

Fire Detection Alarm

Raghav Gupta
Computer Science Engineering
190101069
IOT DIY Project

Abstract—Fire alarm systems detect the fire and trigger a loud alarm to aware everybody. So to inform the authority about any fire incident today we are building a IOT based Fire Alarm system which not only trigger an alarm but also sends a Email alert to concern persons. This method can also be used to inform fire department automatically in case of fire. Here we will use Infrared Flame Sensor to detect the fire and ESP8266 NodeMCU to trigger the alarm and send email via Blynk Cloud.

I. IMPLEMENTATION ATTRIBUTES

- The flame sensor is relaxed in normal state and as soon as it detects the fire, buzzer will starts to alarm and the notification is sent to the user's mail and on the mobile.
- User can turn off the buzzer form the phone via Blynk cloud app, sends the instruction back to NodeMCU and if the fire detects it again it will start the alarm again .
- The fire sensor sense the fire and send data to NodeMCU which in turn sends data to Blynk cloud displays it on to the dashboard and reverts back to NodeMCU then instructs buzzer to alarm.
- When the status from relaxed changes to fire the notification sends to the user.

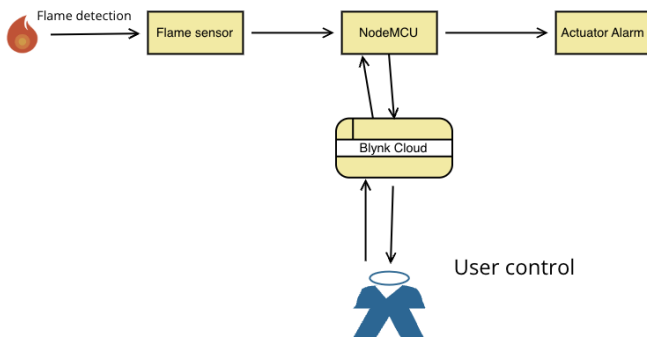


Fig. 1. Data FLOW

II. CONFIGURATION DIAGRAM

A Flame Sensor is a device that can be used to detect presence of a fire source or any other bright light sources. The module uses a LM393 comparator chip to provide a stable digital output signal. This comparator has a driving ability of 15 mA.

All the red and black wired are Vcc and GND connection. Flame sensor is connected to D0 and buzzer to D1

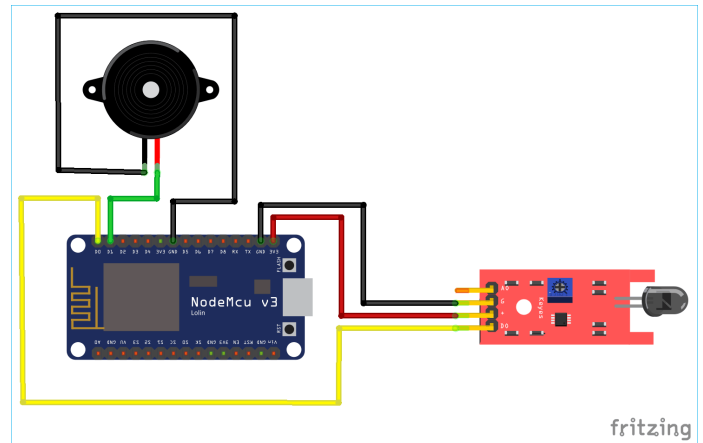


Fig. 2. Circuit Diagram

III. OUTPUT

Initially when the device is online and no fire is there the fire alarm is in the relaxed state

