

# ESP Settings

## *Manual*

---

Ver 1, July 2025



## Table of Contents

1. Ordering Information .....	1
2. ESP Arduino IDE settings .....	2
3. ESP Pin Configuration .....	5
4. Working Reference Code .....	7

# Chapter 1. Ordering Information

S.no	Components	Comment
1	ESP	<a href="#">NodeMCU Mini (200Rs)</a>
2	Additional Information	<ul style="list-style-type: none"> <li>• <a href="https://lastminuteengineers.com/wemos-d1-mini-pinout-reference">lastminuteengineers.com/wemos-d1-mini-pinout-reference</a></li> <li>• <a href="https://randomnerdtutorials.com/esp8266-pinout-reference-gpios">randomnerdtutorials.com/esp8266-pinout-reference-gpios</a></li> <li>• The Reason for Selecting NodeMCU Mini is, It is Compact.</li> <li>• It has all the Basic Features for Normal Usage</li> <li>• It has single UART Tx and Rx for Flashing and Communication.</li> <li>• During flashing, connect the UART TX &amp; RX only to COM Port.</li> <li>• EN Pin must be always High, Some ESP doesnot have EN Pin.</li> <li>• GPIO0-D3 Must be Low During UART Boot Mode, After Reset.</li> <li>• GPIO2-D4 &amp; GPIO15-D8 Can be High/Low during Flashing.</li> <li>• For Safety, Keep GPIO2 &amp; GPIO16-D0 as High and GPIO15 Low.</li> </ul>



ROBOCRAZE **BEST PRICE GUARANTEED**

## ESP-12 D1 Mini V2 Board

★★★★★ 16 reviews | No questions

~~Rs. 244~~ **Rs. 189** Sale 23% off

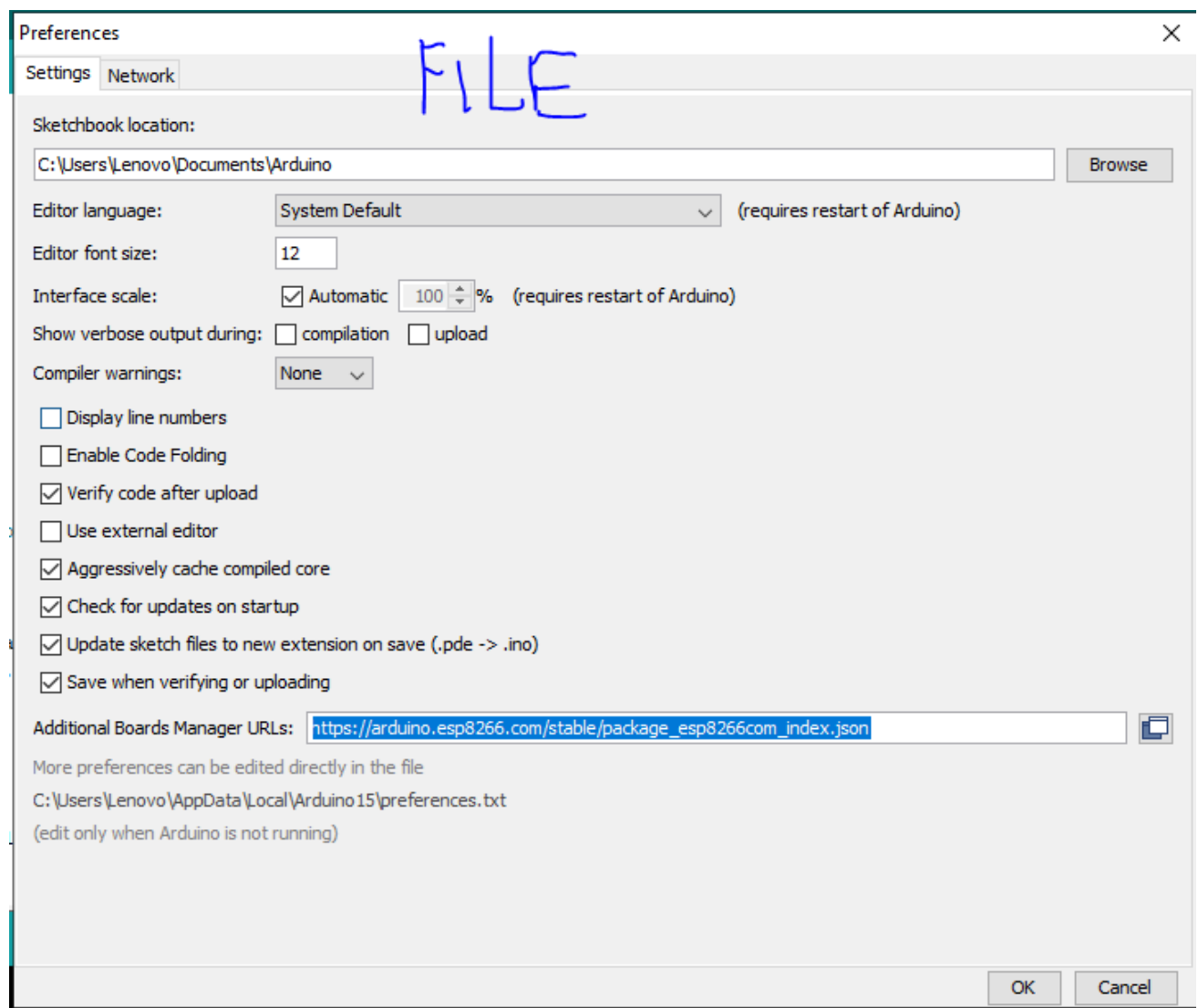
Incl. GST (No Hidden Charges) [Found a better price?](#)

