

# How to create a Smoke Detector using Arduino in Tinkercad

by Monisha Macharla | Jul 5, 2022 | Arduino Uno, arduino uno projects

## Introduction

In this project, we will create a smoke detector setup using Arduino in the Tinkercad platform. The setup is simple and can be made by anyone. Also, the components we use are cheap and the hardware setup can be used in our homes, workplaces, etc. So let's get started with the project and see its simulation.

You can visit the Tinkercad platform by clicking [here](#). If you're unfamiliar with the Tinkercad platform and want to know more about it, you can click [here](#).

## Components Required

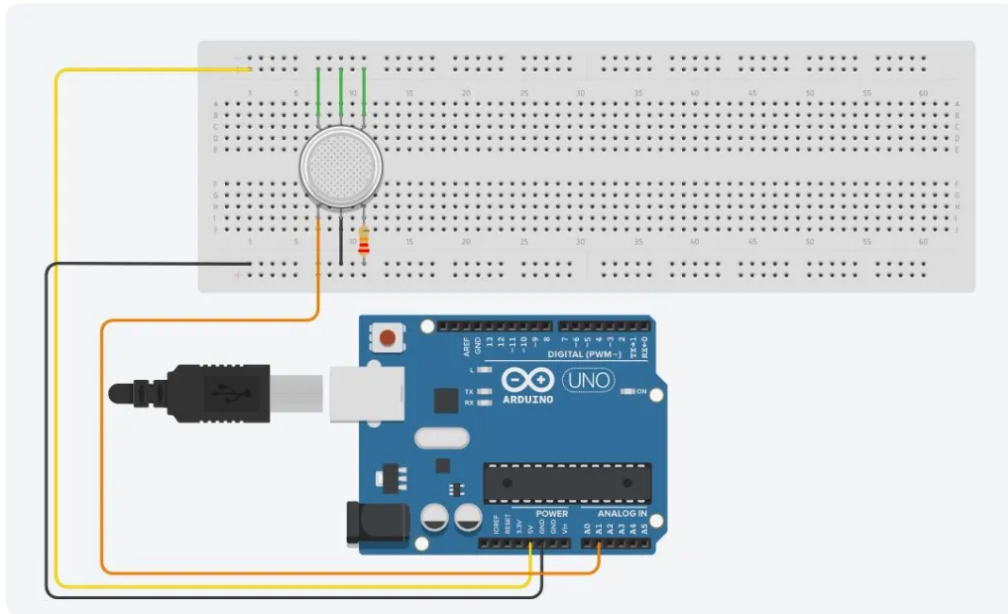
- Arduino UNO R3
- Gas sensor
- Piezo sensor
- RGB LED
- 220Ω resistors (3)
- 1kΩ resistor
- 22kΩ resistor
- Breadboard

## Setting up the Smoke Detector

First and foremost, drag all the above-mentioned components into the Tinkercad workspace.

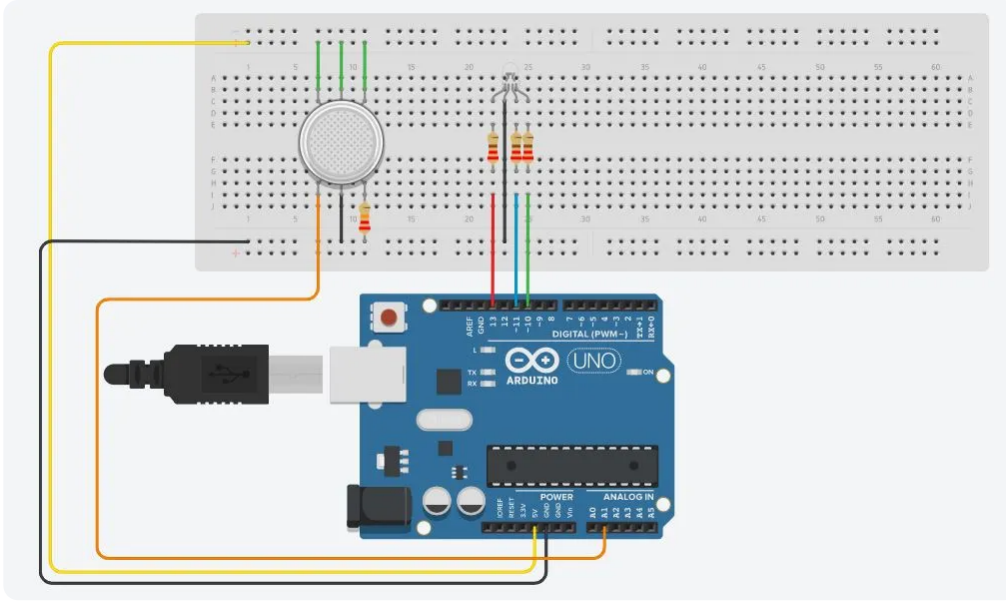
Subsequently, let's start connecting the components one by one. So, let's start by looking into the gas sensor connections. The gas sensor provided in Tinkercad has 6 terminals, namely **A1**, **H1**, **A2**, **B1**, **H2**, and **B2**. Connect the terminals A1, H1, and A2 to the power supply.

