Experiment – 1.4

Student Name: Diwakar Kumar   **UID: 22BCS10849**

**Branch: BE-CSE**   **Section/Group: 22BCS_IOT-640/B**

Semester: 6th   **Date of Performance: 19/02/25**

Subject Name: IOT LAB   **Subject Code: 22CSP-367**

1. **Aim:** Build a security system with any sensor and alerts using Blynk.

2. **Objective:** To design and implement a security system using sensors (e.g., Gas sensor, magnetic door sensor, or ultrasonic sensor) and integrate it with the Blynk platform to send real-time alerts.

3. **Hardware Used:**

   - ESP8266/NodeMCU(or any Wi-Fi-enabled microcontroller)
   - Buzzer/LED (for local alerts, optional)
   - Blynk App (installed on your smartphone)
   - Breadboard and jumper wires
   - Gas Sensor (HC-SR04)

4. **Procedure:**
   a. **Connect the Hardware:**
      i. **Connect the Hardware: MQ-135Gas Sensor Pinout:**
         1) VCC: Connect to 3.3V or 5V (depending on the sensor model).
         2) GND: Connect to GND.
         3) AO(Analog Output): Connect to an analog pin on ESP8266 (e.g., A0).
         4) DO(Digital Output, optional): Connect to a digital pin on ESP8266 (e.g., D6).
      ii. **Wiring Diagram:**
         1) MQ-135 VCC→NodeMCU3.3V/5V
         2) MQ-135 GND→NodeMCUGND
         3) MQ-135 AO→NodeMCUA0
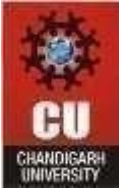
    4) MQ-135 DO(optional) → NodeMCU D6

    5) Buzzer/LED (optional) → D4

  b. **Set Up Blynk:**

- Download and install the Blynk app (iOS/Android).

- Create a new project and select ESP8266 as the device.

- Note down the Auth Token sent to your email.

- Add a Notification Widget in the app for alerts.

  c. **Install Libraries inArduino IDE: → Blynk Library:**

- Go to Tools > Manage Libraries and search for Blynk.

- Install the Blynk library.

  → ESP8266 Board Support:

- Go to File > Preferences and add the following URL to the Additional Boards Manager.

- http://arduino.esp8266.com/stable/package_esp8266com_index.json Go to Tools > Board > Boards Manager and install the ESP8266 package.

5. Code:

```
#define BLYNK_TEMPLATE_ID "TMPL3-ppAtY5O"
#define BLYNK_TEMPLATE_NAME "Pragyan"
#define BLYNK_AUTH_TOKEN "q1UpxNTwWPFEpugYJL2giq8Jv8waiOcC"
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// Blynk and Wi-Fi credentials char auth[] = "
q1UpxNTwWPFEpugYJL2giq8Jv8waiOcC ";  char ssid[] =
"Pragyan OPPO A96";  char pass[] = " c5zp7gzn ";

// PIR sensor pin
int pirPin = D5; int buzzerPin = D2;
void setup() {  Serial.begin(115200);
Blynk.begin(auth, ssid, pass);  pinMode(pirPin,
INPUT);  pinMode(buzzerPin, OUTPUT);
digitalWrite(buzzerPin, LOW);
Serial.println("Security system ready.");
```
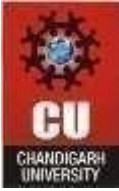
```
}  void
loop() {
Blynk.run();
if (digitalRead(pirPin) == HIGH) {

Serial.println("Motion Detected!");
Blynk.notify("Alert! Motion Detected at Home.");
digitalWrite(buzzerPin, HIGH);
Turn on buzzer/LED delay(5000); // Alert duration  digitalWrite(buzzerPin,
LOW); // Turn off buzzer/LED
}
}
```

