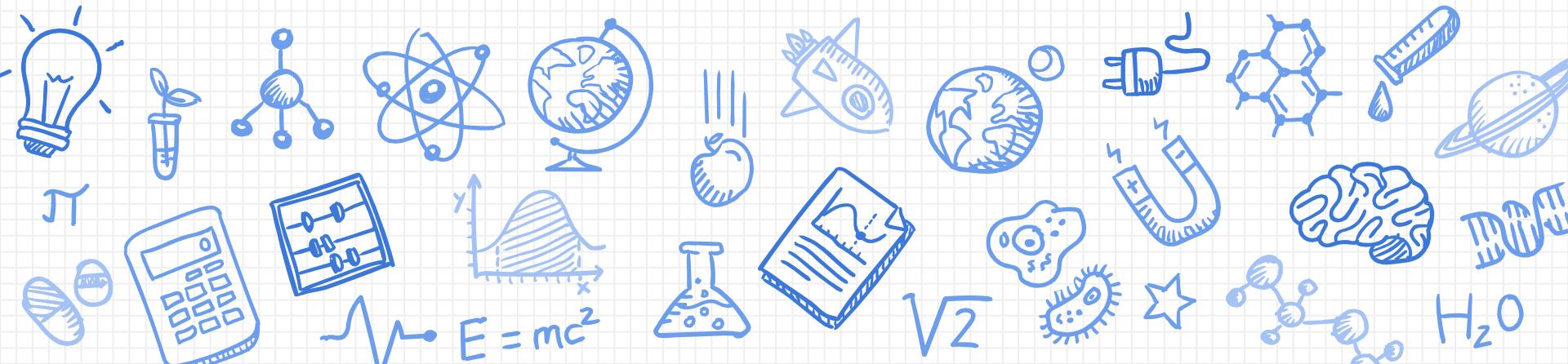
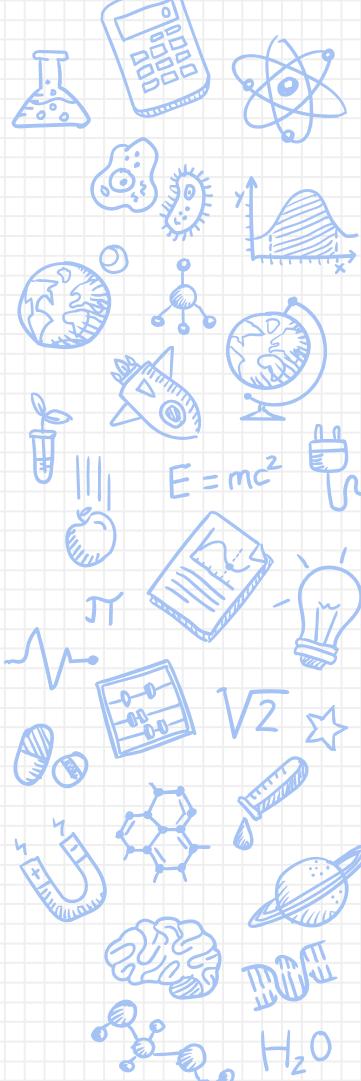
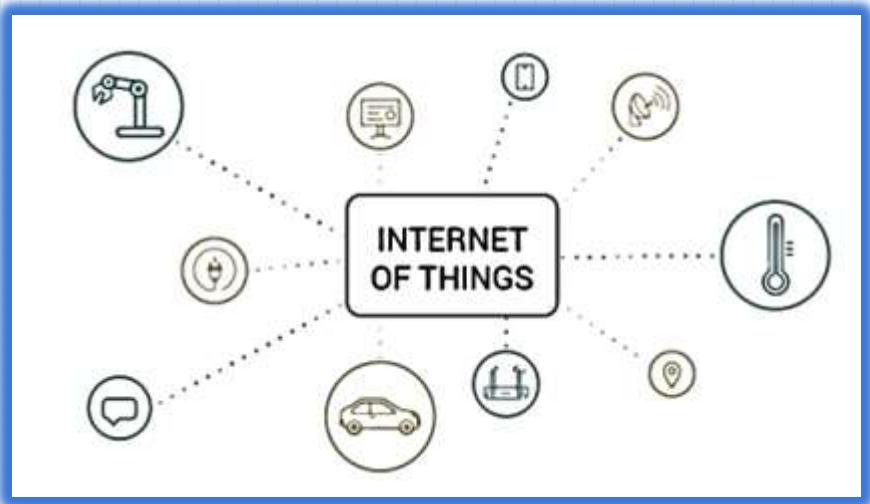


IOT or Internet of THings



What is IoT ?

The Internet of Things (IoT) describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet.



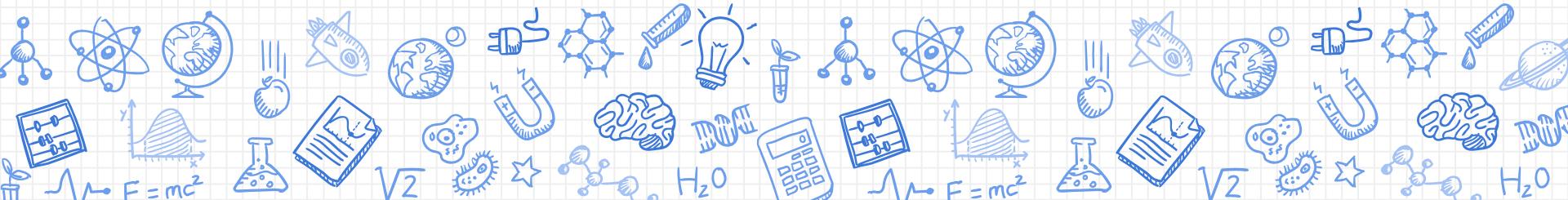


HELLO!

We are Shivam &
Gaurav.

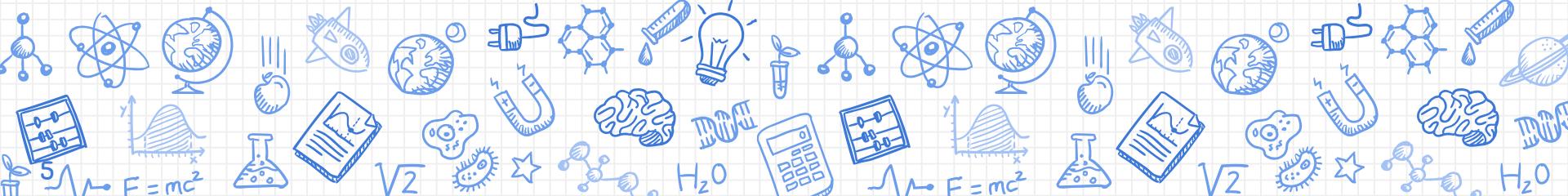
Why IoT?

The internet of things helps people live and work smarter, as well as gain complete control over their lives. In addition to offering smart devices to automate homes, IoT is essential to business.

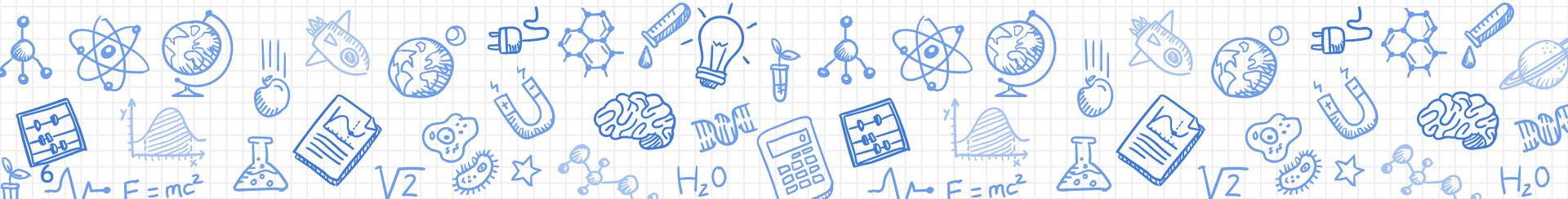


System Requirement

1. Microcontrollers
 2. Sensors
 3. Web & Mobile Applications



Microcontrollers



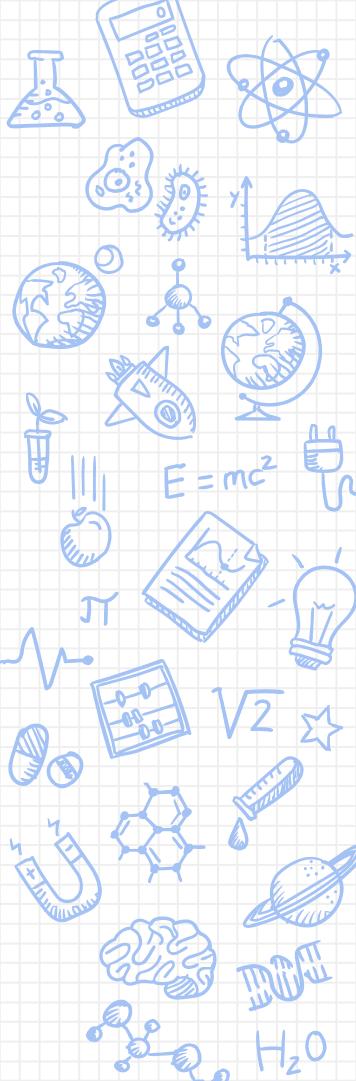
Types of Microcontrollers for IoT.



Node MCU



ESP-32

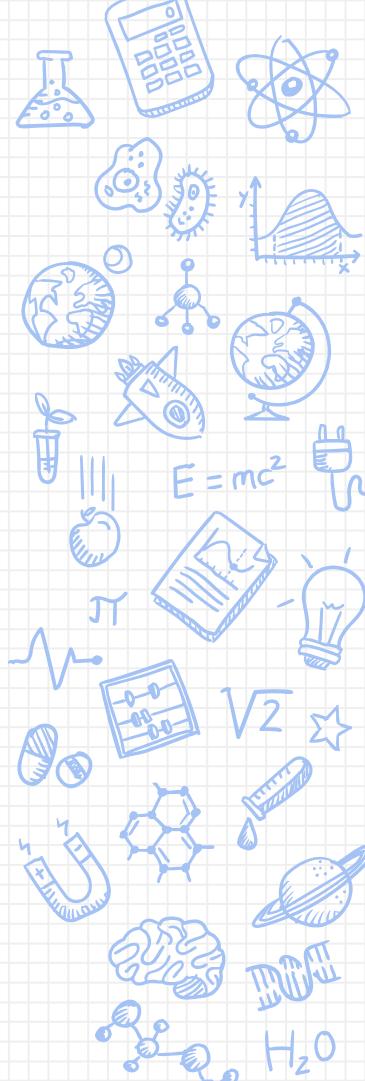


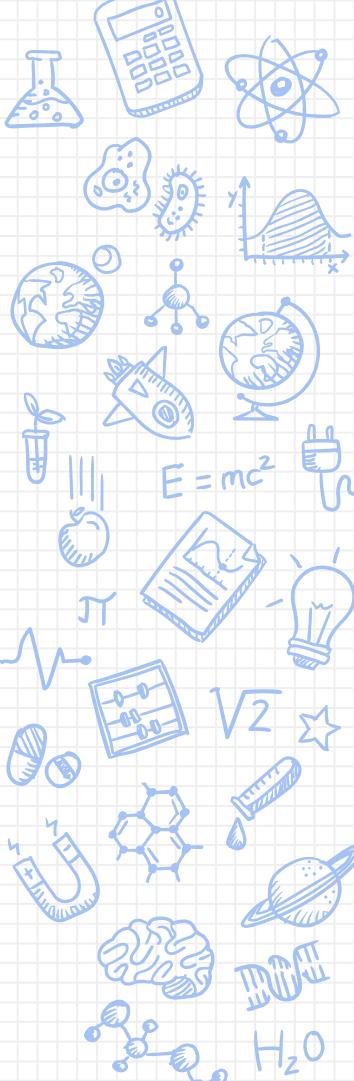
Node MCU



Node MCU

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects





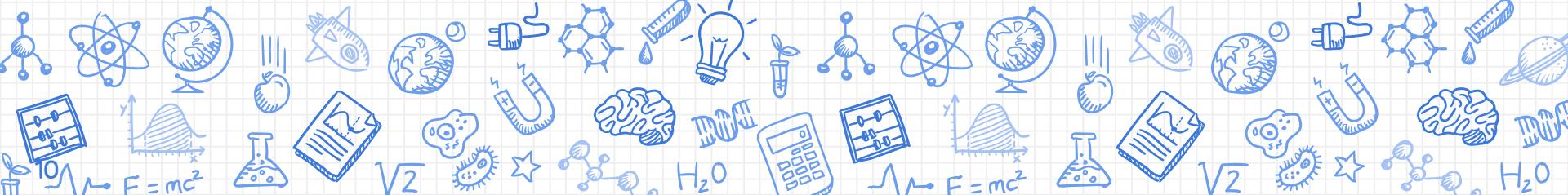
Types of Microcontrollers for IoT.



