



SAKARYA
ÜNİVERSİTESİ

Nesnelerin İnterneti ve Uygulamaları Proje Ödevi

Hazırlayanlar:

Ali Kutay Kılınç B201210071

Berkay Can B211210048

Dersi Verenler:

Prof.Dr. Cüneyt BAYILMIŞ & Dr.Öğr.Üyesi Murat İSKEFİYELİ

Akıllı Maske ve Ev Karantinası Hasta Takip Sistemi

Evinde kendi odasında karantina altında olan bir hastanın odasından çıktıktan sonra ev içinde güvenli hareket edip etmediğini takip eden bir sistem kurduk. Maskeye takılacak ısı sensörü ile hastanın odasından çıkmadan önce maskesini takıp takmadığı kontrol ediliyor. Hastanın nefesi ısı sensörüyle tespit edilecek. MQ-3 alkol sensörü hastanın odadan çıkmadan önce elini kolonya ile dezenfekte ettiğini de tespit ediyoruz.

Hasta mutfığa gidip ocağı kullanmak isteyebilir. Ocağın yakıldığını tespit etmek için yakınına bir ateş sensörü sistemi kurduk. Ocağın yanmadan gazın açık kalması halinde hasta doğalgaz kokusunu fark edemeyeceği için MQ-6 gaz sensörüyle uyarı vermek için buzzer kullandık. Sensörden okunan değer Firebase platformuna aktarılıyor.

Maske ve ocak 2 adet NodeMcu ile internete bağlı.

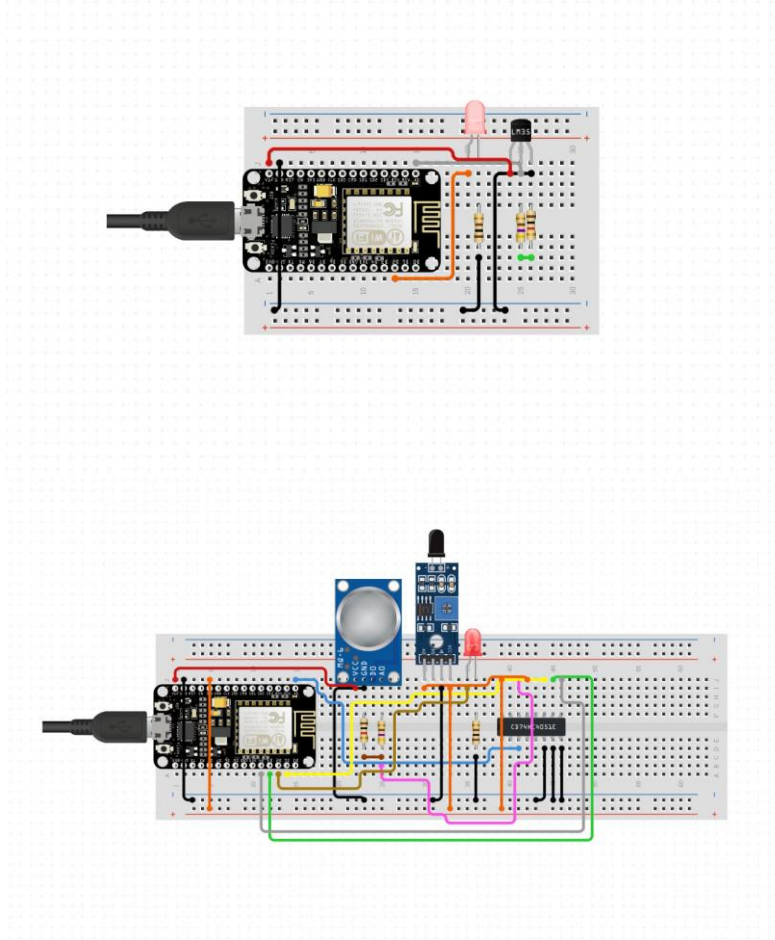
Gerekli Bileşenler

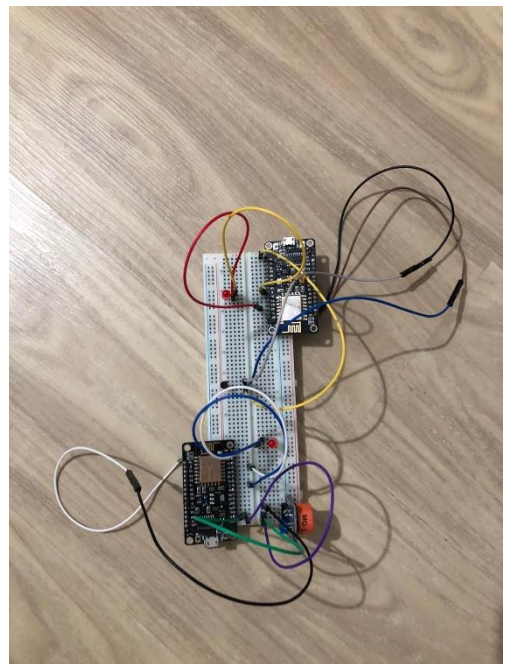
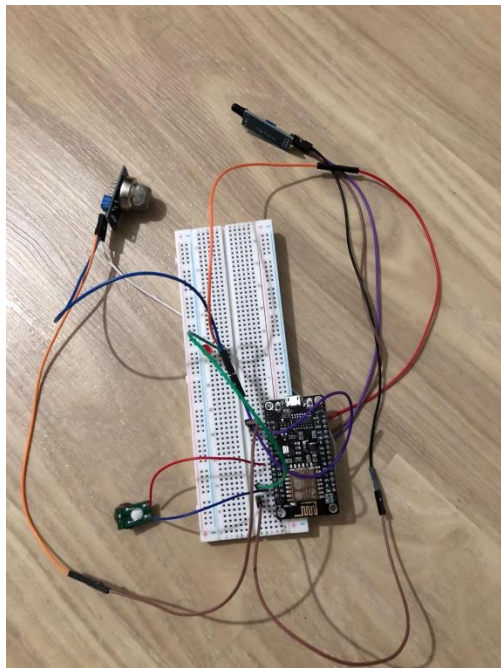
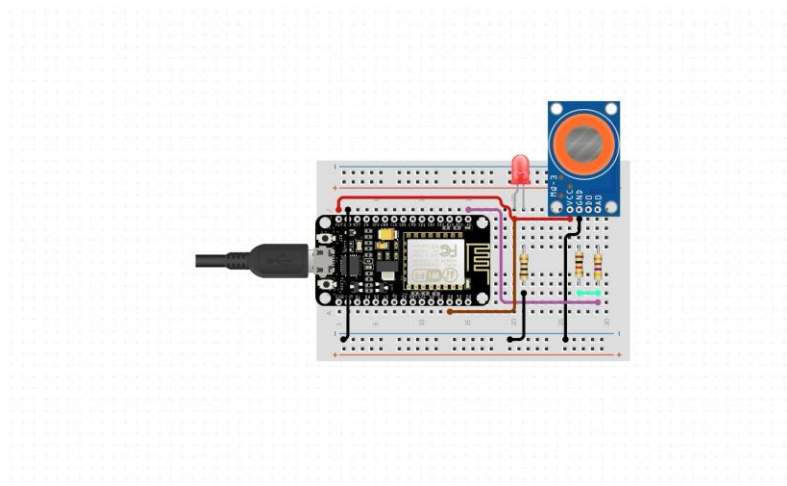
LM35 ısı sensörü,
MQ-3 alkol gaz sensörü
MQ-6 gaz sensörü
Led
Deney Zili-Buzzer
Ateş algılayıcı sensör kartı
NodeMcu

Kullanılan Teknolojiler

FireBase Bulut Platformu
MIT App Inventor
ThingSpeak

DEVRE ŞEMASI





LM35 Sıcaklık Sensörü

LM35 sıcaklık sensörü, hassas sıcaklık ölçümü yapan analog çıkışlı bir sıcaklık sensörüdür. Bu projemizde hastanın maskeyi takıp takmadığını hastanın nefes sıcaklığı ile kontrol ediyor.



Ateş Algılayıcı Sensör

Alev / Ateş Algılayıcı Dedektör 760nm - 1100 nm arasındaki dalga boyuna sahip ateşi tespit etmek için üzerinde bulunan IR alıcı bulunan bir sensör kartıdır. Bu projede ocağın açık unutulması halinde hastayı uyaran görevinde kullanılmıştır.



MQ-3 Alkol Sensörü

MQ3, MQ sensör serisinde en sık kullanılan sensörlerden biridir. Metal Oksit Yarı İletken (MOS) tipi bir sensördür. Metal oksit sensörleri aynı zamanda Chemiresistörler olarak da bilinir, çünkü algılama, alkole maruz kaldığında algılama malzemesinin direncinin değişmesine dayanır. Böylece basit bir voltaj bölücü ağına yerleştirilerek alkol konsantrasyonları tespit edilebilir.

Bu sensör ile hasta odadan çıkmadan önce ellerini dezenfekte edip etmediği kontrol edilecek



MQ-6 Gaz Sensörü

Yapılarının içerisinde gazı algılamaya duyarlı bir tel, ısıtıcı eleman ve bir yük direnci bulunmaktadır. Çalışma prensipleri genel olarak aynıdır. Isıtıcının etkisiyle ısınan metan gazı sensördeki telin üzerinden geçerek telin direncinin değişmesine etki eder. Analog direnç değerini, 0 ile 5V aralığına eş bir değere çevirmek için bir yük direnci kullanılır. Yük direnci pini, Arduino'nun analog giriş pinlerinden birine bağlanarak Arduino'ya bilgi aktarımı sağlanır.

Bu sensör doğal gaz kaçağı olup olmadığını kontrol edecek ve FireBase veritabanına değerleri anlık olarak gönderecek.



ÇALIŞMA ve KOD

NodeMCu'yu internete ve FireBase'e bağlanmak için aşağıdaki kod bloğunu yazıyoruz.

```
#include <Arduino.h>
#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"
#define WIFI_SSID "FiberHGW_ZT2AU5_2.4GHz"
#define WIFI_PASSWORD "h33hTYsADkFc"
#define API_KEY "AIzaSyCXycvlzLgRgWB8R2zYvMi7ReNyGe2iFsM"
#define DATABASE_URL "https://ates-ve-gaz-sensoru-default-rtdb.europe-west1.firebaseio.com/"
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
```

Wifi şebekesine bağlanırken ekrana yazdırılan durum mesajlarını yazan kodu aşağıdaki gibi yazalım

```
void setup() {
  Serial.begin(9600);
  pinMode(fsensor, INPUT);
  pinMode(MQ6sensor, INPUT);
  pinMode(buzzer, OUTPUT);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  config.api_key = API_KEY;
  config.database_url = DATABASE_URL;

  if (Firebase.signUp(&config, &auth, "", "")) {
    Serial.println("ok");
    signupOK = true;
  } else {
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
  }

  config.token_status_callback = tokenStatusCallback;

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}
```

Bu döngüde MQ-6 ve ateş algılayıcı sensörden değerleri okuyup FireBase veritabanına gönderiyoruz.

```
void loop() {
  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 5000 || sendDataPrevMillis == 0)) {
    sendDataPrevMillis = millis();

    int fire = digitalRead(fsensor);

    if (fire != HIGH) {
      fireDetected = true;
      Firebase.RTDB.setBool(&fbdo, "A/Flame", fireDetected);
      Serial.println("Fire Detected");
    }
    // If the MQ6 sensor value exceeds a certain threshold, activate the buzzer
    if (sensorvalue > 600) {
      tone(buzzer, 1000); // Send 1KHz sound signal...
      delay(1000);        // ...for 1 sec
      noTone(buzzer);     // Stop sound...
      delay(1000);        // ...for 1 sec
    } else {
      fireDetected = false;
      Firebase.RTDB.setBool(&fbdo, "A/Flame", fireDetected);
      Serial.println("No Fire");
    }

    // Check if MQ6 sensor value exceeds a certain threshold
    sensorvalue = analogRead(MQ6sensor);
    Serial.println(sensorvalue);
    Firebase.RTDB.setFloat(&fbdo, "A/MQ6", sensorvalue);
  }
}
```

