Home Automation using blynk

# Material used

* 1. Node mcu (esp8266) .
  2. 4- channel relay module.
  3. DHT 11 (temperature and humidity sensor).
  4. 16 x 2 LCD display (with i2c module).
  5. Jumper wires and bread board.

# Features

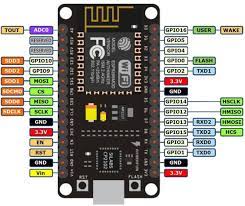
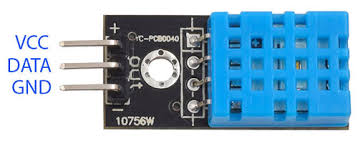
* This is home electronic appliances controlling system with your smart phone app “BLYNK IOT.
* The BLYNK IOT service provide a web dashboard to create or modify and controll the system.
* This system is controlled by two methods.
  + - * + Manual control of the house by 2-way switch.
        + BLYNK IOT app through wifi.
* You need a WIFI connection to access the home appliances using BLYNK IOT APP anywhere around the world in network . whereas the 2-way switch is directly connected with the relay and appliance in case of any malfunction in WIFI you can manully control your appliances as normal in day to day life.
* With the LCD display and DHT11 sensor we can moniter the temperature and humidity send to the LCD DISPLAY and to BLYNK APP also to the WEB DASHBOARD.
* The main advantage of this project that is it can be expanded for our choice . eg: gas sensor , ultrasonic sensor , camera modules and so on. We can add many more features to the Home automation so it is future upgradable.
* It is so easy and simple to make your own and use.

# Steps to make

1. Gather the components required.
2. Arrange the components in the breadboard.
3. Connect the circuit with NODE MCU.
4. Set up the BLYNK dashboard and mobile APP.
5. Connect the NODE MCU to the computer and create a program and upload the code.