After that, connect terminal B1 to the power supply and the terminals H2 and B2 to the ground. In order to increase the accuracy and sensitivity of the gas sensor, we connect a **22k $\Omega$  load resistor** between the terminal B2 and the ground.



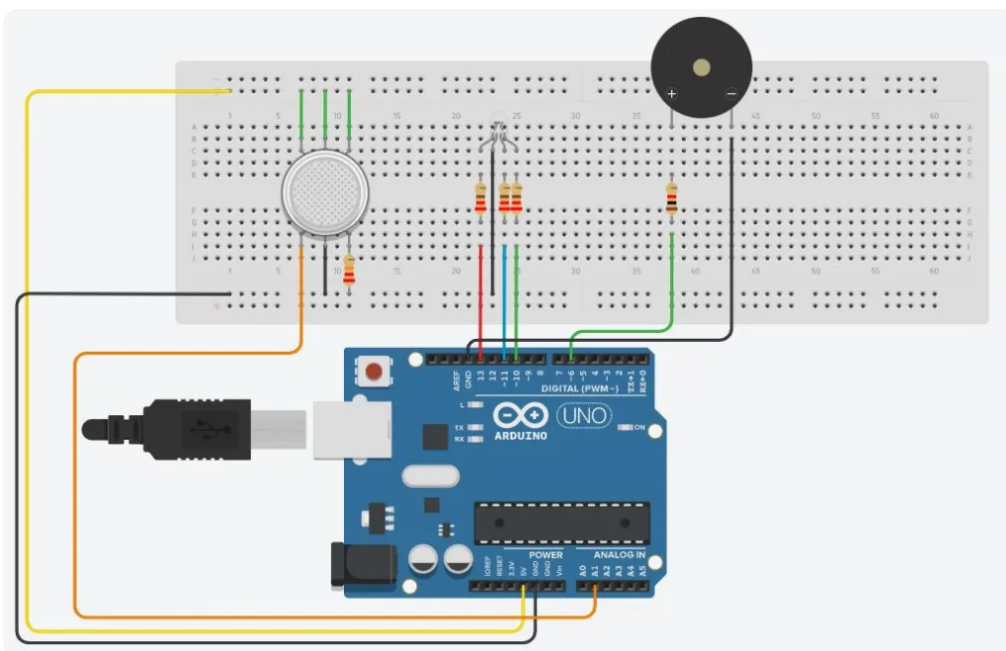
Gas sensor setup

Subsequently, let's connect the RGB LED which acts as an indicator for the smoke detector setup. The RGB LED has 4 terminals, namely **RED, CATHODE, BLUE, AND GREEN**. Connect the cathode terminal to the ground. Next, connect the terminals red, blue, and green to the digital pins 13, 11, and 10 respectively (we have chosen these pins for this project, any of the digital pins can be used for the connection). We use the 1k $\Omega$  resistors in the connection between the terminals and the digital pins. This is done to limit the current through the LED. As a result, the setup would look like this



RGB LED connections

Following this, we connect the piezo sensor to our setup. This is done so as to make a buzzing noise when smoke is detected. The piezo sensor has two terminals namely **POSITIVE(+)** and **NEGATIVE(-)**. Connect the positive terminal to pin number 6 and the negative terminal to the ground. We need to limit the current flow to the sensor. Hence, connect the  $1k\Omega$  resistor in between the connection from the positive terminal to the digital pin. Therefore, the final circuit would look as follows



