IOT PROJECT
CLASS ROOM AUTOMATION

Requirements:

Hardware Used :

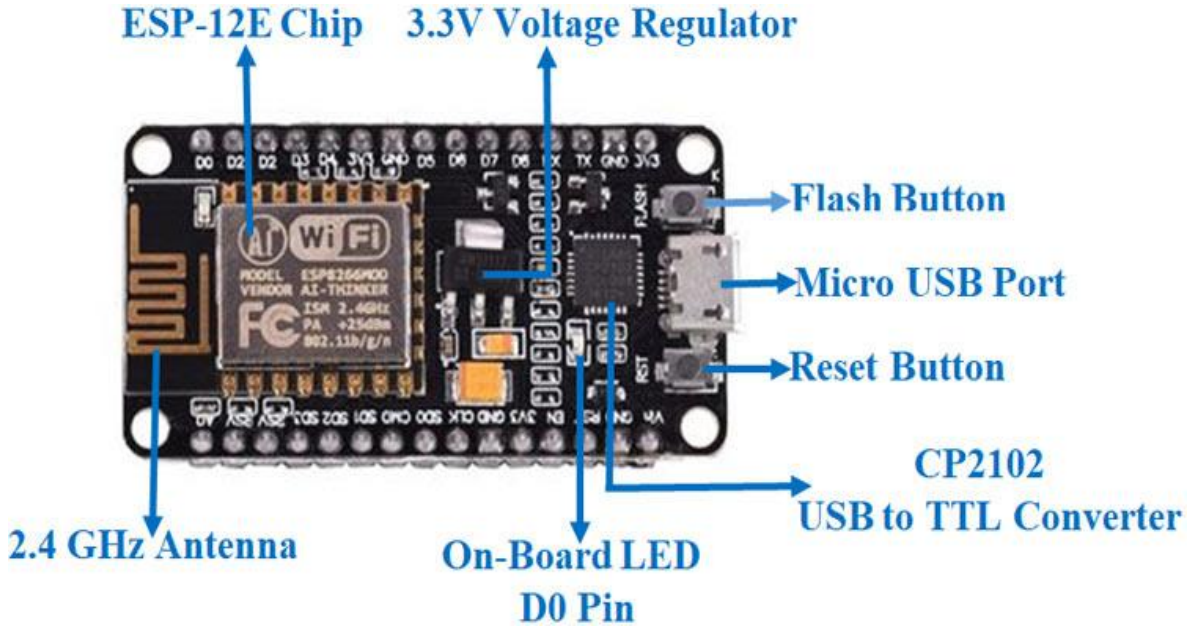
- NodeMcu Esp32
- NodeMcu Esp8266
- Relay Modules
- Jumper wires
- Breadboard
- Switches

Esp32 :-

- The Esp32 is a very Versatile System On a Chip (Soc) that can be used as a general purpose microcontroller with quite an extensive set of peripherals including Wi-Fi and Bluetooth wireless capabilities.
- The Esp32 series employs either a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISC-V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters and power-management modules.

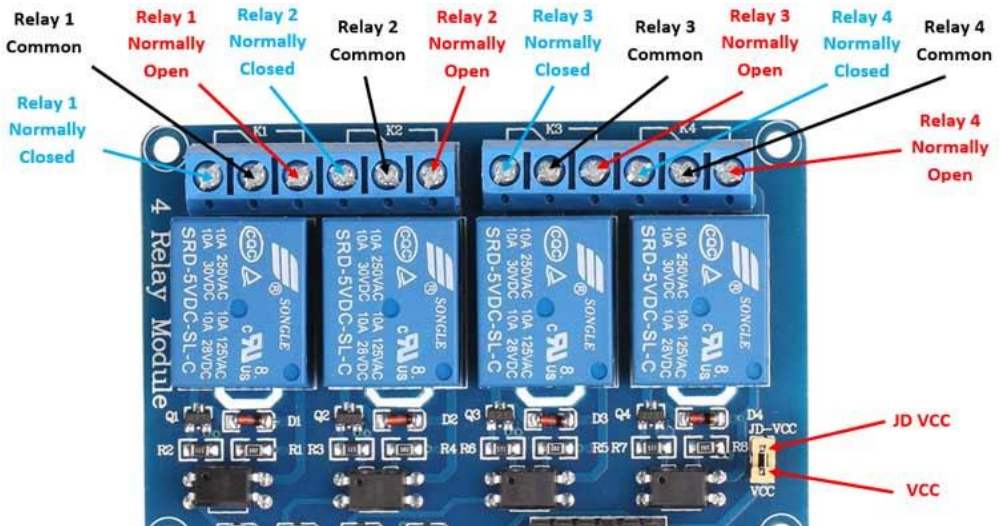
NodeMcu Esp8266 :