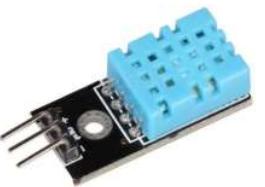
ESP-32

- Microcontroller: Tensilica Xtensa Dual-Core 32-bit LX6 with 600 DMIPS
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 39 (of which 34 is normal GPIO pin)
- Analog Input Pins (ADC): 12-bit, 18 Channel
- UARTs: 3
- SPIs: 3
- I2Cs: 2
- Flash Memory: 4 MB
- SRAM: 520 KB
- Clock Speed: 160 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

Sensors



Types of Sensors for IoT.



DHT-11



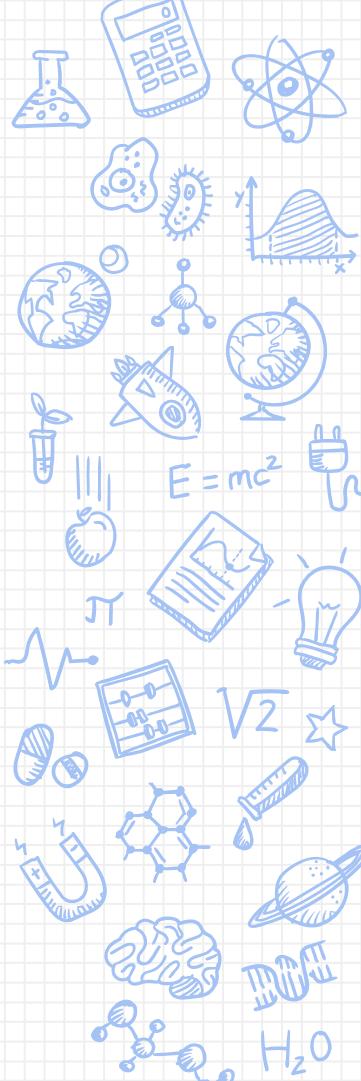
ECG Sensor



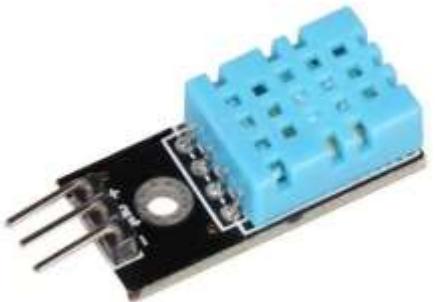
BMP-280



Temperature Sensor

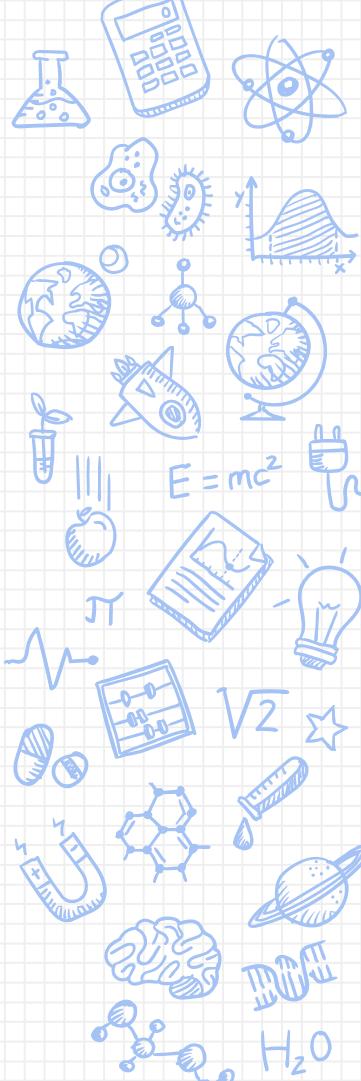


Types of Sensors for IoT.



DHT-11

1. Operating Voltage: 3.5V to 5.5V
2. Operating current: 0.3mA
(measuring) 60uA (standby)
3. Output: Serial data
4. Temperature Range: 0°C to 50°C
5. Humidity Range: 20% to 90%
6. Resolution: Temperature and Humidity both are 16-bit
7. Accuracy: $\pm 1^{\circ}\text{C}$ and $\pm 1\%$

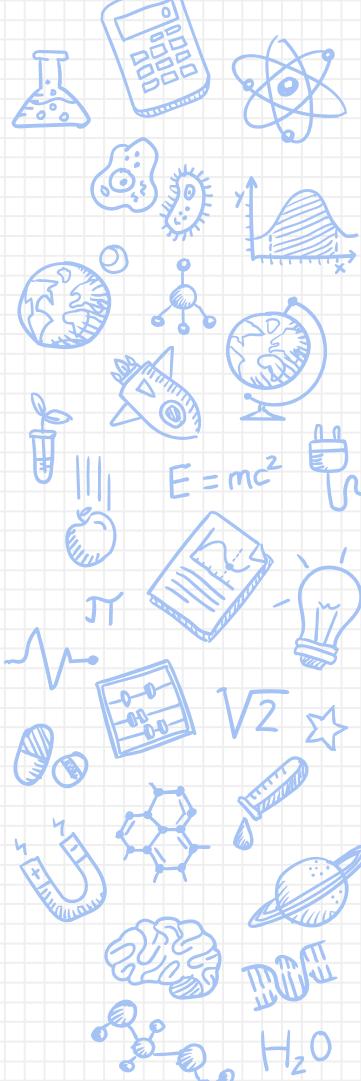


Types of Sensors for IoT.



BMP-280

- Model: GY-BMP280-3.3
- Chip: BMP280
- Power supply: 3V/3.3V DC
- Peak current: 1.12mA
- Air pressure range : 300–1100hPa
(equi. to +9000...–500m above sea level)
- Temperature range: -40 ... +85 °C
- Digital interfaces: I²C (up to 3.4 MHz) and SPI (3 and 4 wire, up to 10 MHz)
- Current consumption of sensor
BMP280: 2.7µA @ 1 Hz sampling rate

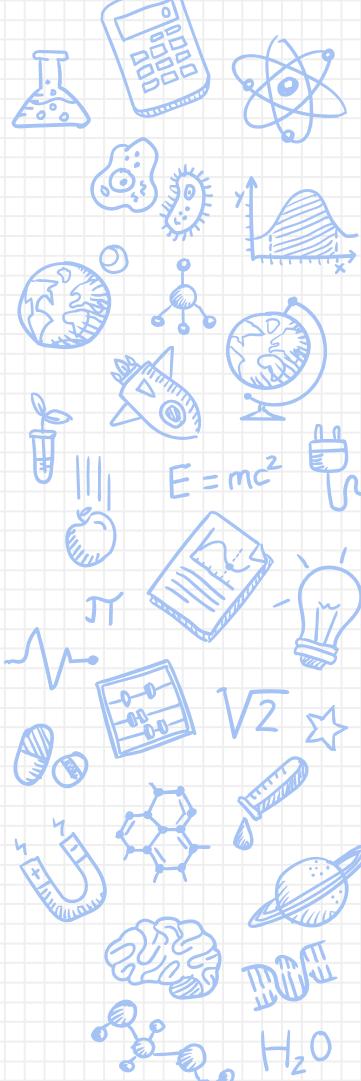


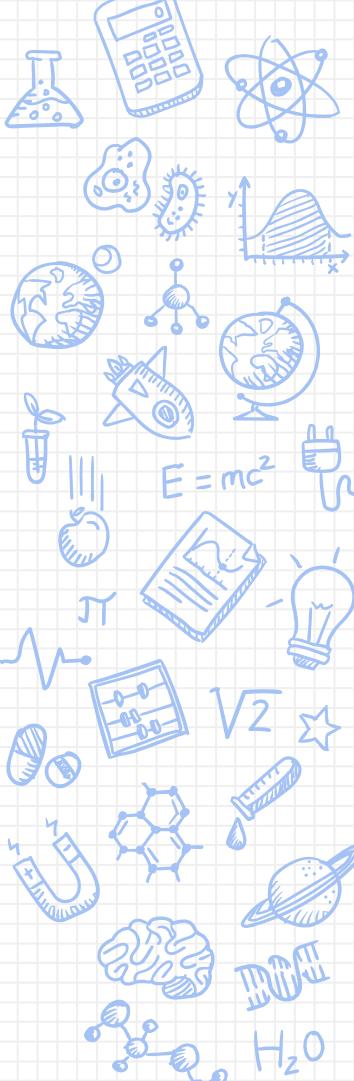
Types of Sensors for IoT.



ECG Sensor

- Fully integrated single-lead ECG front end
- Common-mode rejection ratio: 80 dB (dc to 60 Hz)
- Two or three-electrode configurations
- Qualified for automotive application
- Single-supply operation: 2.0 V to 3.5
- Fast restore feature improves filter settling
- Size: 3.5cm x 3cm





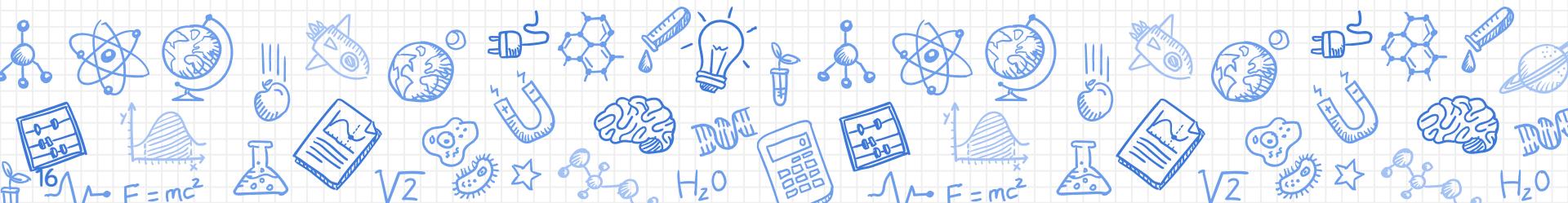
Types of Sensors for IoT.

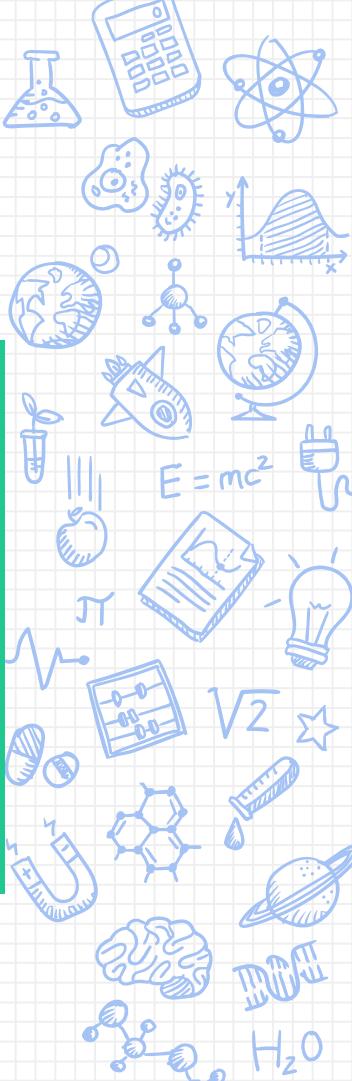


Temperature Sensor

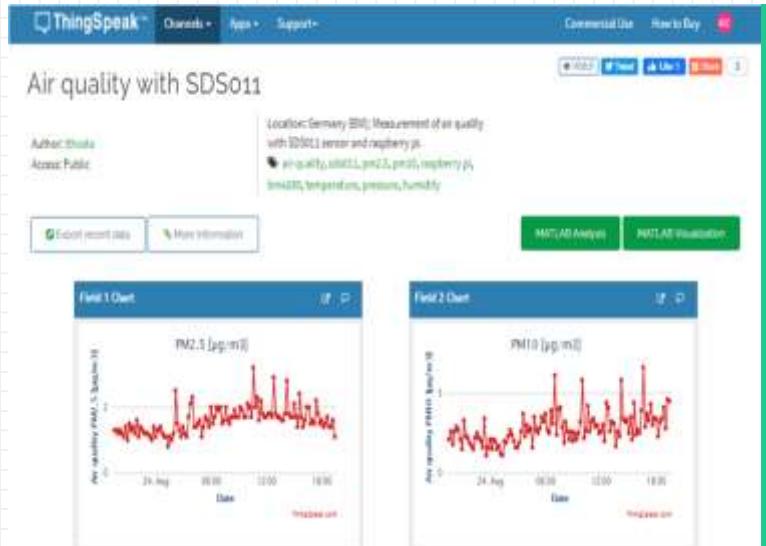
- Programmable Digital Temperature Sensor
- Communicates using 1-Wire method
- Operating voltage: 3V to 5V
- Temperature Range: -55°C to $+125^{\circ}\text{C}$
- Accuracy: $\pm 0.5^{\circ}\text{C}$
- Output Resolution: 9-bit to 12-bit (programmable)
- Unique 64-bit address enables multiplexing
- Conversion time: 750ms at 12-bit
- Programmable alarm options
- Available as To-92, SOP and even as a waterproof sensor

Different Web and Mobile application.





