

# IoT Workshop

## Installation Instructions

### Installing Arduino IDE

GO TO GOOGLE AND SEARCH FOR :

“ARDUINO IDE FOR WINDOWS”

The screenshot shows a Google search interface with the query "Ardino IDE for Windows". The search results list several links from the Arduino website, including "Arduino IDE 1 Installation (Windows)", "Software | Arduino", and "Download and install Arduino IDE". A knowledge panel on the right provides a summary of the Arduino IDE, describing it as an integrated development environment for Arduino hardware, featuring a text editor, message area, text console, and toolbar. The panel also mentions the programming language C++ and includes a link to the Arduino IDE v1 download page.

Google search results for "Ardino IDE for Windows".

Search results include:

- Arduino IDE 1 Installation (Windows)** (2 Jan 2024) — This document explains how to install the **Arduino Software (IDE)** on **Windows** machines. Download the **Arduino Software (IDE)**.
- Software | Arduino** (https://www.arduino.cc > software) — Program using IEC 61131-3 languages and mix **Arduino** sketches through **Arduino PLC IDE!** ... **Windows Arduino PLC IDE**, Win 10 and newer, 64 bits **Windows Arduino PLC** ... [Arduino PLC IDE](#) · [About Arduino](#) · [Donate to Arduino](#) · [GO BACK](#)
- Download and install Arduino IDE** (https://support.arduino.cc > en-us > articles > 3600198...) — 8 days ago — View all download options on the Software page. If you need help, **Arduino IDE 2** runs on the following systems: **Windows**: Win 10 (64-bit) or newer ...

People also ask :

**Arduino IDE** Software :

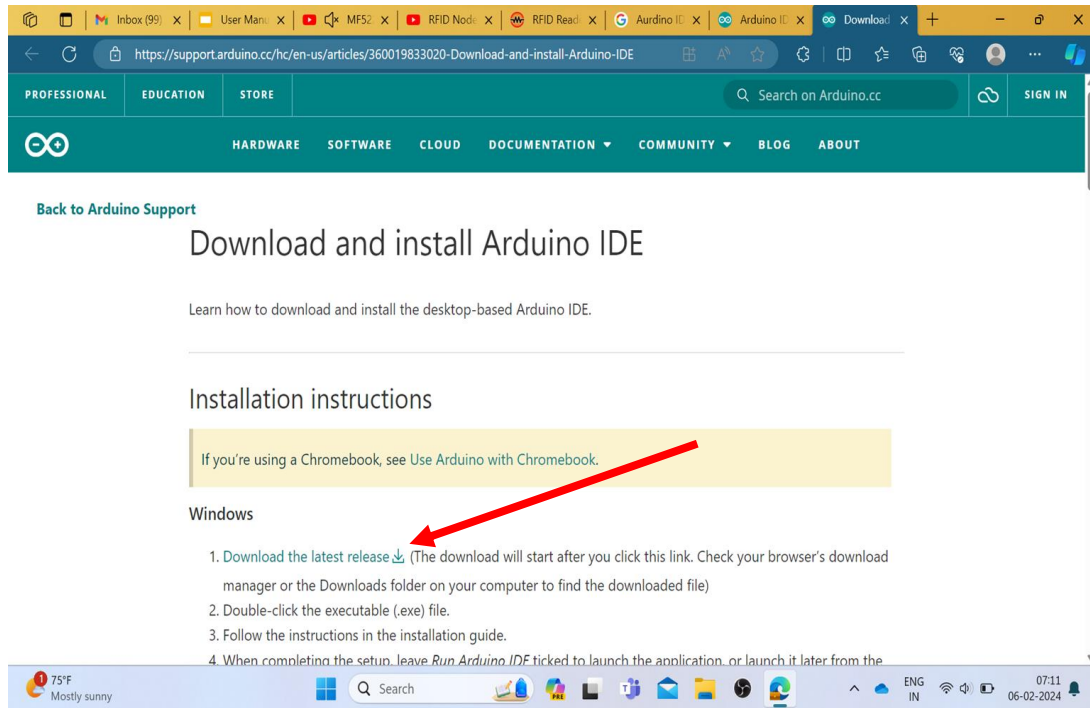
The **Arduino Integrated Development Environment** - or **Arduino Software (IDE)** - contains a **text editor for writing code**, a **message area**, a **text console**, a **toolbar with buttons for common functions** and a **series of menus**. It connects to the **Arduino hardware** to upload programs and communicate with them.