- The ESP8266 is a low-cost Wi-Fi microchip, with built-in TCP/IP networking software, and microcontroller capability.
- In 2020, Espressif announced a new chip ESP-32, which is pin-compatible with ESP8266. It is based on a single core RISC-V 32-bit CPU with a clock speed of up to 160MHz. It includes 400KB of SRAM and 348 KB ROM storage space built-in.
- Microchip mainly used for the development of end-point IOT(Internet of things) applications. It is referred to as a standalone wireless transceiver, available at a very low price.
- It is used to enable the internet connection to various applications of embedded systems.
- Wi-Fi modules or wi-fi microcontrollers are used to send and receive data over wi-fi. They can also accept commands over the Wi-Fi. Wi-Fi modules are used for communication between devices.



4 Channel Relay :-

- The 4 Channel Relay Module is a convenient board which can be used to control high voltage, high current load such as motor, solenoid valves, lamps and AC load. It is designed to interface with microcontroller such as Arduino, PIC and etc. The relays terminal (COM, NO and NC) is being brought out with screw terminal.
- The relay module function is mainly to switch electrical devices and systems on or off. It also serves to isolate the control circuit from the device or system being controlled.
- The four relays on the module are rated for 5V, which means the relay is activated when there is approximately 5V across the coil.
- Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called protective relays.
- The 4 channel Relay Module SPDT (Single Pole Double Throw) can control up to 4 separate devices individually. Relays are typically used to switch devices which uses a higher voltage than what most micro-controllers such as an Arduino or Raspberry Pi can handle. This particular relay module can control typical household appliance up to 10A.



Software Used :

