**Wi-Fi Based Notice Board Using Node MCU**

**Abstract:**

Objective:

Notice boards can be used anywhere and are very useful in Hotels, Malls, colleges, offices to display messages, alerts, offers etc. But it is very tedious task to change the notice every day. So, in this project we will make a **Wi-Fi Based Notice Board** using NodeMCU where you can remotely change the notice using web browser.

Digital advertising boards in public places like railway stations, bus stations, airports etc. But with Internet of Things (IoT), there is a great shift in technology and we can also revolutionize this kind of notice board by making it wireless and controlled from a [web server](https://iotdesignpro.com/tags/webserver).

Working:

A display connected to a server system should continuously listen for the incoming messages from the user, process it, and display it on the LCD screen. The message displayed should be updated every time the user sends new information. Only authenticated people should update the data to be displayed on the LCD.

Components required:

1. Node MCU
2. LCD
3. Potentiometer
4. Resistor
5. Jumper Wires
6. Bread Board
7. Soldering station
8. USB cable

Software:

1. Arduino IDE

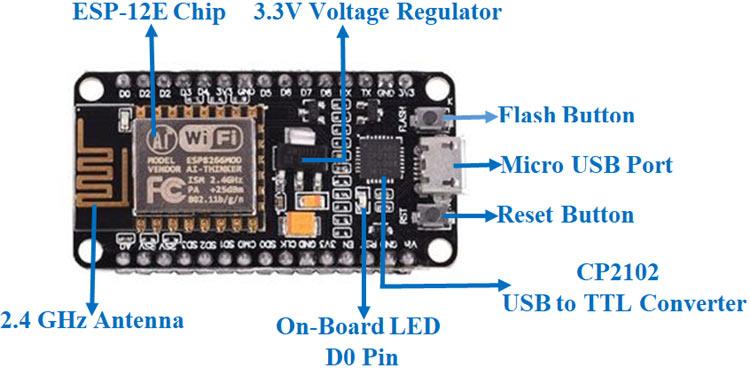
**Node MCU :**

Node MCU is a low-cost open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things) platform. It initially included [firmware](https://en.wikipedia.org/wiki/Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip) from Espressif Systems, and hardware which was based on the ESP-12 module. Later, support for the [ESP32](https://en.wikipedia.org/wiki/ESP32) 32-bit MCU was added.

**Node MCU ESP8266 Specifications & Features**

* Microcontroller: Ten silica 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

Node MCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.



**Code:**

#include <ESP8266WiFi.h>

#include <ESPAsyncTCP.h>

#include <ESPAsyncWebServer.h>

#include <LiquidCrystal.h>

const int rs = D1, en = D0, d4 = D4, d5 = D5, d6 = D6, d7 = D7;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

AsyncWebServer server(80);

const char\* ssid = "iPhone"; // change u r SSID

const char\* password = "raj77777"; //wifi password

const char\* PARAM\_INPUT\_1 = "input1";

const char index\_html[] PROGMEM = R"rawliteral(

<!DOCTYPE HTML><html><head>

<title>Smart Notice Board Secure Space</title>

<meta name="viewport" content="width=device-width, initial-scale=5">

<p> <font size="9" face="sans-serif"> <marquee> <font color="green">Smart Notice Board</font> </marquee> </font> </p>

</head><body><center>

<form action="/get">

Enter Text to Display: <input type="text" name="input1">

<input type="submit" value="Send">

</form><br>

</center></body></html>)rawliteral";

void notFound(AsyncWebServerRequest \*request) {

request->send(404, "text/plain", "Not found");

}

void setup() {

Serial.begin(115200);

lcd.begin(16, 2);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Smart Notice Board");

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

if (WiFi.waitForConnectResult() != WL\_CONNECTED) {

Serial.println("WiFi Failed!");

return;

}

Serial.println();

Serial.print("IP Address: ");

Serial.println(WiFi.localIP());

server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){

request->send\_P(200, "text/html", index\_html);

});

server.on("/get", HTTP\_GET, [] (AsyncWebServerRequest \*request) {

String message;

String inputParam;

if (request->hasParam(PARAM\_INPUT\_1)) {

message = request->getParam(PARAM\_INPUT\_1)->value();

inputParam = PARAM\_INPUT\_1;

lcd.clear();

lcd.setCursor(0,0);

lcd.print(message);

}

else {

message = "No message sent";

inputParam = "none";

}

Serial.println(message);

request->send(200, "text/html", index\_html);

});

server.onNotFound(notFound);

server.begin();

}

void loop() {

for (int positionCounter = 0; positionCounter < 29; positionCounter++) {

//lcd.scrollDisplayLeft();

delay(500);

}

}