Final circuit

## Coding the Smoke Detector

So now that the smoke detector circuit is complete, let's dive into the

coding part. The complete code is as follows

```
1 // C++ code
2 //
3 void setup()
4 {
5     Serial.begin(9600);
6     pinMode(A1,INPUT);
7     pinMode(6,OUTPUT);
8     pinMode(10,OUTPUT);
9     pinMode(11,OUTPUT);
10    pinMode(13,OUTPUT);
11 }
12
13 void loop()
14 {
15     int sensorValue = analogRead(A1);
16     Serial.print("Gas sensor reading = ");
17     Serial.println(sensorValue);
18     if(sensorValue > 800){
19         digitalWrite(13,HIGH);
20         digitalWrite(11,LOW);
21         digitalWrite(10,LOW);
22         digitalWrite(6,HIGH);
23     }
24     else{
25         digitalWrite(13,LOW);
26         digitalWrite(11,LOW);
27         digitalWrite(10,HIGH);
28         digitalWrite(6,LOW);
29     }
30 }
```

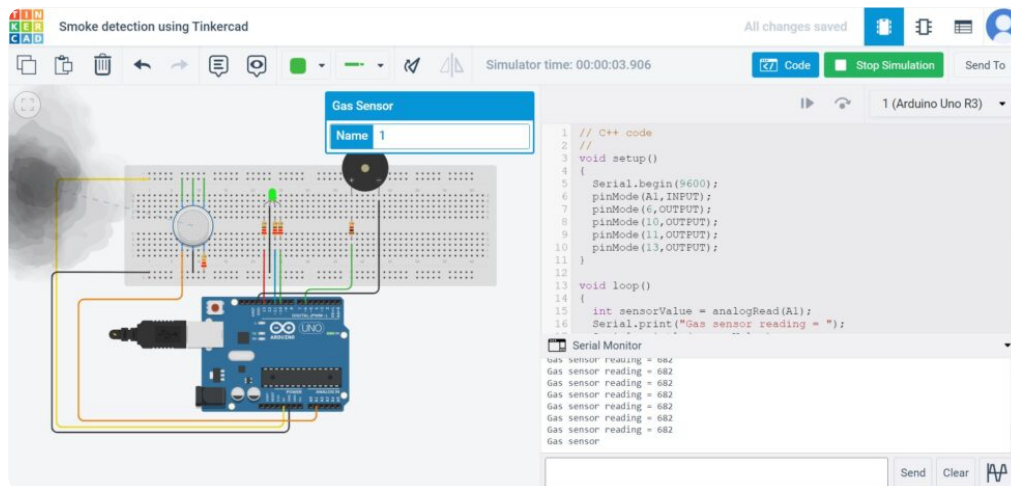
```
1 // C++ code
2 //
3 void setup()
4 {
5     Serial.begin(9600);
6     pinMode(A1,INPUT);
7     pinMode(6,OUTPUT);
8     pinMode(10,OUTPUT);
9     pinMode(11,OUTPUT);
10    pinMode(13,OUTPUT);
11 }
12
13 void loop()
14 {
15     int sensorValue = analogRead(A1);
16     Serial.print("Gas sensor reading = ");
17     Serial.println(sensorValue);
18     if(sensorValue > 800){
19         digitalWrite(13,HIGH);
20         digitalWrite(11,LOW);
21         digitalWrite(10,LOW);
22         digitalWrite(6,HIGH);
23     }
24     else{
25         digitalWrite(13,LOW);
26         digitalWrite(11,LOW);
27         digitalWrite(10,HIGH);
28         digitalWrite(6,LOW);
29     }
30 }
```

## Code for the smoke detector

Firstly, within the **setup()** function, we use **Serial.begin()** function in order to establish a serial communication. We set pin A1 as the input, and pins 6, 10, 11, and 13 as the output using the **pinMode()** function.