Quantity:  Add To Cart Buy it now ADD TO

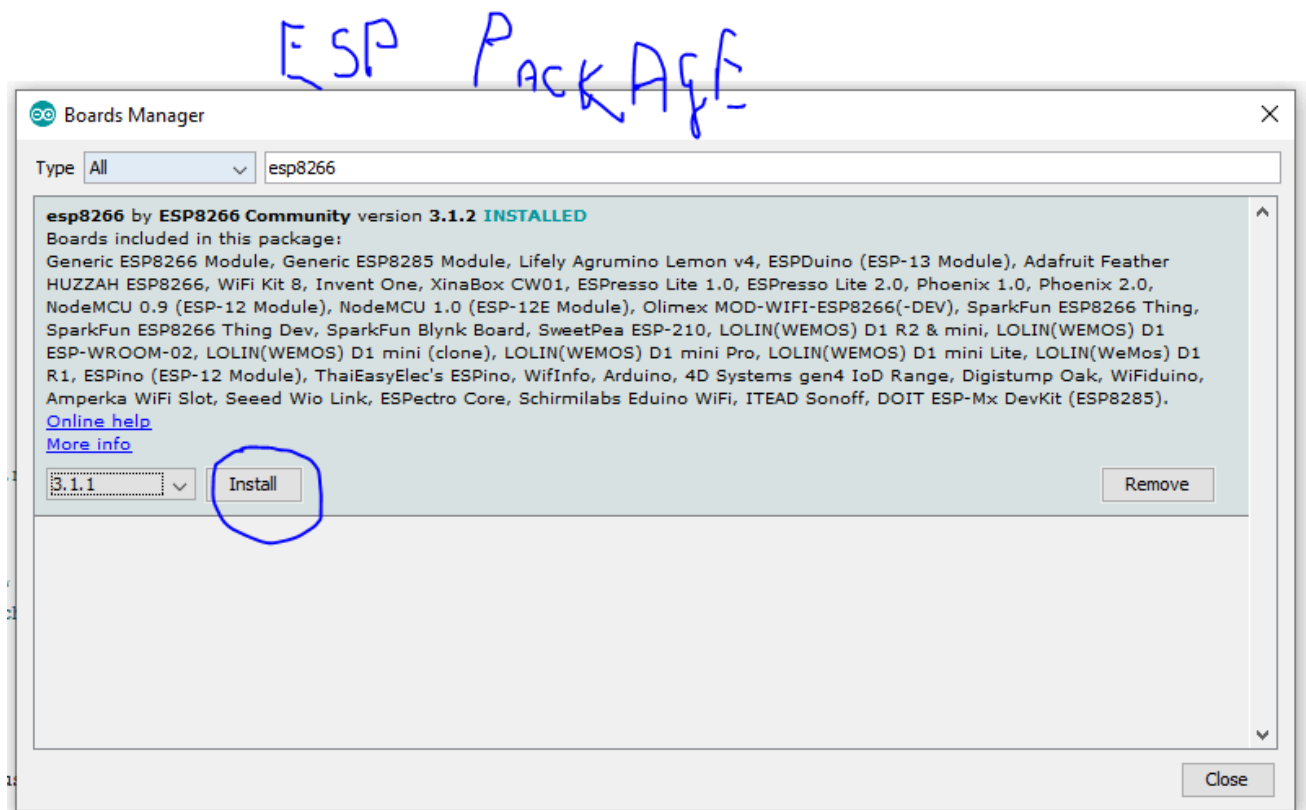
[Submit your Bulk Enquiry Here](#)

Currently, **13** people are actively viewing.