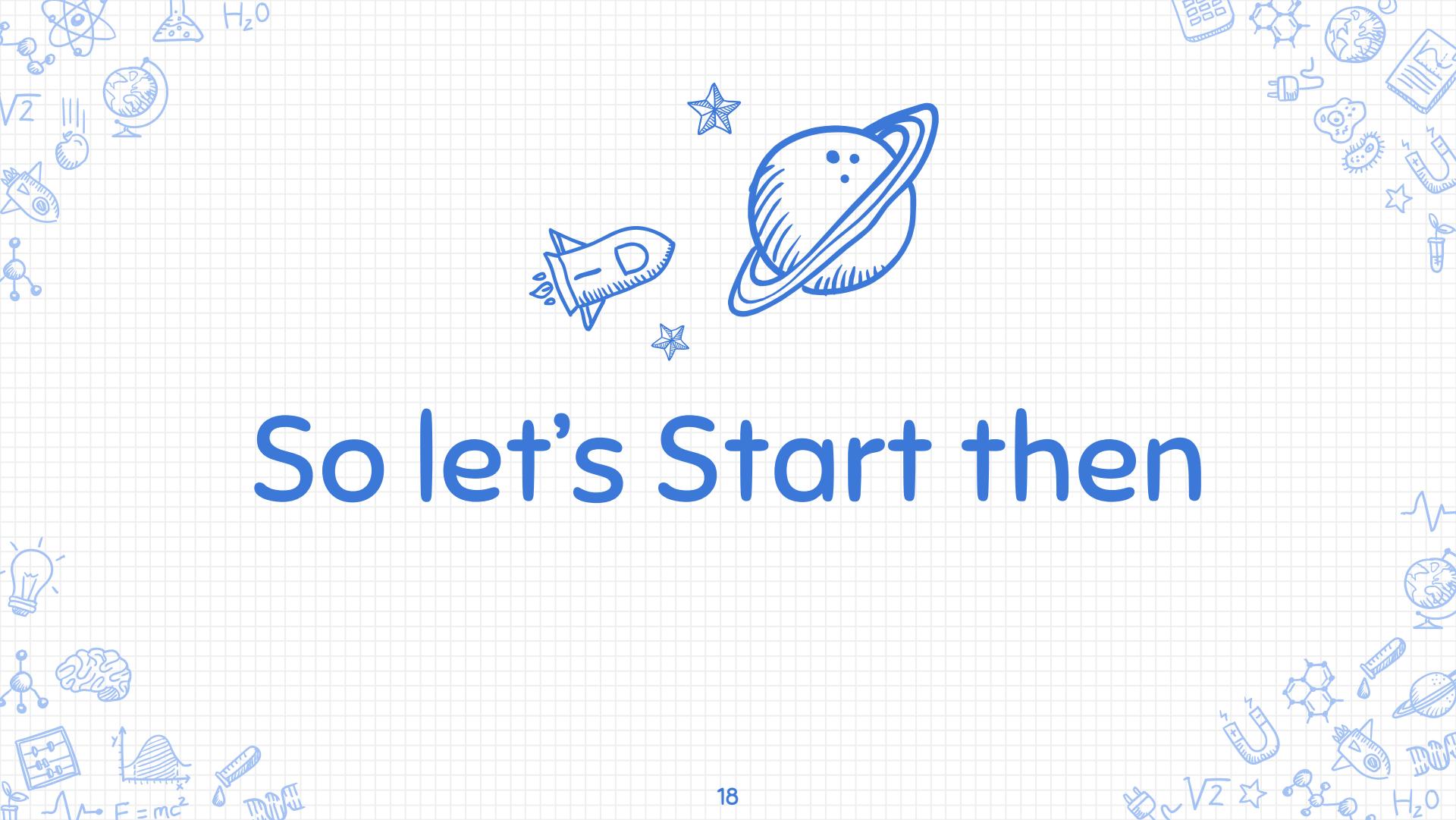
Different Web and Mobile application.



ThingSpeak



Blynk

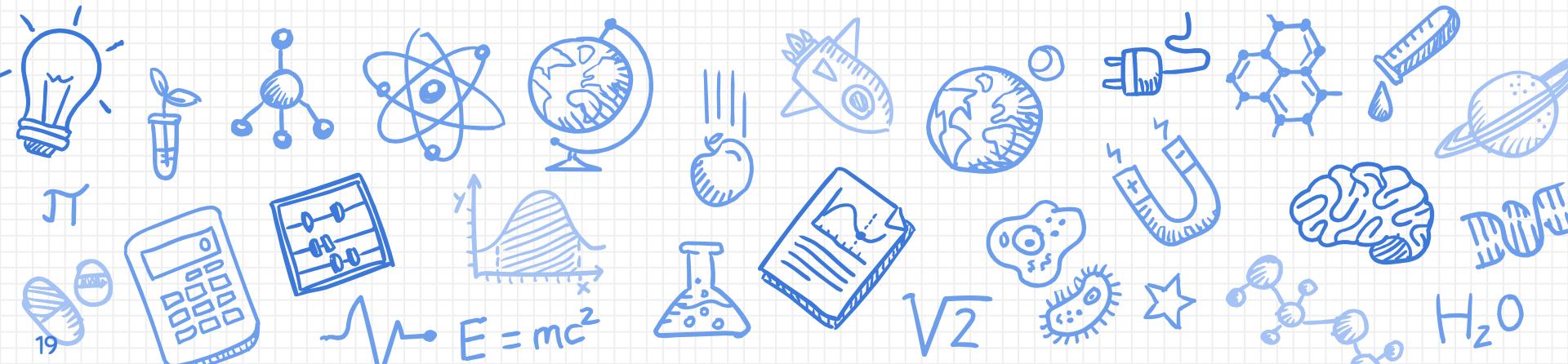


So let's Start then

First Install the Arduino IDE.

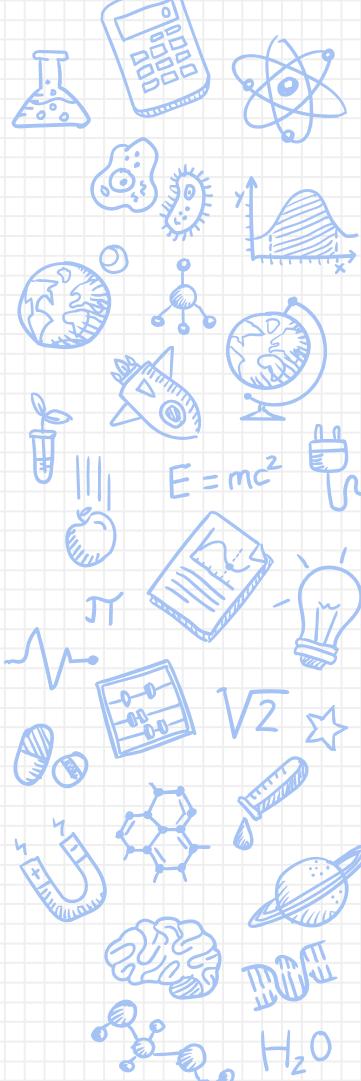
Step 1 - Open this website www.arduino.cc.

Step 2 - Go to software → Downloads → Arduino IDE 1.8.19
→ Windows win 7 and newer.

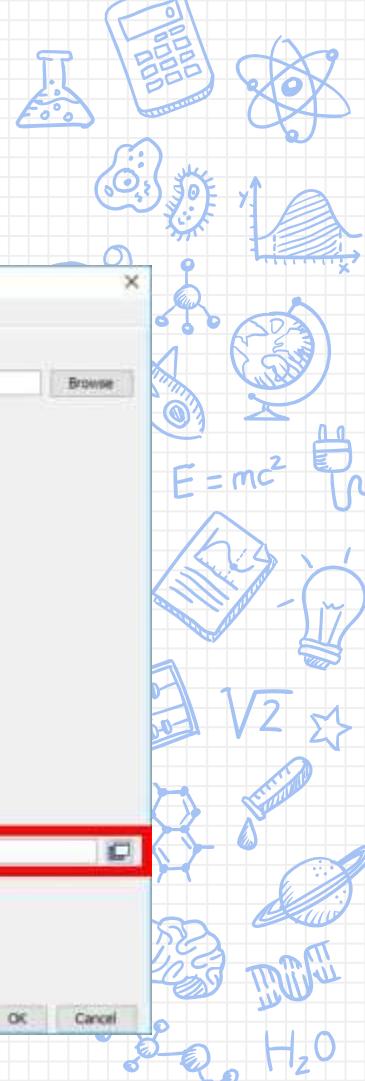


Setup a programming environment for Node MCU in Arduino IDE.

1. Select the *File → Preferences*
2. This will open up a new window, where we set the IDE to point to the configuration file.
3. In the field “Additional Boards Manager URLs:” enter
4. http://arduino.esp8266.com/stable/package_esp8266com_index.json
5. Select the OK button at the bottom of the window when the link address is entered.



How to setup a programming environment for Node MCU in Arduino IDE.



The image shows the Arduino IDE interface with a red box highlighting the "File" menu and the "Preferences" option under "File". The "Preferences" dialog box is open, showing various settings and an "Additional Boards Manager URLs" field which is also highlighted with a red box. The URL entered is https://arduino.esp8266.com/stable/package_esp8266com_index.json.

sketch_sep05c | Arduino 1.8.9

File Edit Sketch Tools Help

New Ctrl+N

Open... Ctrl+O

Open Recent >

Sketchbook >

Examples >

Close Ctrl+W

Save Ctrl+S

Save As... Ctrl+Shift+S

Page Setup Ctrl+Shift+P

Print Ctrl+P

Preferences Ctrl+Comma

Quit Ctrl+Q

Preferences

Settings Network

Sketchbook location: C:\Users\EE35V\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: Automatic 100.0% (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: compilation upload

Compiler warnings: None

Display line numbers

Enable Code Folding

Verify code after upload

Use external editor

Aggressively cache compiled core

Check for updates on startup

Update sketch files to new extension on save (.pde -> .ino)

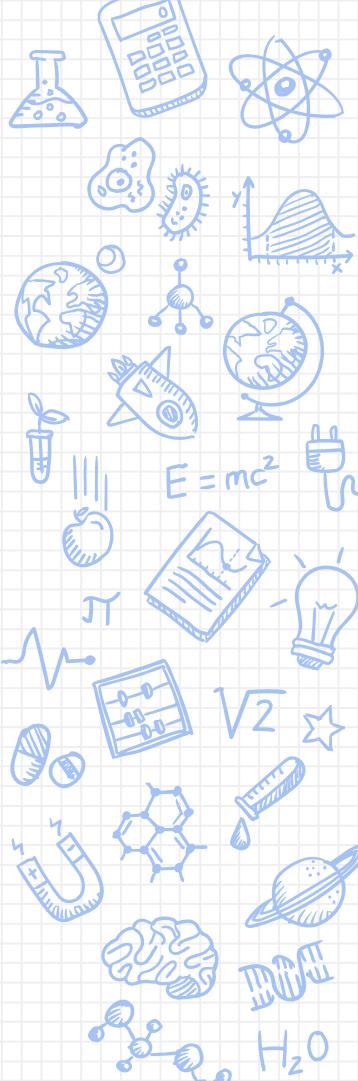
Save when verifying or uploading

Additional Boards Manager URLs: https://arduino.esp8266.com/stable/package_esp8266com_index.json