Blynk Code

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
BlynkTimer timer;
char auth[] = "xxxxx"; //Enter the authentication code sent by Blynk to your Email
char ssid[] = "xxxxx"; //Enter your WIFI SSID  char pass[] = "xxxxx"; //Enter
your WIFI Password  int flag=0;
void notifyOnButtonPress() {  int isButtonPressed
= digitalRead(D1);  if (isButtonPressed==1 &&
flag==0) {  Serial.println("Someone Opened the
door");  Blynk.notify("Alert : Someone Opened
the door");  flag=1; } else if
(isButtonPressed==0){ flag=0; }
} void
setup()
{
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
pinMode(D1,INPUT_PULLUP);
timer.setInterval(16000L,notifyOnButtonPress);
}  voidloop()
{
```
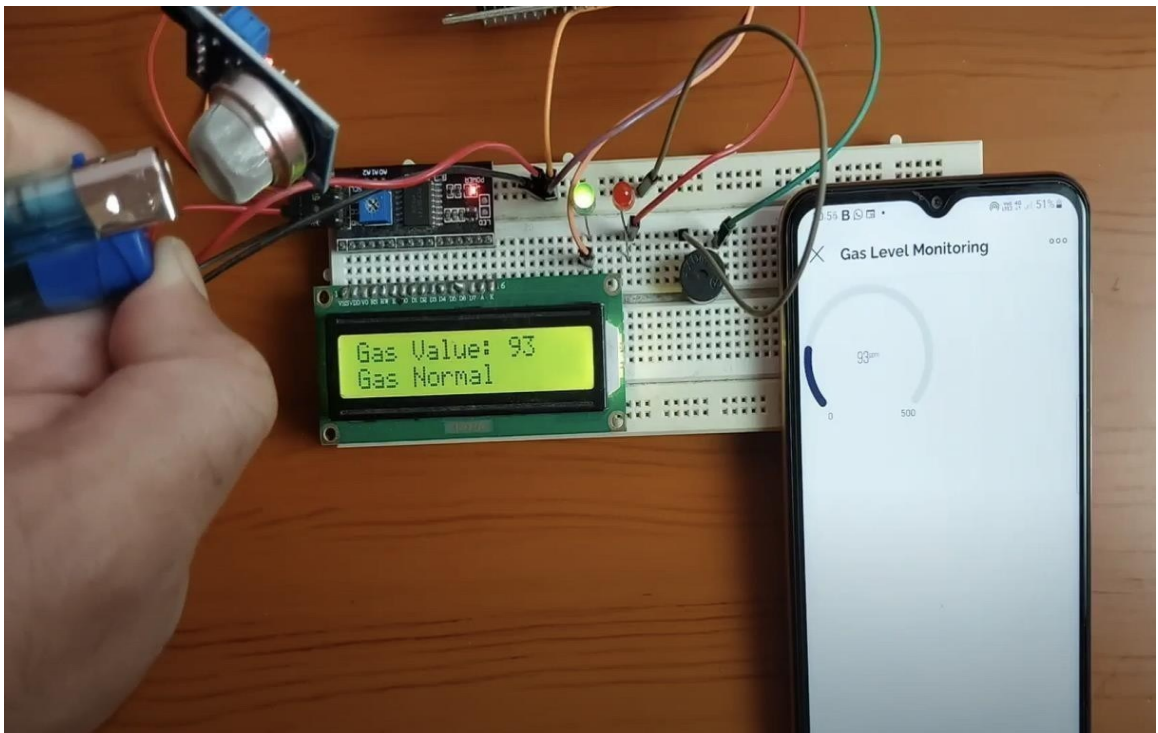
```
Blynk.run();
timer.run();
}
```
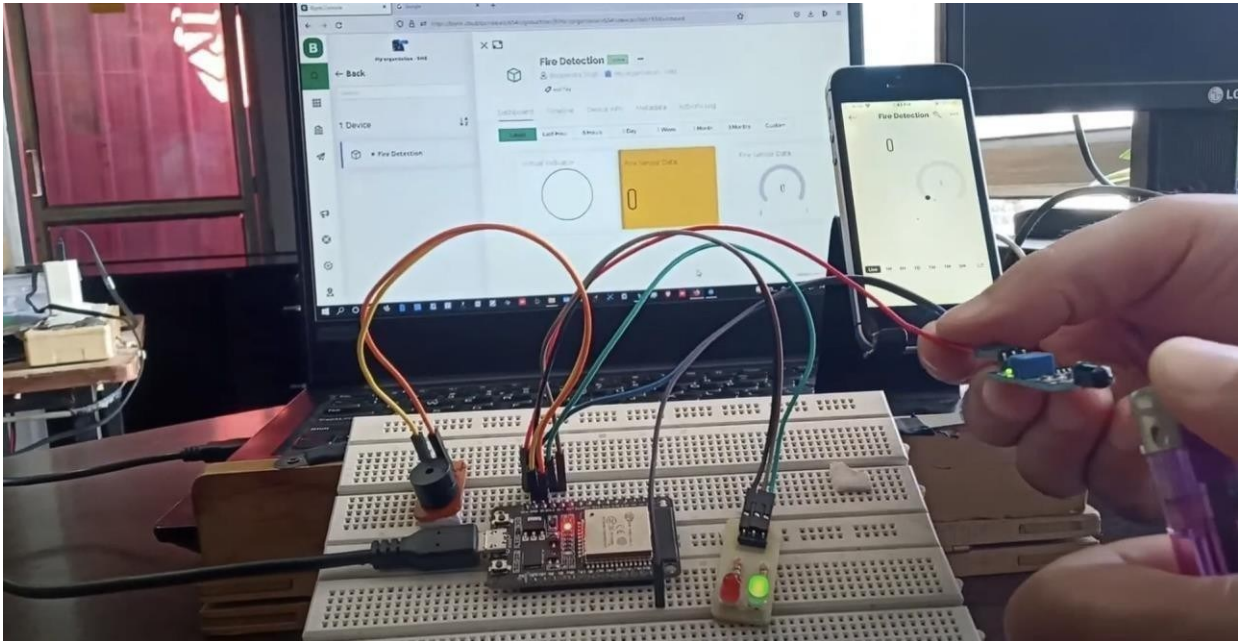
6. **Output:**

Fig 1

Fig 2

7. **Learning Outcome:**

- **IoT and Blynk Integration** – Learn how to connect sensors with Blynk for real time monitoring and remote alerts.

- **Sensor and Hardware Interfacing** – Gain hands-on experience in working with motion, door, or gas sensors and microcontrollers like ESP8266/ESP32.

- **Alert Mechanisms** – Implement real-time notifications via Blynk (push alerts, email, or SMS) and physical alerts using buzzers or LEDs.

- **Embedded Programming** – Develop coding skills in C++ (Arduino IDE) or Micro Python to process sensor data and trigger security actions.