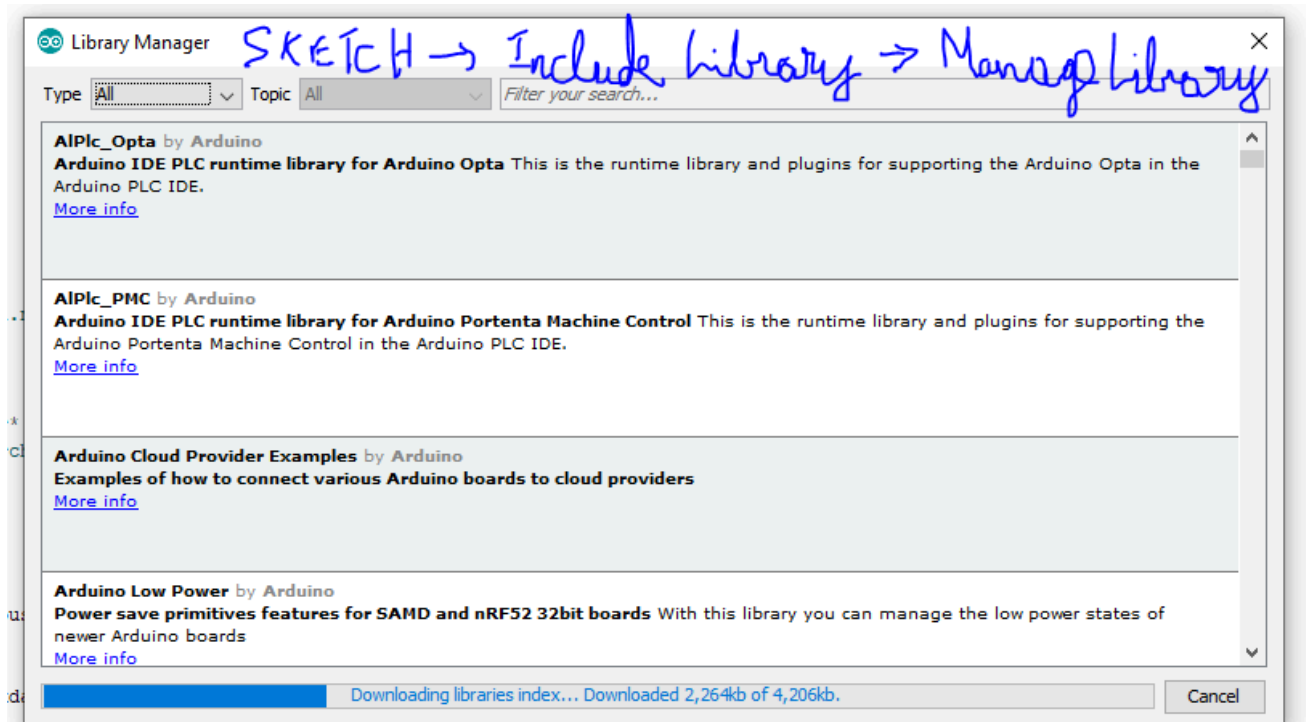
## Chapter 2. ESP Arduino IDE settings



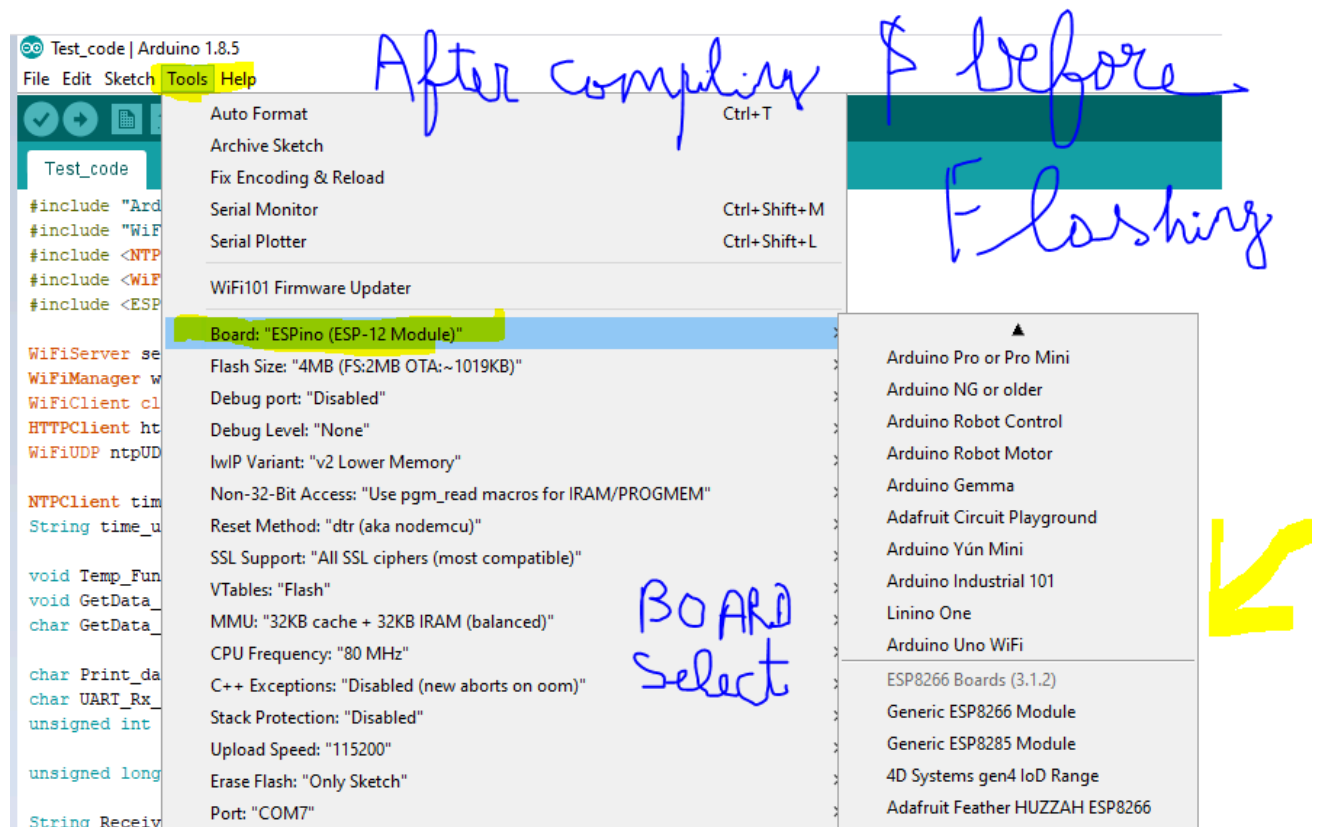
- Step 1 Goto File → Preferences → Settings
- Enter the Below line as shown in the Above Image
- [https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json)
- This URL make sures, ESP module will be compiled using Arduino IDE.



- Step 2 is to install the Packages for the Board which we are using.
- Goto Tools → Board → Board Manager
- Search and Install the Required Package for our Board.



- Step 3 is to install the Library Package for our Code.
- Goto Sketch → Include Library → Manage Library



- Step 4 is the Final Step, After Compiling and Before Flashing
- Goto Tools → Board → Select the Board which I am using.
- Select the Required COM Port and Keep the remaining settings as per the Image.

## Chapter 3. ESP Pin Configuration

*After Reset*

These pins are used to choose the boot mode.

*Mult*

GPIO 0	GPIO 2	GPIO 15	Boot Mode
LOW	HIGH	LOW	UART Bootloader
HIGH	HIGH	LOW	Boot from SPI Flash
x	x	HIGH	Boot from SDIO

- Before Flashing using UART, Make sure GPIO0 is Low to Enter into Boot Mode.
