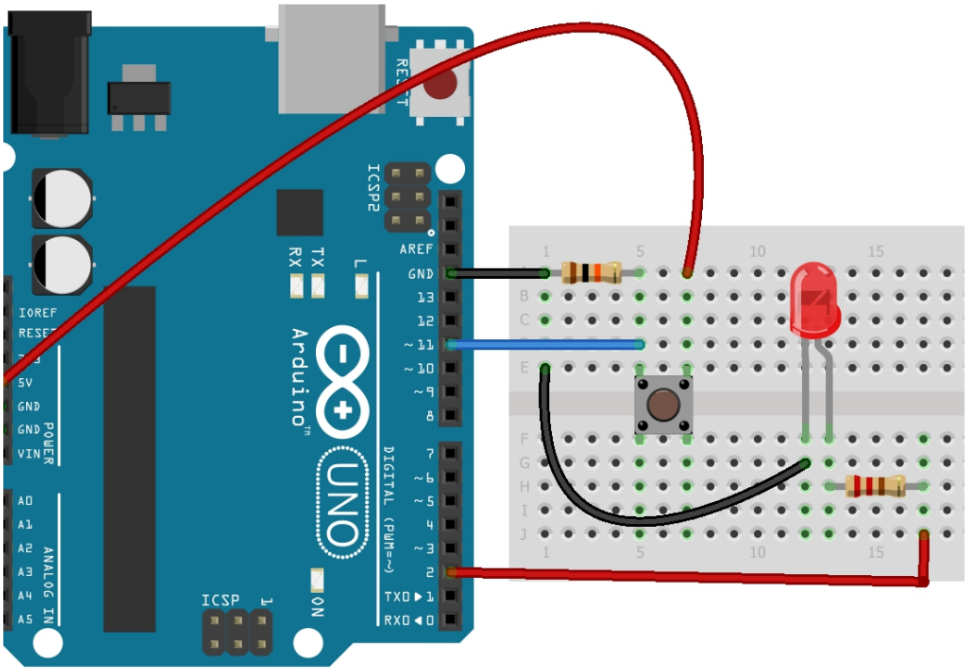
## Components and supplies:

- Arduino UNO

- Pushbutton switch
- Resistor 10k ohm
- Breadboard
- Jumper wires
- LED Lights

## Wiring to an Arduino:

Please follow the diagram.




```python int led1 =2; int led2 =3; int led3 =5; int button =6;
```

- -- setup () { pinMode ( button , --- ); pinMode ( led1 , --- ); pinMode ( led2 , --- ); pinMode ( led3 , --- ); } void loop () { int ButtonStatus = --- ( button ); if ( ButtonStatus == HIGH ) { digitalWrite ( led1 , --- ); digitalWrite ( --- , HIGH ); digitalWrite ( --- , HIGH ); } else { digitalWrite ( led1 , LOW ); digitalWrite ( led2 , ---); digitalWrite ( led3 , LOW ); } }

```
# 7- Color Detection Using TCS3200/230 sensor
We will learn how the color detection sensor works with Arduino in terms of detecting different colors.
# Components and supplies
```

