Home automation system based on IOT

- **Project Team:**

- Rahmah Elsayed Mohamed Ibrahim
- Nourhan Ahmad Mohamed Shams
- Mariam Fathy Ismail Mostafa
- Manar Mamdouh Mohamed Wagdi
- Zahra Badawy Soliman
- Mawada Salah El-Mogy
- Ahmed soliman thabet

overwiew

We make a Full Home Automation system using the Nodemcu board and the New Blynk app. This project aims to monitor the home remotely and works to protect against risks, This system includes Temperature and Humidity monitoring system, GAS level monitoring system, Security system, water level monitoring system, and home appliances controlling system, we can monitor these data on the Blynk

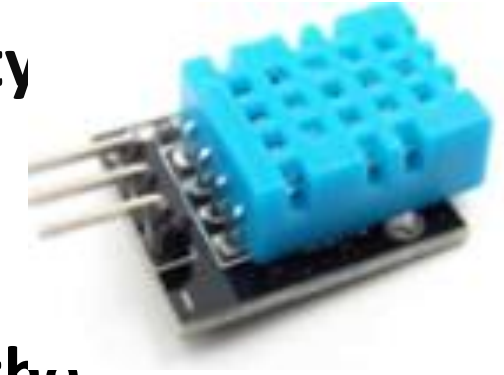
Components

- Node Mcu ESP8266
- Relay Module 2-channel
- Ultrasonic sensor
- PIR sensor
- LCD
- Breadboard
- MQ2 sensor
- DHT11 sensor
- Flame sensor
- Buzzer 5v
- Wire

Temperature and Humidity monitoring system

In this system, the room temperature and humidity are measured, and the output is displayed on the LED, and the output shows on the Blynk app.

We used DHT11 sensor which consists of a capacitive humidity sensing element and a thermistor for sensing temperature.



For measuring the humidity, sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them, Change in the capacitance value occurs with the change in humidity levels.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature.

GAS level monitoring system

In this system, a gas leak or a fire is detected, as soon as either of them occurs, Buzzer makes a sound As a fire alert and the output is displayed on the LED, and the output shows on the Blynk app

we used the MQ-2 sensor through which we can get the LP Gas, Smoke, Hydrogen, Propane, Methane, and Carbon monoxide.

When it works at a temperature of 200 to 300 °C, the tin dioxide adsorbs the oxygen in the air, which reduces the density of electrons on the semiconductor and thus increasing the resistance

When it comes into contact with smoke, if the barriers at grain boundaries are changed by the smoke, the conductivity on the surface is changed accordingly. In this way, smoke can be detected. A higher smoke density indicates a higher conductivity and lower output resistance.

water level monitoring system

In this system, the water level is monitored to avoid drowning of the room by measuring the distance between the water level and the sensor, and the result is shown on the Blynk app.

We used Ultrasonic sensors which works by emitting sound waves at a frequency too high for humans to hear. They then wait for the sound to be reflected back, calculating distance based on the time required.

Home security system

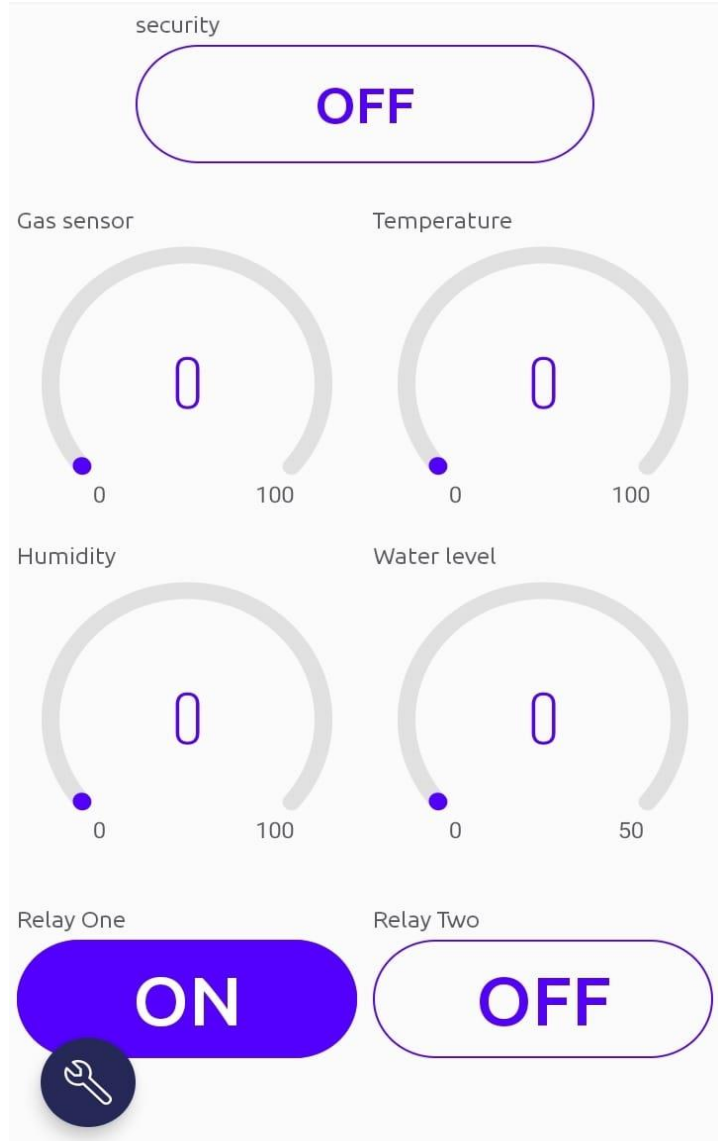
In this system, if we want to secure, for example, the house or anything We used the PIR sensor for detecting motions and makes a sound as an alert of strange movement in the place

