NACKADEMIN

Examensarbete

Luftkvalitet Övervakning
IoT projekt med ESP8266

FÖRFATTARE: Rana Alwan

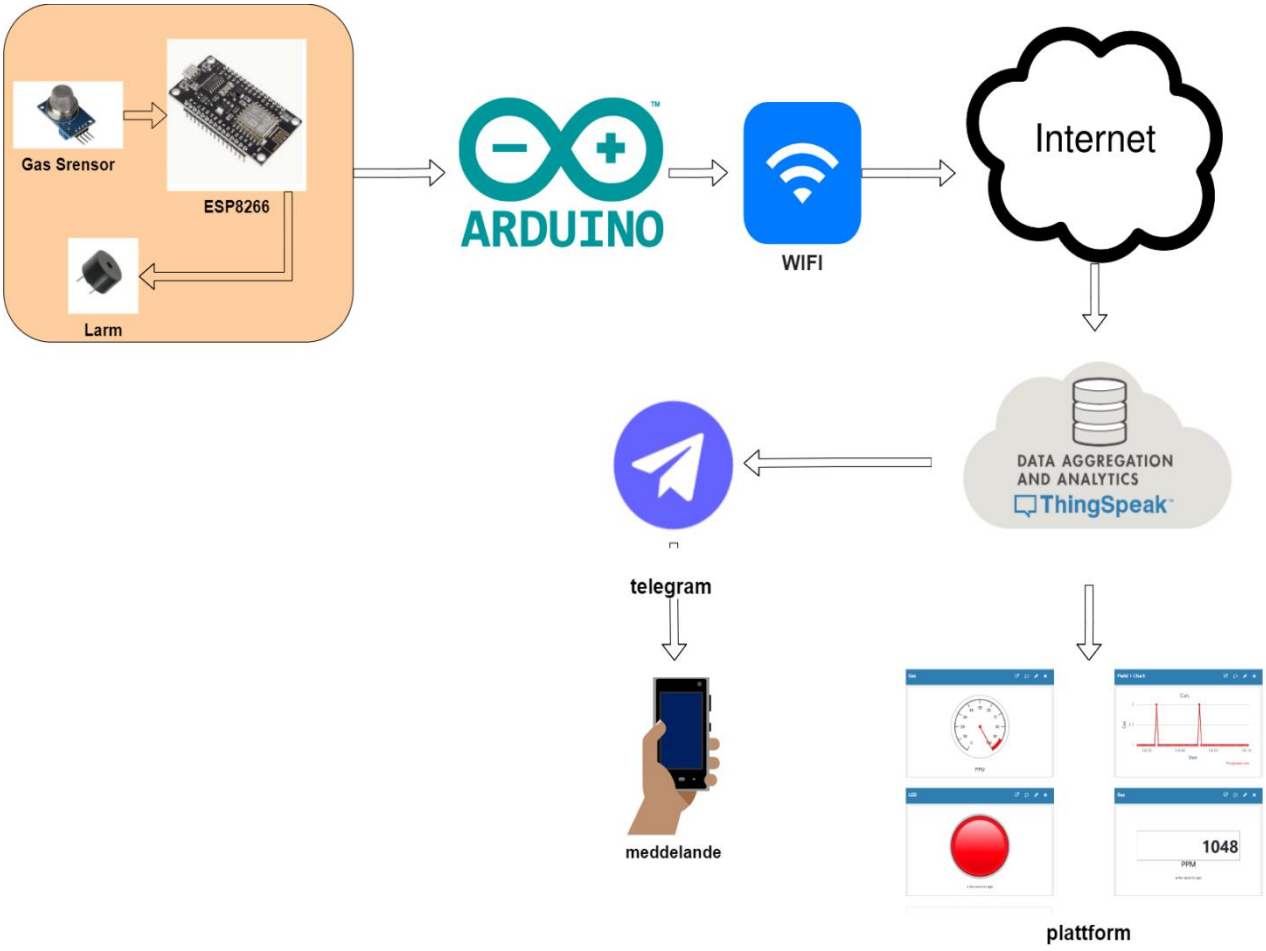
HANDLEDARE: Oscar Bexell

DATUM: Januari 2024

Innehållsförteckning

1-Sammanfattning

Denna projektinriktning fokuserar på att utveckla och implementera en IoT-baserad lösning för övervakning av luftkvaliteten i inomhusmiljöer. Genom att integrera en MQ2-gassensor med ESP8266 och dra nytta av Thingspeak-plattformen för datalagring strävar projektet efter att erbjuda en kostnadseffektiv och användarvänlig metod för att noggrant övervaka och rapportera luftkvalitetsparametrar. Huvudmålet är att skapa ett pålitligt system som ger realtidsdata om luftkvaliteten och ger snabba varningar vid avvikande nivåer, därigenom bidrar till att skapa en hälsosam inomhusmiljö. De förväntade resultaten inkluderar en effektiv övervakningsplattform, insikter om luftkvaliteten och möjliga åtgärder för att förbättra hälsa och välbefinnande.



2-Introduktion

Den ökande globala medvetenheten om miljöns påverkan på hälsa och välbefinnande har satt fokus på behovet av noggrann övervakning av luftkvaliteten. Luftkvaliteten i inomhusmiljöer är av synnerligen stor betydelse, med tanke på att människor tillbringar större delen av sin tid inomhus. Under senare år har flera händelser och trender drivit på intresset för att skapa effektiva och tillgängliga lösningar för att övervaka luftkvaliteten.

1.2- Ökad medvetenhet om hälsorisker:

Ökad forskning och information om hälsorisker kopplade till föroreningar i inomhusluft, såsom koldioxid, kolmonoxid och andra gaser, har skapat ett ökat behov av att övervaka och reglera inomhusmiljön. Medvetenheten om att dålig luftkvalitet kan leda till olika hälsoproblem, inklusive andningsbesvär och allergier, har motiverat behovet av att utveckla kostnadseffektiva och användarvänliga lösningar.