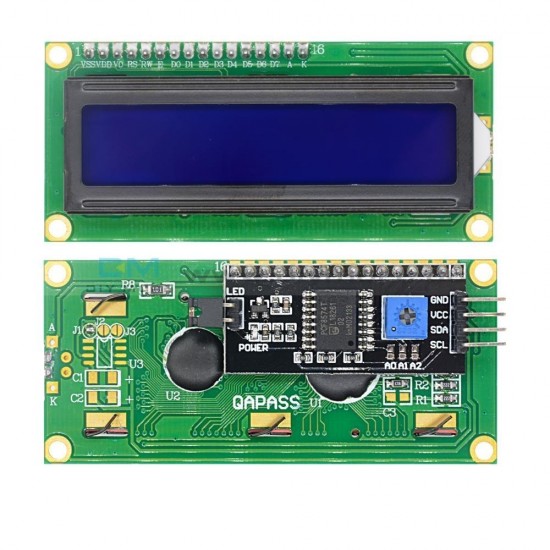
# Components and circuit diagram

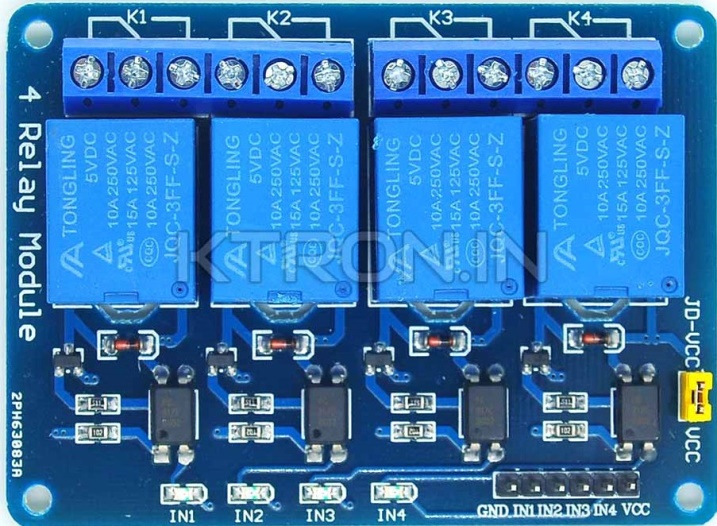
**Node mcu**