2 Jan 2024

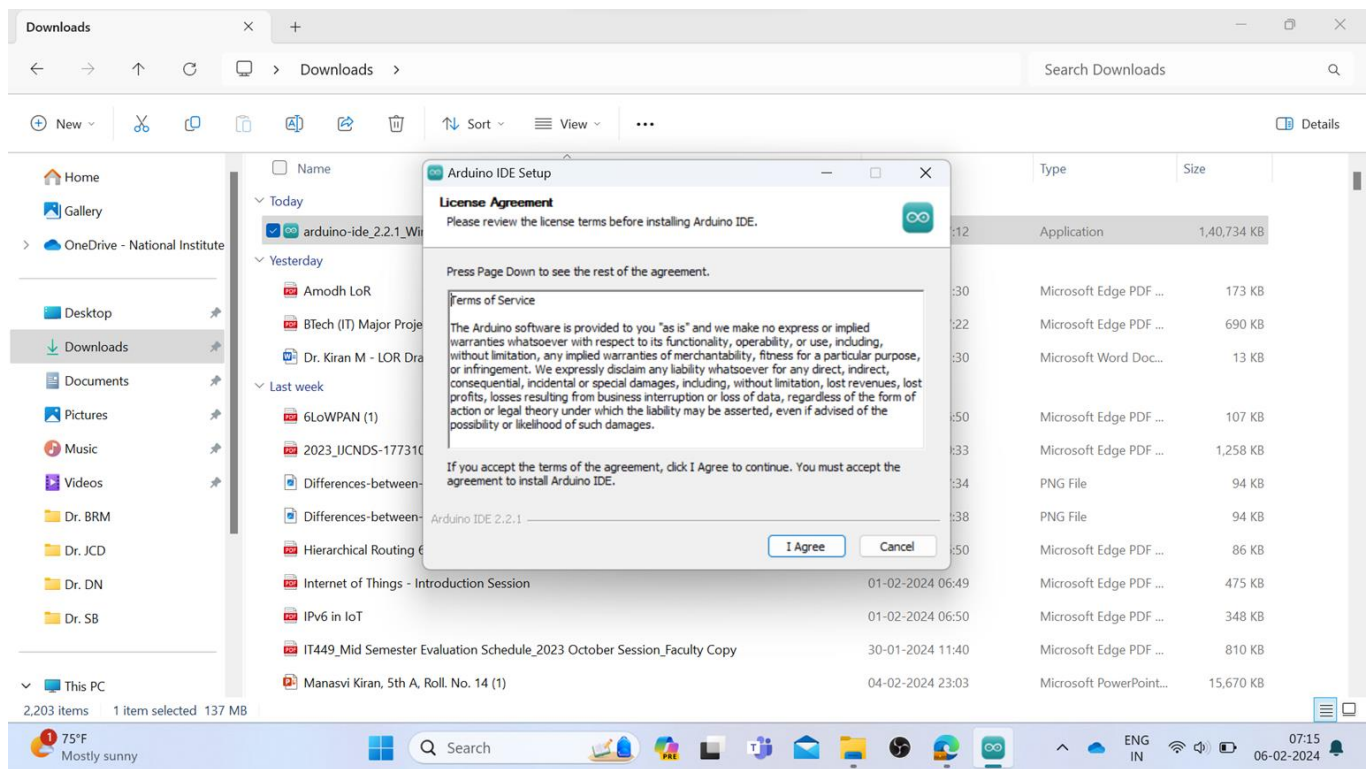
[arduino.cc](#) [https://docs.arduino.cc > software > ide-v1 > tutorials > a...](#)

**Arduino Integrated Development Environment (IDE) v1**

**Programming language:** C++



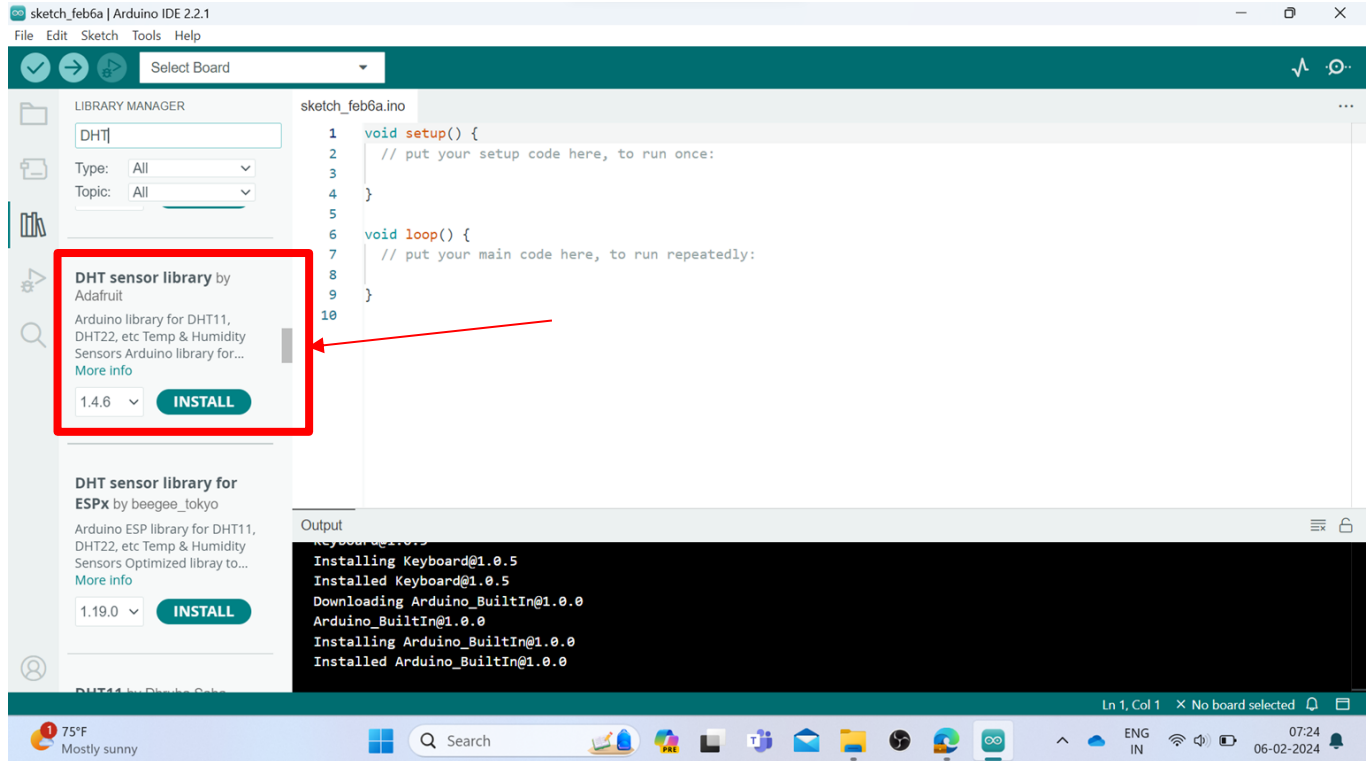
- arduino-ide\_2.2.1\_Windows\_64bit Will get downloaded.
- Double click on the file to install the Aurdino IDE
- Choose the Default options and install. It will take approximately one min to install



