**iot based home automation**

**introduction**

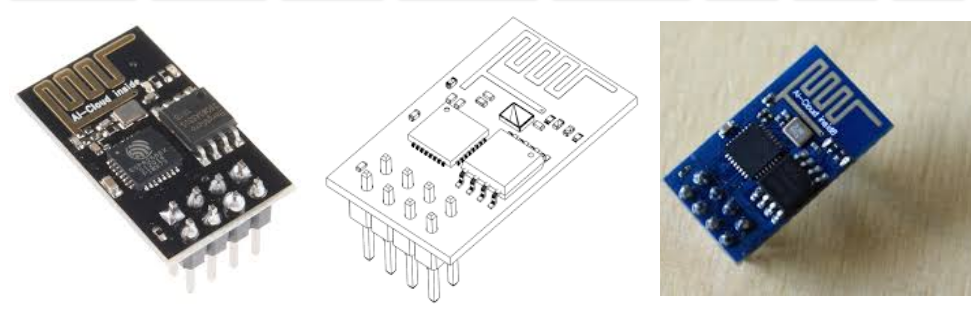
In this project, we will be using the ESP8266 – 01 Wi-Fi Development board to make a small IoT home automation that has Wi-Fi featured in it. The system operates on a local web server and is easy to use for the novice. With this project, we can control at most two AC appliances which suit best for your small IoT projects.

1. When I tap on any button the Blynk server sends the signal to ESP01.

2. Then ESP01 (ESP8266) sends the signal to Arduino Nano through the serial terminal.

3. After receiving the signal from ESP01, the Arduino TURN ON or OFF the respective relay and send the feedback to ESP01 through the serial terminal.

If there is no internet, then we can control the relay module from the IR remote.



**Component**

1. TSOP 1738 IR Receiver (1 no)

2. 100uF Capacitor (1 no)

3. Arduino Nano

4. ESP01 module (ESP8266)

5. Optocoupler PC817 (4 no)

6. Transistor BC547 (4 no)

7. LEDs (1.5) (7 no)

8. Diode 1N4007 (4 no)

9. SPDT Relay 5v (4 no)

10. 220-ohm Resistors (8 no)

11. Resistor (6 no)- 1k, 2k, 4.7k, 10k

15. Male & Female connectors (2mm Pitch Female BERG Strip)

16. AMS1117 3.3V voltage regulator (1no)

17. Push buttons (2 no)

Application

1. Traffic monitoring.

### **2. Fleet management.**

### **3. Agriculture.**

### **4. Smart grid and energy saving.**

### **5.Hospitality.**

Objective

During this activity ,you will help students to achieve following objectives

1. Understanding the principle and operation of ESp8266-01 module
2. Design algorithm and flowchart to home automation system
3. Programming ESp8266-01 module
4. Interfacing ESp8266-01 module

program

#include <IRremote.h>

#include <arduino-timer.h>

auto timer = timer\_create\_default(); // create a timer with default settings

//Define PIN constant

const int RELAY\_PIN\_1 = 12;

const int RELAY\_PIN\_2 = 11;

const int RELAY\_PIN\_3 = 10;

const int RELAY\_PIN\_4 = 9;

const int smode = 7;

const int irled = 6;

const int wifiled = 5;

int RECV\_PIN = 8;

char toggleState\_1 = 'a'; //Define integer to remember the toggle state for switch 1

char toggleState\_2 = 'b'; //Define integer to remember the toggle state for switch 2

char toggleState\_3 = 'c'; //Define integer to remember the toggle state for switch 3

char toggleState\_4 = 'd'; //Define integer to remember the toggle state for switch 4

int ModeFlag = 1;

String pinStatus = "abcd";

//Define IR receiver and Result Objects

IRrecv irrecv(RECV\_PIN);

decode\_results results;

void sendStatus(){

Serial.print(pinStatus);

//true;

}

void relayOnOff(int relay){

switch(relay){

case 1:

if(toggleState\_1 == 'a'){

digitalWrite(RELAY\_PIN\_1, HIGH); // turn on relay 1

toggleState\_1 = 'A';

}

else{

digitalWrite(RELAY\_PIN\_1, LOW); // turn off relay 1

toggleState\_1 = 'a';

}

break;

case 2:

if(toggleState\_2 == 'b'){

digitalWrite(RELAY\_PIN\_2, HIGH); // turn on relay 2

toggleState\_2 = 'B';