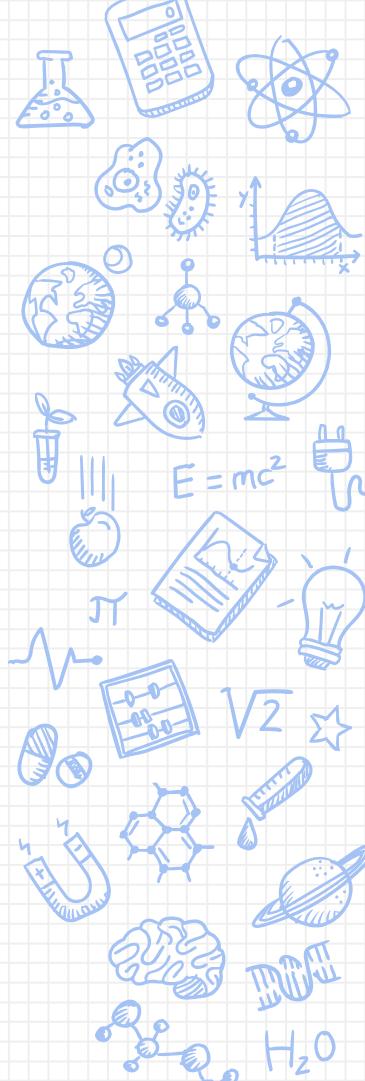
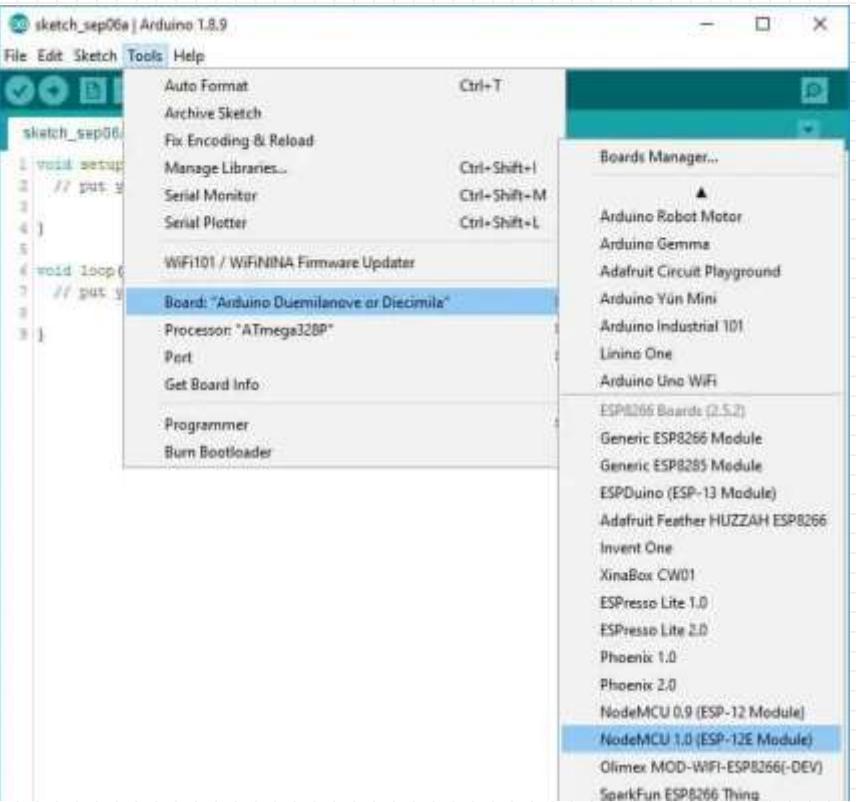
More preferences can be edited directly in the file:
C:\Users\EE35V\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

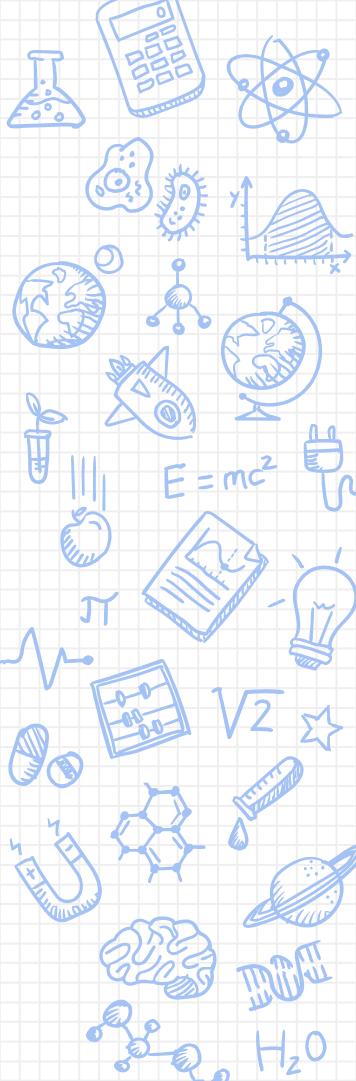
OK Cancel

How to setup a programming environment for Node MCU in Arduino IDE.



How to setup a programming environment for Node MCU in Arduino IDE.





Uploading and Verifying of the sketch.

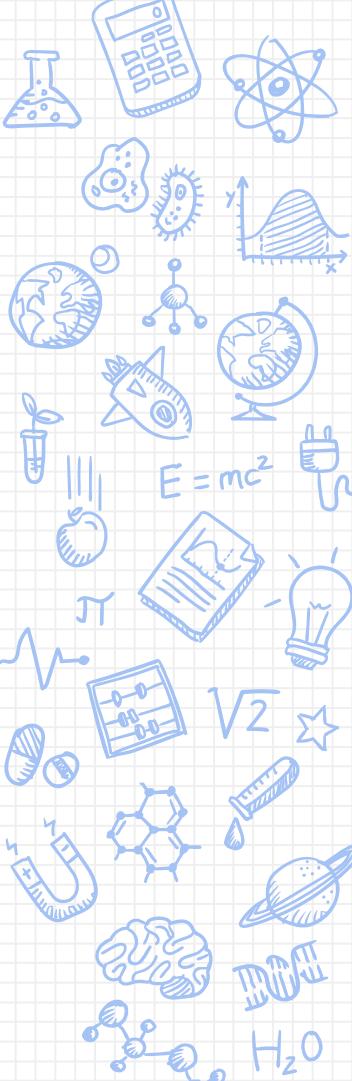
sketch_jun30b | Arduino 1.8.19

[File](#) [Edit](#) [Sketch](#) [Tools](#) [Help](#)



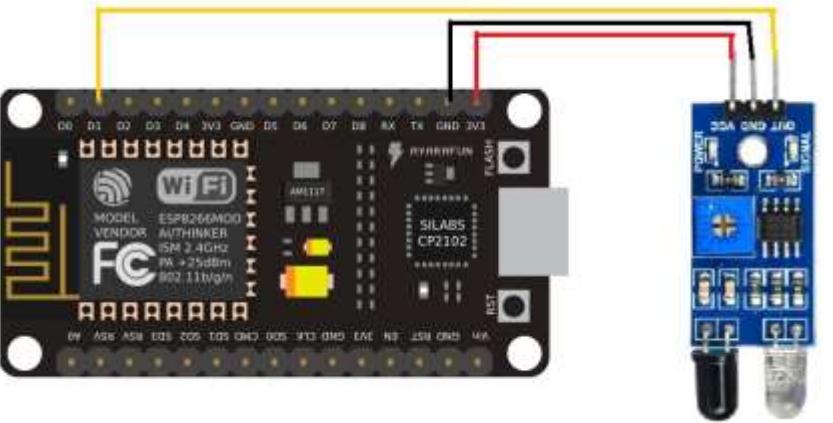
Click this for Verification of the sketch.

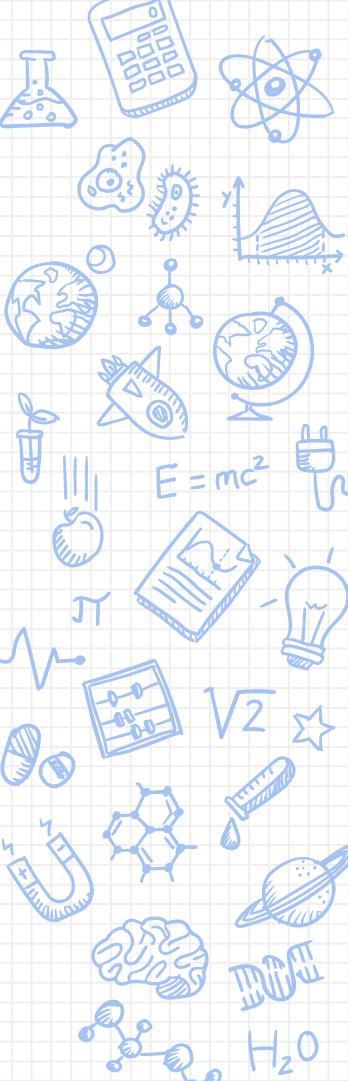
**Click this for Uploading the sketch
to the board.**



Interface sensors with Node MCU.

- X Connect Vcc pin of sensor to 3V3 pin of Node MCU.
- X Connect GND pin of sensor to GND pin of Node MCU.
- X Connect signal pin of sensor to D1 pin of Node MCU.

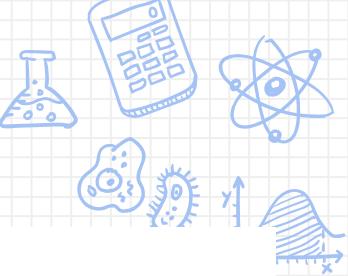




Interface sensors with Node MCU.

Program for IR Sensor :-

```
int IR = D1; //Define the pin Name.  
  
void setup() {  
    Serial.begin(9600); //Begin the serial communication between Node MCU and Serial  
    monitor.  
    pinMode(IR, INPUT); // Declare the sensor as Input.  
}  
  
void loop() {  
    int s = digitalRead(IR); //This will read the sensor data and store it in a variable.  
    Serial.println(s); //This will print the sensor data to serial monitor.  
}
```



Interface sensors with Node MCU.