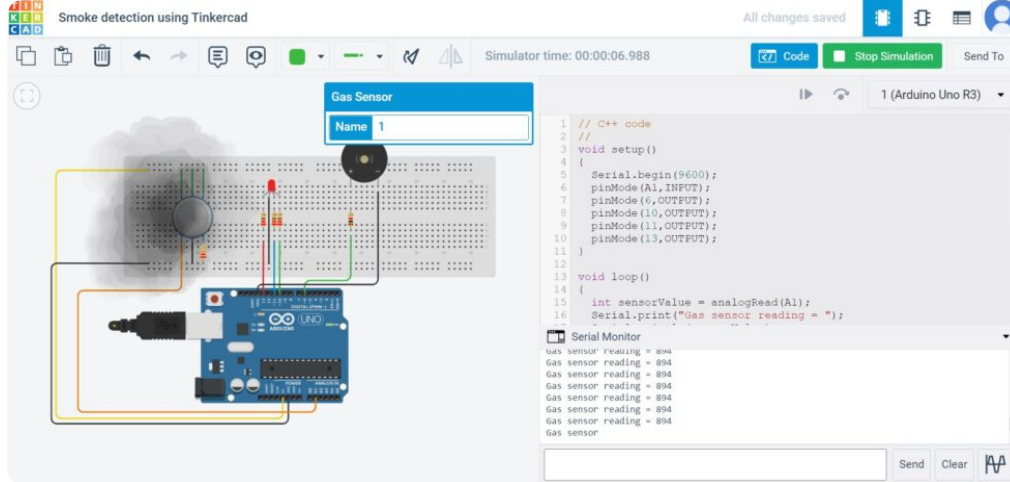
Subsequently, we start coding the **loop()** function. We read the smoke sensor readings from pin A1 using the **analogRead()** function. Then we assign the values to an integer type variable named “sensorValue”. Next, we print the gas sensor readings to the serial monitor using the **Serial.print()** function. For this project, we have set the threshold value for gas sensor readings as 800. So, if the readings are over the threshold value, the LED turns **RED** and the piezo sensor starts **buzzing**. If the readings are below the threshold value, the LED remains **GREEN** and the piezo sensor doesn't make any buzzing noise. We accomplish this by using the **if-else loop** and setting the pins as HIGH/LOW using the **digitalWrite()** functions.

So now, the coding part is complete. Further, let's simulate the circuit and see the results.



When gas sensor values are below the threshold value

Here, we can see that when the gas sensor readings are below the threshold value, the LED remains GREEN and the piezo sensor doesn't make any buzzing noise. The serial monitor continuously displays the gas sensor readings on a real-time basis.



When gas sensor values are above the threshold value

Here, we can see that when the readings are above the threshold value, the LED turns RED and the piezo sensor starts buzzing. The serial monitor value updates on moving the smoke nearer to the gas sensor.

## Conclusion

So with this, we have completed our smoke detector setup using the Tinkercad platform. Hope that this project was informative as well as interesting.

Happy Learning!

## Categories

Applications of IoT (43)

Arduino Quiz (3)

Arduino Uno (17)

arduino uno projects (13)

Articles (63)

Communication Protocols (18)

Development Board (7)

IoT Basics (6)

IoT Cloud (26)

IoT Connectivity (26)

IoT Continuous Integration (21)

IoT Programming (20)

IoT Security (9)

IoT\_Basics\_Advanced\_Quiz (3)

MQTT (20)

Node-Red (6)

NodeMCU (7)

nodemcu projects (6)

NodeMCU\_Quiz (3)

Projects (27)

Python (26)

Python Quiz (3)

Python Tkinter (6)

Python Tkinter Examples (5)

Raspberry Pi (31)

Raspberry Pi projects (23)

Raspberry Pi Quiz (4)

Sensors (12)

Uncategorized (2)

## Recent Posts

Alternatives to Docker for Containerization and Deployment January 24, 2023

Getting started with ReactJS and MQTT – IoT Dashboard December 15, 2022

IoT Smart Gardening System – ESP8266, MQTT, Adafruit IO December 13, 2022

MCP3008 with ESP8266 for Analog Moisture Sensors – SPI December 7, 2022

NodeMCU and RGB LED Strip with Adafruit IO – Arduino IDE December 4, 2022

How to control NEMA Stepper Motor with Arduino and MicroStep Driver October 24, 2022

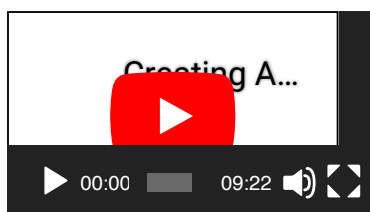
How to push a Docker Image to the Docker Hub using Jenkins Pipeline – CI CD August 16, 2022

