3rd Year CSE, Zagazig University
Course: Computer Organization 2
Professor: Dr. Tamer Sami

Fire detection and extinguishing robot using IOT (Project 6)

Team:

Nour El-Din Mustafa Ibrahim Nafea

Alzahraa Mohamed Abdel Hamid Shaheen

Ferial Ahmed El-Sayed Amr

Salma Mohamed Fathy Ali

Eman Mohamed Ahmed Gomaa

Eman Abdelghany Eldeeb

Mariam Magdy Abdel Masih Michael

Nourhan Salem Abdel Hamid

Nada Ashraf Abdel Moneim Hashem

Mohamed Fatouh Mohamed Saeed

Abstract

This project aims to make us explore IOT field and apply what we have learned on one of its popular applications. Our project solves fire spread problem that happened because of late detection and response, so we use sensor to discover the fire at the beginning and send Realtime data to indicate the robot to extinguish the fire.

Introduction

Internet of Things (IOT) is a preferred system for Internet-connected devices with the ability to collect data from users or the environment without human interaction.

The device or “thing” in IOT can be with any device embedded in electronics, software, and sensors such as a smart refrigerator, smart air conditioner, lamps at home, designed safety systems, or even a person with a heart monitor. Internet of Things can create information about the connected objects, analyze it, and make decisions.

Project description

The project has two parts:

- Fire detection part: we have used a Flame sensor with NodeMcu to detect fire existing and send data to firebase. When there is a fire, the flame state variable in the firebase will be set to true as an indicator to the website and the robot that connected with the firebase.
- Fire extinguish part: we have made a robot controlled by another NodeMcu to receive the real-time data and drive the robot when the flame state is true and turn on a water pump (we replaced the pump with a led).

Discussion

Hardware Components:

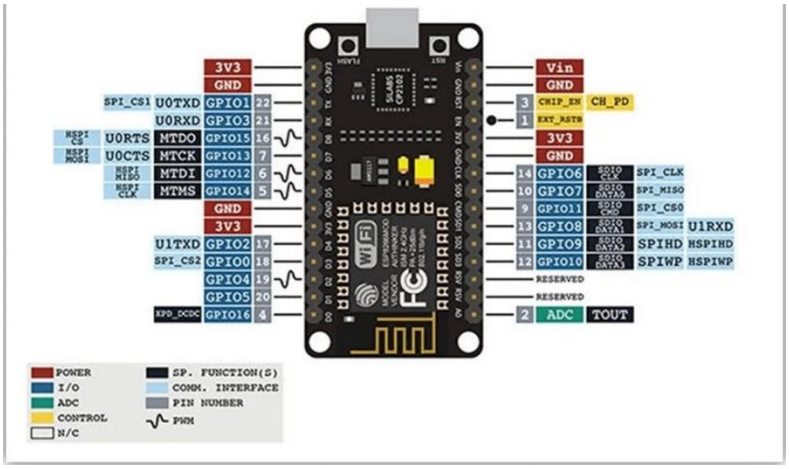
- NodeMcu

  NodeMCU is an open-source development board and firmware based in the widely used ESP8266 -12E Wifi module. It allows you to program the ESP8266 Wifi module with the simple and powerful LUA programming language or Arduino IDE. With just a few lines of code you can establish a

Wifi connection and define input/output pins according to your needs exactly like arduino, turning your ESP8266 into a web server and a lot more. It is the Wifi equivalent of ethernet module. Now you have internet of things (iot) real tool. With its USB-TTL, the nodeMCU Dev board supports directly flashing from USB port. It combines features of WIFI access point and station + microcontroller. These features make the NodeMCU extremely powerful tool for Wifi networking. It can be used as access point and/or station, host a webserver or connect to internet to fetch or upload data.

