Session 5

Agenda

1

What is IOT ?

- The History of IoT
- Main categories of IoT

2

How IOT work ?

- The mobile value chain
- The IoT value chain

3

Types of ESP & NodeMCU

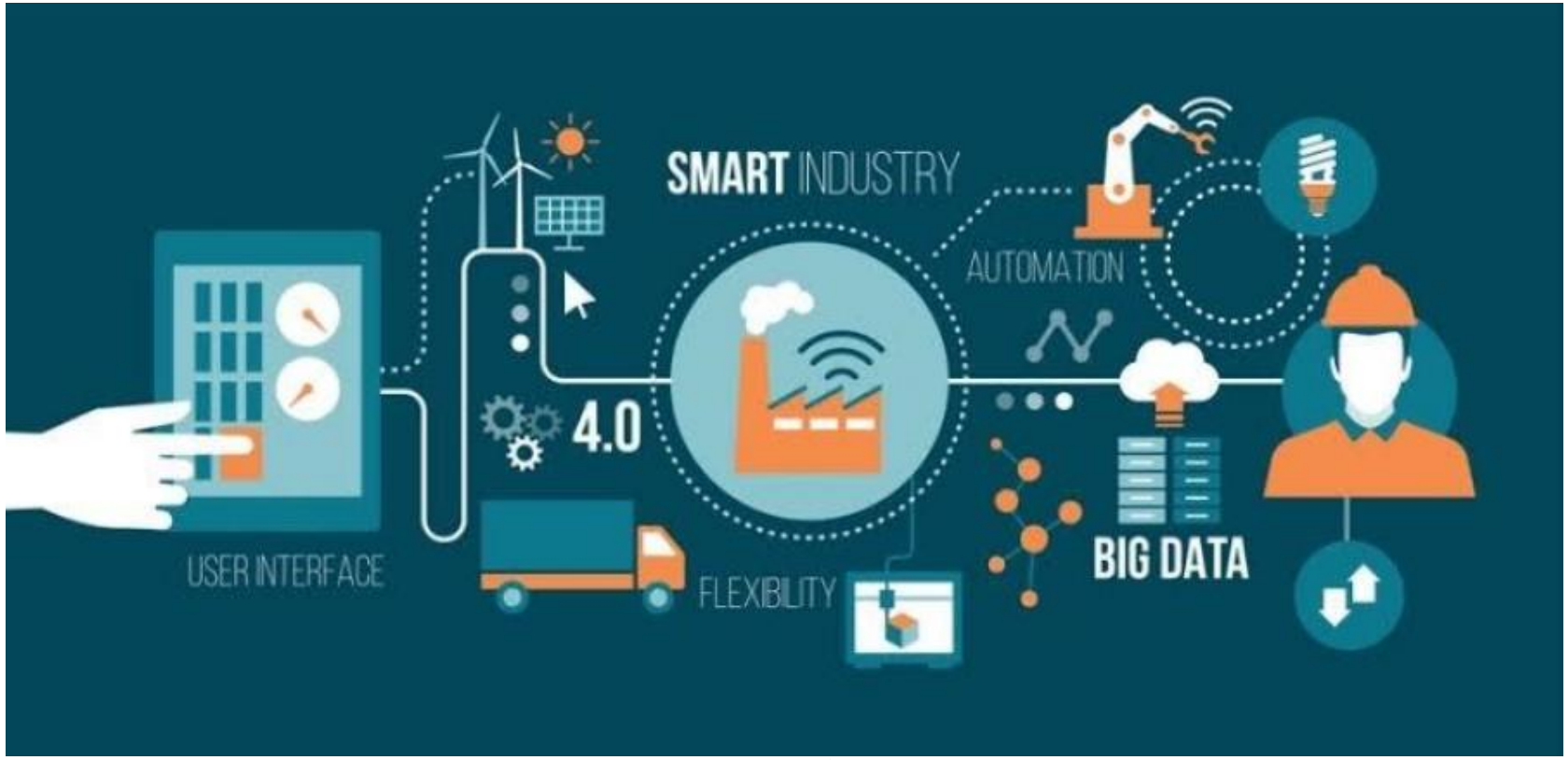
- Types of ESPs modules
- NodeMCU
- Comparing Arduino and NodeMCU

4

NodeMCU

- Pinout
- How to connect ESP and NodeMCU
- Install ESP libraries

Main categories of IoT



1- Home consumer

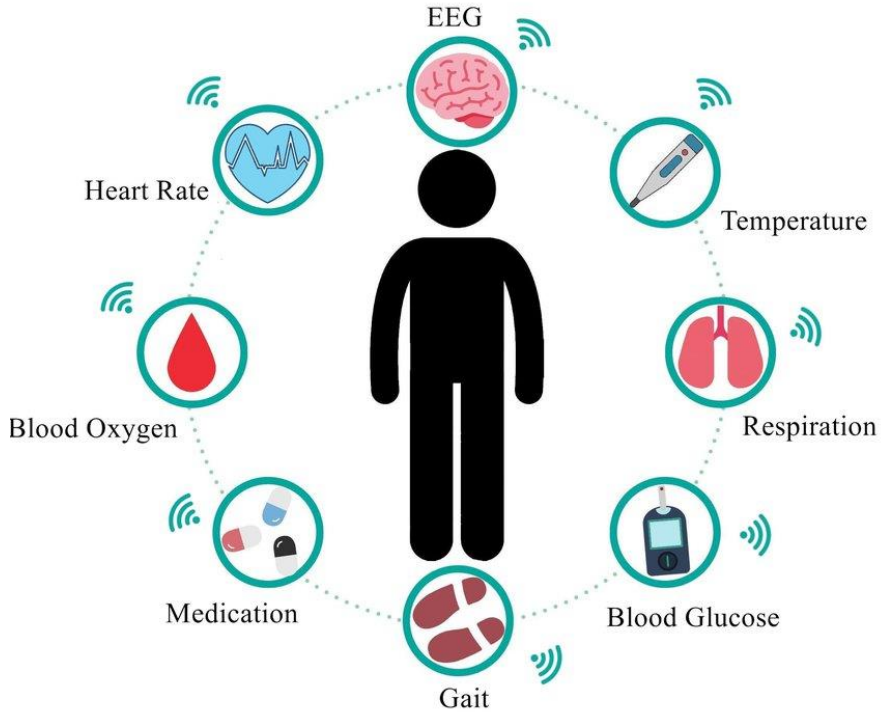
- **Serving many devices**
- **Low revenues**

2- Cities/Industry

- **High revenues**

3- Health Body

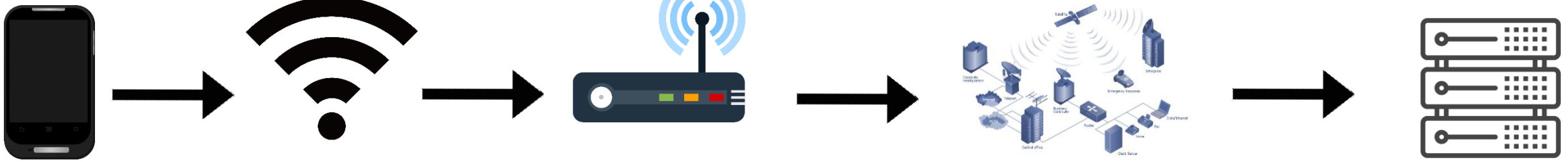
- **Devisees requires long life batteries**
- **Serving few devices**