What is Edge Intelligence: Architecture and Use Cases August 14, 2022

Getting Started with Bash Script : A Simple Guide August 14, 2022

How to Extract REST API Data using Python August 14, 2022

## Creating a multiplication Skill in Alexa using python



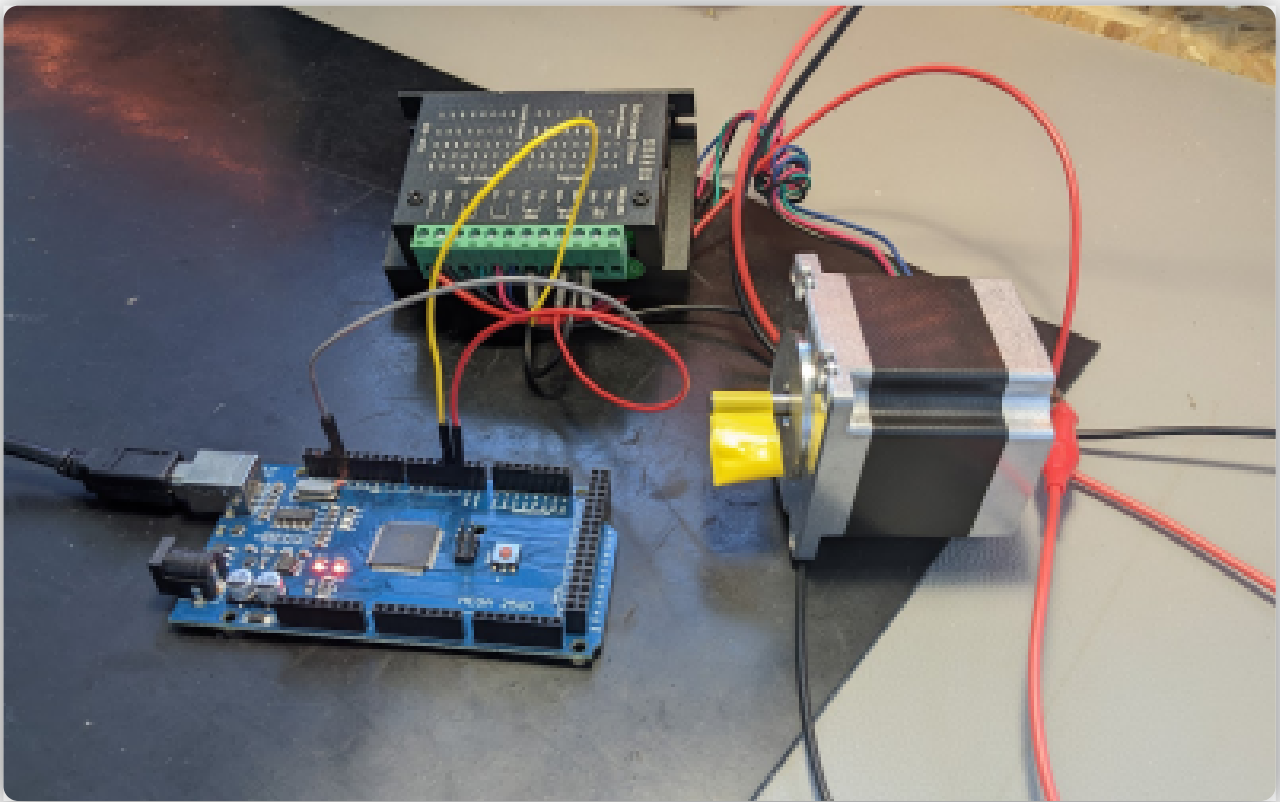


### **Written By Monisha Macharla**

Hi, I'm Monisha. I am a tech blogger and a hobbyist. I am eager to learn and explore tech related stuff! also, I wanted to deliver you the same as much as the simpler way with more informative content. I generally appreciate learning by doing, rather than only learning. Thank you for reading my blog! Happy learning!