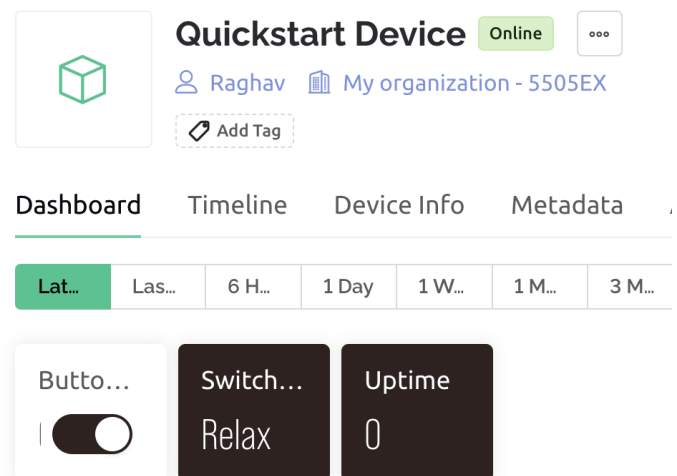


Fig. 3. Dashboard without flame detection

When Fire is Detected is updates its status to Flame!!! and trigger the event of sending notification

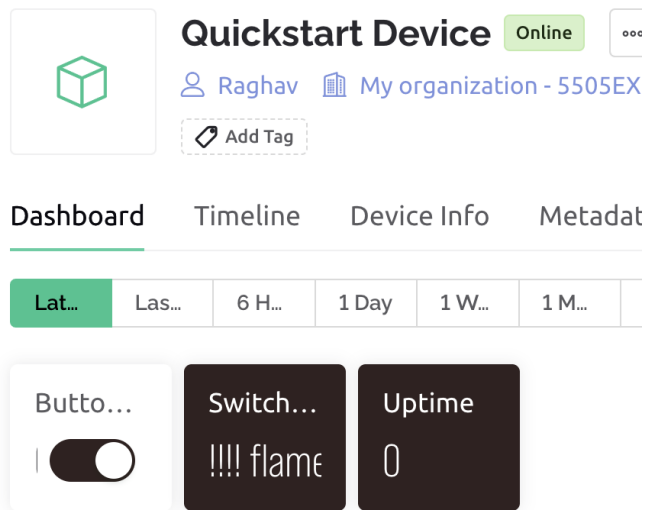


Fig. 4. Dashboard with flame detected

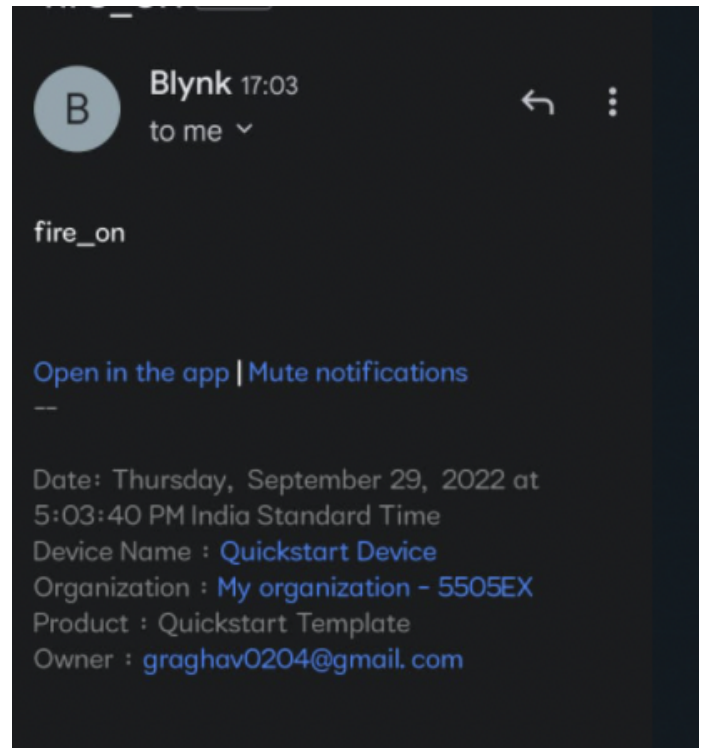


Fig. 6. E-mail notification

Alarm is beeping and notification is sent on phone and email ,i.e. triggering the event and user can turn off the alarm by pressing the alarm button.