4- Transport Mobility

- **Challenge : location based services.**

Mobile app value chain



Device

Wifi

Wifi router

Carrier Network

Server

IoT value chain

Sensor Device

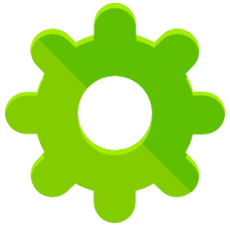
Wireless Sensor Networks

IoT Gateways

Carrier Networks

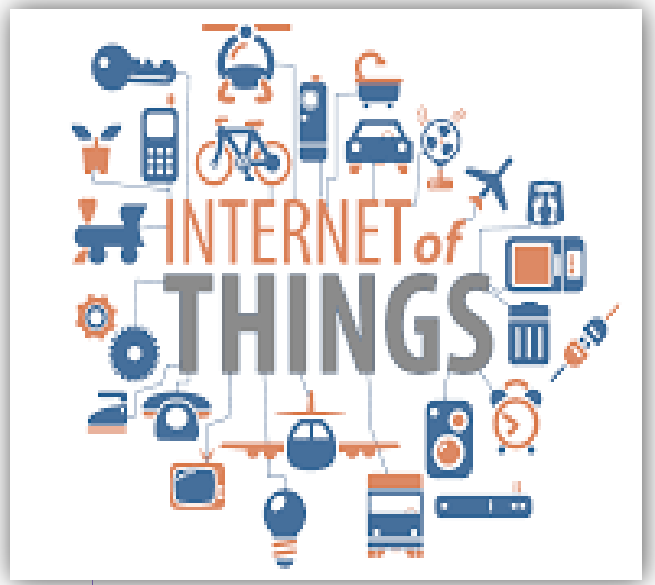
Application Enablement Platforms

Big Data Analytics Platforms



Device and Gateway remote management

IoT with hardware



Types of ESP

Types of ESP

- **Now a days ESP8266 chip based various kind of module boards are available. We will talk about AI-Thinker modules which are named as ESP-01 to ESP-13. NodeMCU boards extend upon the AI-Thinker modules.**

Types of ESP

- **ESP modules mainly differ in:**
  - **physical layout**
  - **form factor flash**
  - **Wi-Fi antenna: ceramic or on PCB**

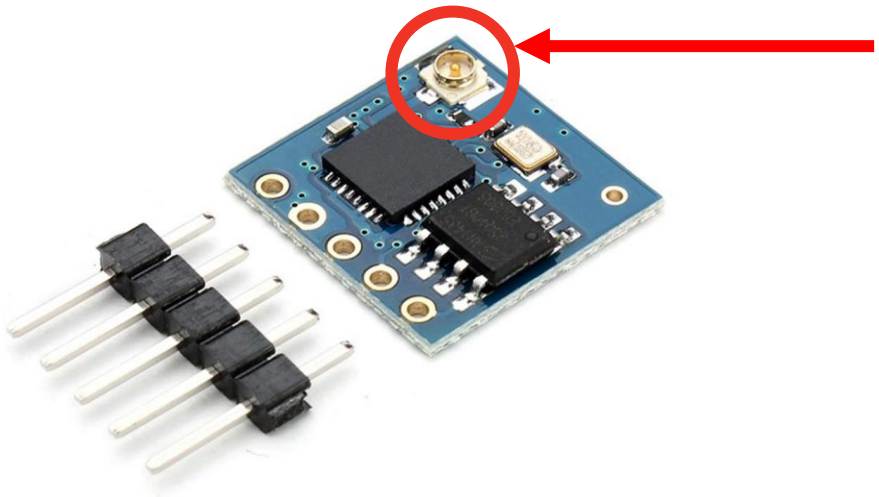
ESP-01

General-purpose input/output (GPIO) is a pin on an IC. It can be either an input pin or output pin, whose behavior can be controlled at the run time

Firmware is a specific class of computer software that provides the low-level control for a device's specific hardware. AT firmware is a special software for ESP modules

- This is probably one of the most popular modules
- It has 2 GPIO pins
- It comes with one of the different versions of the AT firmware
- One of the disadvantages of this module is the placement of the pins

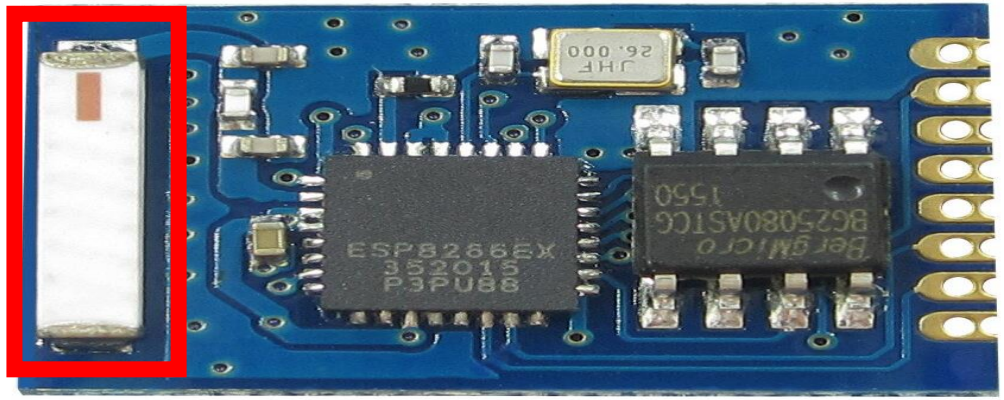
ESP-05



- It used as mini-Wi-Fi shield together with Arduino
- It only requires two wires (TX/RX) to communicate between a microcontroller (e.g. Arduino) and Wi-Fi
- This module has no antenna
- Has no GPIO pins
- Not all boards come with the same AT firmware version

ESP-11

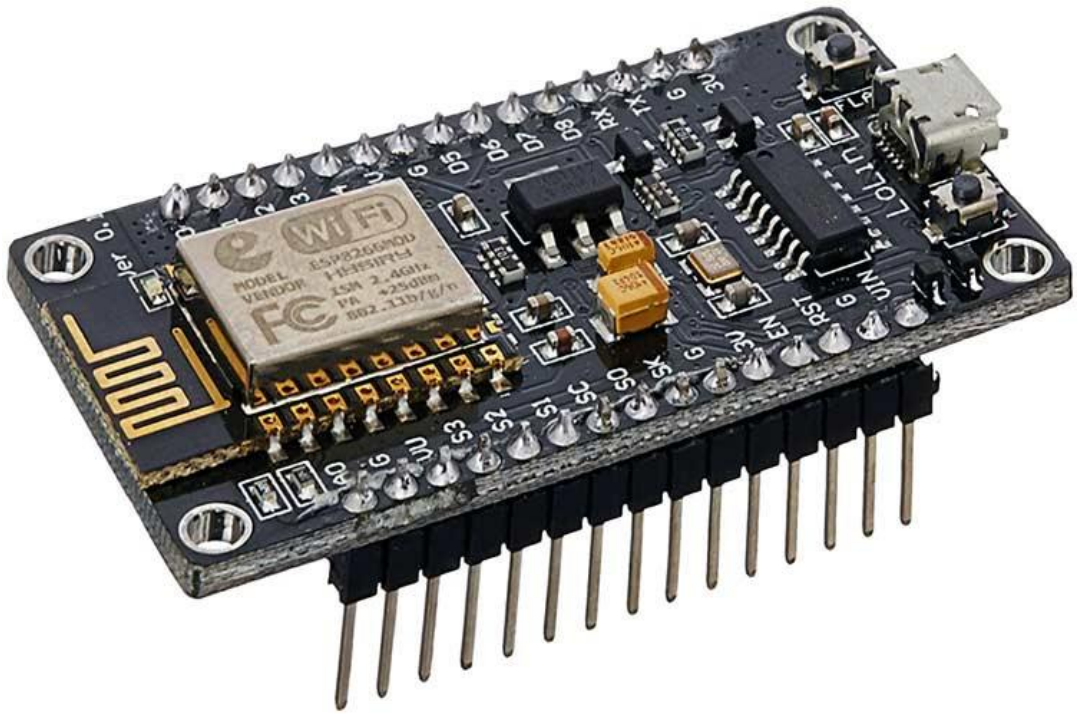
- **This module comes with ceramic antenna**



ESP-12

- It has 11 GPIO pins
- one analog-to-digital converter (ADC) with a 10-bit resolution
- The antenna is a track on the PCB which delivers good results for Wi-Fi sensitivity
- The difference between ESP-12 versions are in certificates, number of pins, or the patterns of the PCB trace antenna.