Bu döngüde LM35 sensöründeki değeri okuyup FireBase ve ThingSpeak platformuna gönderiyoruz. Değer eşik değerini geçince led yanıyor.

```
void loop() {
  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 30000 || sendDataPrevMillis == 0)) {
    sendDataPrevMillis = millis();

    int sensorValue = analogRead(LM35sensor);
    Serial.println(sensorValue);

    // Firebase'e veri gönderme
    Firebase.RTDB.setFloat(&fbdo, "A/LM35", sensorValue);

    // Belirli bir eşik değerini aştığında LED'i yak
    if (sensorValue > sensorValueThreshold) {
      digitalWrite(ledPin, HIGH); // LED'i yak
    } else {
      digitalWrite(ledPin, LOW); // LED'i söndür
    }

    // ThingSpeak'e veri gönderme
    ThingSpeak.setField(1, sensorValue);
    int tsStatusCode = ThingSpeak.writeFields(TS_CHANNEL_ID, TS_API_KEY);

    if (tsStatusCode == 200) {
      Serial.println("ThingSpeak Data Sent Successfully!");
    } else {
      Serial.println("Error Sending Data to ThingSpeak");
    }
  }
}
```


Bu döngüde MQ-3 alkol sensörü çalışıyor. Değer belli bir değeri aşınca led yanıyor.

```
void loop() {  
  // MQ-3 sensöründen okuma yap  
  int sensorValue = analogRead(mq3Pin);  
  
  // Okunan değeri seri monitöre yazdır  
  Serial.println("Alkol Sensör Değeri: " + String(sensorValue));  
  
  // Belirli bir eşik değerini kontrol et  
  if (sensorValue > 380) {  
    digitalWrite(ledPin, HIGH); // LED'i yak  
    Serial.println("Alkol Algılandı! LED Yandı!");  
  } else {  
    digitalWrite(ledPin, LOW); // LED'i söndür  
  }  
  
  delay(1000); // 1 saniye bekleme  
}
```

MIT App Inventor

The image displays two screenshots of the MIT App Inventor web interface, showing the 'Designer' view.

Top Screenshot: This view shows the 'Designer' tab with a 'Blocks' palette on the left and a 'Viewer' area in the center. The 'Blocks' palette includes categories like Text, Lists, Dictionaries, Colors, Variables, Procedures, and Media. The 'Viewer' area displays a complex block-based code editor with several scripts. These scripts involve Firebase database operations (e.g., 'when FirebaseDB1 is changed', 'do', 'get tag', 'call FirebaseDB1 StoreValue'), notifications (e.g., 'call FirebaseDB1 ShowAlert'), and UI interactions (e.g., 'when MaskedTextBox Click', 'call FirebaseDB1 Getvalue').

Bottom Screenshot: This view shows the 'Designer' tab with a 'Palette' on the left and a 'Viewer' area in the center. The 'Palette' includes categories like User Interface, Layout, Media, and Drawing and Animation. The 'Viewer' area displays a mobile device preview showing a form with fields for 'MaskedTextBox', 'Ver AI', 'Ver AI', 'Ver AI', 'Ver AI', and 'Ver AI'. Below the preview, there are 'Non-visible components' including 'FirebaseDB1', 'Notifier1', and 'Sound1'. The 'Properties' panel on the right shows the 'Behavior' section for 'Sound1' with properties like 'MinimumInterval' and 'Source'.

KAYNAKÇA

- Küçük, Z. K. & Ekren, N. (2020). Akıllı mutfak için tasarlanmış sistemler üzerine bir derleme. International Periodical of Recent Technologies in Applied Engineering, 2(1), ss. 25- 34, 2020.
- Özdoğan, E. & Daş, R. (2021). IoT based a Smart Home Automation System Design: Simulation Case. Balkan Journal of Electrical and Computer Engineering, 9(3), pp. 297-303.
- <https://randomnerdtutorials.com/esp8266-nodemcu-firebase-realtime-database/>
- <https://arduinodestek.com/arduino-ile-alkol-gaz-sensorumq-3-kullanimi/>