The PIR sensor use a pair of pyroelectric sensors to detect heat energy in the surrounding environment. These two sensors sit beside each other, and when the signal differential between the two sensors changes it triggers an alarm, notifies authorities, IR radiation focuses on each of the two pyroelectric sensors using a series of lenses constructed as the sensor's housing

Blynk app

We use the Blynk app which was designed for IOT or Android smartphones. It can control hardware remotely, it can display sensor data, it store data, visualize it, It allows you to create amazing interfaces for your projects using various widgets which are provided. It is responsible for all the communications between the smartphone and hardware.

sensors and actuators are connected to NodeMCU which are run in client mode. This MCU node is connected to raspberry pi as a wireless server. Furthermore, the Raspberry pi is connected to the internet network via a LAN cable. On raspberry pi the application runs connected to the Blynk server via the internet. after all is connected, the smart phone can be seen on the widget to adjust the actuator and the sensors reading.


security
OFF
Gas sensor
0
0
100
Temperature
0
0
100
Humidity
0
0
100
Water level
0
0
50
Relay One
ON
Relay Two
OFF

code

```
/*Full home automation with the New Blynk app*/
//Include the library files
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
//Initialize the LCD display
LiquidCrystal_I2C lcd(0x27, 16, 2);
char auth[] = "";//Enter your Auth token
char ssid[] = "";//Enter your WIFI name
char pass[] = "";//Enter your WIFI password
DHT dht(D3, DHT11); //(sensor pin,sensor type)
BlynkTimer timer;
bool pirbutton = 0;
```

```
// Define component pins
#define Buzzer D0
#define MQ2 A0
#define trig D4
#define echo D5
#define PIR D6
#define relay1 D7
#define relay2 D8
//Get buttons values
BLYNK_WRITE(V0) {
  pirbutton = param.asInt();}
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
pinMode(Buzzer, OUTPUT);
  pinMode(PIR, INPUT);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
```

```
digitalWrite(relay1, HIGH);
  digitalWrite(relay2, HIGH);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  dht.begin();

  lcd.setCursor(0, 0);
  lcd.print("Home Automation");
  lcd.setCursor(4, 1);
  lcd.print("System");
  delay(4000);
  lcd.clear();

//Call the functions
  timer.setInterval(100L, gassensor);
  timer.setInterval(100L, DHT11sensor);
  timer.setInterval(100L, pirsensor);
  timer.setInterval(100L, ultrasonic);
}
```

```
//Get the MQ2 sensor values
void gassensor() {
  int value = analogRead(MQ2);
  Serial.println(value);
  value = map(value, 0, 1024, 0, 100);
  if (value <= 55) {
    digitalWrite(Buzzer, LOW);
  } else if (value > 55) {
    Blynk.notify("Warning! Gas leak detected");
    digitalWrite(Buzzer, HIGH);
  }
  Blynk.virtualWrite(V1, value);
  lcd.setCursor(0, 0);
  lcd.print("G:");
  lcd.print(" ");
  lcd.print(value);
}
```

```
//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(V2, t);
  Blynk.virtualWrite(V3, h);
  lcd.setCursor(8, 0);
  lcd.print("T:");
  lcd.print(t);
  lcd.setCursor(0, 1);
  lcd.print("H:");
  lcd.print(h);
```

```
//Get the PIR sensor values
void pirsensor() {
  bool value = digitalRead(PIR);
  if (pirbutton == 1) {
    if (value == 0) {
      digitalWrite(Buzzer, LOW);
    } else if (value == 1) {
      Blynk.notify("Warning! Please check your security system");
      digitalWrite(Buzzer, HIGH);}}}
//Get the ultrasonic sensor values
void ultrasonic() {
  digitalWrite(trig, LOW);
  delayMicroseconds(4);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  long t = pulseIn(echo, HIGH);
  long cm = t / 29 / 2;
```

```
  Blynk.virtualWrite(V4, cm);
  lcd.setCursor(8, 1);
  lcd.print("W:");
  lcd.print(cm);
  lcd.print("  ");
}

//Get buttons values
BLYNK_WRITE(V5) {
bool RelayOne = param.asInt();
  if (RelayOne == 1) {
    digitalWrite(relay1, LOW);
  } else {
    digitalWrite(relay1, HIGH);
  }
}
```

```
//Get buttons values
BLYNK_WRITE(V6) {
bool RelayTwo = param.asInt();
  if (RelayTwo == 1) {
    digitalWrite(relay2, LOW);
  } else {
    digitalWrite(relay2, HIGH);
  }
}


void loop() {
  Blynk.run();//Run the Blynk library
  timer.run();//Run the Blynk timer
}
```