Applications of NodeMCU:

- Prototyping of IoT devices
- Low power battery operated applications
- Network projects
- Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities



[Datasheet link]

- ## Flame Sensor

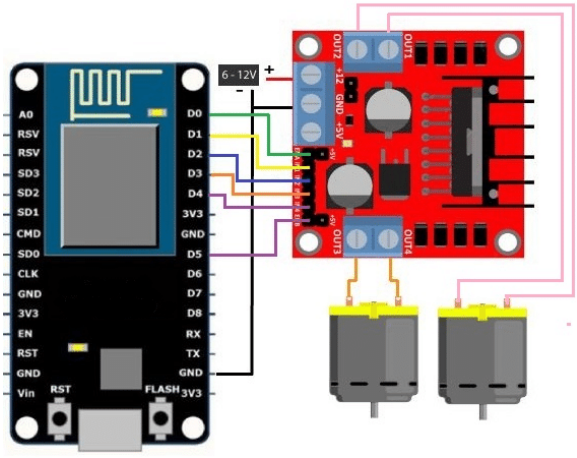
  A flame-sensor is one kind of detector which is mainly designed for detecting as well as responding to the occurrence of a fire or flame. The flame detection response can depend on its fitting. It includes an alarm system, a natural gas line, propane & a fire suppression system. The response of these sensors is faster as well as more accurate compare with a heat/smoke detector because of its mechanism while detecting the flame.

  This sensor/detector can be built with an electronic circuit using a receiver like electromagnetic radiation. This sensor uses the infrared flame flash method, which allows the sensor to work through a coating of oil, dust, water vapor, otherwise ice.

  [Datasheet link]

- ## Robot Parts

  Batteries- Wheels- Base- Motors- Motor Driver l298n



Software Parts

- Firebase Realtime Database

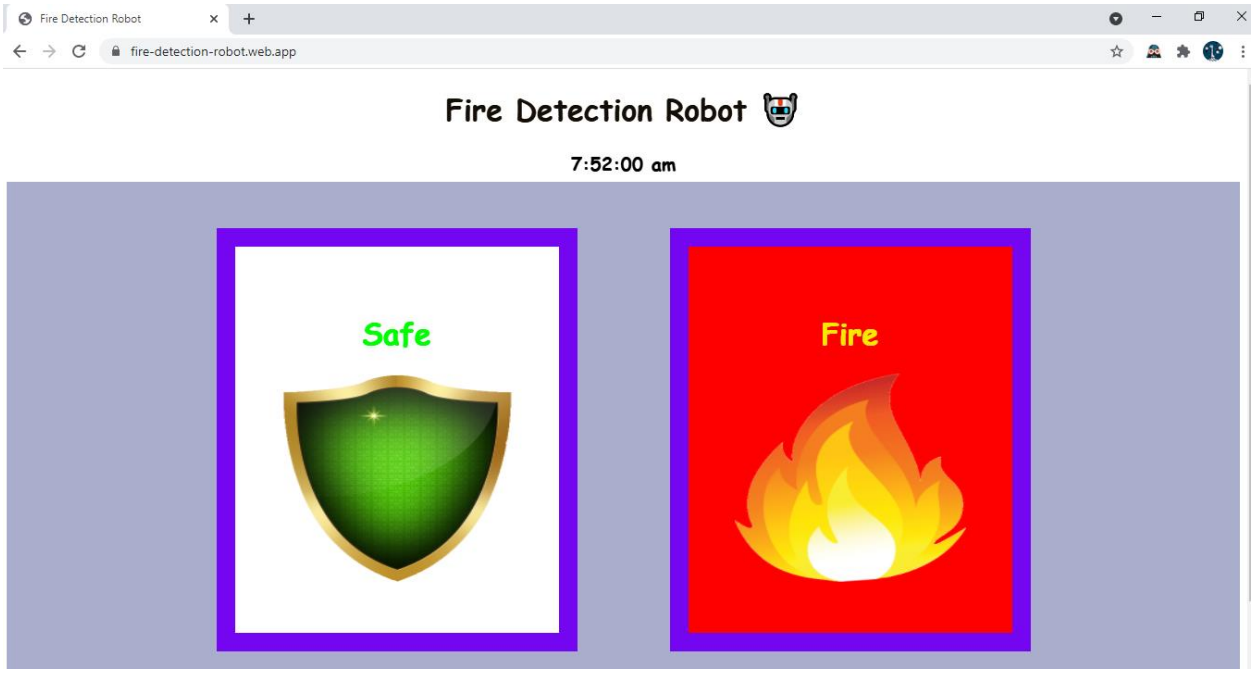
  Most of IoT projects require some way of communication between the different endpoints. These endpoints can be anything from devices and services to applications, and eventually data needs to be stored somewhere for further processing and analyzing.