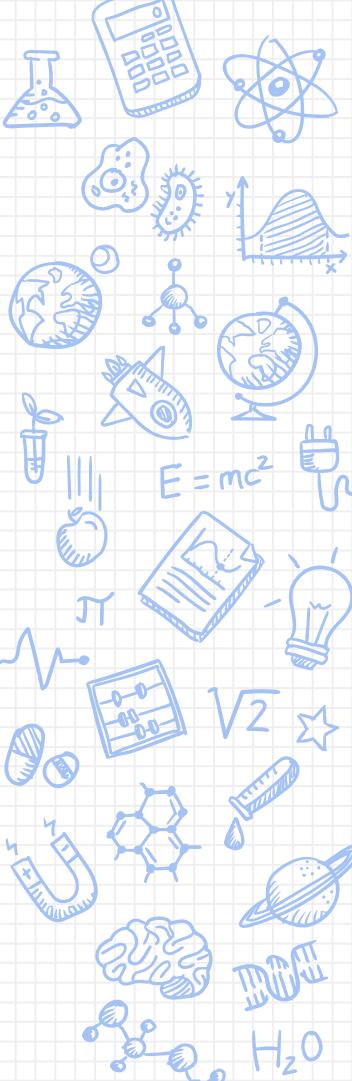
IR_with_Node_MCU | Arduino 1.8.19

File Edit Sketch Tools Help



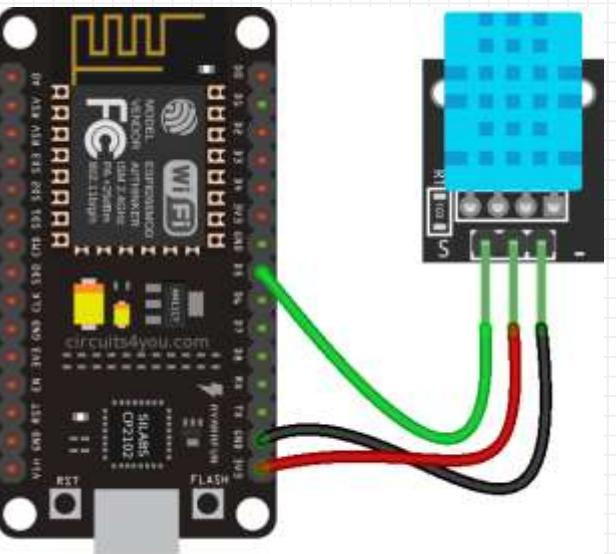
IR_with_Node_MCU \$

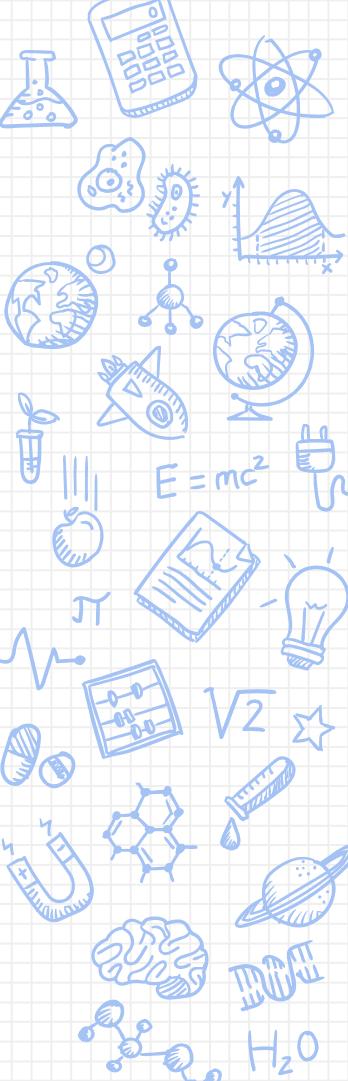
```
1 int IR = D1; //Define the pin Name.  
2 void setup() {  
3   Serial.begin(9600); //Begin the serial communication between Node MCU and Serial monitor.  
4   pinMode(IR, INPUT); // Declare the sensor as Input.  
5 }  
6  
7 void loop() {  
8   int s = digitalRead(IR); //This will read the sensor data and store it in a variable.  
9   | Serial.println(s); //This will print the sensor data to serial monitor.  
10 }
```



Interface sensors with Node MCU.

- X Connect Vcc pin of sensor to 3V3 pin of Node MCU.
- X Connect GND pin of sensor to GND pin of Node MCU.
- X Connect signal pin of sensor to D5 pin of Node MCU.



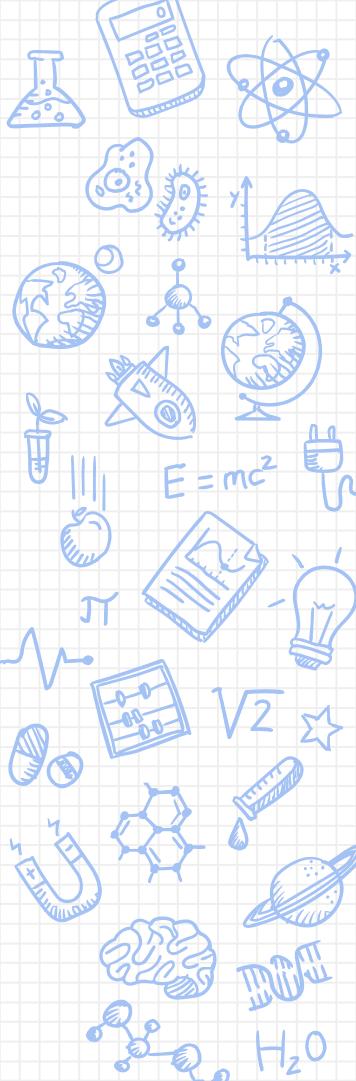
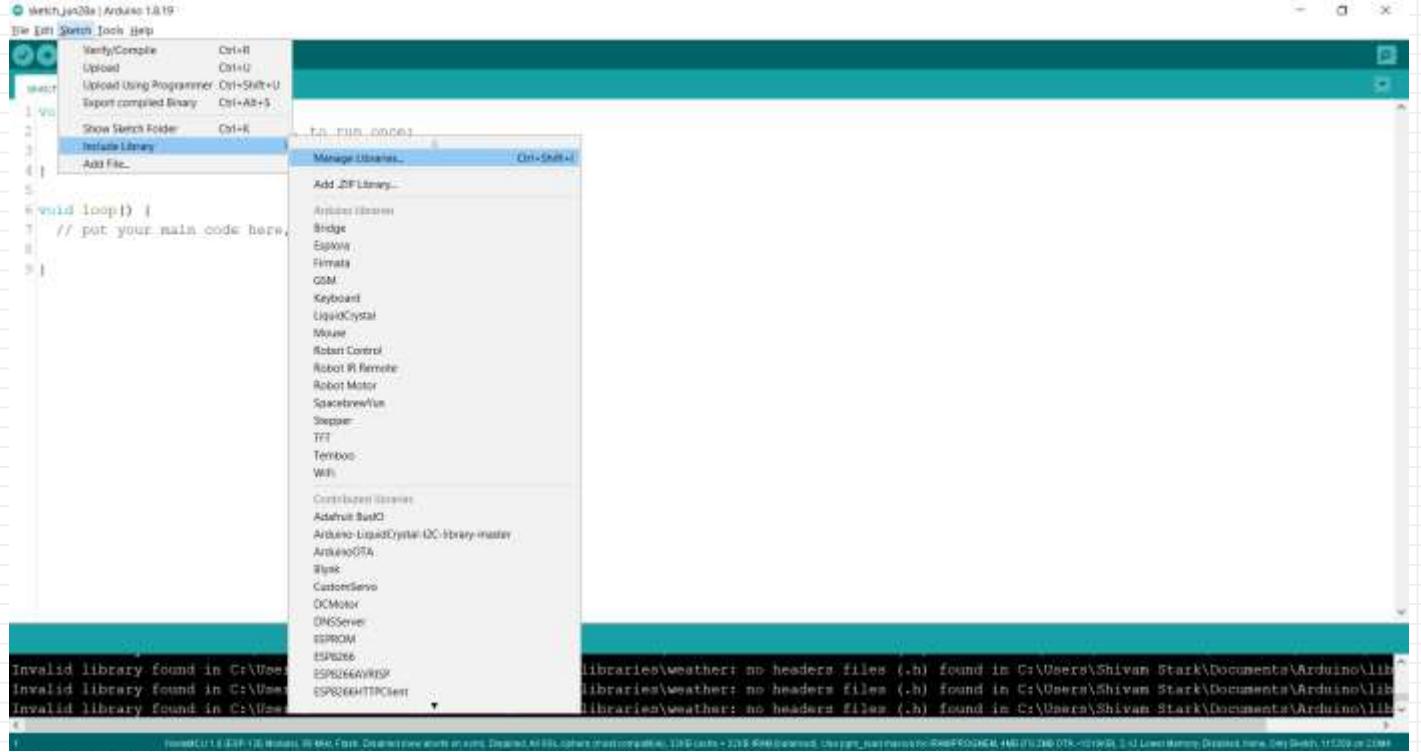


Interface sensors with Node MCU.

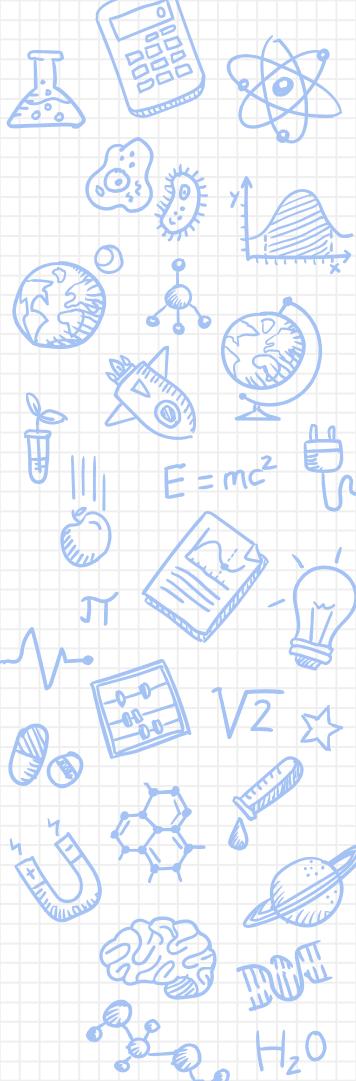
Install DHT Library.

Go to menu **Sketch** → **Include Library** → **Manage Library**

Interface sensors with Node MCU.



Interface sensors with Node MCU.



Sketch: DHT11 | Arduino 1.8.9

File Edit Sketch Tools Help

sketch_jun18

```
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 }
```

Library Manager

Type: All Topic: All DHT 11

DHT Sensor Library
by Adafruit Version 1.4.4 INSTALLED
Arduino Library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino Library for DHT11, DHT22, etc Temp & Humidity Sensors
[More info](#)

Select version: 1.4.4

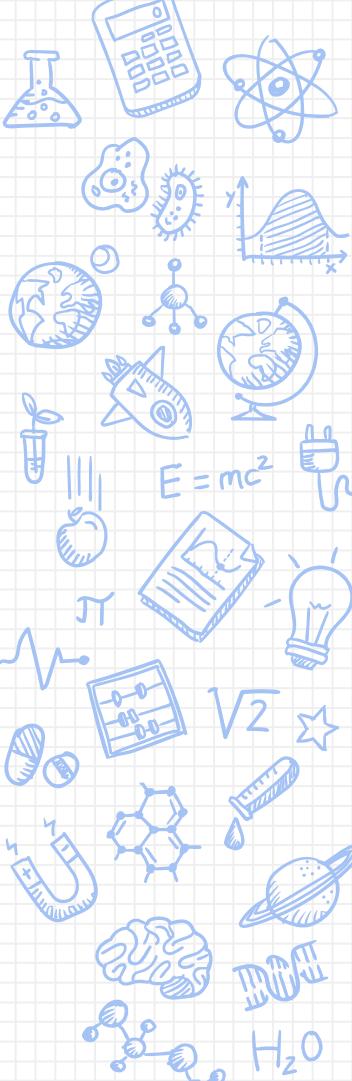
DHT Sensor Library for ESP8266
by koumoo, 10 hrs ago
Arduino ESP8266 Library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized library to match ESP8266 requirements. Last changes: Fix negative temperature problem (credit: @holjungy)
[More info](#)

DHTlib
by Rob Tillaart
Arduino Library for DHT temperature and humidity sensor, with automatic sensortype recognition. Types supported: DHT11, DHT22, DHT34, DHT44, AM2301, AM2302, AM2303; autodetect, iPin, interrupt, powerdown
[More info](#)

DHTStable
by Rob Tillaart
Stable version of library for DHT Temperature & Humidity Sensor DHT11 and DHT22 and equivalents.
[Close](#)

Invalid library found in C:\Users\Shivam Stark\Documents\Arduino\libraries\weather: no headers files (.h) found in C:\Users\Shivam Stark\Documents\Arduino\libs
Invalid library found in C:\Users\Shivam Stark\Documents\Arduino\libraries\weather: no headers files (.h) found in C:\Users\Shivam Stark\Documents\Arduino\libs

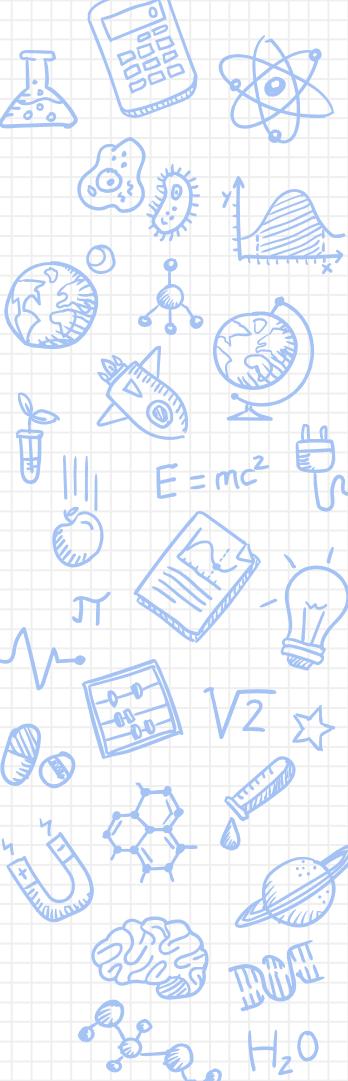
NodeMCU 1.0 (ESP-12E Module), CH340G, Diskless (firmware on chip), Disabed, At 800, Update proceed requirement, CPU 200MHz standard, 1.8V logic, 3.3V processor, PA1/PD1, WIFIT0, 2MB SPI-128MB, 2.02 Libre Monitor, Standard, None, Gray Sketch, 112206 94 COM4



Interface sensors with Node MCU.