- After Flashing, Make sure GPIO0 is High after Pressing the Reset Button.

The EN pin on ESP32 and ESP8266 micro-controllers is the enable pin. When pulled HIGH, it enables the chip, and when pulled LOW, it disables the chip, effectively putting it into a low-power or standby state.





Label	GPIO	Input	Output	Notes
D0	GPIO16	no interrupt	no PWM or I2C support	<b>HIGH at boot</b> used to wake up from deep sleep
D1	GPIO5	OK	OK	often used as <b>SCL</b> (I2C)
D2	GPIO4	OK	OK	often used as <b>SDA</b> (I2C)
D3	GPIO0	pulled up	OK	connected to FLASH button boot
D4	GPIO2	pulled up	OK	<b>HIGH at boot</b> connected to on-board LED boot
D5	GPIO14	OK	OK	<b>SPI</b> (SCLK)
D6	GPIO12	OK	OK	<b>SPI</b> (MISO)
D7	GPIO13	OK	OK	<b>SPI</b> (MOSI)
D8	GPIO15	pulled to GND	OK	<b>SPI</b> (CS) Boot fails if pulled HIGH
RX	GPIO3	OK	RX pin	<b>HIGH at boot</b>
TX	GPIO1	TX pin	OK	<b>HIGH at boot</b> debug output at boot, boot fails if
A0	ADC0	Analog Input	X	

- Keep the GPIO0 is Low to Enter into Boot Mode, After Reset.
- Dont Disturb the Other Pins Line GPIO2, GPIO15 and GPIO16.
- IF ESP does not going to Boot Mode then, Configure the Pins as per the Image.