  Firebase provides a quick way to keep sensory data collected at the device level, we can connect the device to the Firebase Realtime Database and make it push data periodically to the database. On the other part of the system, we have a web application which will be connected to the same service as the device and will receive new data whenever there is a change in the database.

- Website Dashboard

  Firebase offers a hosting service that we can use to host our website and dealing with deployment and networking configurations.

  We have created a dashboard to indicate if there is a fire or safe.



[Fire detection robot dashboard link]

We have configured firebase in java script code to toggle between states corresponding to firebase.

```
(function(){
    var firebaseConfig = {

        apiKey: "eV6sxUs1lTPkcyQphKwf9z1u4rlOxb3nlEmMNDWS",
          authDomain: "fire-detection-robot-default-rtdb.firebaseio.com",
          databaseURL: "https://fire-detection-robot-default-rtdb.firebaseio.com",
          projectId: "fire-detection-robot",
          storageBucket: "fire-detection-robot.appspot.com",
          messagingSenderId: "1098199709707"
      };

    //   Initialize Firebase
    firebase.initializeApp(firebaseConfig);

            //   Initialize Firebase
            firebase.initializeApp(firebaseConfig);

            var database = firebase.database(); //get reference to the database



            firebase.database().ref("FLAME STATE").on("value", (snapshot)=>{
              const val = snapshot.val();
              var element1 = document.getElementById("box1");
              var element2 = document.getElementById("box2");
              if (val == true) //fire is detected
              {
                console.log(val);
                element1.style.backgroundColor = "white";
                element2.style.backgroundColor = "red";
              }

              else
              {
                element1.style.backgroundColor = "green";
                element2.style.backgroundColor = "white";
                console.log(val);
              }

            })
            }())
```

html code shows the current real time and two states safe with green and fire with red.

```
<header>
    <h1 style="color:rgb(20, 14, 7);">Fire Detection Robot 👹</h1>
    <iframe src="https://free.timeanddate.com/clock/i7spz91u/n4501/tleg/fn7/fs20/tct/pct/ftb/th2" frameborder="0" widtl
</header>

<div class="box-row">

    <div id="box1">
        <h1 style="color:rgb(0, 255, 0);">Safe</h1>
        <img src="shield-logo.png" style="width:250px;height:225px;">
    </div>

    <div id="box2">
        <h1 style="color:rgb(250, 233, 0);">Fire</h1>
        <img src="fire-logo.png" style="width:250px;height:225px;">
    </div>

</div>
```

- Programming NodeMcu

  We have used Arduino IDE for programming NodeMcu boards.

  We have two codes :

  - Code 1: Reads from flame sensor and sends data to firebase.
  - Code 2: Receives data from firebase and controls the robot motors and the pump.

There is a common function in both codes this to connect NodeMcu board to Wifi.

```
void connectToWiFi(char const *ssid, char const *password)
{
  delay(10);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.print(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println("");
}
```

Code 1:

FlameSend | Arduino 1.8.13

File Edit Sketch Tools Help

FlameSend §

```
//first we include the libraries of Wi-Fi, firebase
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <FirebaseArduino.h>

//Wi-Fi name and password
#define SSID "space1"
#define PASSWORD "@lmor@ba3"

//firebase information
#define FIREBASE_HOST "fire-detection-robot-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "eV6sxUs1lTPkcyQphKwf9z1u4rlOxb3nlEmMNDWS"

void connectToWiFi(char const *ssid, char const *password); // function used to connect to Wi-Fi information

const int flame = 5; //--> The pin used for the FLAME sensor is Pin D1 = GPIO5
bool flame1 = 0; //state flame {False}
int isFlame = LOW; // HIGH when Flame Exposed

void setup()
{
  Serial.begin(115200);
  connectToWiFi(SSID, PASSWORD);
  //starting firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  //set flame sensor to be input
  pinMode(flame, INPUT);
}
```