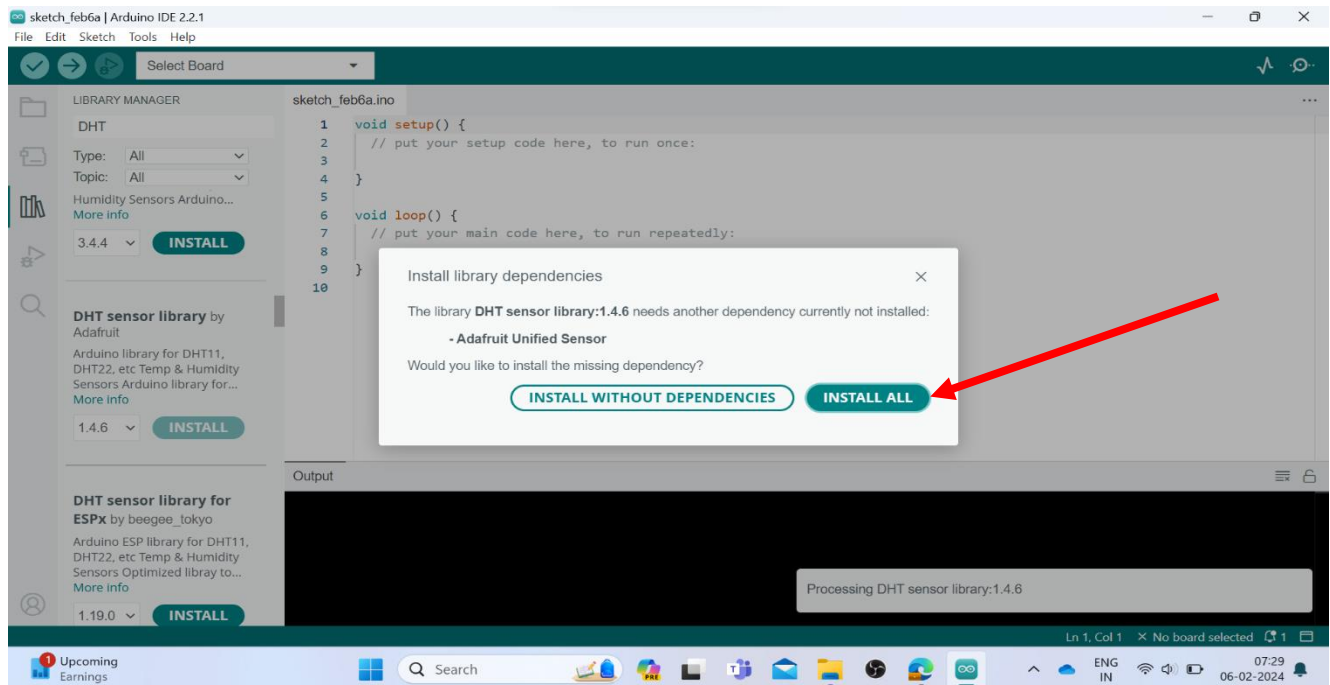
- During the installation process it will ask for the libraries to be installed. "Allow" it to install. And choose "YES" to make changes in the system.

# Installation of Required Libraries

## DHT11 Library



1. Open Arduino IDE and go to Sketch > Include Library > Manage Libraries.
2. The Library Manager should open. Search for “DHT” on the Search box and install the DHT library from Adafruit.



## ESP8266 Installation Procedure

In your Arduino IDE, go to File> Preferences.