}

else{

digitalWrite(RELAY\_PIN\_2, LOW); // turn off relay 2

toggleState\_2 = 'b';

}

break;

case 3:

if(toggleState\_3 == 'c'){

digitalWrite(RELAY\_PIN\_3, HIGH); // turn on relay 3

toggleState\_3 = 'C';

}

else{

digitalWrite(RELAY\_PIN\_3, LOW); // turn off relay 3

toggleState\_3 = 'c';

}

break;

case 4:

if(toggleState\_4 == 'd'){

digitalWrite(RELAY\_PIN\_4, HIGH); // turn on relay 4

toggleState\_4 = 'D';

}

else{

digitalWrite(RELAY\_PIN\_4, LOW); // turn off relay 4

toggleState\_4 = 'd';

}

break;

default : break;

}

delay(200);

}

void setup()

{

Serial.begin(9600);

irrecv.enableIRIn(); // Enable the IR receiver

pinMode(RELAY\_PIN\_1, OUTPUT);

pinMode(RELAY\_PIN\_2, OUTPUT);

pinMode(RELAY\_PIN\_3, OUTPUT);

pinMode(RELAY\_PIN\_4, OUTPUT);

pinMode(irled, OUTPUT);

pinMode(wifiled, OUTPUT);

pinMode(smode, INPUT);

digitalWrite(irled, HIGH); // turn on IRLED

digitalWrite(wifiled, HIGH);

// call the toggle\_led function every 1000 millis (1 second)

timer.every(2000, sendStatus);

}

void loop() {

if (digitalRead(smode) == HIGH){

if (ModeFlag == 0){

ModeFlag = 1;

digitalWrite(irled, HIGH); // turn on IRLED

digitalWrite(wifiled, HIGH); //turn on WIFILED

delay(200);

}

else if (ModeFlag == 1) {

ModeFlag = 0;

digitalWrite(wifiled, LOW); // turn off WIFILED

digitalWrite(irled, HIGH); // turn on IRLED

delay(200);

}

//delay(1000);

}

if (ModeFlag == 1) { //Both WiFi & IR control

//WiFi Control

if(Serial.available()>0)

{

char c\_val= Serial.read(); // reading the data received from the bluetooth module

switch(c\_val)

{

case 'A': digitalWrite(RELAY\_PIN\_1, HIGH); toggleState\_1 = 'A'; break; // when A is pressed on the app Turn on Pin 12

case 'a': digitalWrite(RELAY\_PIN\_1, LOW); toggleState\_1 = 'a'; break; // when a is pressed on the app Turn off Pin 12

case 'B': digitalWrite(RELAY\_PIN\_2, HIGH); toggleState\_2 = 'B'; break; // when B is pressed on the app Turn on Pin 11

case 'b': digitalWrite(RELAY\_PIN\_2, LOW); toggleState\_2 = 'b'; break; // when b is pressed on the app Turn off Pin 11

case 'C': digitalWrite(RELAY\_PIN\_3, HIGH); toggleState\_3 = 'C'; break; // when C is pressed on the app Turn on Pin 10

case 'c': digitalWrite(RELAY\_PIN\_3, LOW); toggleState\_3 = 'c'; break; // when c is pressed on the app Turn off Pin 10

case 'D': digitalWrite(RELAY\_PIN\_4, HIGH); toggleState\_4 = 'D'; break; // when D is pressed on the app Turn on Pin 9

case 'd': digitalWrite(RELAY\_PIN\_4, LOW); toggleState\_4 = 'd'; break; // when d is pressed on the app Turn off Pin 9

default : break;

}

pinStatus = String(toggleState\_1) + String(toggleState\_2) + String(toggleState\_3) + String(toggleState\_4);

}

delay(100);

//IR Remote Control

if (irrecv.decode(&results)) {

switch(results.value){

case 0x80BF49B6: relayOnOff(1); break;

case 0x80BFC936: relayOnOff(2); break;

case 0x80BF33CC: relayOnOff(3); break;

case 0x80BF718E: relayOnOff(4); break;

default : break;