FlameSend | Arduino 1.8.13

File Edit Sketch Tools Help

FlameSend §

```
void loop(){
  isFlame = digitalRead(flame);  //reading flame value

  if (isFlame == HIGH) {
    flame1 = 1;
      //printing on serial monitor
    Serial.println("FLAME, FLAME, FLAME");
      //write in the firebase that the flame detect a fire
    Firebase.setString("FLAME", "FLAME DETECTED");
    //--> Command or code for sending data (int data type) to Firebase Realtime Database.
    Firebase.set("FLAME STATE", flame1);

   //else there is no fire write that on the firebase
  } else {
    flame1 = 0;
    Serial.println("no flame");
    Firebase.setString("FLAME", "NOT DETECTED");
    Firebase.set("FLAME STATE", flame1);
  }
  Serial.println(flame1);

  //handling error
  if (Firebase.failed())
  {
    Serial.print("setting /message failed:");
    Serial.println(Firebase.error());
    return;
  }
```

[Code 1 link]

Code 2:

Run | Arduino 1.8.13

File Edit Sketch Tools Help

Run §

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <FirebaseArduino.h>
#define SSID "space5"
#define PASSWORD "@lmor@ba3"
#define FIREBASE_HOST "fire-detection-robot-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "eV6sxUs1lTPkcyQphKwf9zlu4rlOxb3nlEmMNDWS"

void connectToWiFi(char const *ssid, char const *password);

#define LM1 4       // left motor D2
#define LM2 5       // left motor D1
#define RM1 12       // right motor D5
#define RM2 14       // right motor D6
#define pump 15     // D8