Enter [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

into the “Additional Boards Manager URLs” field. Then, click the “OK” button:

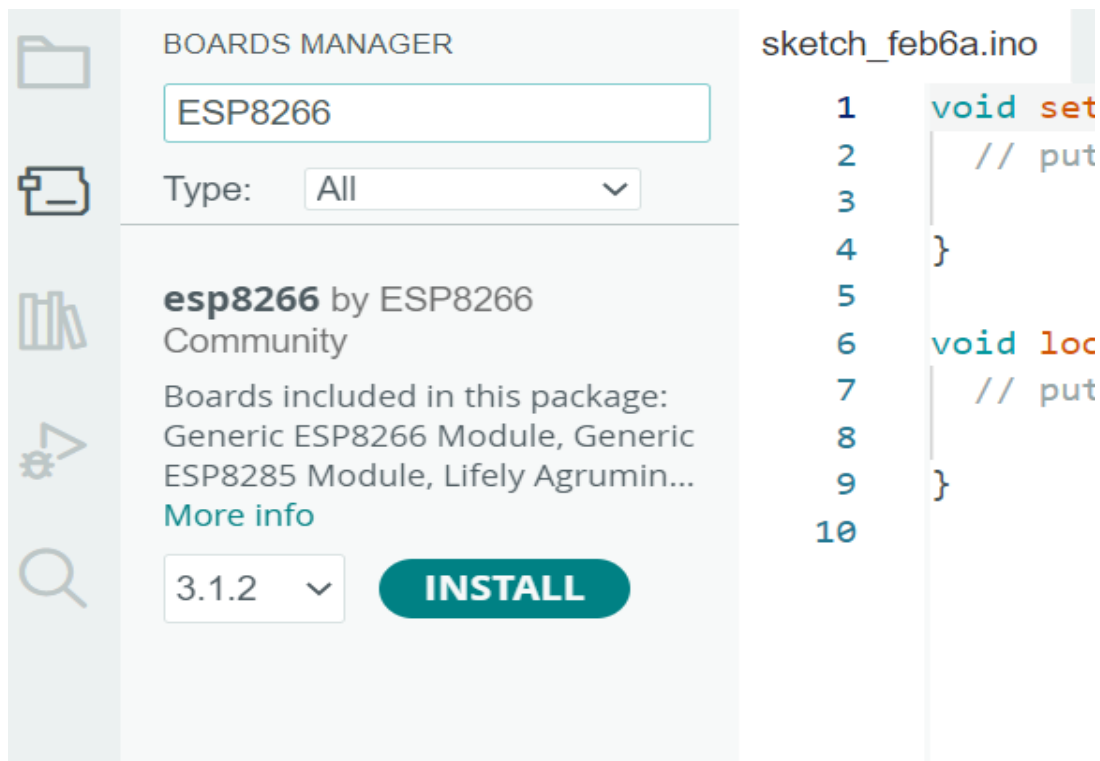
Open the Boards Manager.

Go to

**Tools > Board > Boards Manager**

**Search for ESP8266 and press install button for the**

**“ESP8266 by ESP8266 Community“:**



## Driver Installation

Driver cp210x:

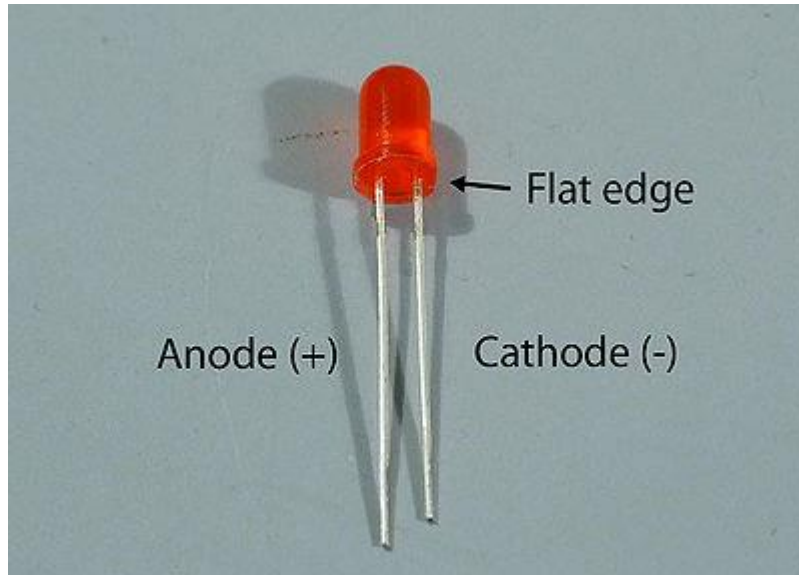
<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>

Extract and install cp210 windows driver

Driver cp340:

<https://sparks.gogo.co.nz/ch340.html>

## BLINK LED



CODE:

```
#define LED D0          // Led in NodeMCU at pin GPIO16 (D0).

void setup() {

  pinMode(LED, OUTPUT); // LED pin as output.

}

void loop() {

  digitalWrite(LED, HIGH); // turn the LED off. (Note that LOW is the voltage level but actually
                           // the LED is on; this is because it is active low on the ESP8266.

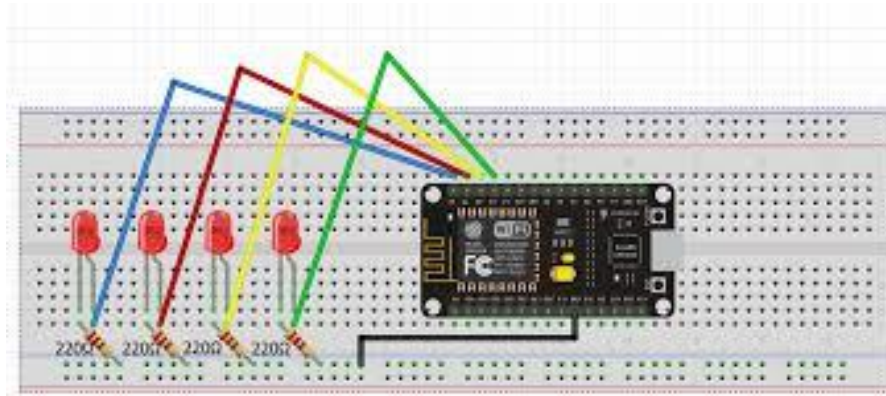
  delay(1000);           // wait for 1 second.

  digitalWrite(LED, LOW); // turn the LED on.

  delay(1000); // wait for 1 second.

}
```

## BLINKING TRAFFIC LED



CODE:

```
#define LED1 D0

#define LED2 D1

#define LED3 D2

void setup() {

pinMode(LED1, OUTPUT);

pinMode(LED2, OUTPUT);

pinMode(LED3, OUTPUT);  // LED pin as output.

}

void loop() {

digitalWrite(LED1, HIGH); // turn the LED off.(Note that LOW is the voltage level but actually

                           //the LED is on; this is because it is active low on the ESP8266.

delay(1000);

digitalWrite(LED1, LOW); // turn the LED on.

delay(1000);

digitalWrite(LED2, HIGH); // turn the LED off.(Note that LOW is the voltage level but actually

                           //the LED is on; this is because it is active low on the ESP8266.

delay(1000);
```



```
digitalWrite(LED2, LOW); // turn the LED on.

delay(1000);

digitalWrite(LED3, HIGH); // turn the LED off. (Note that LOW is the voltage level but actually

//the LED is on; this is because it is active low on the ESP8266.

delay(1000);

// wait for 1 second.

digitalWrite(LED3, LOW); // turn the LED on.

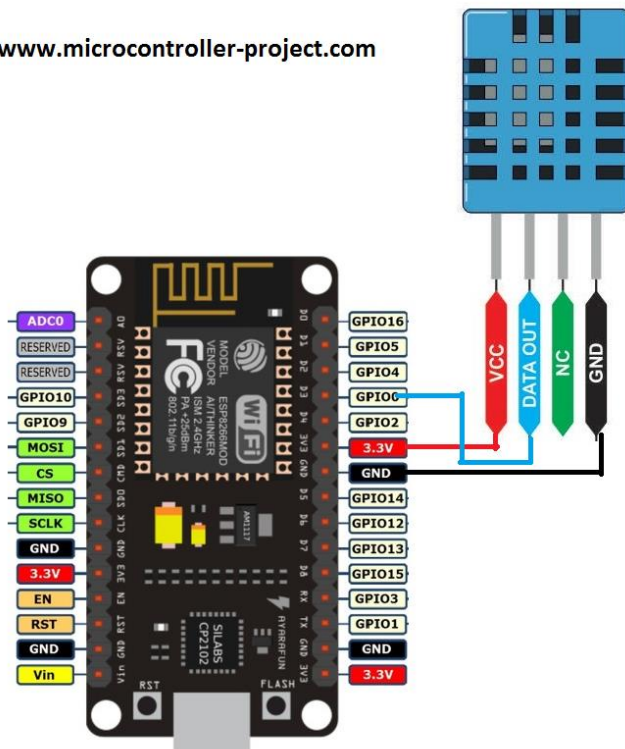
delay(1000); // wait for 1 second.

}
```

## DHT11

For CH340

[www.microcontroller-project.com](http://www.microcontroller-project.com)

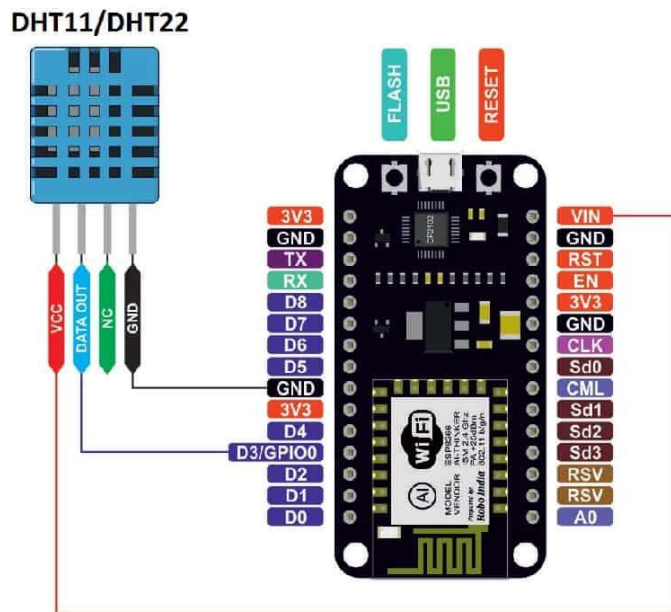


VCC (+) to 3V

GND (-) to GND

DATA (OUT) to D3

For cp210



VCC (+) to VIN

GND (-) to GND

DATA (OUT) to D3

CODE:

```
#include "DHT.h"      // including the library of DHT11 temperature and humidity sensor
```

```
#define DHTTYPE DHT11 // DHT 11
```

```
#define dht_dpin 0
```

```
DHT dht(dht_dpin, DHTTYPE);
```

```
void setup(void)
```

```
{
```

```
  dht.begin();
```

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



```

Serial.println("Humidity and temperature\n\n");

delay(700);

}

void loop() {

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    Serial.print("Current humidity = ");

    Serial.print(h);

    Serial.print("% ");

    Serial.print("temperature = ");

    Serial.print(t);

    Serial.println("C ");

    delay(800);

}

```

## DHT11 TO THINKSPEAK CODE:

```

#include "DHT.h"    // including the library of DHT11 temperature and humidity sensor

#define DHTTYPE DHT11 // DHT 11

#define dht_dpin 0

DHT dht(dht_dpin, DHTTYPE);

void setup(void)

{

```

```

dht.begin();

Serial.begin(9600);

Serial.println("Humidity and temperature\n\n");

delay(700);

}

void loop() {

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    Serial.print("Current humidity = ");

    Serial.print(h);

    Serial.print("% ");

    Serial.print("temperature = ");

    Serial.print(t);

    Serial.println("C ");

    delay(800);

}

```

## MOBILE APP FOR LED

### Required Libraries

<https://drive.google.com/drive/folders/1FJ7dczRCDbwVu3AmH3BHaZhSoBZfYG6M?usp=sharing>

### CODE:

```

#define BLYNK_TEMPLATE_ID "TMPL3HinybI08"

#define BLYNK_TEMPLATE_NAME "LED Control"

```

```
#define BLYNK_AUTH_TOKEN "0Aj_8y-nliORfbxQnHADNFsXzvrcu3bH"
```

```
#define BLYNK_PRINT Serial
```

```
#include <ESP8266WiFi.h>
```

```
#include <BlynkSimpleEsp8266.h>
```

```
char auth[] = BLYNK_AUTH_TOKEN;
```

```
char ssid[] = " "; //Enter your WIFI name
```

```
char pass[] = " "; //Enter your WIFI password
```

```
//Get the button value
```

```
BLYNK_WRITE(V0) {
```

```
    digitalWrite(D0, param.asInt());
```

```
}
```

```
void setup() {
```

```
    //Set the LED pin as an output pin
```

```
    pinMode(D0, OUTPUT);
```

```
    //Initialize the Blynk library
```

```
    Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
```

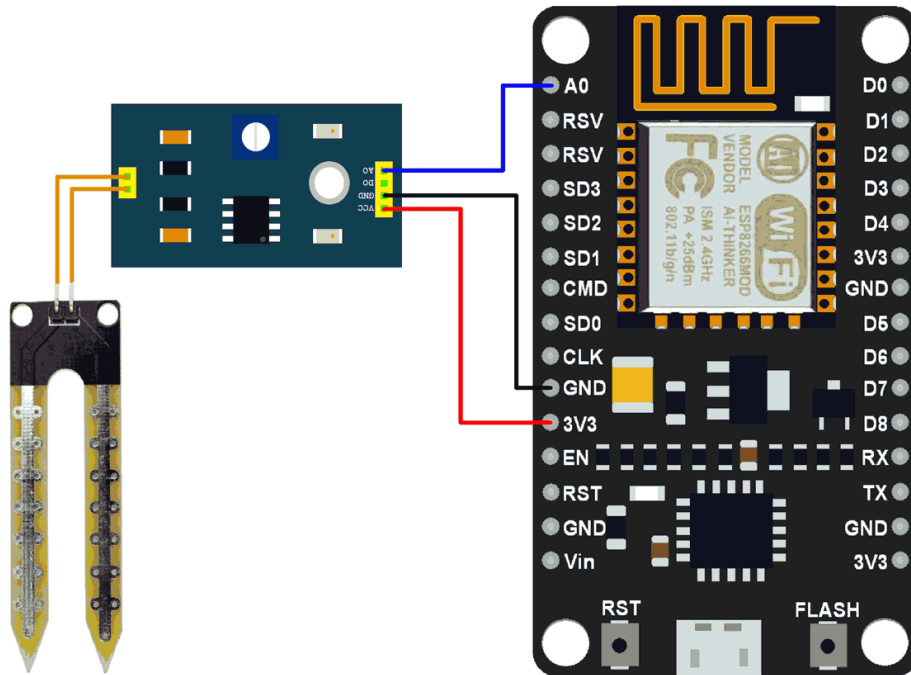
```
}
```

```
void loop() {
```

```
    //Run the Blynk library
```

```
Blynk.run();  
  
}
```