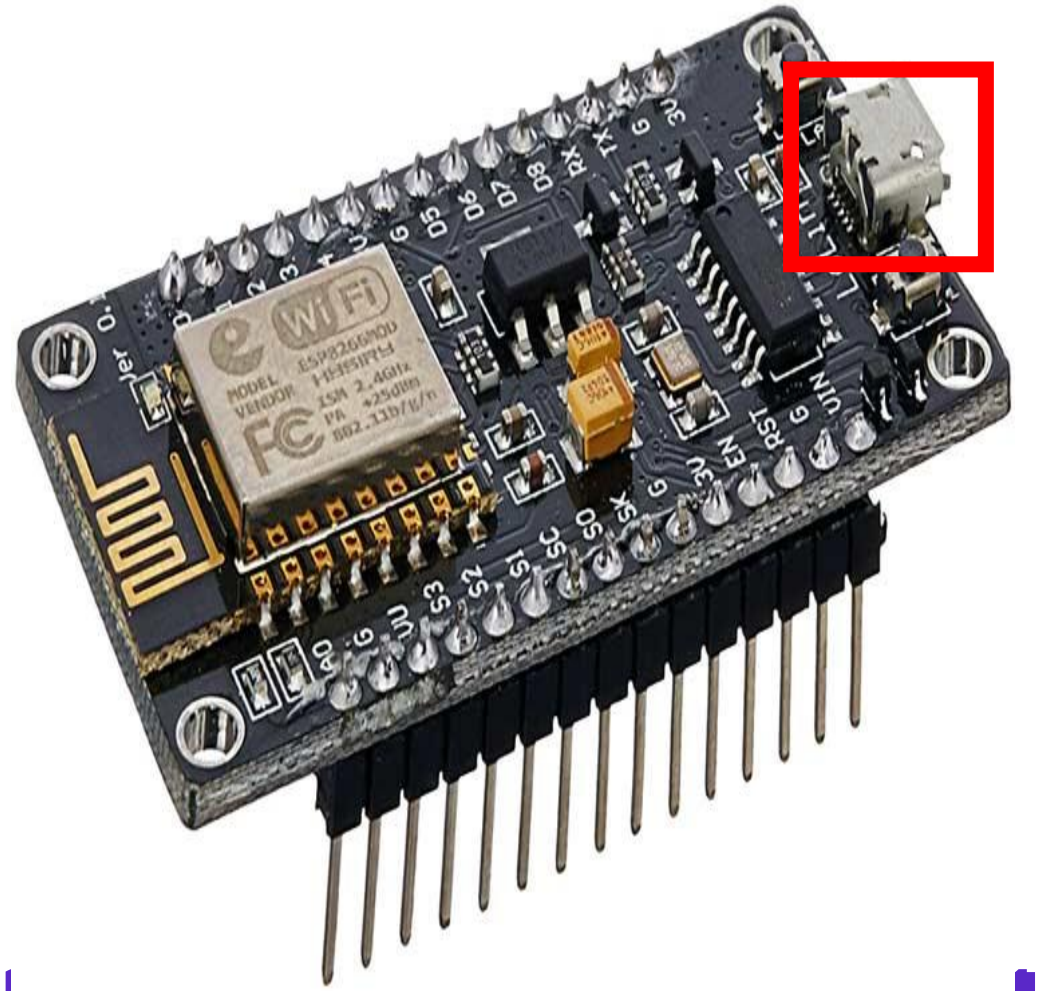
NodeMCU



- **NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications.**

- **It provides some of the most important features of microcontrollers such as GPIO, PWM, ADC (containing 13 GPIO pins, 10 PWM channels, I2C, SPI, ADC, UART).**

NodeMCU

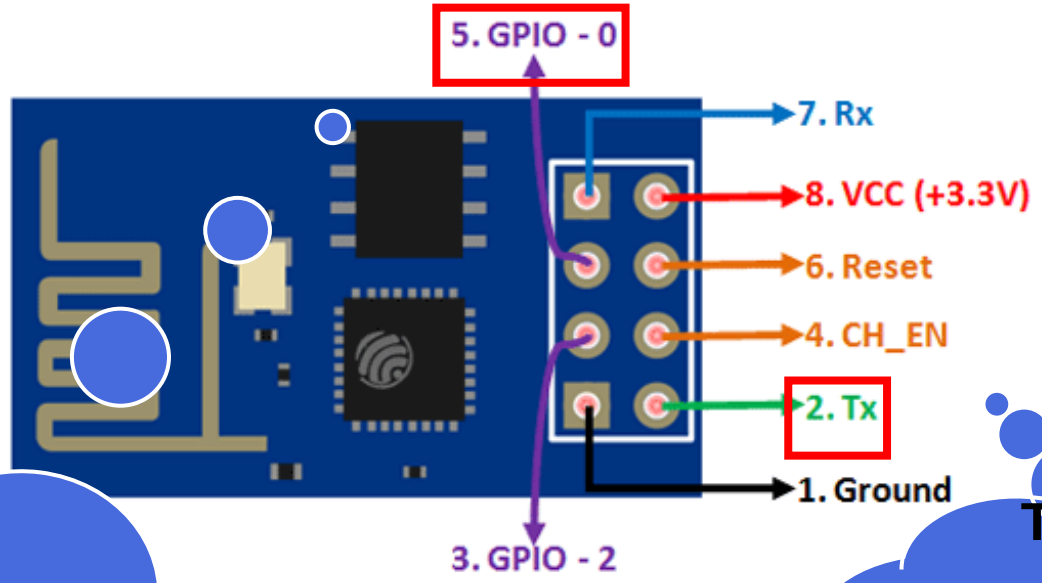


- **Unlike ESP NodeMCU comes with built-in serial-to-USB adapter and comes with a micro-USB plug for power supply and for programming the module.**

- ❖ **Operating Voltage: 3.3V**
- ❖ **Input Voltage: 7-12V**

How to pin out ESPs & **NodeMCU**?

ESP-01 pinout



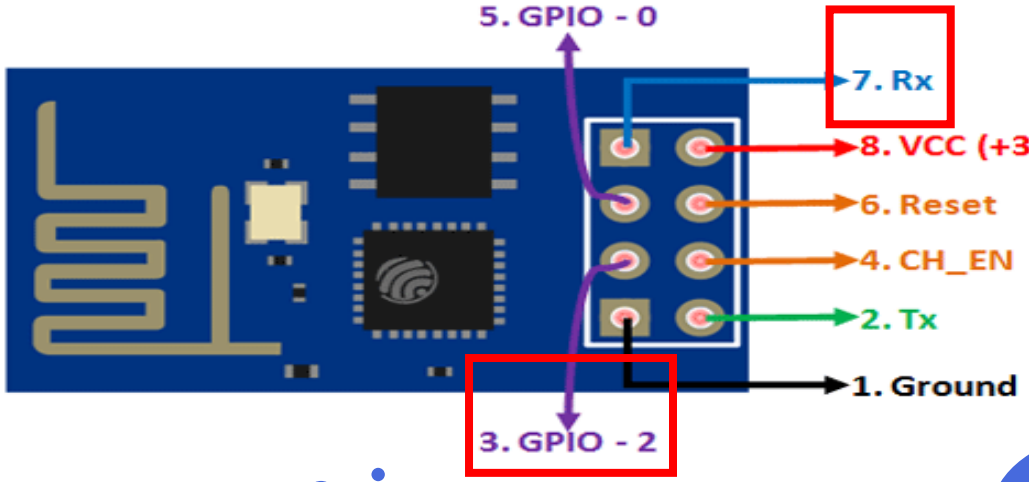
GPIO-0 (Flash)