void setup()
{
  Serial.begin(9600);
  connectToWiFi(SSID, PASSWORD);
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  //set motors and pump to be outputs
  pinMode(LM1, OUTPUT);
  pinMode(LM2, OUTPUT);
  pinMode(RM1, OUTPUT);
  pinMode(RM2, OUTPUT);
  pinMode(pump, OUTPUT);
}
```

Run | Arduino 1.8.13

File Edit Sketch Tools Help

Run §

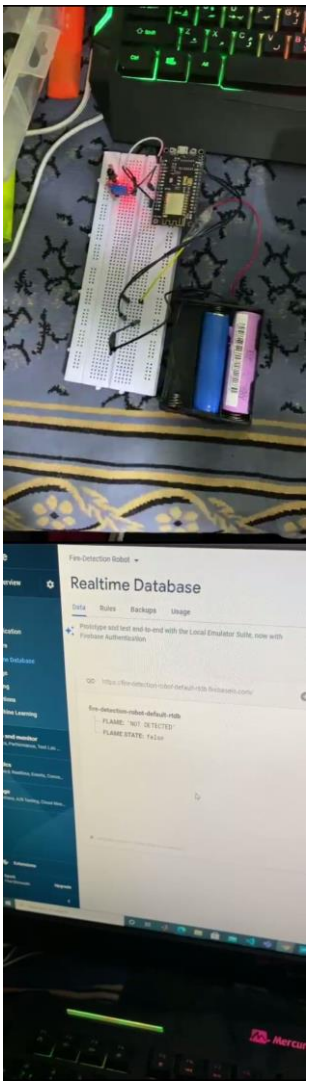
```
void loop(){
   int isFire = Firebase.getInt("FLAME STATE"); //get data integer from firebase
  //handling error
  if (Firebase.failed())  {
    Serial.print("setting /message failed:");
    Serial.println(Firebase.error());
    return;  }
   if (isFire == 1) {
    Serial.println("PUMP On");
    Serial.println();
       //Move the robot forward
    digitalWrite(LM1, HIGH);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, HIGH);
    digitalWrite(RM2, LOW);
    //calling function to extinguish the fire
    put_off_fire(); }
  else{
    Serial.println("PUMP Off");
    Serial.println();
     //Do not move the robot
    digitalWrite(LM1, LOW);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, LOW);
    digitalWrite(RM2, LOW);
   }
  delay(500);
}
```

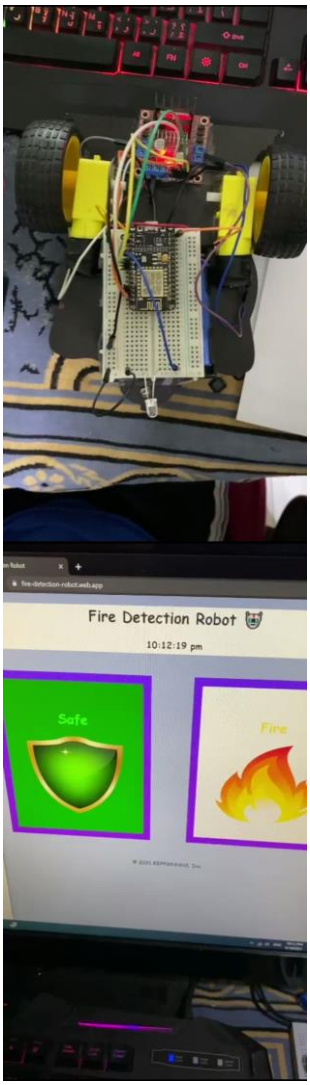
```
void put_off_fire(){
    //go forward untill delay finish
    delay (1000);
    //stop
    digitalWrite(LM1, LOW);
    digitalWrite(LM2, LOW);
    digitalWrite(RM1, LOW);
    digitalWrite(RM2, LOW);
    //turn on the pump for the dalay
   digitalWrite(pump, HIGH);
   delay(1000);
   //turn off the pump
   digitalWrite(pump,LOW);
}
```

Code 2 link

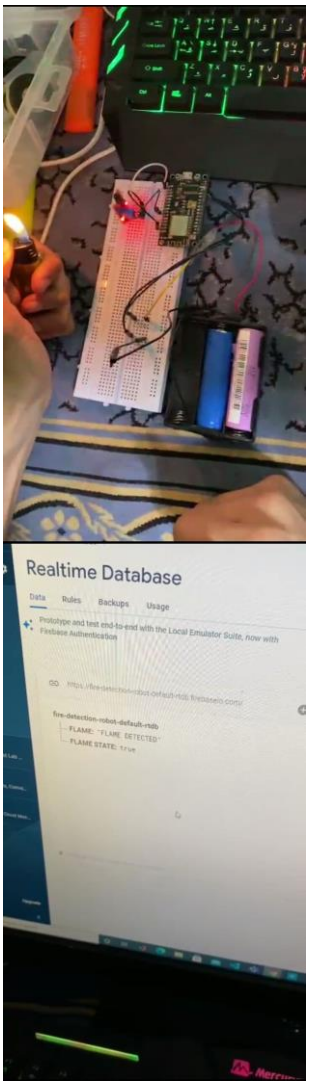
Result :

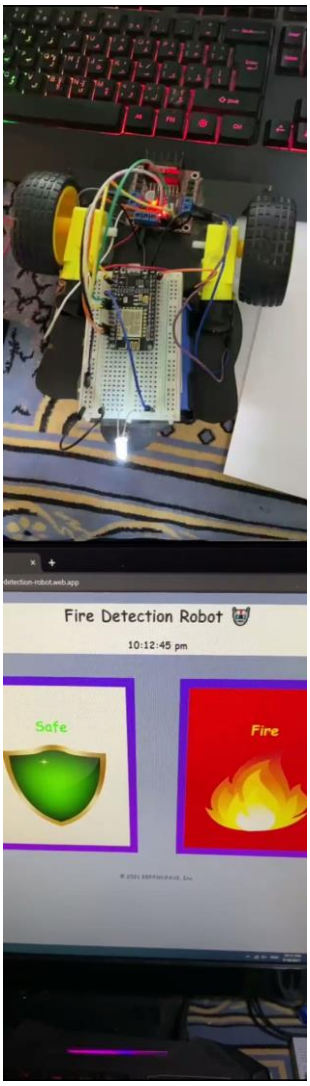
- Case 1: No Fire Detected





- Case 2: Fire Detected





Video Result Link

GitHub Repositories :

firebase-web-demo

IOT-Firebase-Demo

IOT-robot-DEMO

Problems we faced:

- To define fire location, we have to use GPS modules, but we can't get it.
  - ✓ We have drive the robot to move forward only with putting the fire in its way.
- To extinguish the fire, we have to use a water pump or a fan, but we haven't the component and time is tight.
  - ✓ We have used led as an indicator to the pump.
- We can't gather all team members to work on hardware because of covid.
  - ✓ We divide tasks software and hardware.

Future work:

- Use different sensors to define the type of fire to better dealing with.
- Use different methods to extinguish the fire depend on the type.
- Use GPS for the location of fire.
- Use ultrasonic for obstacles.

References:

iot-prototyping-with-firebase-doing-more-with-less

IoT - NodeMCU (ESP8266)

Intelligent IoT Projects in 7 Day