## Soil Moisture Sensor



## Code For Soil Moisture Sensor:

```
const int sensor_pin = A0; /* Connect Soil moisture analog sensor pin to A0 of NodeMCU */  
  
void setup() {  
  Serial.begin(9600); /* Define baud rate for serial communication */  
}  
  
void loop() {  
  float moisture_percentage;  
  
  moisture_percentage = ( 100.00 - ( (analogRead(sensor_pin)/1023.00) * 100.00 ) );  
  
  Serial.print("Soil Moisture(in Percentage) = ");
```

```

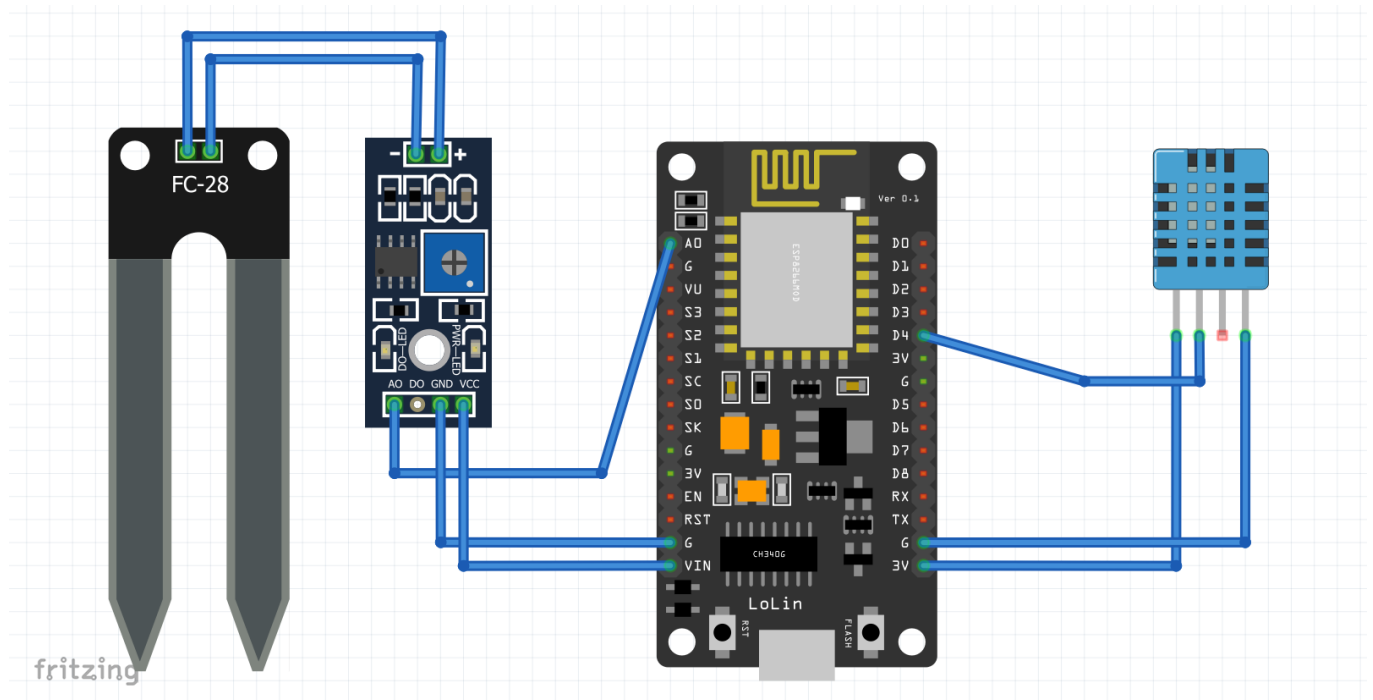
Serial.print(moisture_percentage);

Serial.println("%");

delay(1000);
}

```

## DHT11 AND SOIL MOISTURE



## CODE :

```

#include <WiFi.h>

#include <WiFiMulti.h>

#include <HTTPClient.h>

#include <Adafruit_Sensor.h>

#include <DHT.h>

#include <DHT_U.h>

```

```

WiFiMulti WiFiMulti;

```

HTTPClient ask;

// user config: TODO

#define MOISTURE\_THRESHOLD 55 // moisture alert threshold

const char\* ssid = "\*\*\*\*"; // enter wifi user name

const char\* password = "\*\*\*\*"; // enter wifi password

const char\* apiKeyIn = "E5IiZ2835YBOgiiQwKjEErx7gaiaVIbw"; // Write your api key given from ask sensor

const unsigned int writeInterval = 25000; // write interval (in ms)

// ASKSENSORS config.

const char\* host = "api.asksensors.com"; // ASKSENSORS API host name

const int httpPort = 80; // port

// DHT config.

#define DHTPIN 4 //d4 Pin which is connected to the DHT sensor.

// Uncomment the type of sensor in use:

#define DHTTYPE DHT11 // DHT 11

// #define DHTTYPE DHT22 // DHT 22 (AM2302)

// #define DHTTYPE DHT21 // DHT 21 (AM2301)

DHT\_Unified dht(DHTPIN, DHTTYPE);

uint32\_t delayMS;

int status = WL\_IDLE\_STATUS;

float myTemperature = 0, myHumidity = 0;



```
int moisture_Pin= A0; // Soil Moisture Sensor input at Analog PIN vp

int moisture_value= 0, moisture_state = 0xFF;


// create ASKSENSORS client

//

void setup() {

  // open serial

  Serial.begin(115200);


  Serial.println("Wait for WiFi... ");


  // connecting to the WiFi network

  WiFiMulti.addAP(ssid, password);

  while (WiFiMulti.run() != WL_CONNECTED) {

    Serial.print(".");

    delay(500);

  }

  // connected

  Serial.println("WiFi connected");

  Serial.println("IP address: ");

  Serial.println(WiFi.localIP());


  // Initialize device.

  dht.begin();

  Serial.println("DHTxx Unified Sensor Example");
```

```

// Print temperature sensor details.

sensor_t sensor;

dht.temperature().getSensor(&sensor);

Serial.println("-----");

Serial.println("Temperature");

Serial.print ("Sensor:   "); Serial.println(sensor.name);

Serial.print ("Driver Ver: "); Serial.println(sensor.version);

Serial.print ("Unique ID: "); Serial.println(sensor.sensor_id);

Serial.print ("Max Value: "); Serial.print(sensor.max_value); Serial.println(" C");

Serial.print ("Min Value: "); Serial.print(sensor.min_value); Serial.println(" C");

Serial.print ("Resolution: "); Serial.print(sensor.resolution); Serial.println(" C");

Serial.println("-----");

// Print humidity sensor details.

dht.humidity().getSensor(&sensor);

Serial.println("-----");

Serial.println("Humidity");

Serial.print ("Sensor:   "); Serial.println(sensor.name);

Serial.print ("Driver Ver: "); Serial.println(sensor.version);

Serial.print ("Unique ID: "); Serial.println(sensor.sensor_id);

Serial.print ("Max Value: "); Serial.print(sensor.max_value); Serial.println("%");

Serial.print ("Min Value: "); Serial.print(sensor.min_value); Serial.println("%");

Serial.print ("Resolution: "); Serial.print(sensor.resolution); Serial.println("%");

Serial.println("-----");

// Set delay between sensor readings based on sensor details.

delayMS = sensor.min_delay / 1000;

```

```
}
```

```
void loop() {
```

```
    Serial.println("      Temperature, Humidity and Soil Moisture sensors ");
```

```
        Serial.print("Moisture level : ");
```

```
    moisture_value= analogRead(moisture_Pin);
```

```
    moisture_value= moisture_value/10;
```

```
    Serial.println(moisture_value);
```

```
    if(moisture_value > MOISTURE_THRESHOLD) moisture_state = 0;
```

```
    else moisture_state = 1;
```

```
// Read data from DHT
```

```
// Delay between measurements.
```

```
    delay(delayMS);
```

```
// Get temperature event and print its value.
```

```
sensors_event_t event;
```

```
dht.temperature().getEvent(&event);
```

```
if (isnan(event.temperature)) {
```

```
    Serial.println("Error reading temperature!");
```

```
}
```

```
else {
```

```
    // Update temperature and humidity
```

```
    myTemperature = (float)event.temperature;
```

```
    Serial.print("Temperature: ");
```

```
    Serial.print(myTemperature);
```

```
    Serial.println(" C");

}

// Get humidity event and print its value.

dht.humidity().getEvent(&event);

if (isnan(event.relative_humidity)) {

    Serial.println("Error reading humidity!");

}

else {

    myHumidity = (float)event.relative_humidity;

    Serial.print("Humidity: ");

    Serial.print(myHumidity);

    Serial.println("%");

}


// Use WiFiClient class to create TCP connections

WiFiClient client;


if (!client.connect(host, httpPort)) {

    Serial.println("connection failed");

    return;

}else {

    // Create a URL for the request

    String url = "https://api.asksensors.com/write/";
```

```
url += apiKeyIn;
```

```
url += "?module1=";
```

```
url += myTemperature;
```

```
url += "&module2=";
```

```
url += myHumidity;
```

```
url += "&module3=";
```

```
url += moisture_value;
```

```
url += "&module4=";
```

```
url += moisture_state;
```

```
Serial.print(" requesting URL: ");
```

```
Serial.println(url);
```

```
ask.begin(url); //Specify the URL
```

```
    Serial.println("Temperature, Humidity and Soil moisture data sent to Asksensors");
```

```
//Check for the returning code
```

```
int httpCode = ask.GET();
```

```
if (httpCode > 0) {
```

```
    String payload = ask.getString();
```

```
    Serial.println(httpCode);
```

```
    Serial.println(payload);
```

```
    } else {
```

```
Serial.println("Error on HTTP request");
```

```
}
```

```
ask.end(); //End
```

```
Serial.println(" End ");
```

```
}
```

```
client.stop(); // stop client
```

```
delay(writeInterval); // delay
```

```
}
```