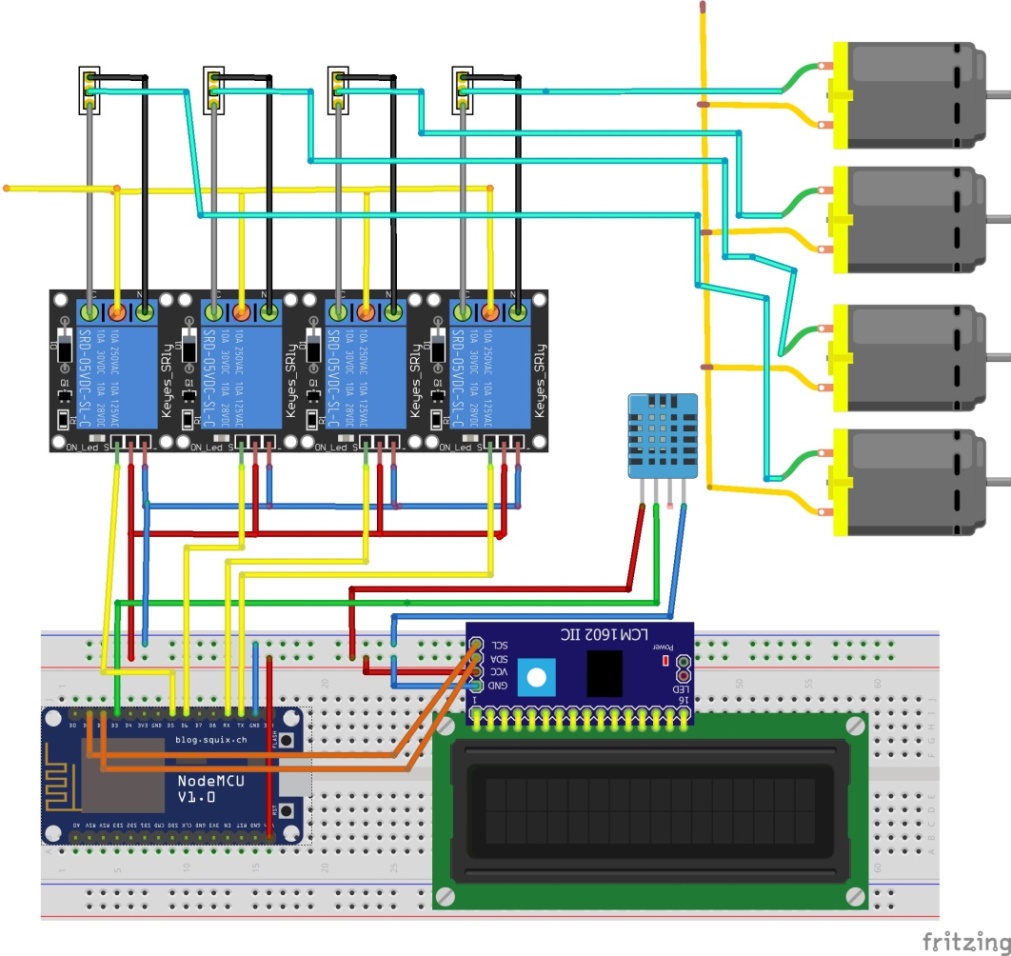


**DHT 11**

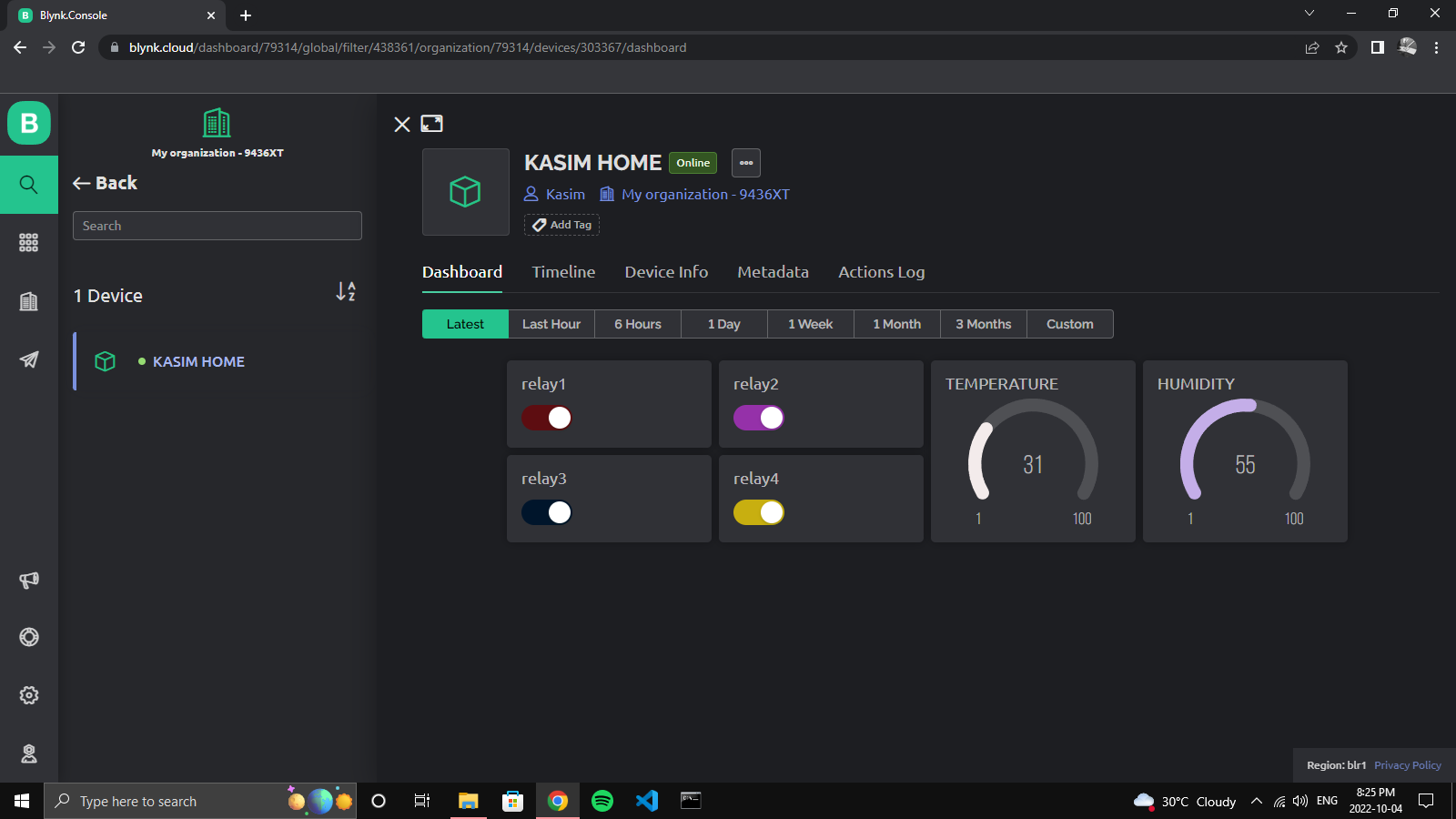
**4-channel relay module** **16x2 LED Display with i2c module**