- **Normal use:** GPIO pin
- **Alternate use:** Takes module into serial programming when held low during start up

TX (GPIO-1)

- **Normal use:** Connected to Rx pin of programmer/uC to upload program
- **Alternate use:** GPIO pin

ESP-01 pinout



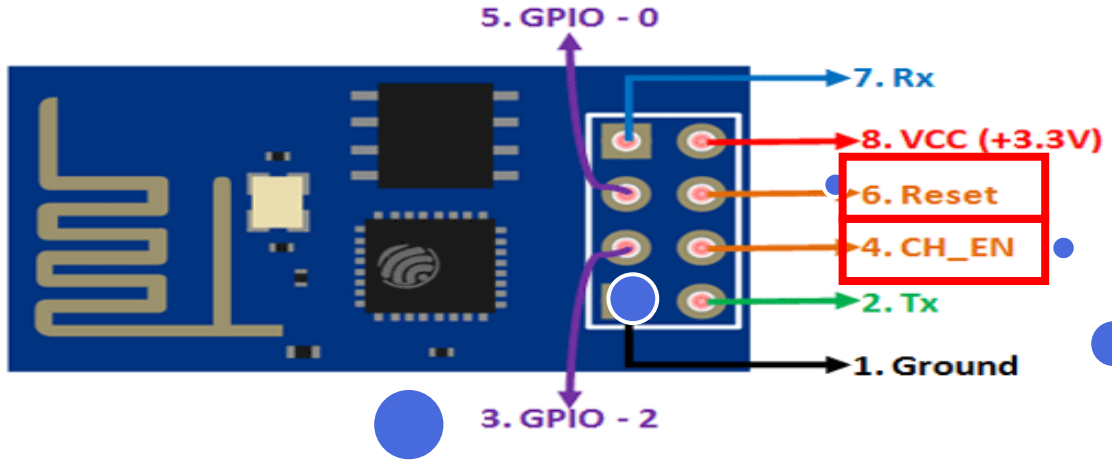
GPIO-2

- **Normal use:** GPIO pin

RX (GPIO-3)

- **Normal use:** Connected to Tx pin of programmer/uC to upload program
- **Alternate use:** GPIO pin