}

pinStatus = String(toggleState\_1) + String(toggleState\_2) + String(toggleState\_3) + String(toggleState\_4);

irrecv.resume(); // Receive the next value

}

}

else if (ModeFlag == 0) { //only IR control

//IR Remote Control

if (irrecv.decode(&results)) {

switch(results.value){

case 0x80BF49B6: relayOnOff(1); break;

case 0x80BFC936: relayOnOff(2); break;

case 0x80BF33CC: relayOnOff(3); break;

case 0x80BF718E: relayOnOff(4); break;

default : break;

}

pinStatus = String(toggleState\_1) + String(toggleState\_2) + String(toggleState\_3) + String(toggleState\_4);

irrecv.resume(); // Receive the next value

}

}

timer.tick(); // tick the timer

}

Program for ES01 module

//#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

int pinState = 0;

#define VPIN\_BUTTON\_1 V1

#define VPIN\_BUTTON\_2 V2

#define VPIN\_BUTTON\_3 V3

#define VPIN\_BUTTON\_4 V4

// You should get Auth Token in the Blynk App.

char auth[] = "AUTH TOKEN";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "WIFI NAME";

char pass[] = "WIFI PASSWORD";

bool connectedFlag = false;

String pinStatus = "";

// When App button is pushed - switch the state

BLYNK\_WRITE(VPIN\_BUTTON\_1) {

pinState = param.asInt();

if(pinState == 1){

Serial.write('A');

}

else if(pinState == 0){

Serial.write('a');

}

}

BLYNK\_WRITE(VPIN\_BUTTON\_2) {

pinState = param.asInt();

if(pinState == 1){

Serial.write('B');

}

else if(pinState == 0){

Serial.write('b');

}

}

BLYNK\_WRITE(VPIN\_BUTTON\_3) {

pinState = param.asInt();

if(pinState == 1){

Serial.write('C');

}

else if(pinState == 0){

Serial.write('c');

}

}

BLYNK\_WRITE(VPIN\_BUTTON\_4) {

pinState = param.asInt();

if(pinState == 1){

Serial.write('D');

}

else if(pinState == 0){

Serial.write('d');

}

}

void updateState(String state){

if (state.substring(0,1) == "A"){

Blynk.virtualWrite(VPIN\_BUTTON\_1, HIGH);

}

else{

Blynk.virtualWrite(VPIN\_BUTTON\_1, LOW);

}

if (state.substring(1,2) == "B"){

Blynk.virtualWrite(VPIN\_BUTTON\_2, HIGH);

}

else{

Blynk.virtualWrite(VPIN\_BUTTON\_2, LOW);

}

if (state.substring(2,3) == "C"){

Blynk.virtualWrite(VPIN\_BUTTON\_3, HIGH);

}

else{

Blynk.virtualWrite(VPIN\_BUTTON\_3, LOW);

}

if (state.substring(3,4) == "D"){

Blynk.virtualWrite(VPIN\_BUTTON\_4, HIGH);

}

else{

Blynk.virtualWrite(VPIN\_BUTTON\_4, LOW);

}

}

void setup()

{

// Debug console

Serial.begin(9600);

Blynk.begin(auth, ssid, pass);

}

void loop()

{

while(Serial.available())

{

pinStatus = Serial.readString();

updateState(pinStatus);

connectedFlag = true;

}

if (connectedFlag){

Blynk.run();

}

}

Hardware

1. In the circuit, the RX and TX pins of Arduino Nano is connected with the ESP01 (ESP8266), So that the Arduino and ESP01 can communicate with each other and we can monitor the real-time status in the Blynk.
2. A 1738 IR detector is connected with the D8 pin of Arduino to control the relay module from any IR remote (Ex TV Remote).
3. There is also a push-button to select the MODE connect this to D7. MODE-1 we can control the relay module from both Blynk App and IR remote.
4. And in MODE-2 we can control the relay module from only IR Remote.
5. Connect four relay pin to D12,D111,D10,D9 of arduino nano
6. Connect two LED to D5 and D6 of arduino

Steps to start blynk app

1. Enter the WiFi credentials,**(WiFi Name & Password).**

2. Then enter the **AUTH TOKEN** sent by the Blynk.

3. Connect the USB to TTL interface board (**FTDI232**) with **ESP01**.

4. **Download the Arduino sketch** (attached) for the ESP8266 board.

5. Select the Generic **ESP8266**board and proper PORT. Then upload the code.

Then add all the 4 buttons with virtual pins **V1, V2, V3, V4.**