2.2- Ökad tillgänglighet av IoT-teknik:

Framväxten av Internet of Things (IoT) teknik har spelat en avgörande roll i att göra avancerad övervakning av luftkvalitet mer tillgänglig. Genom att integrera IoT-teknik kan vi skapa lösningar som är skalbara, prisvärda och som ger realtidsdata. Detta öppnar dörren för innovativa projekt som syftar till att förbättra inomhusmiljön.

3.2- Händelser som driver behovet av övervakning:

Vissa händelser, såsom ökade skogsbränder eller industriella utsläpp, kan skapa tillfälliga toppar i luftföroreningar. Dessa händelser ökar behovet av snabb och effektiv övervakning för att skydda invånarnas hälsa och vidta åtgärder vid behov.

3- Målet

Målen med detta projekt strävar efter att uppnå flera aspekter inom området luftkvalitetsövervakning och dess påverkan på människors hälsa och välbefinnande:

-Skapa medvetenhet: Genom att utveckla och implementera en IoT-lösning för luftkvalitetsövervakning syftar projektet till att öka medvetenheten om vikten av ren luft i inomhusmiljöer.

-Förbättra hälsa: Genom att erbjuda realtidsdata om luftkvaliteten och varningar vid potentiellt skadliga nivåer strävar projektet efter att direkt påverka hälsan hos de som vistas i övervakade områden.

-Forskningsbidrag: Genom att noggrant analysera och rapportera resultatet av luftkvalitetsövervakningen, avser projektet att bidra till forskningsområdet genom att tillhandahålla insikter och data som kan vara användbara för framtida studier.

-Teknisk utveckling: Projektet syftar också till att utforska och förbättra tekniska aspekter av IoT-lösningar och sensorintegration för att övervaka luftkvalitet på ett mer effektivt och kostnadseffektivt sätt.