ESP-01 pinout



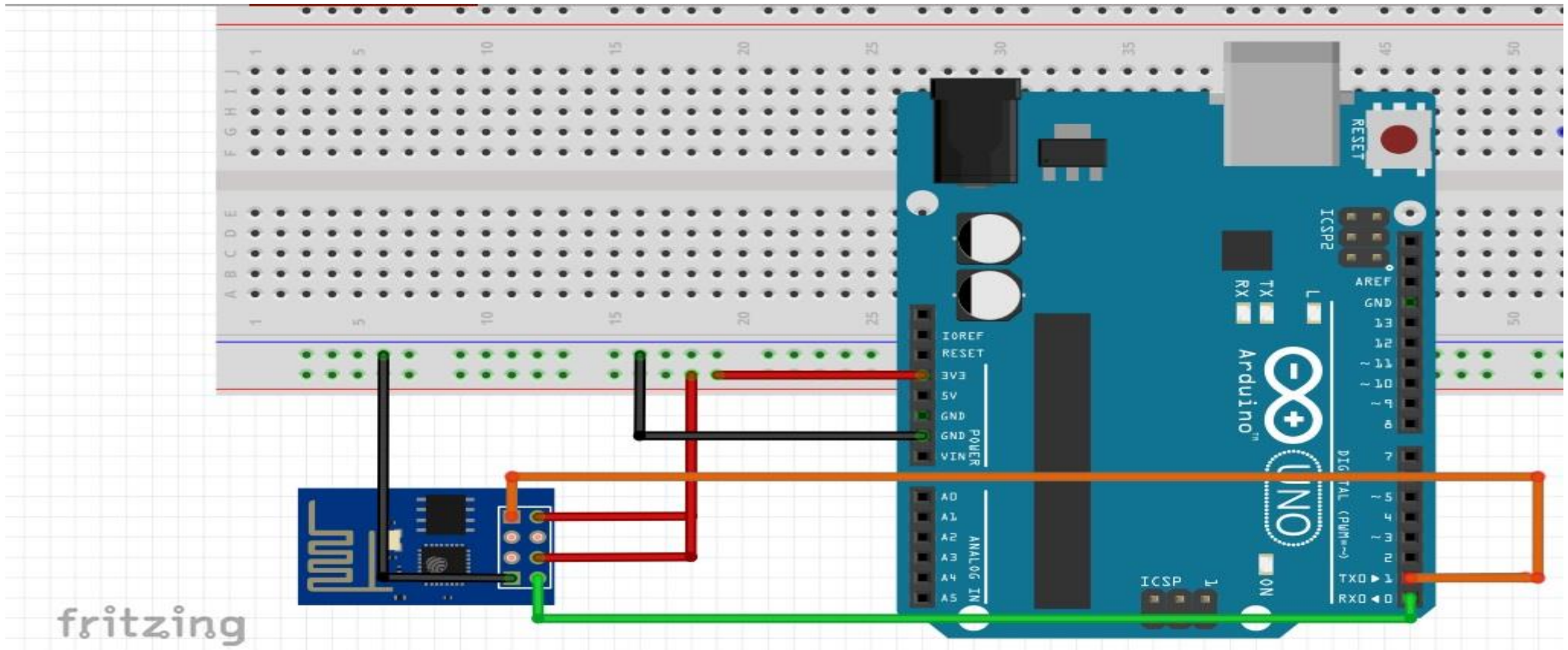
RESET

- **Normal use:** Reset the module

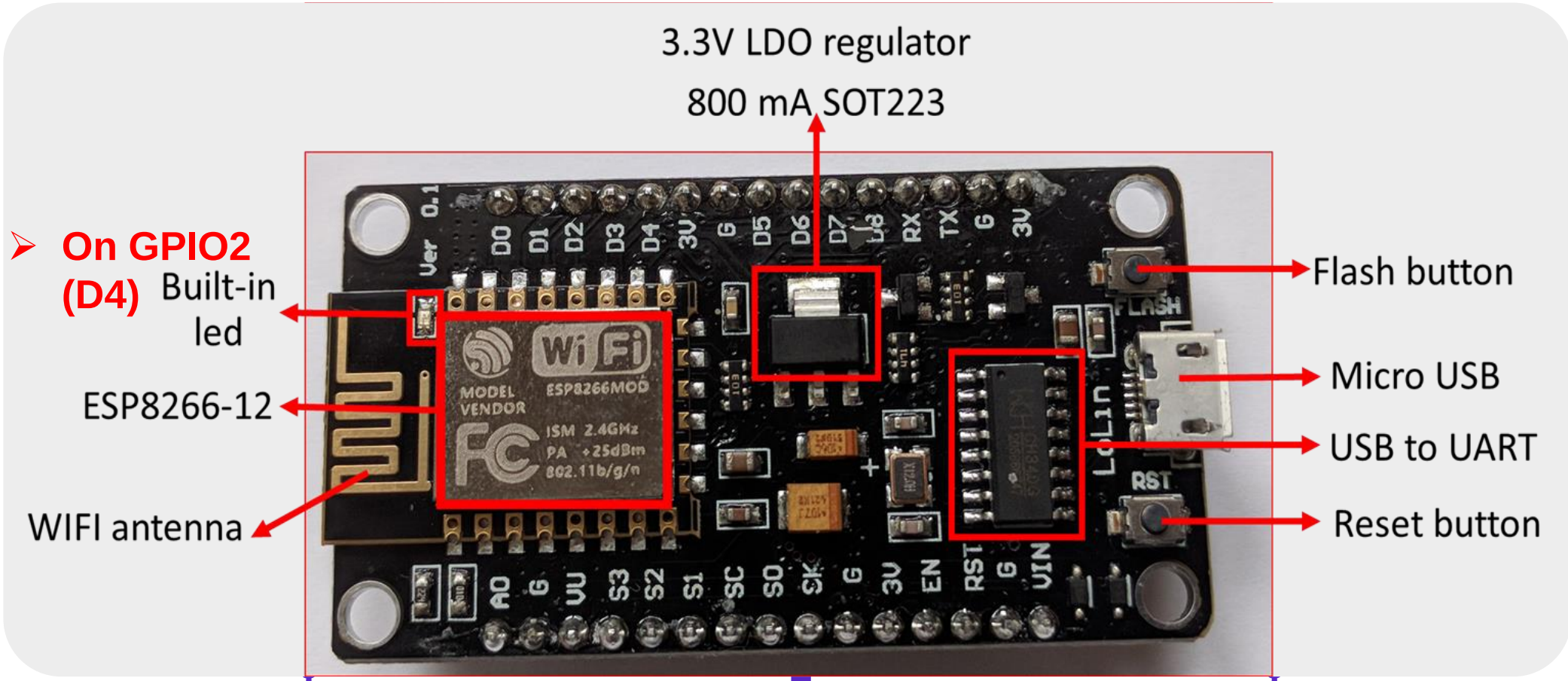
CH-EN

- **Normal use:** Chip Enable – Active high

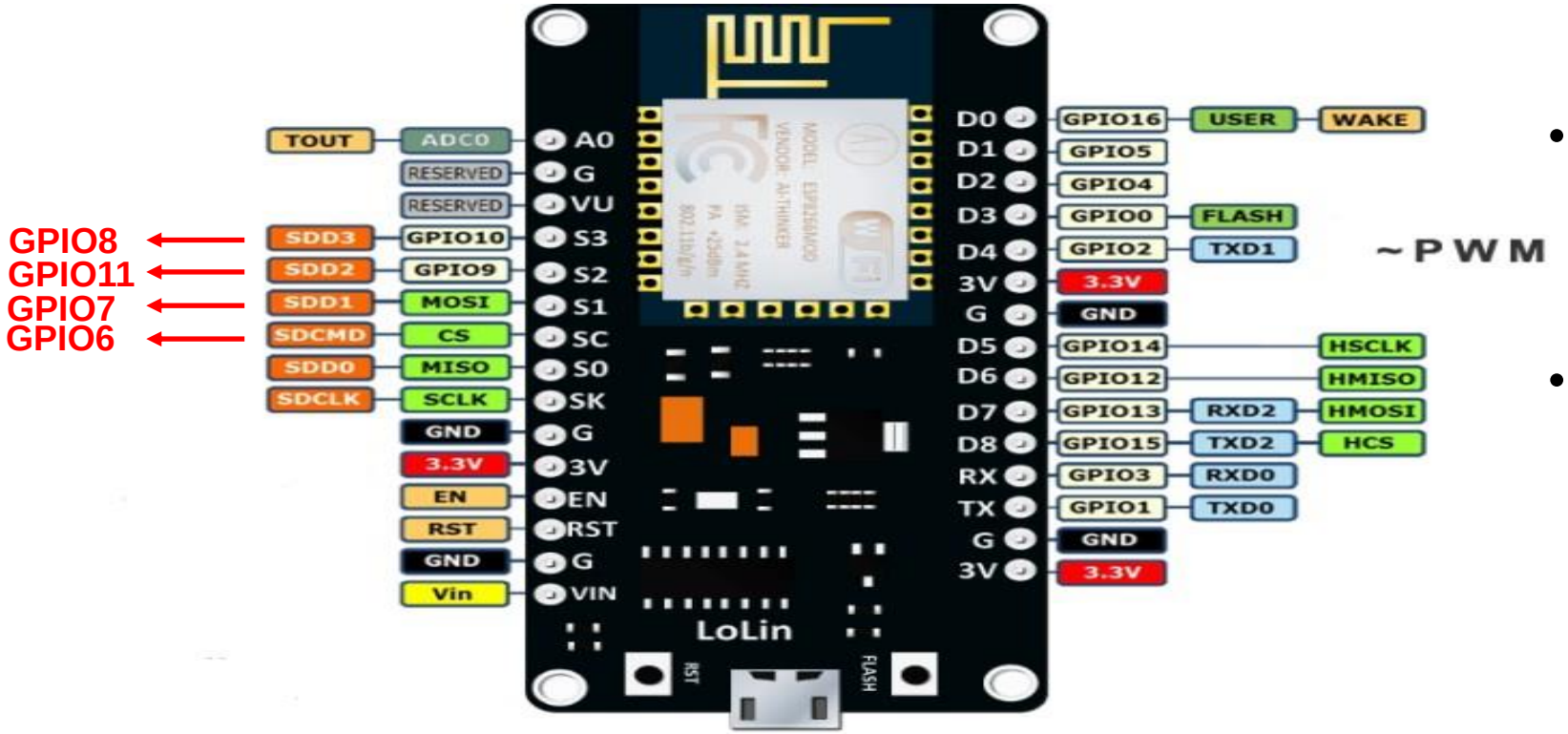
ESP-01 connection to Arduino



NodeMCU pinout

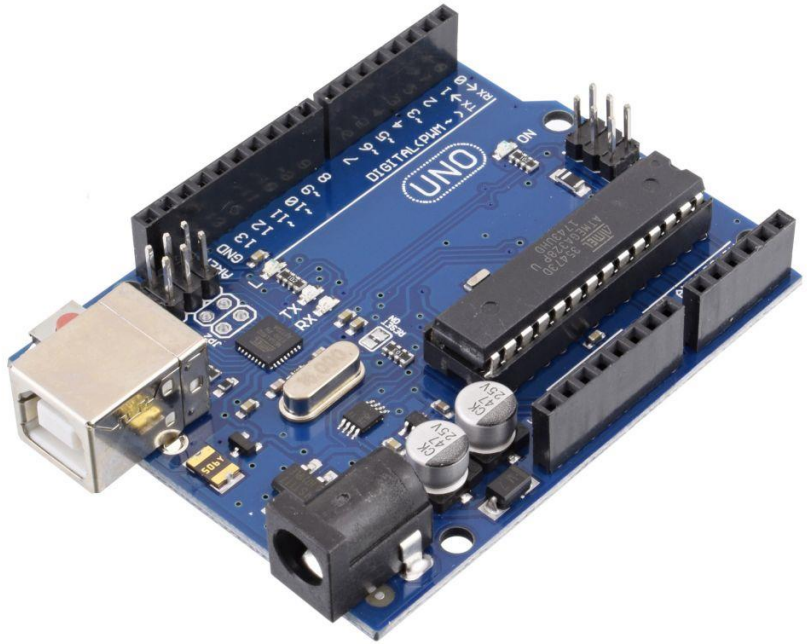


NodeMCU pinout

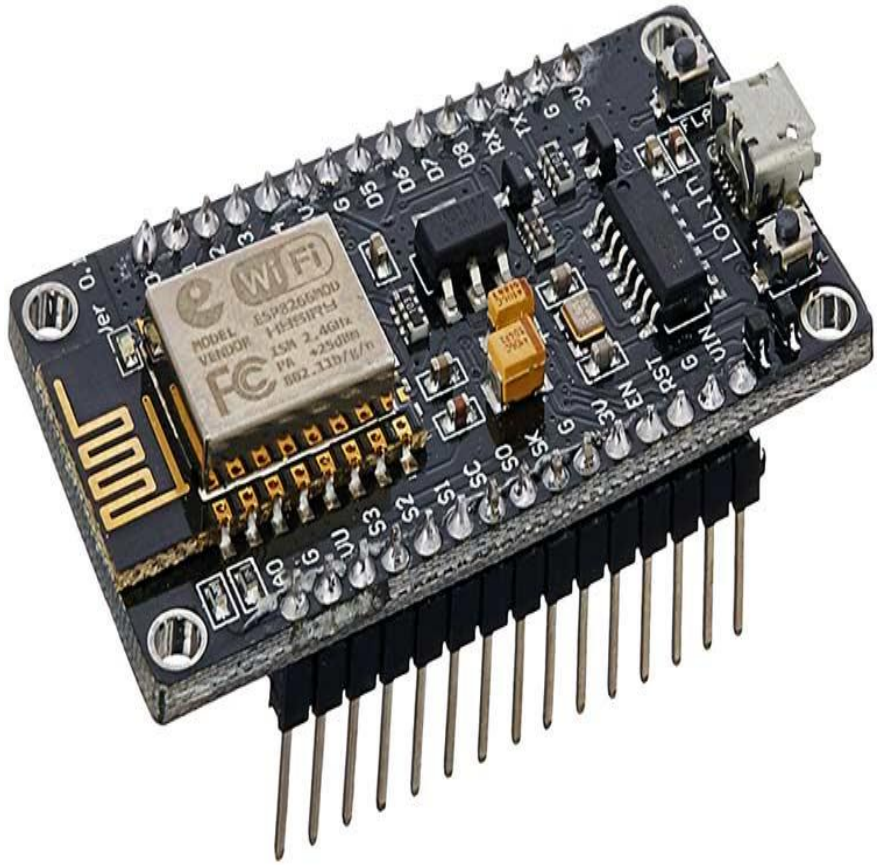


- We can use GPIO pins [0 – 16] (D0 ~ D8, S2, S3) as digital I/O and PWM.
- There is only analog read pin (A0).

Arduino & Nodemcu



VS



Arduino & Nodemcu

GPIO pins :	20	18
Digital pins :	14	16
Analogue input pins :	6	1
PWM pins :	6	16

Arduino & Nodemcu

Output :	**5V / 3.3V**	**3.3V**
Interfacing :	**USB type B**	**Micro USB**
Current Consumption :	**45 mA ~ 80 mA**	**15 µA ~ 400 mA**

Communication Protocols : **SPI - I2C - I2S – UART**

Arduino & Nodemcu

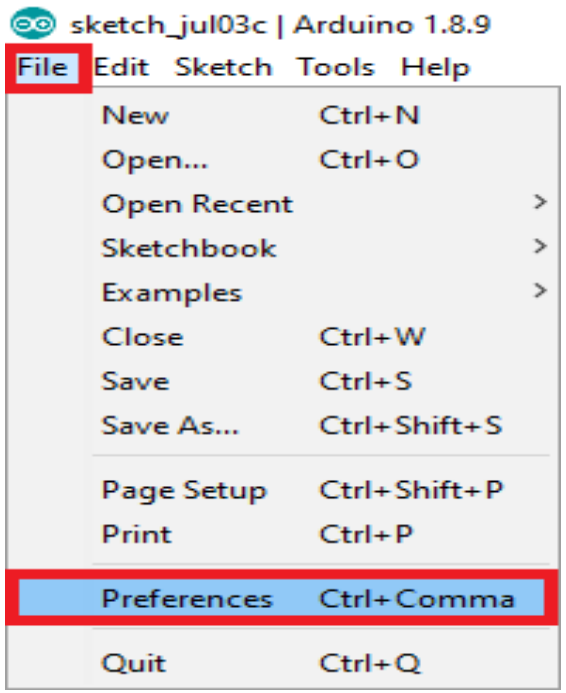
Flash Memory :	32 KB	4 MB
SRAM :	2 KB	64 KB
EEPROM :	1 M	512 B
Clock (Instructions per second) :	16 Hz	52 ~ 160 MHz

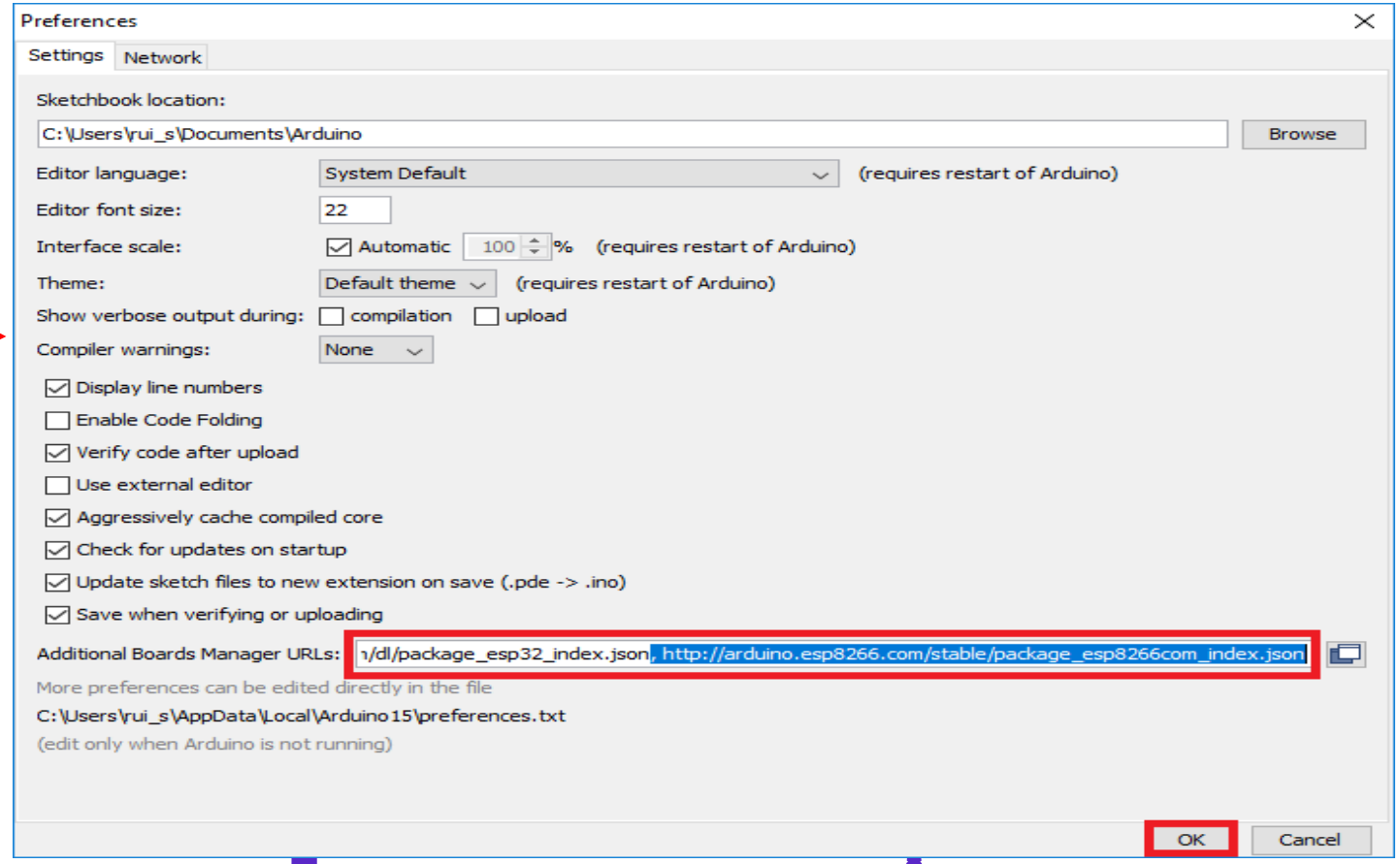
Flash memory

SRAM (static random-access

EEPROM is memory space that programmers can use to store long-term information

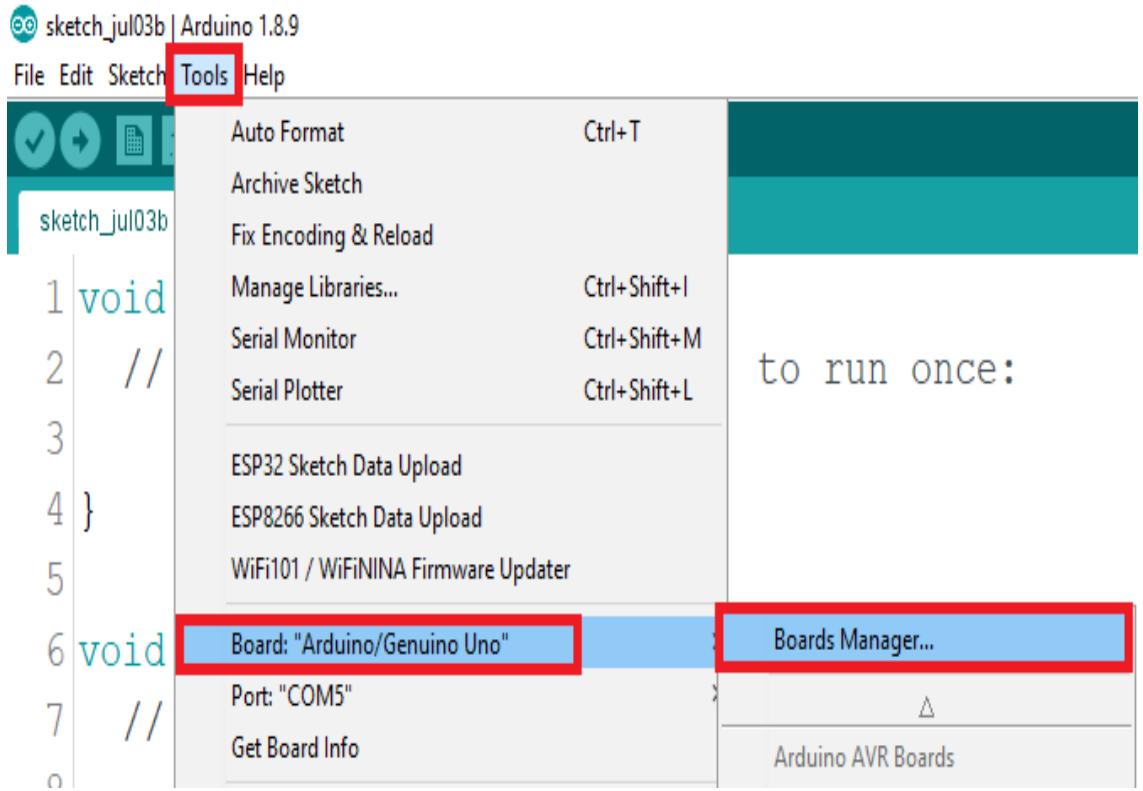
Install ESP Board

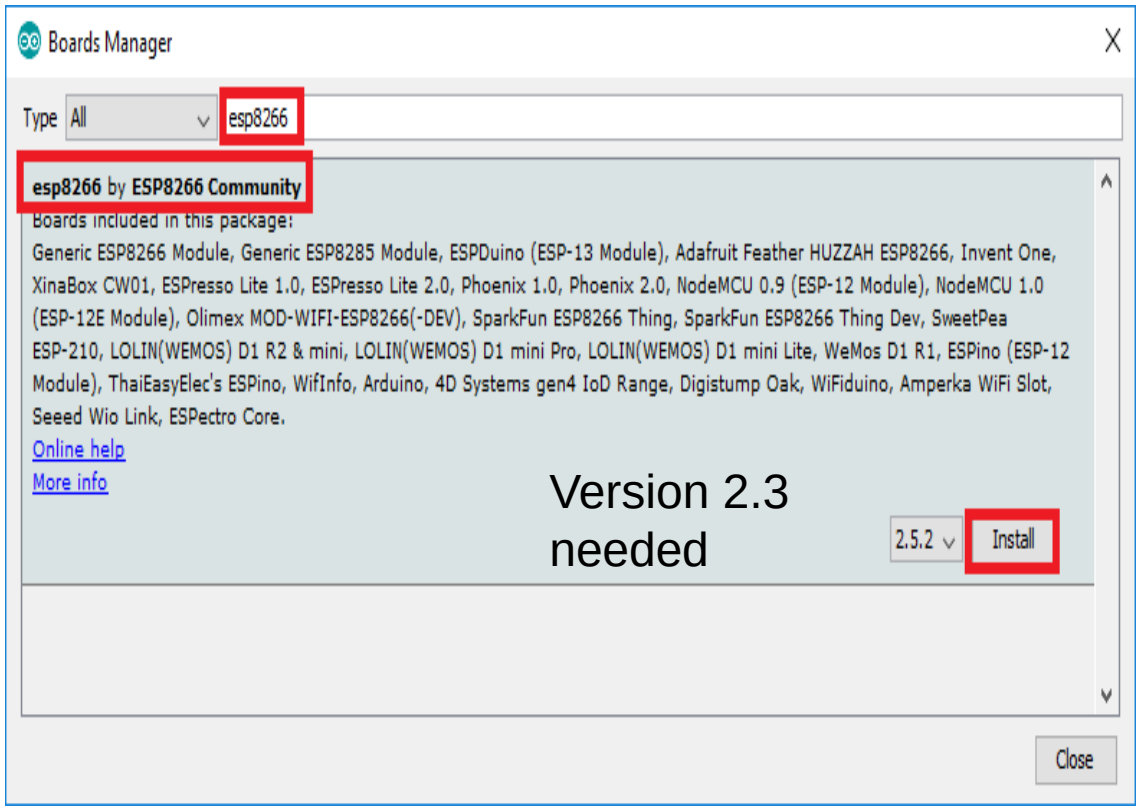




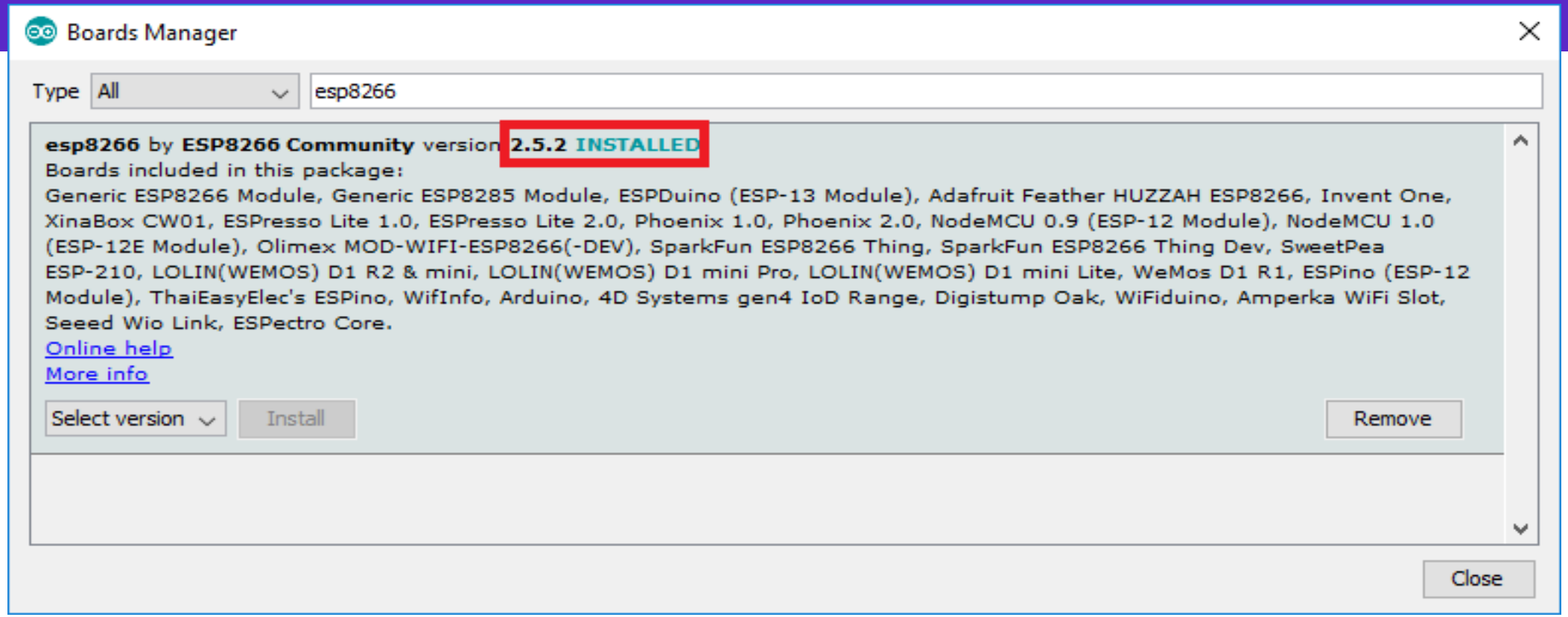
https://arduino.esp8266.com/stable/package_esp8266com_index.json

Install ESP library





Install ESP library



Installed Successfully ☺

Control led on NodeMcu

Control the internal LEDs in nodeMCU

```
void setup() {
  // put your setup code here, to run once:
  pinMode(D0, OUTPUT);// LED on NodeMCU GPIO16
  pinMode(D4, OUTPUT);// LED on ESP GPIO2
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(D0, HIGH);
  digitalWrite(D4, LOW);
  delay(1000);

  digitalWrite(D4, HIGH);
  digitalWrite(D0, LOW);
  delay(1000);

}
```

ESP8266WiFi

WiFi.begin(wifiname,wifipassword):connects to the wifi

WiFi.status().

WiFi.localIP():returns IP address

WIFIUDP

Creates a named instance of the WiFi UDP class that can send and receive UDP messages

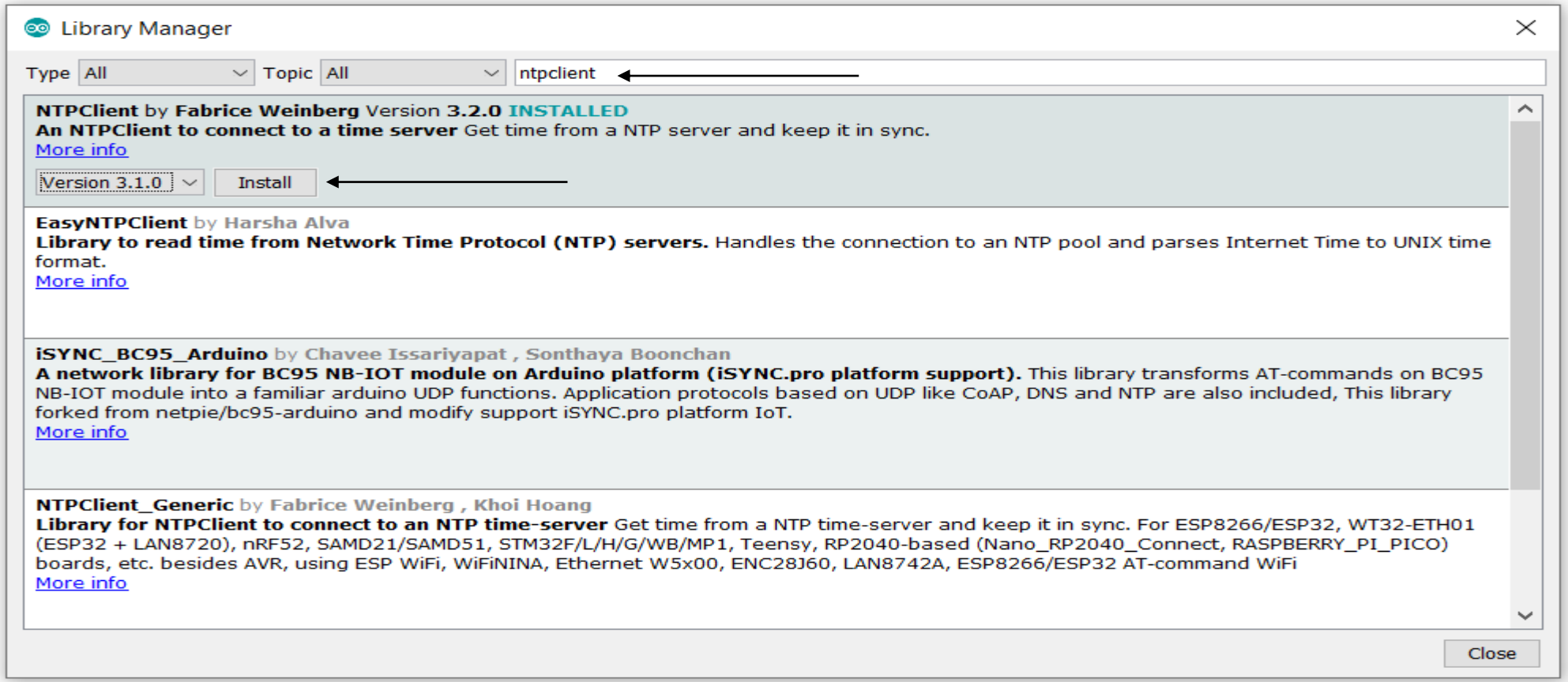
NTPClient

An NTPClient to connect to a time server Get time from a NTP server and keep it in sync.

Install NTPClient library

Install NTPClient library

Install NTPClient library



NTPClient functions

NTPClient time(UDP, Server_name, 7200);
time.begin():Intializes time object
time.update():updates time
time.getEpochTime:returns number of seconds
since 1970
time.getFormattedTime:returns curre

Time difference between gmt in seconds

TimeLib

The Time library adds timekeeping functionality to Arduino with or without external timekeeping hardware

TimeLib functions

setTime(seconds):coverts number of seconds from 1970 to a date
day():returns current day
month():returns current month
year():returns current year

Practice: Time Now

- **We will get the current time from the internet via NTP (Network Time Protocol)**

1. **From Arduino IDE in Manage libraries download NTPClient & WiFiUDP libraries**

2. **Download TimeLib library and include it in Arduino IDE.**

3. **Using National Institute of Standards and** Technology | NIST (time.gov) **we will get the time and date.**

4. **Code**

Practice: Time Now

Time_Now

```
#include <ESP8266WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <TimeLib.h>

char ssid[] = "TE-Data-2CBDF1";
char pass[] = "27419711";

WiFiUDP udp;
NTPClient timeClient(udp,"time.nist.gov",7200);
void setup() {

  Serial.begin(115200);

  WiFi.begin(ssid,pass);

  while(WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print('.');
  }
   Serial.println("Connected");
   Serial.println(WiFi.localIP());

   timeClient.begin();
}
```

```
void loop() {

    timeClient.update();
    Serial.println("Time Now: " + timeClient.getFormattedTime());

    unsigned long sec = timeClient.getEpochTime();
    setTime(sec);

    String Date = String(day()) + " / " + String(month()) + " / " + String(year());
    Serial.println("Date Now: " + Date);

    delay(1000);

}
```

The End

THANK YOU