Note: While Flasing The UART pins must be connected only to the COM port.

## Chapter 4. Working Reference Code

```
#include <ESP8266WiFi.h>
#include "stdint.h"
#include "string.h"
#include "Arduino.h"
#include "WiFiManager.h"
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <ESP8266HTTPClient.h>
```

```
WiFiServer server(80);
WiFiManager wifiManager;
WiFiClient client;
HTTPClient http;
WiFiUDP ntpUDP;
```

```
NTPClient timeClient(ntpUDP, "asia.pool.ntp.org", 19800, 60000);
String time_update;
void Temp_Function(void);
void GetData_Server_Function(const char* Local_Server_Name);
// switch get link
char GetData_ServerName[] = "http://purchase.indrainsignia.co.in/test/get1.php";
```

```
char Print_data[1000];
char UART_Rx_Data[1000] ;
unsigned int UART_Rx_Data_Inc =0 ;
unsigned long currentMillis = 0, previousMillis = 0;
String Receive_Server_Parse;
char *Receive_Pointer , Store_Server_Rxdata[1000] ;
```

```
/******Wifi Configuration using Mobile******/
void Wifi_ConfigMode_Callback (WiFiManager *myWiFiManager)
{
    Serial.println("Entered config mode");
    Serial.println(WiFi.softAPIP());
    Serial.println(myWiFiManager->getConfigPortalSSID());
}
/*******/
```

```
void setup()
{
    pinMode(LED_BUILTIN, OUTPUT);
    pinMode(12, OUTPUT);
    Serial.begin(115200); //Tx to STM
    wifiManager.autoConnect("TEST WIFI");
    wifiManager.setAPCallback(Wifi_ConfigMode_Callback);
    //Loop Waits for more than 15min, after some time breaks
    delay(10);
    Serial.print("\nConnecting to ");
```

```
for(char x=0; x<=20;x++)
{
    Serial.print(".");
    if((WiFi.status() == WL_CONNECTED))
    {
        Serial.println("\nWL_CONNECTED");
        server.begin();
        break;
    }
    delay(100);
}
```

```
    if((WiFi.status() != WL_CONNECTED))
    {
        sprintf(Print_data, "\nTime Taken:%d \n" ,millis()) ;
        Serial.print(Print_data);
        Serial.print("Not Connected to any Wifi");
        ESP.reset();
    }
}
```

```
void loop()
{
    while(millis() < 500)
    {
        previousMillis = 0;
    }
    currentMillis = millis();
```

```
    if((WiFi.status() == WL_CONNECTED))
    {
        if((currentMillis - previousMillis) >= 500)
        {
            digitalWrite(12, HIGH);
            digitalWrite(LED_BUILTIN, LOW);
            GetData_Server_Function (GetData_ServerName);
            previousMillis = millis();
        }
        Temp_Function();
    }
    else
    {
        digitalWrite(12, LOW);
        digitalWrite(LED_BUILTIN, HIGH);
    }
}
```

```
void Temp_Function(void)
{
```

```
}
```

```
void GetData_Server_Function(const char* Local_Server_Name)
{
    http.begin(client, Local_Server_Name);
    delay(10);

    /*****/
    timeClient.update();
    time_t epochTime = timeClient.getEpochTime();
    time_update = timeClient.getFormattedTime();
    //Get a time structure
    struct tm *ptm = gmtime ((time_t *)&epochTime);
    int monthDay = ptm->tm_mday;
    int currentMonth = ptm->tm_mon+1;
    int currentYear = ptm->tm_year+1900;
    String currentDate = String(currentYear) + "-" +
                        String(currentMonth) + "-" + String(monthDay);

    Serial.print("Date: ");
    Serial.println(currentDate);
    Serial.print("Time: ");
    Serial.println(time_update);//Send to STM
    /*****/

    if(http.GET() > 0)
    {
        memset(Store_Server_Rxdata ,0x00, sizeof(Store_Server_Rxdata));
        Receive_Server_Parse = http.getString();
        Receive_Server_Parse.toCharArray ( Store_Server_Rxdata ,
                                           Receive_Server_Parse.length() );

        Serial.print("Control: ");
        Serial.print(Store_Server_Rxdata); //Send to STM
        Serial.print(" End \n\r");
    }
    delay(10);
    http.end();
}
```