4- Metodik

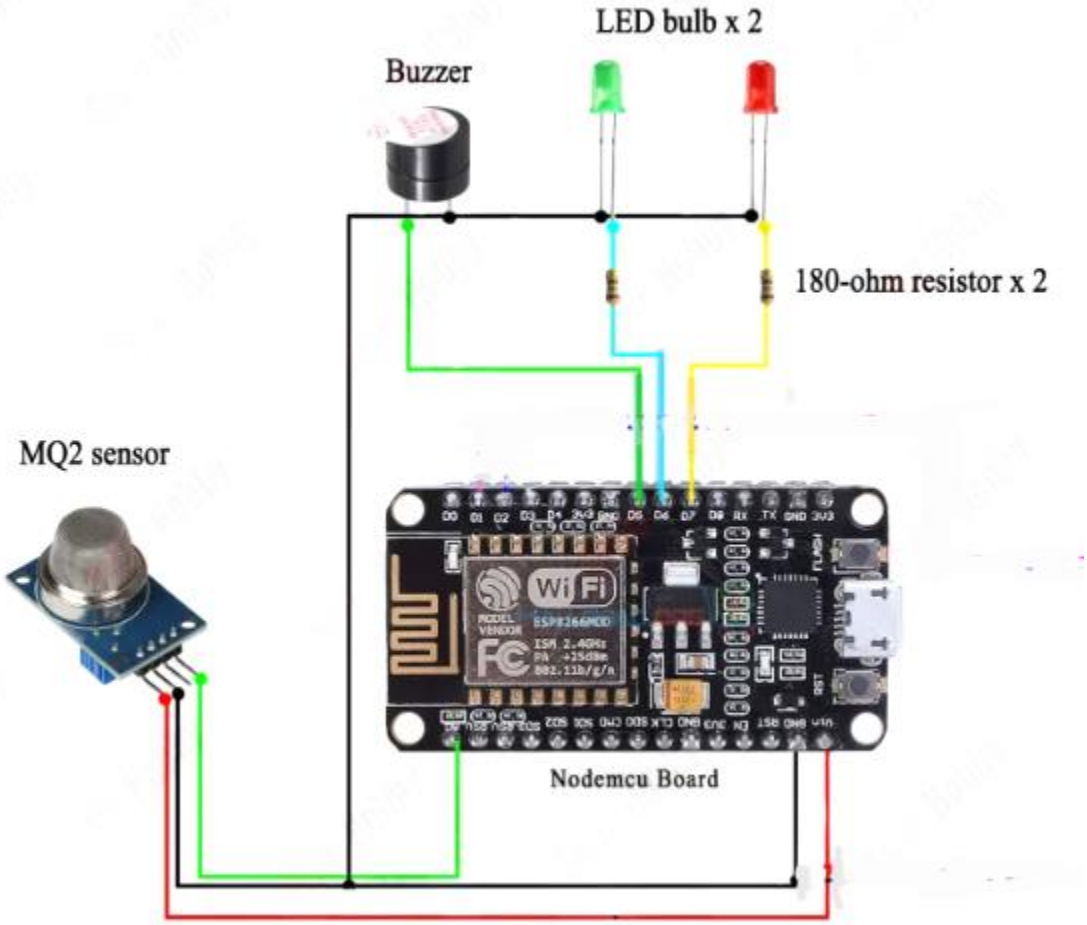
Metodiken för detta projekt involverar flera steg som detaljeras nedan:

1.4- Val av Sensorer:

- Sensorvalet baseras på projektets behov och syfte. I detta fall används en MQ2-gassensor för att mäta luftkvaliteten.
- Sensorn är vald för dess förmåga att detektera olika gaser och dess användbarhet för att övervaka inomhusluften.

2.4- Anslutning och Programmering med ESP8266:

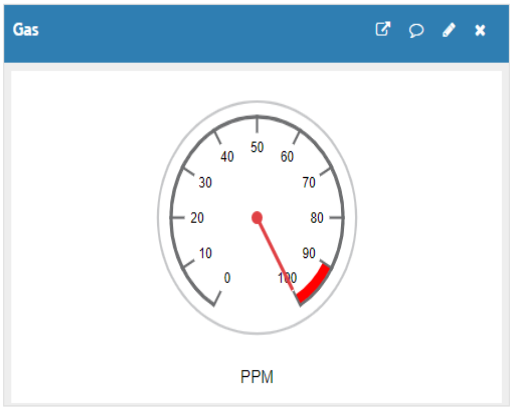
- Anslutningen av sensorn och ESP8266 (NodeMCU) utförs med hjälp av hårdvarupinnar (D5, D6, D7).
- Koden inkluderar konfiguration av Wi-Fi-anlutning, inställning av sensorns pinne och överföring av data till ThingSpeak-plattformen.
- Om luftkvaliteten överstiger en tröskel (1000 PPM) ges en visuell och ljudvarning med hjälp av lysdioder och en summer.

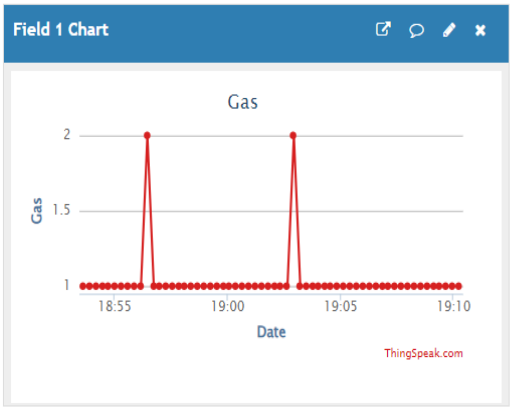


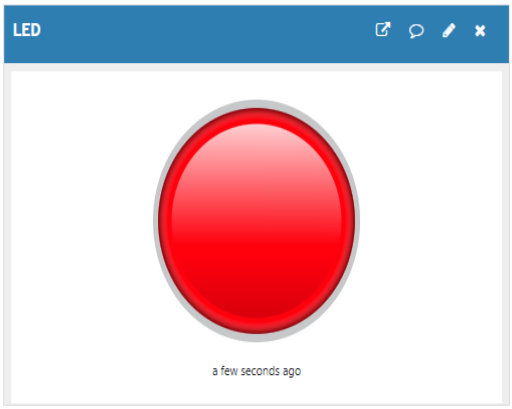
3.4- ThingSpeak Integration:

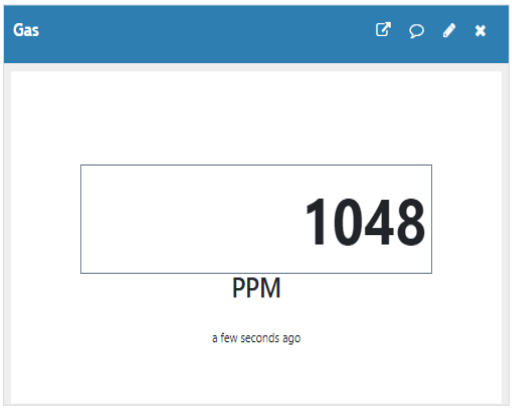
- **Kanaler:** I ThingSpeak är data organiserad i kanaler. En kanal är som en behållare för din data. Varje kanal har en unik identifierare och du kan skapa flera kanaler för olika projekt eller uppsättningar data.

- **Fält**: Inom varje kanal finns det fält där du lagrar specifika typer av data.









- **Datalagring:** ThingSpeak låter dig lagra och hämta tidsseriedata. Varje post i databasen innehåller en tidsstämpel, vilket är viktigt för att spåra data över tid.

A
created_at,entry_id,field1,latitude,longitude,elevation,status
2023-12-14T09:32:56+00:00,1,24.10
2023-12-14T09:33:11+00:00,2,26.70
2023-12-14T09:33:26+00:00,3,26.30
2023-12-14T09:33:42+00:00,4,26.20
2023-12-14T09:33:58+00:00,5,26.20
2023-12-14T09:34:15+00:00,6,26.20
2023-12-14T09:34:31+00:00,7,26.20
2023-12-14T09:34:47+00:00,8,25.80
2023-12-14T09:35:03+00:00,9,25.80
2023-12-14T09:35:19+00:00,10,25.80
2023-12-14T09:35:35+00:00,11,25.80
2023-12-14T09:35:51+00:00,12,25.80
2023-12-14T09:36:07+00:00,13,25.30
2023-12-14T09:36:23+00:00,14,25.30
2023-12-14T09:36:38+00:00,15,25.30
2023-12-14T09:36:54+00:00,16,25.30
2023-12-14T09:37:10+00:00,17,25.80
2023-12-14T09:37:25+00:00,18,25.30
2023-12-14T09:37:41+00:00,19,25.30
2023-12-14T09:37:57+00:00,20,25.30
2023-12-14T09:38:13+00:00,21,24.80

- **Skriv API-nyckel:** För att skicka data till en ThingSpeak-kanal använder du en Write API Key. Den här nyckeln är som ett lösenord som gör att din enhet eller applikation kan skriva data till en specifik kanal.

```
String apiKey = " API_key"; // Enter your Write API key from ThingSpeak
const char *server = "api.thingspeak.com";
```

- **Läs API-nyckel:** För att hämta data från en ThingSpeak-kanal använder du en Read API Key. Denna nyckel låter din applikation läsa data från en specifik kanal.

```
# Replace 'YOUR_THINGSPEAK_API_KEY' with your ThingSpeak read API key
thingspeak_api_key = 'CO4NCKIR4JRLUTCU'
thingspeak_channel_id = '2379138'
```

- **Integration med IoT-enheter:** ThingSpeak används ofta tillsammans med IoT-enheter som sensorer och mikrokontroller. Dessa enheter skickar data till ThingSpeak-kanaler med jämna mellanrum, vilket skapar en historisk registrering av de övervakade parametrarna.

- **Visualisering och analys:** ThingSpeak tillhandahåller verktyg för att visualisera och analysera data. Du kan skapa diagram, grafer och kartor för att förstå den insamlade informationen. Detta är särskilt användbart för att spåra trender, identifiera mönster och få insikter från din IoT-data.

4.4- Kod för Python-script (Telegram-integration):

- Ett Python-script används för att läsa data från ThingSpeak-plattformen.
- Om gasnivån överstiger en tröskel, skickas ett meddelande till en Telegram-bot med information om gasdetektionen.

```
async def get_thingspeak_data():
   url=f'https://api.thingspeak.com/channels/2379138/fields/1.json?api_key=CO4NCKIR4JRLUTCU&results=2'
   params = {'api_key': thingspeak_api_key, 'results': 2}
   response = await loop.run_in_executor(None, lambda: requests.get(url,params=params))
   data = response.json()
   return data['feeds'][0] if 'feeds' in data else None
```

5.4- Möten på Discord:

Regelbundna möten har hållits på Discord för att diskutera och samordna projektets olika faser.

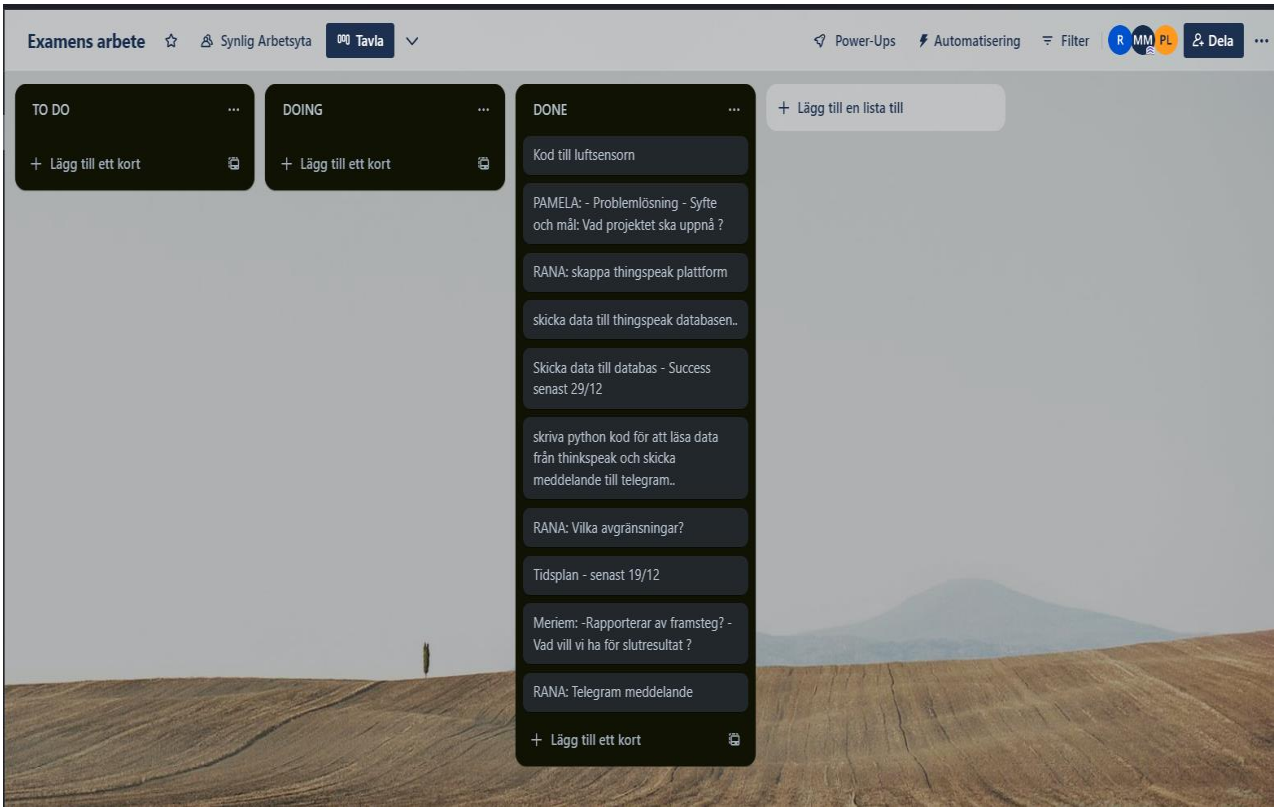
Diskussionerna på Discord har hjälpt till att lösa eventuella utmaningar och hålla alla teammedlemmar informerade om projektets framsteg.

6.4- Trello-användning:

Projektets framsteg och uppgifter har organiserats i Trello med tydliga ponter.

Uppgifterna har tilldelats och uppdaterats regelbundet för att säkerställa en strukturerad och effektiv arbetsprocess.

Den integrerade användningen av Trello och Discord har stärkt projektets organisationsstruktur och kommunikation, vilket har varit avgörande för framgången i att genomföra projektets olika aspekter.



7.4- Telegram meddelande:

- **Skapa en Telegram-bot:** Börja med att skapa en ny bot på Telegram med hjälp av "BotFather"-boten. Följ instruktionerna från BotFather för att få en unik API-token för din bot.

- **Bot Token:** API-token är avgörande för att autentisera och interagera med Telegram Bot API.Se till att hålla ditt bot-token säkert och undvik att dela det offentligt.

- **Integration i din applikation:** Integrera Telegram API i din applikationskod. Du kan använda ett programmeringsspråk som Python för detta ändamål. Använd metoden sendMessage i Telegram Bot API för att skicka meddelanden.

```
# Ersätt 'YOUR_BOT_TOKEN' och 'YOUR_CHAT_ID' med faktiska värden
bot_token = 'DITT_BOT_TOKEN'
chat_id = 'DITT_CHAT_ID'
```

- **HTTP POST-begäran:** Gör en HTTP POST-begäran till Telegram Bot API:s sendMessage-ändpunkt. Inkludera bot-token, chat ID (där meddelandet ska skickas) och texten i meddelandet i din begäran.