## **RELATED POSTS**





## How to control NEMA Stepper Motor with Arduino and MicroStep Driver

Oct 24, 2022 | [Arduino Uno](#), [arduino uno projects](#)

# *Alternatives to Arduino IDE*

## Top 10 Alternatives to Arduino IDE that you must know

Nov 6, 2020 | [Arduino Uno](#)



## Simple Android App to control Devices On Arduino -Webserver using MIT App Inventor

Jun 20, 2020 | [arduino uno projects](#)



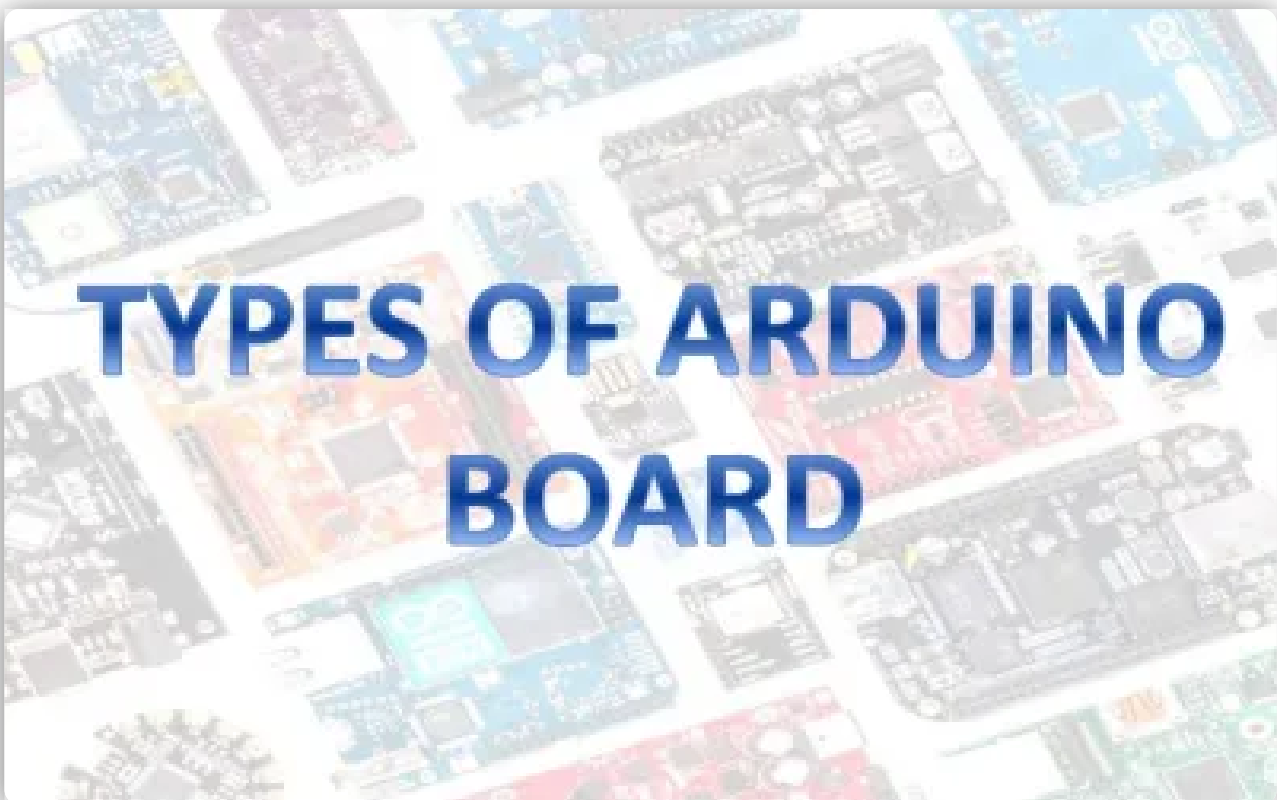
## Installing DHT11/ DHT22 libraries for Arduino/ NodeMCU

Jun 13, 2020 | [Arduino Uno](#), [NodeMCU](#)



## The Ultimate Guide to Arduino Simulation using Tinkercad

Jul 5, 2022 | [Arduino Uno](#), [arduino uno projects](#)



## Arduino and its various development boards

Oct 4, 2020 | [Arduino Uno](#)