Program

Go to menu **File >> Example >> DHT sensor library >> DHT_Unified_Sensor**



Interface sensors with Node MCU.

DHT_Untitled_Sensor | Arduino 1.8.19

File Edit Sketch Tools Help

New Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Page Setup Ctrl+Shift+F
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q

SPISlave
TFT_Touch_Shield_V2
Ticker
Wire

Examples from Core Libraries

Ardublock BMP280 Library
Ardublock BusIO
Ardublock GFX Library
Ardublock NeoMatrix
Ardublock NeoPixel
Ardublock SDI1338
Ardublock TCS34725
Ardublock Untitled Sensor:
Arduino-LiquidCrystal-16x2-I2C-master
Blynk

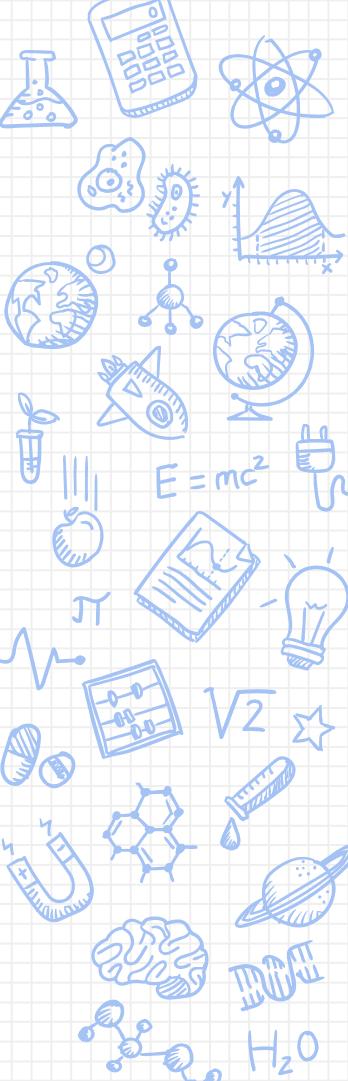
DHT sensor library

FantZD
GY6850
LcdControl
LMIC_Arduino
MPU6050
OttoPiUp
RadioHead
RcSwitch
RF24
SparkFun BME280
ThingSpeak
TimerOne-H1
TinyMepalDC
TinyKunchuk
TinyWire-master
TinyWireM

DHT_Untitled_Sensor

DHTHeader

Processor: ATmega328P - 32 MHz, 16 MHz, 18 MHz or 20 MHz | Clock: 16 MHz | RAM: 2 KB | Flash: 32 KB | EEPROM: 1 KB | Sram: 1 KB | Libraries: 100 | Sketch Size: 10.0 KB | File Size: 1.0 KB | Board: Arduino Uno | Port: COM3 | USB: 100% | Power: 100% | CPU: 100% | Memory: 100% | Disk: 100% | IDE: 100%



Interface sensors with Node MCU.

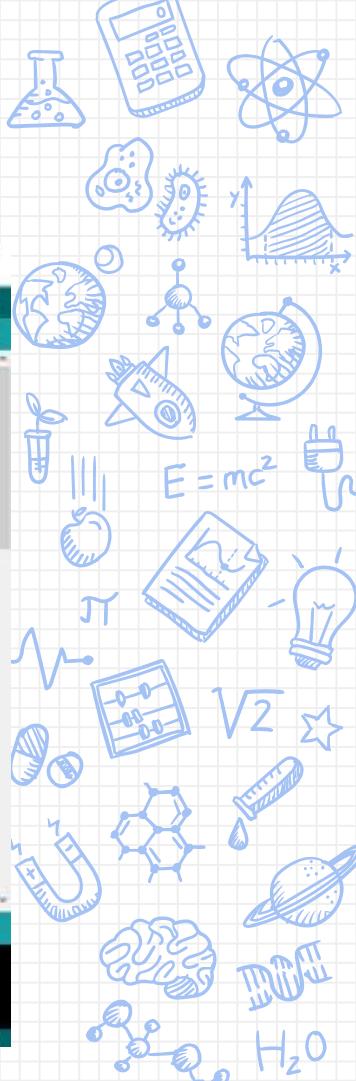
1. Comment the 20th line in the code.
2. Uncomment the 19th line in the code.
3. Change the pin number from 2 to 5 in 14th line of the code.

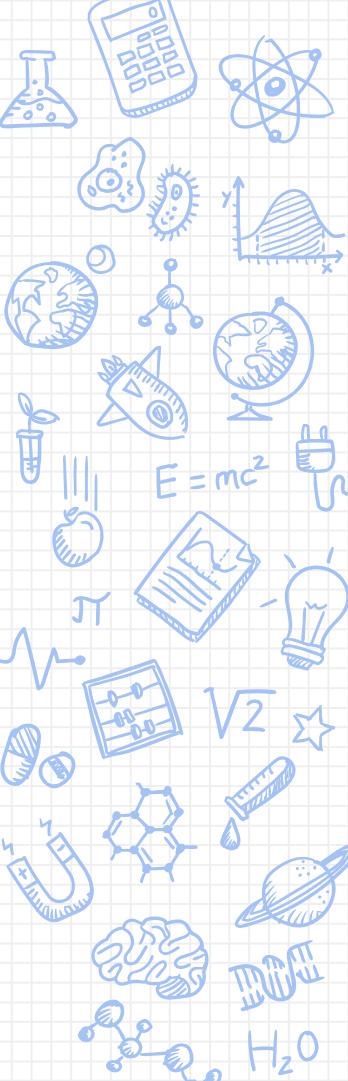
Interface sensors with Node MCU.

```
① DHT_United_Sensor | Arduino 1.8.9
File Edit Sketch Tools Help


DHT_United_Sensor.ino

1 // DHT Temperature + Humidity Sensors
2 // Unified Sensor Library Examples
3 // Written by Tony DiCola for Adafruit Industries
4 // Released under an MIT license.
5
6 // REQUIRES the following Arduino libraries:
7 // - DHT sensor library: https://github.com/adafruit/DHT-sensor-library
8 // - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit Unified Sensor Library
9
10 #include <Adafruit_Sensor.h>
11 #include <DHT.h>
12 #include <DHT_U.h>
13
14 #define DHTPIN 5 // Digital pin connected to
15 // Feather HUZZAH ESP8266 note: use pins 3, 4,
16 // pin 15 can work but DHT must be disconnected
17
18 // Uncomment the type of sensor in use:
19 #define DHTTYPE DHT11 // DHT 11
20 // #define DHTTYPE DHT22 // DHT 22 (AM2302)
21 // #define DHTTYPE DHT21 // DHT 21 (AM2301)
22
23 // See guide for details on sensor wiring and setup
24 // https://learn.adafruit.com/dht-temperature-humidity-sensor-tutorial
25
26 DHT_United_Sensor(dht(DHTPIN, DHTTYPE));
27
28 uint32_t delaysMS;
29
30 void setup() {
```

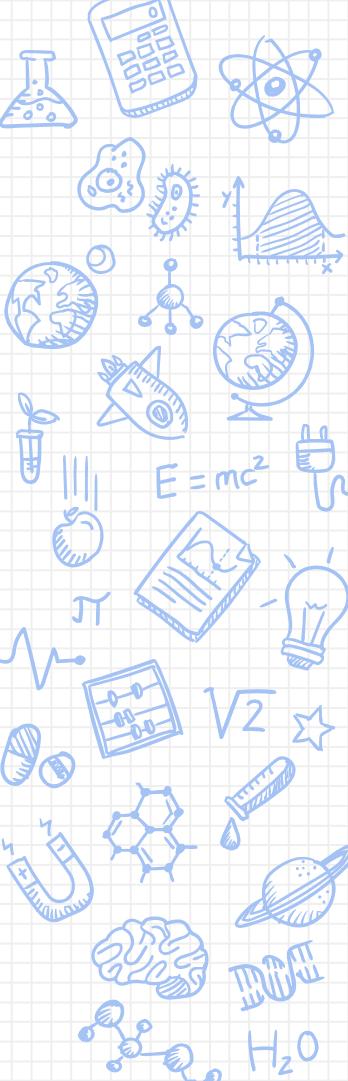




Connect Node MCU with Internet.

Program

```
#include<ESP8266WiFi.h>
const char* ssid= "xxxxxxxxxx"; // Name of the Network.
const char* password= "xxxxxxxxxx"; // Password of the Network.
void setup0{
    Serial.begin(9600);
    WiFi.begin(ssid,password); // This will connect the NodeMCU with WiFi.
    while(WiFi.status() != WL_CONNECTED){
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println(WiFi.localIP()); // This will print the Local IP address of the Network.
} void loop0{
    // put your main code here, to run repeatedly:
}
```



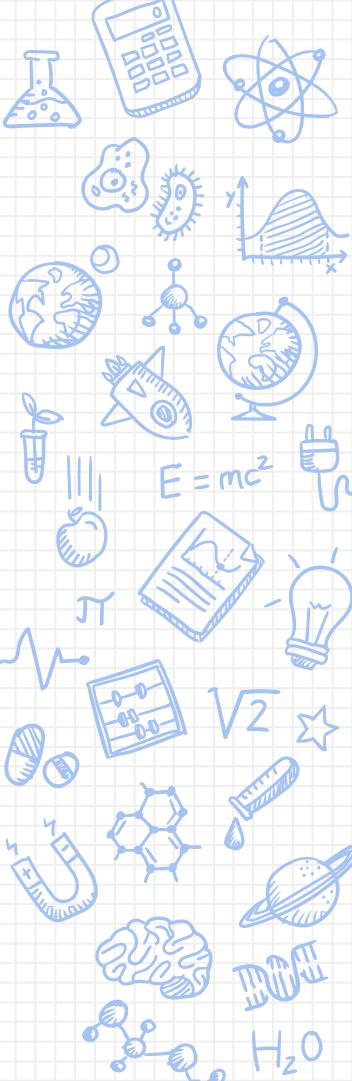
Connect Node MCU with Internet.

Node_MCU_WiFi_Code | Arduino 1.8.19

File Edit Sketch Tools Help



```
Node_MCU_WiFi_Code
1 #include <ESP8266WiFi.h>
2 const char* ssid = "xxxxxxxxxx"; // Name of the Network.
3 const char* password = "xxxxxxxxxx"; // Password of the Network.
4 void setup() {
5   Serial.begin(9600);
6   WiFi.begin(ssid, password); // This will connect the Node MCU with WiFi.
7   while(WiFi.status() != WL_CONNECTED) {
8     delay(500);
9     Serial.print(".");
10 }
11 Serial.println("");
12 Serial.println("WiFi connected");
13 Serial.println(WiFi.localIP()); //This will print the Local IP address of the Network.
14 }
15
16 void loop() {
17   // put your main code here, to run repeatedly:
18
19 }
```

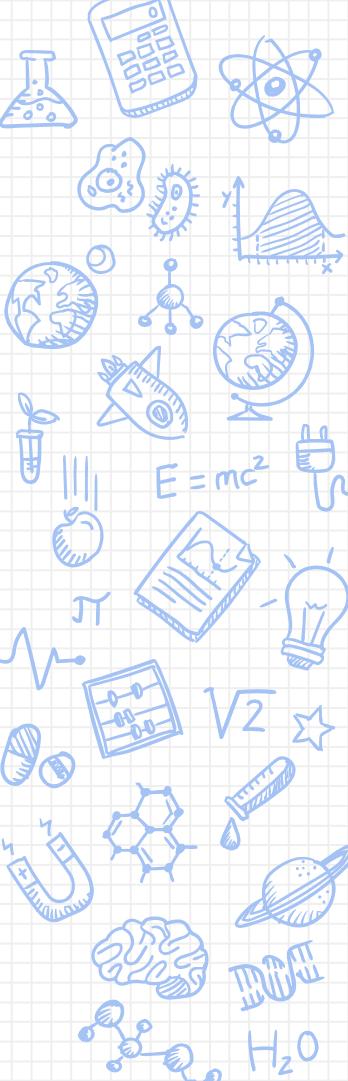


Send DHT-11 sensor Data to ThingSpeak.

Step 1 – Open this website www.thingspeak.com

Step 2 – Click on Get Started For Free.





Send DHT-11 sensor Data to ThingSpeak.

Step 3 – Click on Create one! To create a login account.

The screenshot shows a web browser window for ThingSpeak IoT. The URL is <https://thingspeak.com/login?skipSSOCheck=true>. The page displays a message about signing in with existing accounts or creating a free non-commercial account. It also mentions MATLAB analysis features and a link to send more data. Below this, there's a MathWorks sign-in form with fields for Email and Password. A red box highlights the "Create one!" button under the "No account?" link. At the bottom, there's a note about agreeing to privacy policy and a "Next" button.

To use ThingSpeak, you must sign in with your existing account.
Non-commercial users may use ThingSpeak for free. For full access to the MATLAB analysis features on ThingSpeak, sign in with your account.
To send data faster to ThingSpeak or to send more data, sign in with your account.

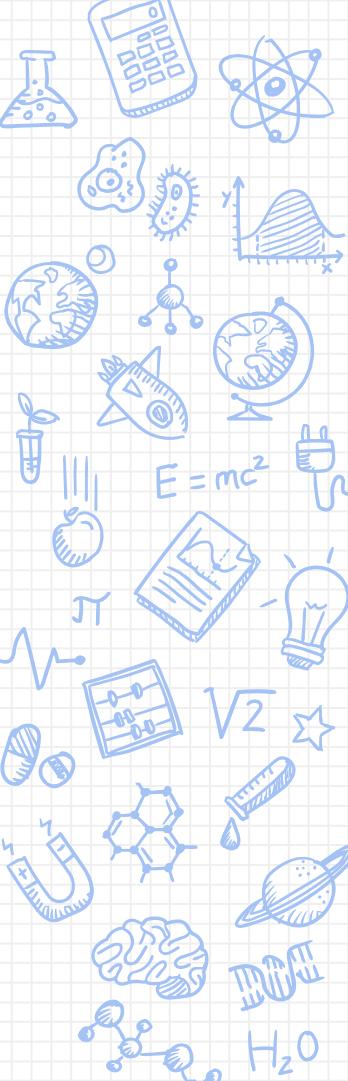
MathWorks®

Email

No account? [Create one!](#)

By signing in, you agree to our [privacy policy](#).

[Next](#)



Send DHT-11 sensor Data to ThingSpeak.

Step 4 – Click on Channels and then click on My Channels

ThingSpeak™

Accelerometer

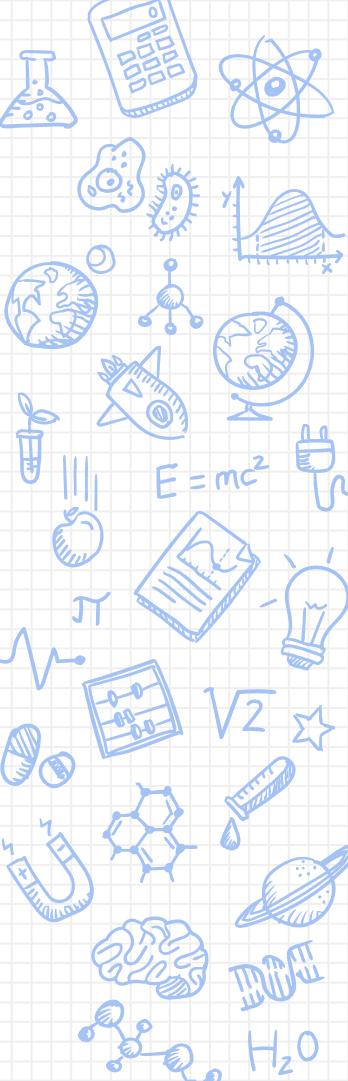
Channel ID: 1780836
Author: stark42
Access: Private

Channels ▾ Apps ▾ Devices ▾ Support ▾

- My Channels
- My Image Channels
- Watched Channels
- Public Channels

Reading from GY-61 Accelerometer

Private View Public View Channel Settings Sharing API Keys Data Impo



Send DHT-11 sensor Data to ThingSpeak.

Step 5 – Click on New Channel.

ThingSpeak™

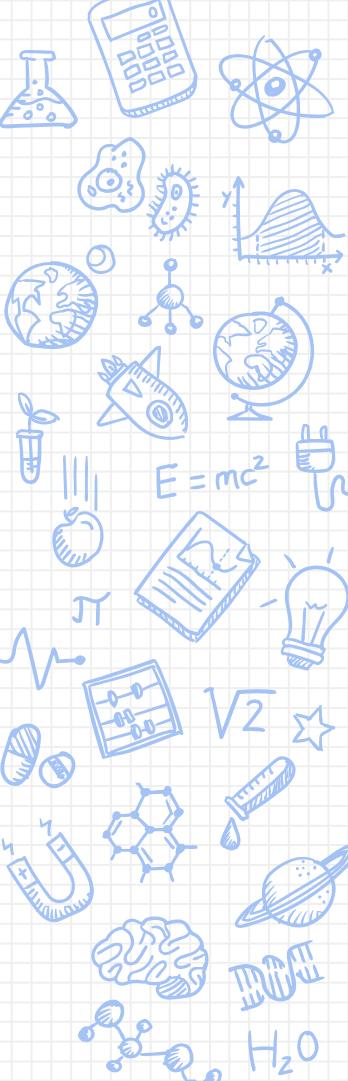
Channels ▾ Apps ▾ Devices ▾ Support ▾

My Channels

New Channel

Search by tag

Name	Created	Upd
Accelerometer Reading	2022-06-25	2022-06-25



Send DHT-11 sensor Data to ThingSpeak.

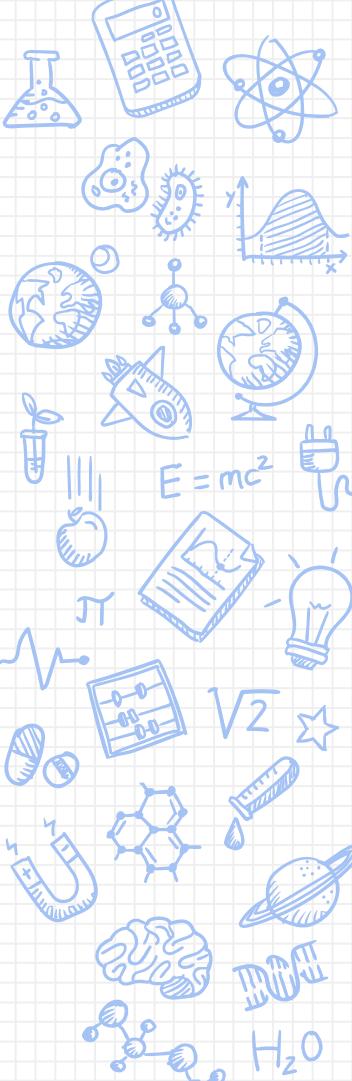
Step 6 – Write the Name of the channel and field .

ThingSpeak™

Channels Apps Devices Support

New Channel

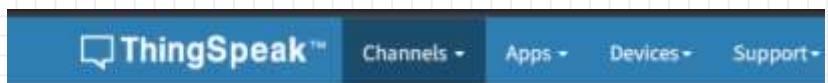
Name	DHT-11 Reading
Description	
Field 1	Humidity <input checked="" type="checkbox"/>
Field 2	



Send DHT-11 sensor Data to ThingSpeak.

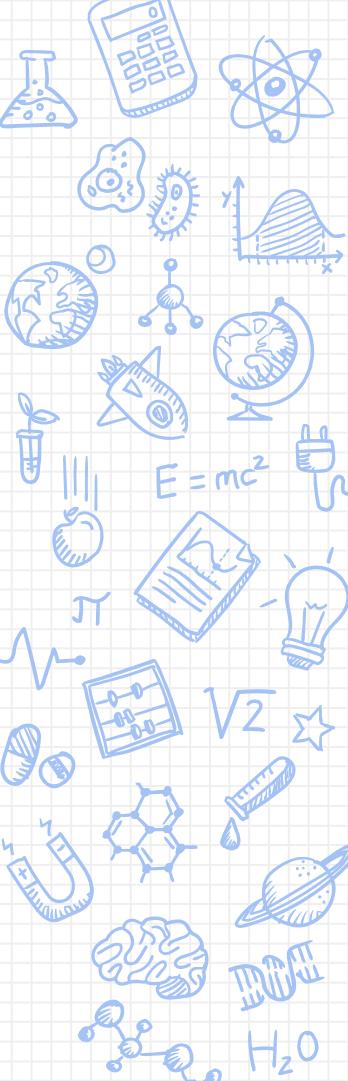
Step 7 – Write the Name of the channel and field .

Step 8 – Click the Save Channel button to create and save your channel.



New Channel

Name	DHT-11 Reading
Description	
Field 1	Humidity <input checked="" type="checkbox"/>
Field 2	



Send DHT-11 sensor Data to ThingSpeak.

Step 9 – Go to your Private View tab and click on the edit icon.

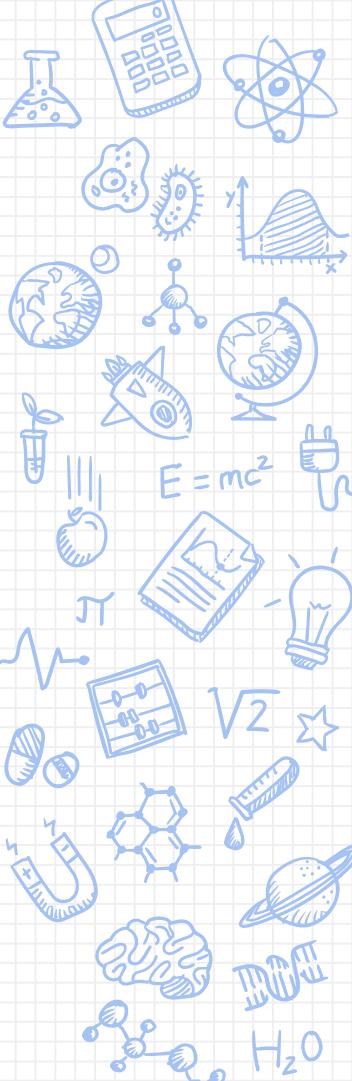
Private View Public View Channel Settings Sharing API Keys

Add Visualizations Add Widgets Export recent data

Channel Stats
Created: less than a minute ago
Entries: 0

Field 1 Chart Edit

DHT-11 Reading



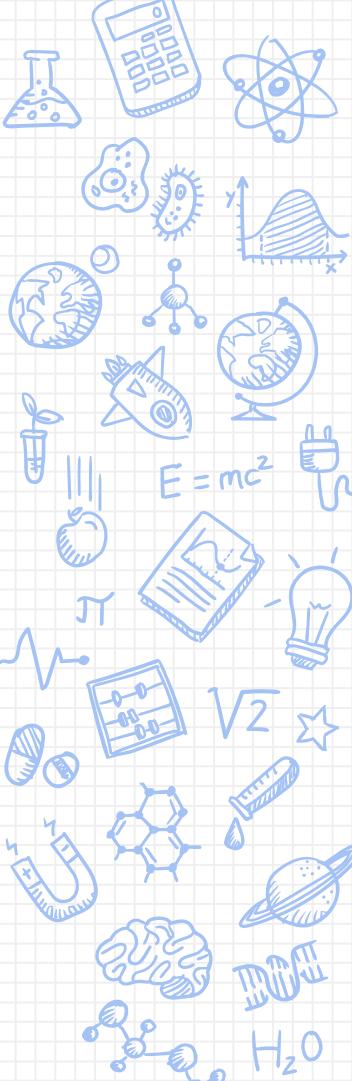
Send DHT-11 sensor Data to ThingSpeak.

Step 10 – You can give a title to your chart, customize the background color, x and y axis, and much more.