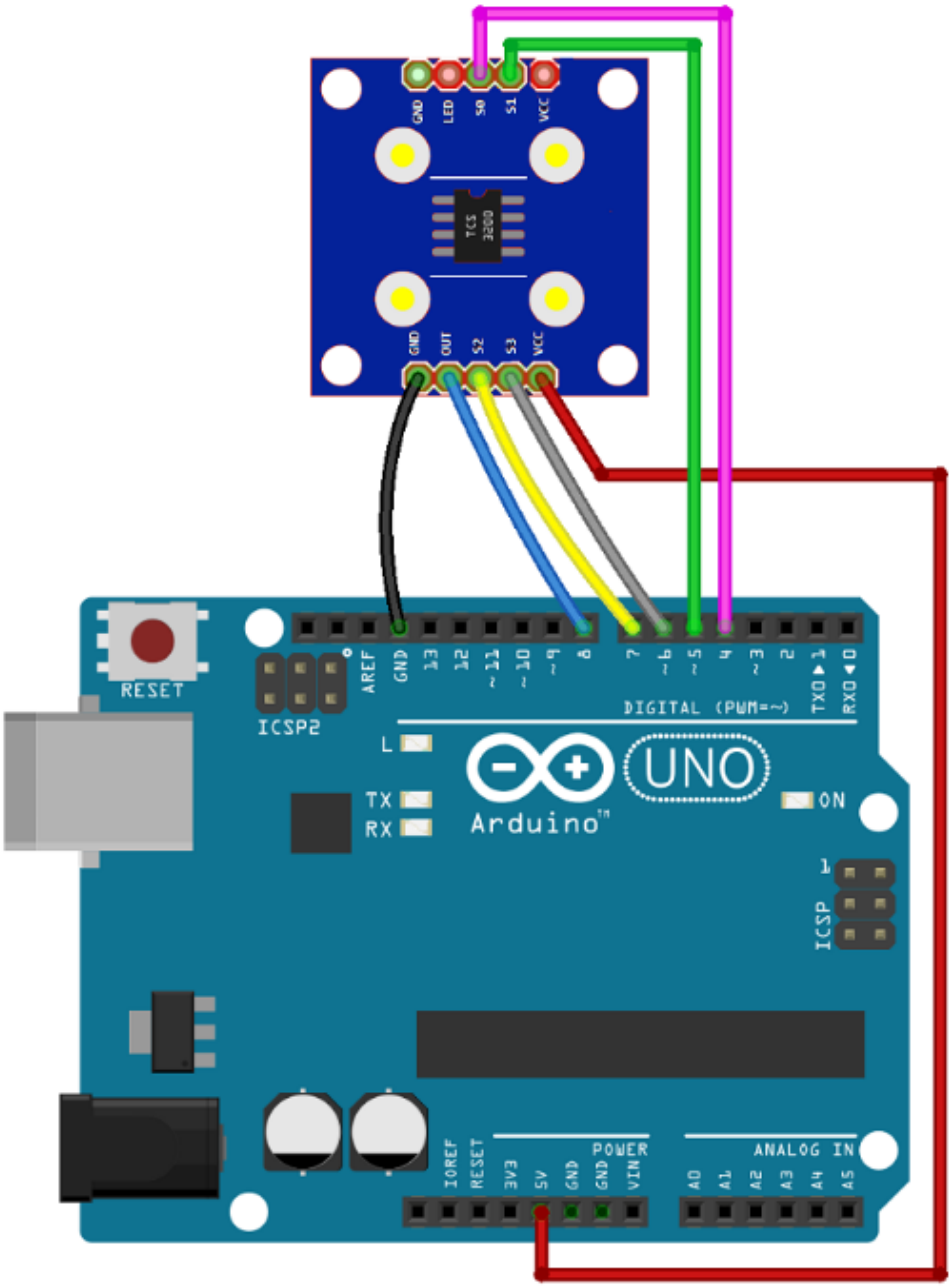
- Arduino


Color Detection module

- TCS3200/TCS230
- RGB Diffused Common Cathode
- Jumper wires

## Wiring to an Arduino:

Please follow the diagram




```
# define s0 8
# define s1 9
# define s2 10
# define s3 11
# define out 12
int data =0;
void setup ()
{
pinMode ( s0 , OUTPUT );
pinMode ( s1 , ---);
```

```
pinMode ( s2 , OUTPUT );
pinMode ( s3 , OUTPUT );
pinMode ( out , --- );
s
Serial . begin (9600);
digitalWrite ( s0 , HIGH );
digitalWrite ( s1 , HIGH );
}
void loop ()
{
digitalWrite ( s2 , LOW );
digitalWrite ( s3 , LOW );
Serial . print ( " Red value = " );
GetData ();
--- ( s2 , LOW );
--- ( s3 , HIGH );
Serial . print ( " Blue value = " );
GetData ();
--- ( s2 , HIGH );
digitalWrite ( s3 , HIGH );
Serial . print ( " Green value = " );
--- ();
Serial . println ();
delay (2000);
}
void GetData (){
data = pulseIn ( out , LOW );
Serial . print ( data );
Serial . print ( " \ t " );
delay (20);
}
```

## License

MIT

**Free Software, Hell Yeah!**