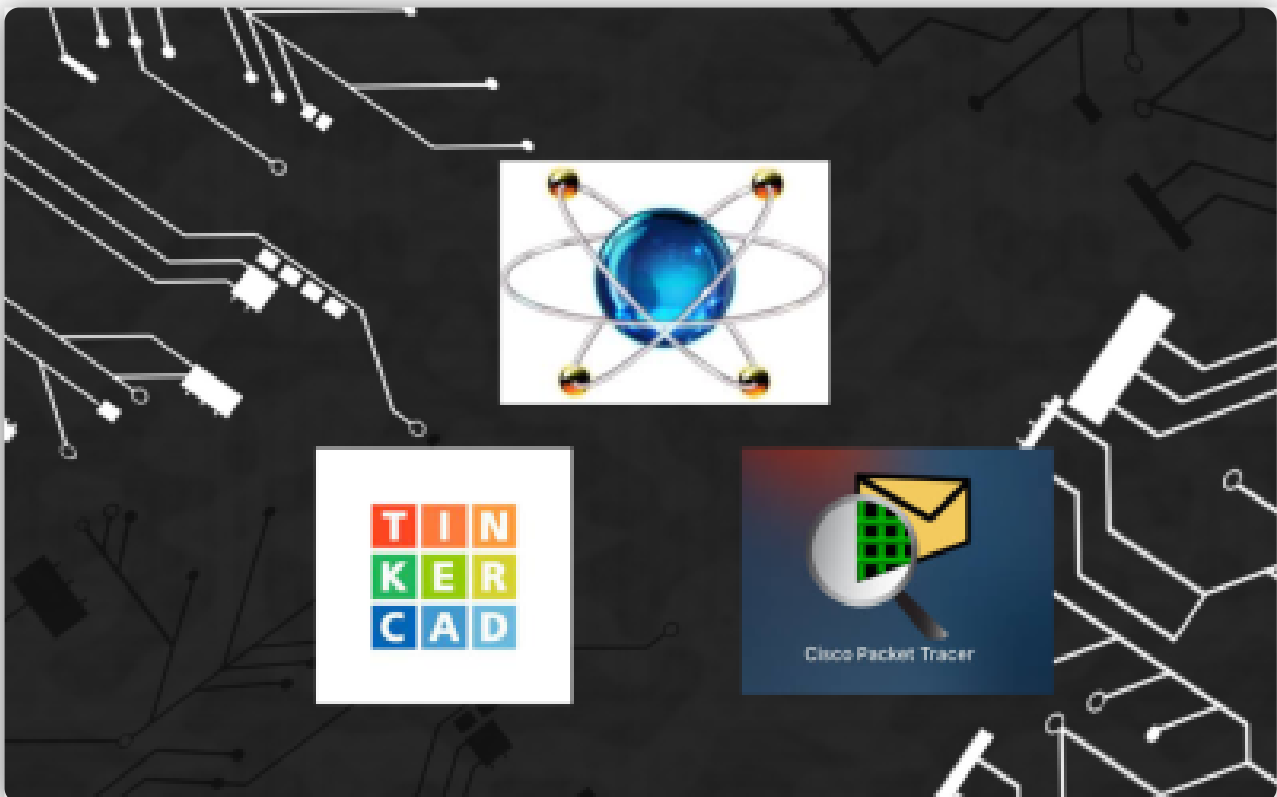


## Arduino Compatible Temperature Sensors

Jun 15, 2020 | [Arduino Uno](#), [Sensors](#)

## What is UART protocol and how to setup it with Arduino?

May 25, 2020 | [Arduino Uno](#), [IoT Connectivity](#)



## How to use simulators to build Arduino Projects

Jun 2, 2021 | [Arduino Uno](#)



MIT app inventor with Arduino

Jul 1, 2020 | [Arduino Uno](#)



ESP8266 OTA (Over the Air) Updates With Arduino IDE

Jun 13, 2020 | [Arduino Uno](#), [NodeMCU](#)

ARDUINO MKR 1000

May 25, 2020 | [Arduino Uno](#)

## VIDEOS – FOLLOW US ON YOUTUBE

Smart Gardening System - Intern



Creating Amazon Alexa Skills usi



Air Quality Monitoring using Node




Internet of Things Project - Impro



## EXPLORE OUR IOT PROJECTS

[CLICK HERE](#)

## Translate

Select Language 

Powered by  Translate

## About Us

IoTEDU is committed to writing blogs and tutorials on IoT, from basic to advanced topics to make the learners understand easily. IoTEDU is considered a one-stop for blogs, tutorials, projects, the latest software, and hardware update for the learners to motivate them to learn more and more to enrich their knowledge. Started in 2019, we proudly say that we achieved a place in the IoT's learners community.

## Follow Us



[Home Page](#)

[IoT Blog](#)

[IoTEDU Innovation Lab](#)

[IoTEDU Training Academy](#)