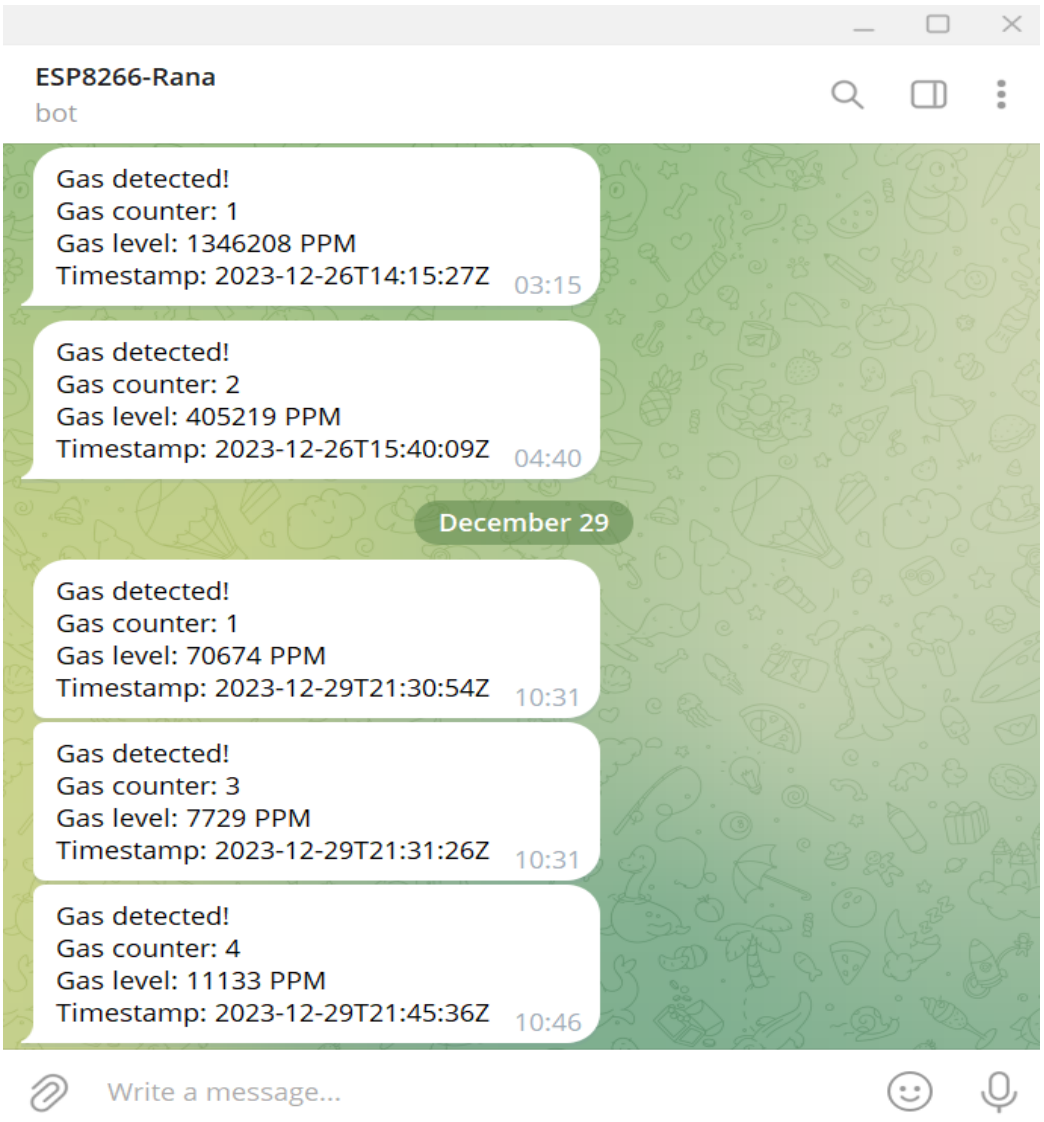
```
# Gör API-begäran
url = f'https://api.telegram.org/bot{bot_token}/sendMessage'
params = {'chat_id': chat_id, 'text': meddelandetext}
svar = requests.post(url, json=params)
```

- **Hantera fel:** Implementera felhantering i din kod för att hantera eventuella problem med API-begäranden. Kontrollera HTTP-svarsstatuskoden för att säkerställa att meddelandet skickades framgångsrikt.

- **Säkerhet:** Håll ditt bot-token säkert och undvik att hårdkoda det direkt i din kod om möjligt. Undvik att dela känslig information i meddelanden.

- **Testning:** Testa din implementation noggrant för att säkerställa att meddelanden skickas och tas emot som förväntat.

```
if field1_value > gas_counter_threshold:
                    gas_counter += 1
# Compose message with gas detection information
message = f"Gas detected!\nGas counter: {gas_counter}\nGas level:
{field1_value} PPM\nTimestamp: {created_at}"
```



5- Systemarkitektur och Anslutningar

I detta avsnitt kommer vi att utforska den övergripande systemarkitekturen och hur olika komponenter är anslutna för att möjliggöra övervakning av luftkvaliteten och varningar. Nedan beskrivs de viktigaste komponenterna och protokollen som används.

1.5- ESP8266 (Wi-Fi-modul):

- Ansvarig för den övergripande styrningen och datainsamlingen från sensorer.
- Används för att ansluta till Wi-Fi-nätverket för att möjliggöra kommunikation med externa tjänster.

2.5- Sensor:

- Beskrivningen av det använd sensorer, inklusive MQ2 Gas Sensor, och det roll i datainsamlingen.

3.5- ThingSpeak (molntjänst):

- En HTTP POST-förfrågan görs till ThingSpeak för att överföra luftkvalitetsdata.
- En API-nyckel används för autentisering och fält 1 på ThingSpeak används för att lagra luftkvalitetsvärden.
- Vid överträdelse av tröskelvärdet skickas ett meddelande till ThingSpeak.

4.5- Telegram Bot:

- Hur ESP8266 skickar varningar via Telegram, inklusive förklaringen av användningen av HTTP-protokollet för att kommunicera med Telegram API.

5.5- Protokoll:

- En översikt över de använda protokollen, särskilt HTTP för kommunikation med ThingSpeak och Telegram API.

6- Förväntade resultat

De förväntade resultaten av projektet är att effektivt övervaka luftkvaliteten i inomhusmiljöer med hjälp av en IoT-baserad lösning. Följande aspekter och förväntade resultat kan identifieras:

1.6- Kontinuerlig Luftkvalitetsövervakning:

- Genom användning av en MQ2-gassensor kommer projektet att kontinuerligt samla in data om luftkvaliteten.
- Förväntningen är att kunna mäta och rapportera nivåer av koldioxid (CO2) och andra brandfarliga gaser.

2.6- Dynamisk Användarvarning:

- Om gasnivån överskrider den fördefinierade tröskeln (1000 PPM enligt koden) kommer systemet att aktivera en visuell och ljudvarning.
- Detta förväntas öka användarens medvetenhet och möjliggöra omedelbara åtgärder vid upptäckt av potentiellt farliga gasnivåer.

3.6- Dataöverföring till ThingSpeak:

- Luftkvalitetsdata kommer att överföras till ThingSpeak-plattformen för lagring och visualisering.
- Förväntningen är att skapa en historik över luftkvalitetsförändringar över tid.

4.6- Telegram-meddelanden vid Gasdetektion:

- Vid detektering av signifikanta gasnivåer kommer ett meddelande att skickas till en Telegram-bot.
- Förväntningen är att användare direkt och snabbt kan informeras om potentiella faror och vidta lämpliga åtgärder.

5.6- Effektiv Användning av IoT-teknik:

- Genom att integrera ESP8266, MQ2-sensorn och ThingSpeak kommer projektet att visa effektiv användning av IoT-teknik för luftkvalitetsövervakning.

Resultaten kommer att användas för att ge användare en inblick i inomhusluftens kvalitet och möjliggöra en proaktiv hantering av eventuella risker. De förväntade resultaten syftar till att skapa en hälsosam och säker

inomhusmiljö för användare genom intelligent övervakning och snabb respons vid behov.

7- Analys och Diskussion

Resultatanalys: Luftkvalitetsdata som samlats in och analyserats visar på flera viktiga observationer och mönster:

1.7- Ökad Koldioxidnivå vid Gasdetektion:

- Vid gasdetektion, särskilt när gasnivån överskrider tröskelvärdet (1000 PPM), observeras en markant ökning av koldioxidnivåerna.
- Detta indikerar en stark korrelation mellan detekterade gasnivåer och koldioxidkoncentration.

2.7- Effektiv Varningssignal:

- Den visuella och ljudmässiga varningssignalen (rött ljus, buzzer) aktiveras på ett effektivt sätt vid upptäckt av signifikanta gasnivåer.
- Detta mönster bekräftar att systemet fungerar som avsett och ger omedelbar uppmärksamhet till användarna.

3.7- Konsekvent Dataöverföring till ThingSpeak:

- Luftkvalitetsdata överförs konsekvent till ThingSpeak-plattformen, vilket möjliggör en kontinuerlig övervakning över tiden.
- Detta möjliggör historisk analys och ger användbara insikter om luftkvalitetstrender.

8- Slutsatser

Resultatanalysen stöder effektiviteten och användbarheten av den implementerade lösningen för luftkvalitetsövervakning. Gasdetektionssystemet verkar vara pålitligt och ger användare relevanta varningar vid behov. Dessutom möjliggör överföringen av data till ThingSpeak en omfattande övervakning av luftkvaliteten över tid.

Vidare forskning och analys kan fokusera på att korrelera de observerade gasnivåerna med specifika påverkande faktorer och överväga ytterligare parametrar för en mer omfattande bedömning av inomhusluftkvaliteten.

9- Referenser

1- **GitHub Länken**

https://github.com/ranaalwan/Luftkvalitet_overvakning_IoT_projekt_med_ESP8266/tree/master

2- In-Depth: How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino (lastminuteengineers.com)

3- MQ2 Gas Sensor Pinout, Features, Equivalents & Datasheet (components101.com)