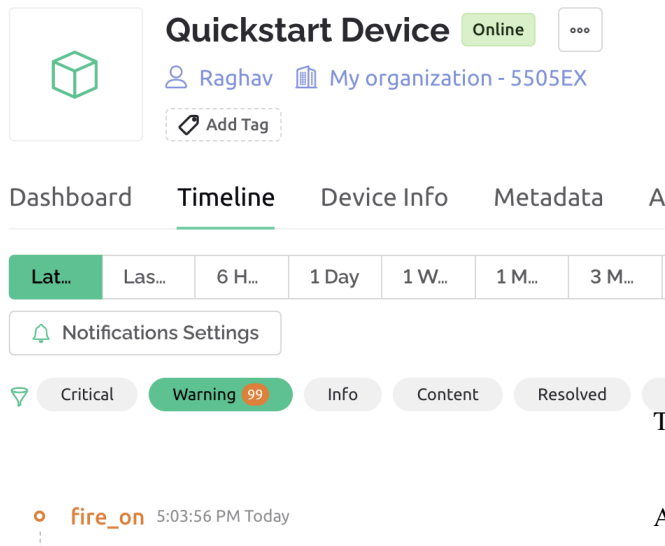


Fig. 5. events Timeline

```
1 #define BLYNK_TEMPLATE_ID "TMPLOFXDMJWH"
2 #define BLYNK_DEVICE_NAME "Quickstart Device"
3 #define BLYNK_AUTH_TOKEN "aEkoWYIx0oD_oQL8FrKvqA8R1MmQCuCh"
4
5
6 // Comment this out to disable prints and save space
7 #define BLYNK_PRINT Serial
8
9
10 #include <ESP8266WiFi.h>
11 #include <BlynkSimpleEsp8266.h>
12
13 char auth[] = BLYNK_AUTH_TOKEN;
14
15 // Your WiFi credentials.
16 // Set password to "" for open networks.
17 char ssid[] = "";
18 char pass[] = "";
19 bool flag1=1;
20 bool flag2=1;
```

Fig. 7. Blynk connection

Then the setup phase which include the initial setup code which includes initialising all the variables

At last we are defining the loop phase which will run continuously till getting power and in this we are detecting and triggering our events and all conditions

notification on the mail and on the mobile. the NodeMCU is basically sending the data to Blynk cloud which after that is triggering an event in which it is sending the notification.

IV. CODE SNIPPET

The initials for setting up with Blynk cloud which also include to connect with the WiFi and sync. with Blynk

```

21 void setup() {
22     // put your setup code here, to run once:
23     pinMode(14, OUTPUT);
24     pinMode(12, INPUT);
25     pinMode(D2, OUTPUT);
26     Serial.begin(9600);
27     Blynk.begin(auth, ssid, pass);
28     Blynk.virtualWrite(V0, 1);
29 }
30 BLYNK_WRITE(V0) {
31     if(param.asInt() == 1){
32         flag1=1;
33     } else{
34         flag1=0;
35         flag2=1;
36     }
37 }

```

Fig. 8. Setup Section

```

} void loop() {
    Blynk.run();
    // put your main code here, to run repeatedly:
    int flamesensor=digitalRead(12); // reading the output of flame sensor |
    Serial.print("Flame Status:");
    Serial.println(flamesensor);
    if(!flamesensor&&flag1&&flag2){
        Blynk.notify("Alert Flame Detected"); // first time flame detected
        flag2=0;
    }
    if(!flag2){ // after restart
        Blynk.virtualWrite(V1, "!!!! flame detected"); // detecting fire
        Blynk.logEvent("fire_on"); // notification sending event
        digitalWrite(14,HIGH);
        digitalWrite(D2,HIGH);
        delay(50);
        digitalWrite(14,LOW);
        digitalWrite(D2,LOW);
        delay(50);
    } else {
        Blynk.virtualWrite(V1, "Relax"); // not detecting fire
    }
}
}

```

Fig. 9. loop section