Field 1 Chart Options

Title:	DHT-11 Humidity	Timescale:	Average
X-Axis:	Timestamp	Median:	
Y-Axis:	Humidity	Sum:	
Color:	#D62020	Rounding:	
Background:	#FFF	Data Min:	
Type:	line	Data Max:	
Dynamic?:	true	Y-Axis Min:	
Days:		Y-Axis Max:	
Results:	60		

Save Cancel



Send DHT-11 sensor Data to ThingSpeak.

Step 11 – To send values from the Node MCU to ThingSpeak, you need the Write API Key. Open the “API Keys” tab and copy the Write API Key.

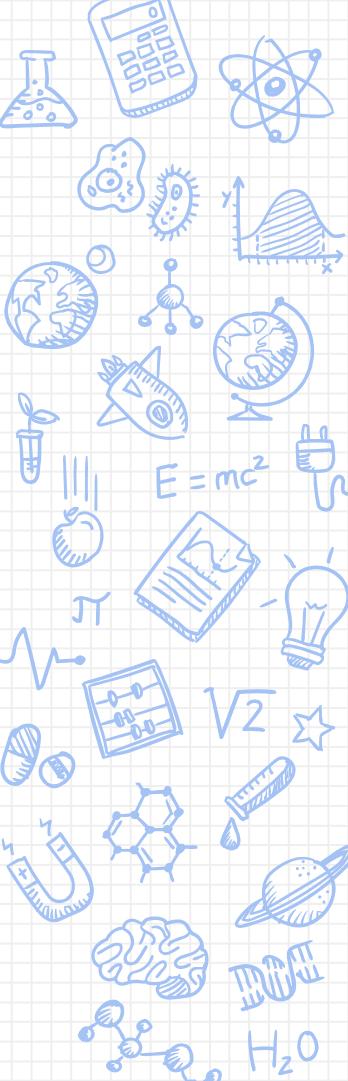
Private View Public View Channel Settings Sharing **API Keys**

Write API Key

Key **X8KSMRXT1XZ61GI2**

Generate New Write API Key

A screenshot of a web interface showing the "API Keys" tab selected. The tab bar includes "Private View", "Public View", "Channel Settings", "Sharing", and "API Keys". Below the tabs, the title "Write API Key" is displayed. A red box highlights the "Key" field, which contains the value "X8KSMRXT1XZ61GI2". At the bottom, a red box highlights the "Generate New Write API Key" button.



Send DHT-11 sensor Data to ThingSpeak.

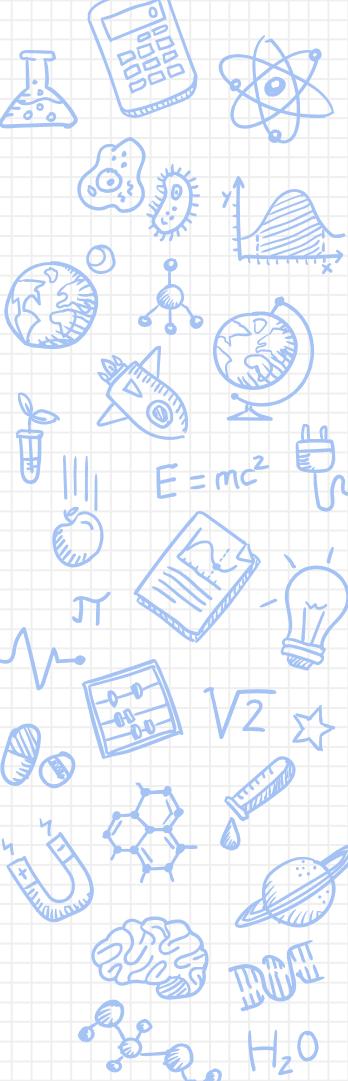
Program

```
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"
#include <Adafruit_BMP280.h>
#include <DHT.h>

const char* ssid = "xxxxxxxxxx"; // Name of the Network.
const char* password = "xxxxxxxxxx"; // Password of the Network.

WiFiClient client; // Create a Wi-Fi client to connect to ThingSpeak.

unsigned long myChannelNumber=1; // Channel No
const char * myWriteAPIKey="xxxxxxxxxx"; // API write key.
```



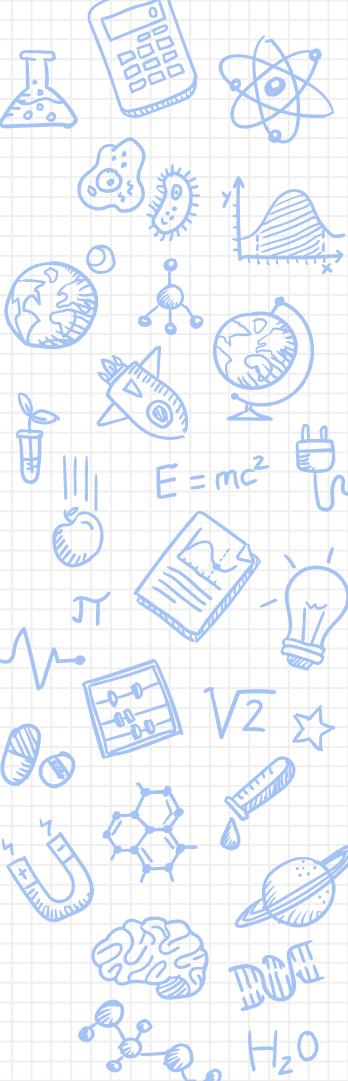
Send DHT-11 sensor Data to ThingSpeak.

```
#define DHTPIN D5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password); // This will connect the Node MCU with WiFi.
  WiFi.mode(WIFI_STA); // Set the Node MCU as a Wi-Fi station.
  ThingSpeak.begin(client); // Initialize ThingSpeak.

  dht.begin();
}

void loop() {
  while (WiFi.status()!=WL_CONNECTED) {
    Serial.print(".");
  }
```



Send DHT-11 sensor Data to ThingSpeak.

```
delay(500);
}

float h = dht.readHumidity();
Serial.println("\nConnected.");
ThingSpeak.writeField(myChannelNumber, 1, h, myWriteAPIKey);
//the channel number.
//the field number (in our case, we just have one field).
//the value you want to publish (x).
//your write API key.
}
```

Send DHT-11 sensor Data to ThingSpeak.

Node_MCU_Thingpeak_code | Arduino 1.8.19

File Edit Sketch Tools Help



```
1 // Node_MCU_Thingpeak_code | Arduino 1.8.19
2
3 // This sketch connects a NodeMCU to WiFi and sends DHT11 sensor data to ThingSpeak.
4 // It uses the Adafruit_BMP280 library for pressure and temperature.
5 // It also includes a basic DHT11 library.
6
7 // You will need to change the SSID and password to your WiFi network.
8 // You will also need to change the myWriteAPIKey to your own ThingSpeak API key.
9
10 // WiFi client to connect to Thingpeak.
11
12 // Channel Number
13 // WiFi Write API Key
14
15 #define DHTPIN D5
16 #define DHTTYPE DHT11
17 DHT dht(DHTPIN, DHTTYPE);
18
19 void setup() {
20   Serial.begin(115200);
21   WiFi.begin(ssid, password); // This will connect the NodeMCU with WiFi.
22   WiFi.mode(WIFI_STA); // Set the NodeMCU as a WiFi station.
23   ThingSpeak.begin(client); // Initialize ThingSpeak.
24
25   dht.begin();
26 }
27
28 void loop() {
29   while (WiFi.status() != WL_CONNECTED) {
30     Serial.print(".");
31   }
32
33   delay(500);
34
35   float h = dht.readHumidity();
36   Serial.println("\nConnected. ");
37   ThingSpeak.writeField(myChannelNumber, 1, h, myWriteAPIKey);
38   //the channel number.
39   //the field number (in our case, we just have one field).
40   //the value you want to publish (x).
41   //your write API key.
42 }
```

Done Saving

```
31   delay(500);
32
33   float h = dht.readHumidity();
34   Serial.println("\nConnected. ");
35   ThingSpeak.writeField(myChannelNumber, 1, h, myWriteAPIKey);
36   //the channel number.
37   //the field number (in our case, we just have one field).
38   //the value you want to publish (x).
39   //your write API key.
40 }
```

Done Saving

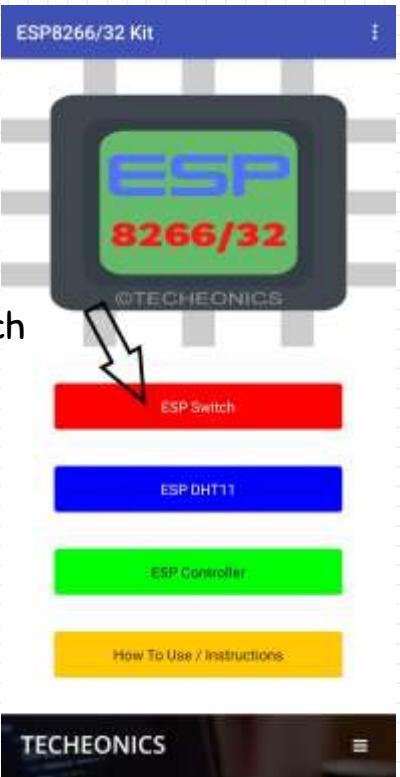


Download mobile application for ESP 8266/32.

Download ESP 8266/32 KIT APP –

[GitHub - gauravk5/ESP-8266-32-KIT-APP: ESP 8266/32 KIT APP](#)

HOW TO USE ESP SWITCH FEATURE

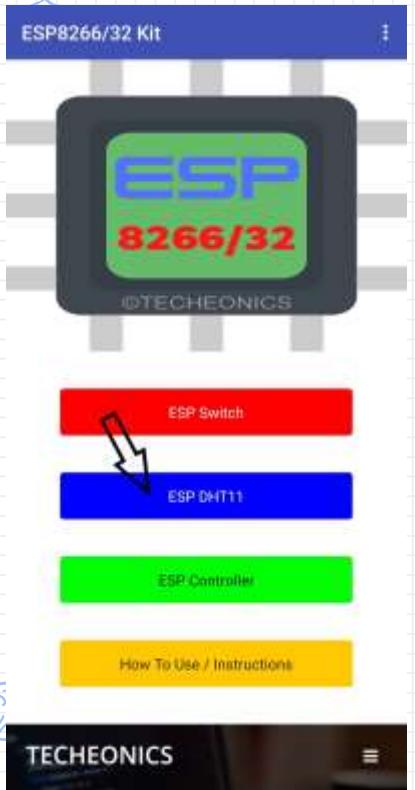


To use the switch
feature click on
the ESP Switch
button

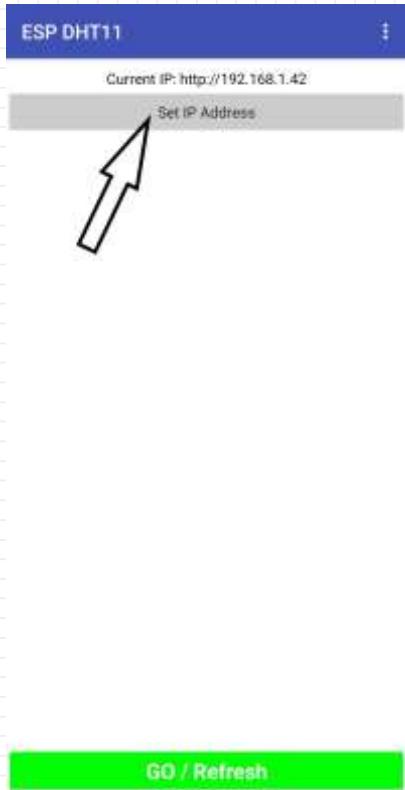


Now put the IP
address of you
Node MCU and Save
it.

HOW TO USE ESP DHT11 FEATURE



To use the Sensor Dashboard feature click on the ESP DHT11 button



Now Click on the Set IP Address Button



Now put the IP
address of your Node
MCU and Save it.



Click on the GO button



ESP8266 DHT Sensor Output

Temperature 25.30 °C

Humidity 48.00 %

Powered By Techeonics

GO / Refresh

Final Result
Dashboard
Sensor Data
Output



Any Questions

??

Reference - <https://techeonics.com/>

Projects - https://techeonics.com/?page_id=9

Youtube - <https://www.youtube.com/c/THETECHEONICS>

Contact us-

Gaurav Kumar - <https://linktr.ee/Gauravk5>

Shivam - <https://www.linkedin.com/in/shivam-4267a31a8/>