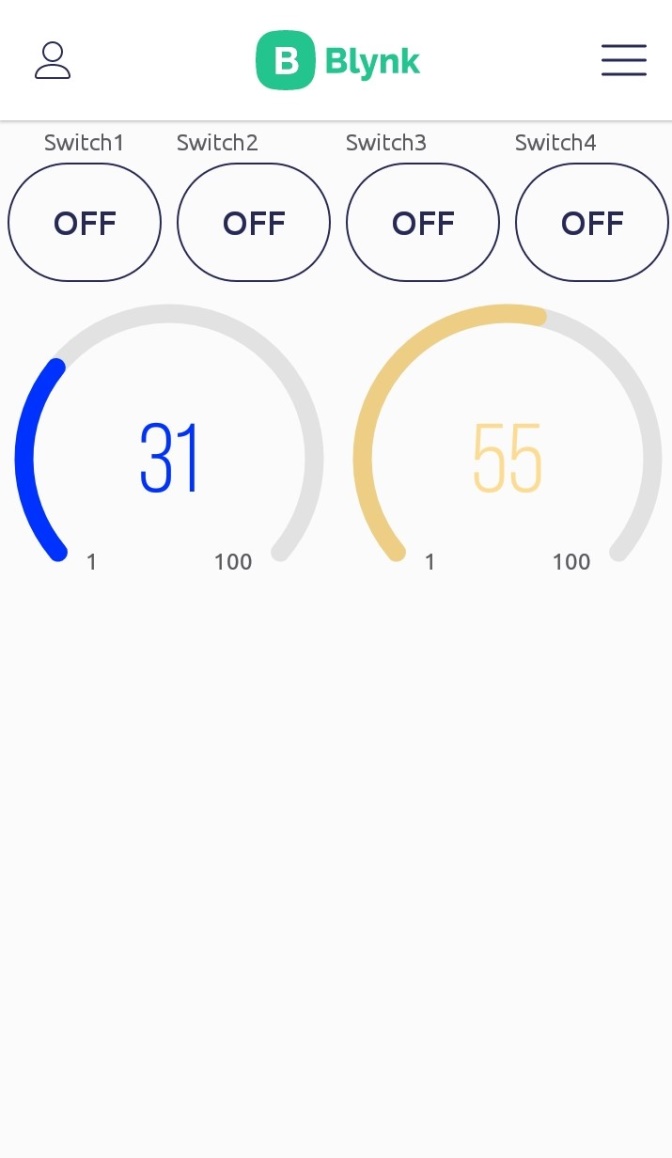


**CIRCUIT DIAGRAM**

**WEB DASHBOARD**

****

**MOBILE APP DASHBOARD**

****

* We can entirely change the interface in both the dashboard for our needs which is very useful. This BLYNK IOT service provides free account and some limited widgets for templates. Also we can create multiple templates in this and create multiple projects. Here the time delay and latency when controlling through app is very low it also may buffer due to wifi.
* BLYNK IOT is a paid service but they also provide free service which is useful for developers.
* The source code and programs for the project is completely open source and free,

**CODE FOR THIS PROJECT.**

* The node mcu can programmed using specific software named “ARDUINO IDE” which is open source and free to use.
* The programe is written in c++ language.

#include <LiquidCrystal\_I2C.h>

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <DHT.h>

char auth[] = "UtOLhoshQ2GSAlI9rhUl52I6R4pCOTPp"; //Enter your Auth token

char ssid[] = "opensource"; //Enter your WIFI name

char pass[] = "12345678"; //Enter your WIFI password

LiquidCrystal\_I2C lcd(0x27, 16, 2);

DHT dht(D3, DHT11); //(sensor pin,sensor type)

BlynkTimer timer;

//gpio pin connected

#define relay1 14

#define relay2 12

#define relay3 13

#define relay4 2

void setup() {

Serial.begin(9600);

lcd.init();

lcd.backlight();

pinMode(relay1, OUTPUT);

pinMode(relay2, OUTPUT);

pinMode(relay3, OUTPUT);

pinMode(relay4, OUTPUT);

digitalWrite(relay1, HIGH);

digitalWrite(relay2, HIGH);

digitalWrite(relay3, HIGH);

digitalWrite(relay4, HIGH);

Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

dht.begin();

lcd.setCursor(0, 0);

lcd.print("Home Automation");

lcd.setCursor(0, 1);

lcd.print("KASIM FUADH");

delay(4000);

lcd.clear();

//Call the functions

timer.setInterval(100L, DHT11sensor);

}

//Get the DHT11 sensor values

void DHT11sensor() {

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(h) || isnan(t)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

Blynk.virtualWrite(V5, t);

Blynk.virtualWrite(V6, h);

lcd.setCursor(0, 0);

lcd.print("T:");

lcd.print(t);

lcd.setCursor(0, 1);

lcd.print("H:");

lcd.print(h);

}

BLYNK\_CONNECTED()

{

Blynk.syncVirtual(V1);

Blynk.syncVirtual(V2);

Blynk.syncVirtual(V3);

Blynk.syncVirtual(V4);

}

//Get buttons values

BLYNK\_WRITE(V1) {

bool RelayOne = param.asInt();

if (RelayOne == 1) {

digitalWrite(relay1, LOW);

} else {

digitalWrite(relay1, HIGH);

}

}

//Get buttons values

BLYNK\_WRITE(V2) {

bool RelayTwo = param.asInt();

if (RelayTwo == 1) {

digitalWrite(relay2, LOW);

} else {

digitalWrite(relay2, HIGH);

}

}

//Get buttons values

BLYNK\_WRITE(V3) {

bool RelayThree = param.asInt();

if (RelayThree == 1) {

digitalWrite(relay3, LOW);

} else {

digitalWrite(relay3, HIGH);

}

}

//Get buttons values

BLYNK\_WRITE(V4) {

bool RelayFour = param.asInt();

if (RelayFour == 1) {

digitalWrite(relay4, LOW);

} else {

digitalWrite(relay4, HIGH);

}

}

void loop() {

Blynk.run();//Run the Blynk library

timer.run();//Run the Blynk timer

}