1.Arduino IDE

2.Blynk Interface

Arduino Program:

```
#define BLYNK_TEMPLATE_ID "TMPLTiPE10Xt"

#define BLYNK_DEVICE_NAME "Class Room"

#define BLYNK_AUTH_TOKEN
"FxGXLYwznRVovFThefpk93MU7_ZT9ZP"


// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "Karthik";

char pass[] = "eswar@12345";


bool fetch_blynk_state = true;


//#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>
```

```
// define the GPIO connected with Relays and switches
#define RelayPin1 5  //D1
#define RelayPin2 4  //D2
#define RelayPin3 14 //D5
#define RelayPin4 12 //D6

#define SwitchPin1 10  //SD3
#define SwitchPin2 D3   //D3
#define SwitchPin3 13  //D7
#define SwitchPin4 3   //RX

#define wifiLed   16   //D0

//Change the virtual pins according the rooms
#define VPIN_BUTTON_1    V5
#define VPIN_BUTTON_2    V6
#define VPIN_BUTTON_3    V7
#define VPIN_BUTTON_4    V8

// Relay State
bool toggleState_1 = LOW;
bool toggleState_2 = LOW
bool toggleState_3 = LOW;
```

```
bool toggleState_4 = LOW

// Switch State
bool SwitchState_1 = LOW;
bool SwitchState_2 = LOW;
bool SwitchState_3 = LOW;
bool SwitchState_4 = LOW;

int wifiFlag = 0;

char auth[] = BLYNK_AUTH_TOKEN;

BlynkTimer timer;

// When App button is pushed - switch the state

BLYNK_WRITE(VPIN_BUTTON_1) {
  toggleState_1 = param.asInt();
  digitalWrite(RelayPin1, !toggleState_1);
}

BLYNK_WRITE(VPIN_BUTTON_2) {
  toggleState_2 = param.asInt();
  digitalWrite(RelayPin2, !toggleState_2);
```

```
}

BLYNK_WRITE(VPIN_BUTTON_3) {
  toggleState_3 = param.asInt();
  digitalWrite(RelayPin3, !toggleState_3);
}

BLYNK_WRITE(VPIN_BUTTON_4) {
  toggleState_4 = param.asInt();
  digitalWrite(RelayPin4, !toggleState_4);
}

void checkBlynkStatus() { // called every 3 seconds by SimpleTimer

  bool isconnected = Blynk.connected();
  if (isconnected == false) {
    wifiFlag = 1;
    Serial.println("Blynk Not Connected");
    digitalWrite(wifiLed, HIGH);
  }
  if (isconnected == true) {
    wifiFlag = 0;
    if (!fetch_blynk_state){
    Blynk.virtualWrite(VPIN_BUTTON_1, toggleState_1);
```

```
    Blynk.virtualWrite(VPIN_BUTTON_2, toggleState_2);
    Blynk.virtualWrite(VPIN_BUTTON_3, toggleState_3);
    Blynk.virtualWrite(VPIN_BUTTON_4, toggleState_4);
    }
    digitalWrite(wifiLed, LOW);
    Serial.println("Blynk Connected");
  }
}

BLYNK_CONNECTED() {
  // Request the latest state from the server
  if (fetch_blynk_state){
    Blynk.syncVirtual(VPIN_BUTTON_1);
    Blynk.syncVirtual(VPIN_BUTTON_2);
    Blynk.syncVirtual(VPIN_BUTTON_3);
    Blynk.syncVirtual(VPIN_BUTTON_4);
  }
}

void manual_control()
{
  if (digitalRead(SwitchPin1) == LOW && SwitchState_1 == LOW) {
    digitalWrite(RelayPin1, LOW);
    toggleState_1 = 1;
```

```
  SwitchState_1 = HIGH;
  Blynk.virtualWrite(VPIN_BUTTON_1, toggleState_1);
  Serial.println("Switch-1 on");
 }
 if (digitalRead(SwitchPin1) == HIGH && SwitchState_1 == HIGH) {
  digitalWrite(RelayPin1, HIGH);
  toggleState_1 = 0;
  SwitchState_1 = LOW;
  Blynk.virtualWrite(VPIN_BUTTON_1, toggleState_1);
  Serial.println("Switch-1 off");
 }
 if (digitalRead(SwitchPin2) == LOW && SwitchState_2 == LOW) {
  digitalWrite(RelayPin2, LOW);
  toggleState_2 = 1;
  SwitchState_2 = HIGH;
  Blynk.virtualWrite(VPIN_BUTTON_2, toggleState_2);
  Serial.println("Switch-2 on");
 }
 if (digitalRead(SwitchPin2) == HIGH && SwitchState_2 == HIGH) {
  digitalWrite(RelayPin2, HIGH);
  toggleState_2 = 0;
  SwitchState_2 = LOW;
  Blynk.virtualWrite(VPIN_BUTTON_2, toggleState_2);
  Serial.println("Switch-2 off");
```

```
}
if (digitalRead(SwitchPin3) == LOW && SwitchState_3 == LOW) {
  digitalWrite(RelayPin3, LOW);
  toggleState_3 = 1;
  SwitchState_3 = HIGH;
  Blynk.virtualWrite(VPIN_BUTTON_3, toggleState_3);
  Serial.println("Switch-3 on");
}
if (digitalRead(SwitchPin3) == HIGH && SwitchState_3 == HIGH) {
  digitalWrite(RelayPin3, HIGH);
  toggleState_3 = 0;
  SwitchState_3 = LOW;
  Blynk.virtualWrite(VPIN_BUTTON_3, toggleState_3);
  Serial.println("Switch-3 off");
}
if (digitalRead(SwitchPin4) == LOW && SwitchState_4 == LOW) {
  digitalWrite(RelayPin4, LOW);
  toggleState_4 = 1;
  SwitchState_4 = HIGH;
  Blynk.virtualWrite(VPIN_BUTTON_4, toggleState_4);
  Serial.println("Switch-4 on");
}
if (digitalRead(SwitchPin4) == HIGH && SwitchState_4 == HIGH) {
  digitalWrite(RelayPin4, HIGH);
```

```
    toggleState_4 = 0;
    SwitchState_4 = LOW;
    Blynk.virtualWrite(VPIN_BUTTON_4, toggleState_4);
    Serial.println("Switch-4 off");
  }
}

void setup()
{
  Serial.begin(9600);

  pinMode(RelayPin1, OUTPUT);
  pinMode(RelayPin2, OUTPUT);
  pinMode(RelayPin3, OUTPUT);
  pinMode(RelayPin4, OUTPUT);

  pinMode(wifiLed, OUTPUT);

  pinMode(SwitchPin1, INPUT_PULLUP);
  pinMode(SwitchPin2, INPUT_PULLUP);
  pinMode(SwitchPin3, INPUT_PULLUP);
  pinMode(SwitchPin4, INPUT_PULLUP);

  //During Starting all Relays should TURN OFF
```

```
  digitalWrite(RelayPin1, !toggleState_1);
  digitalWrite(RelayPin2, !toggleState_2);
  digitalWrite(RelayPin3, !toggleState_3);
  digitalWrite(RelayPin4, !toggleState_4);

  digitalWrite(wifiLed, HIGH);

  //Blynk.begin(auth, ssid, pass);
  WiFi.begin(ssid, pass);
  timer.setInterval(2000L, checkBlynkStatus); // check if Blynk server is connected every 2 seconds
  Blynk.config(auth);
  delay(1000);

  if (!fetch_blynk_state){
    Blynk.virtualWrite(VPIN_BUTTON_1, toggleState_1);
    Blynk.virtualWrite(VPIN_BUTTON_2, toggleState_2);
    Blynk.virtualWrite(VPIN_BUTTON_3, toggleState_3);
    Blynk.virtualWrite(VPIN_BUTTON_4, toggleState_4);
  }
}

void loop()
{
```

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

CIRCUIT DIAGRAM: