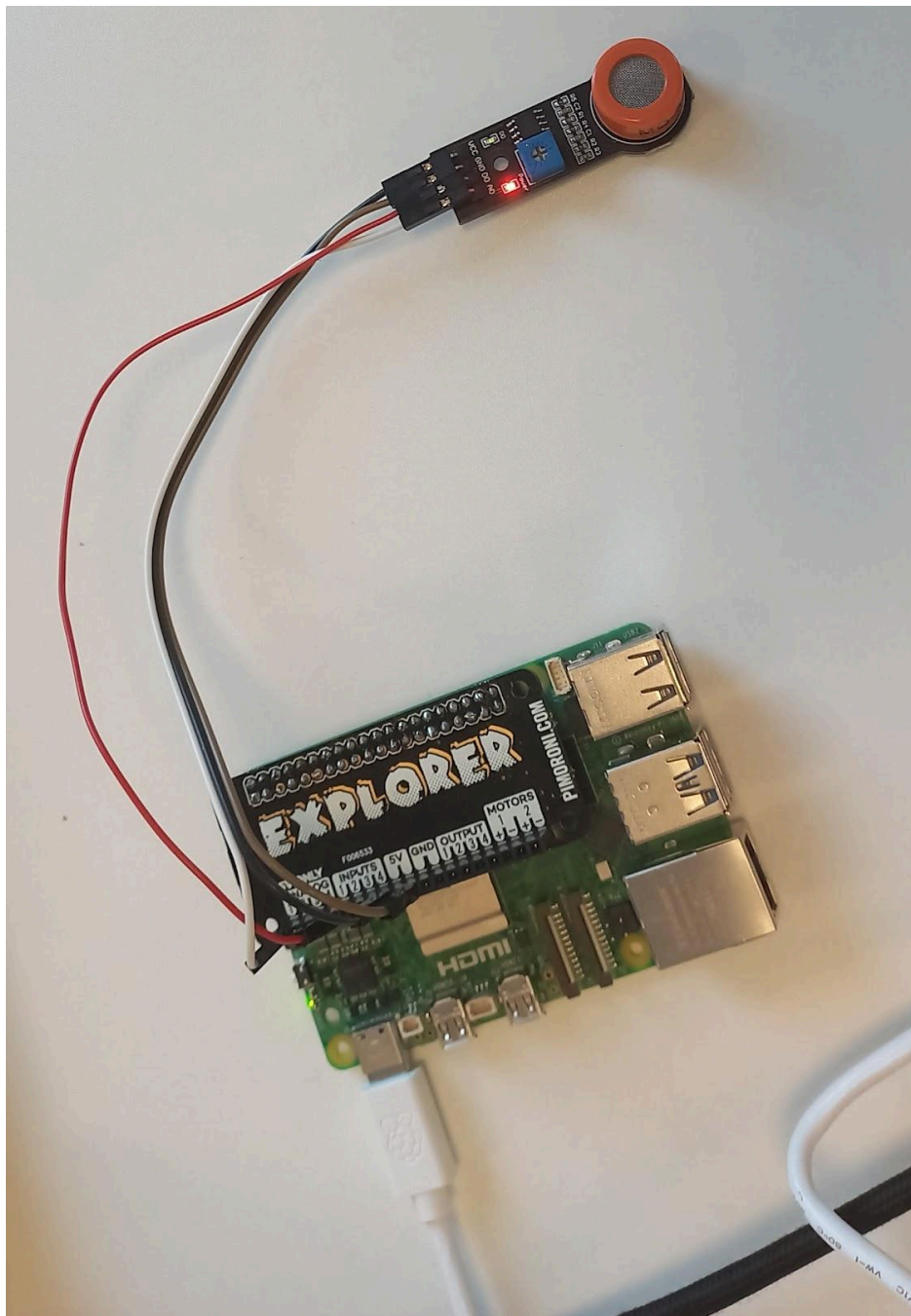


Promille partner

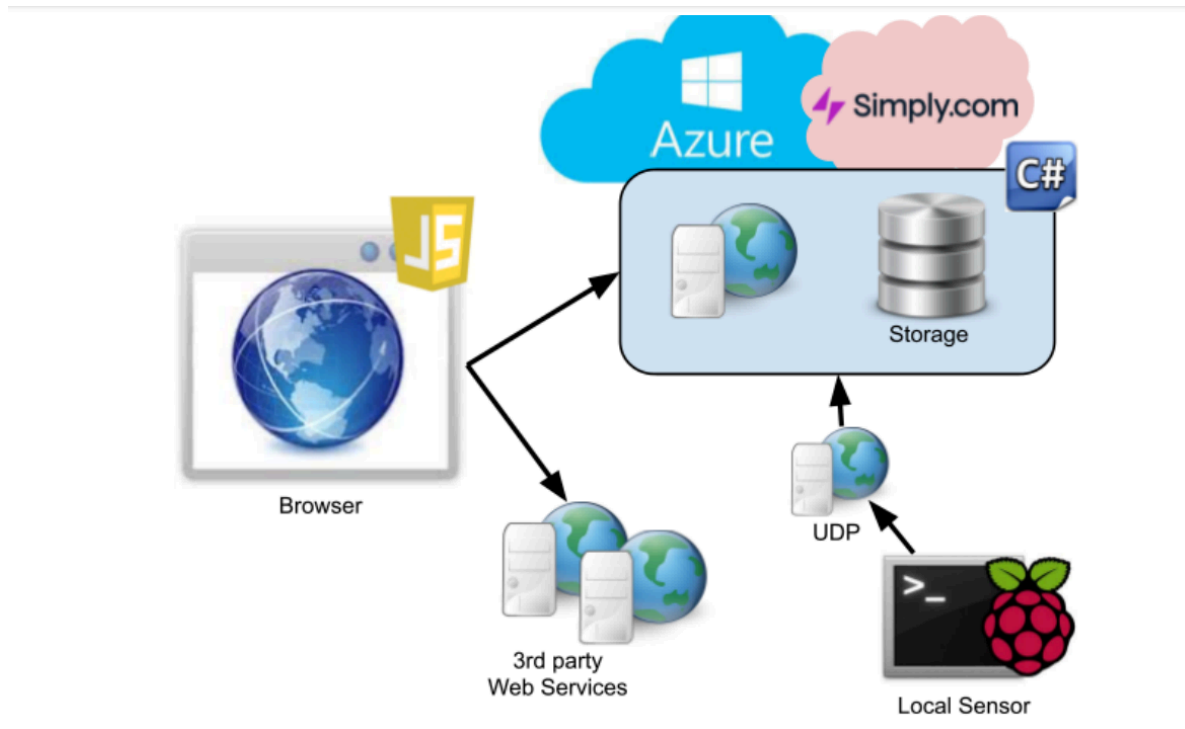


Link til github med al den nødvendige kode - <https://github.com/MathiasNoergaard/PromillePartnerSamlet>

Indledning

Promille partner er en hjemmeside hvor du kan få genereret en personlig drikkeplan for f.eks. studenterture så du kan få en hyggelig tur hvor du kan huske hvad der skete. Du skal bare indtaste nogle få oplysninger, vælge hvad du vil drikke og voila en drikkeplan. Derudover er der en med følgende alkometer så du kan se om du på lovlignvis kan køre hjem selv.

Vores produkt arkitektur



Selve hosting

- Webinterface hvor druk planen genereres hostet fra Azure
- REST API som interface til database, også her vores egen validering af data sker før det bliver lagt over til databasen, der er hostet fra Azure
- Database som REST API kommunikerer med hostet fra Simply
- ~~En proxy server som bindeled mellem REST API og Raspberry pi~~
- En Raspberry pi til at måle promille og afgive visuelle notifikationer til brugeren

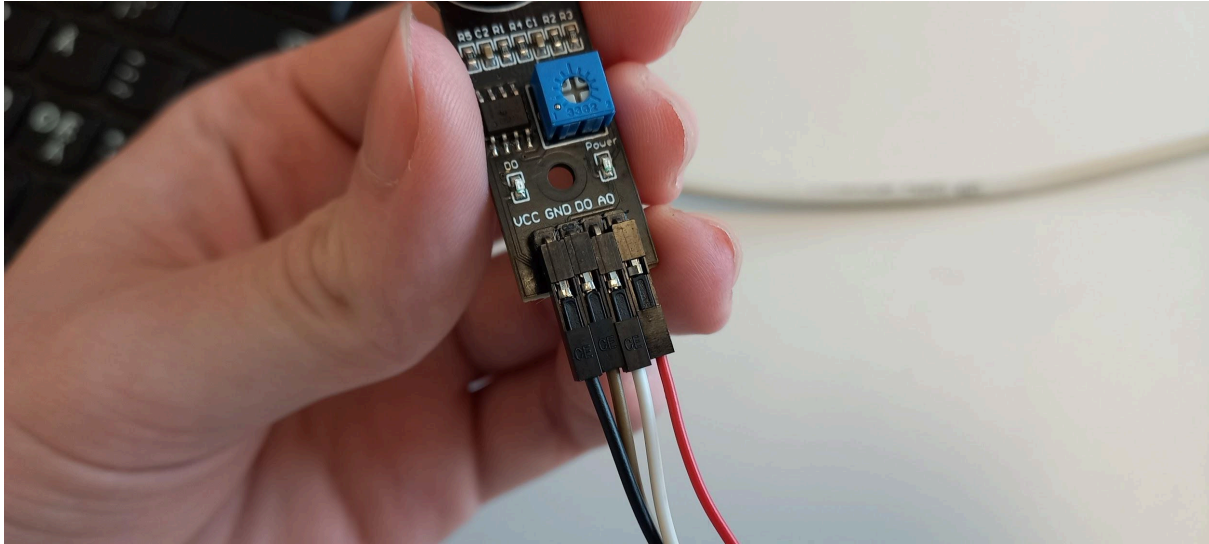
Ingrediens liste

Raspberry pi

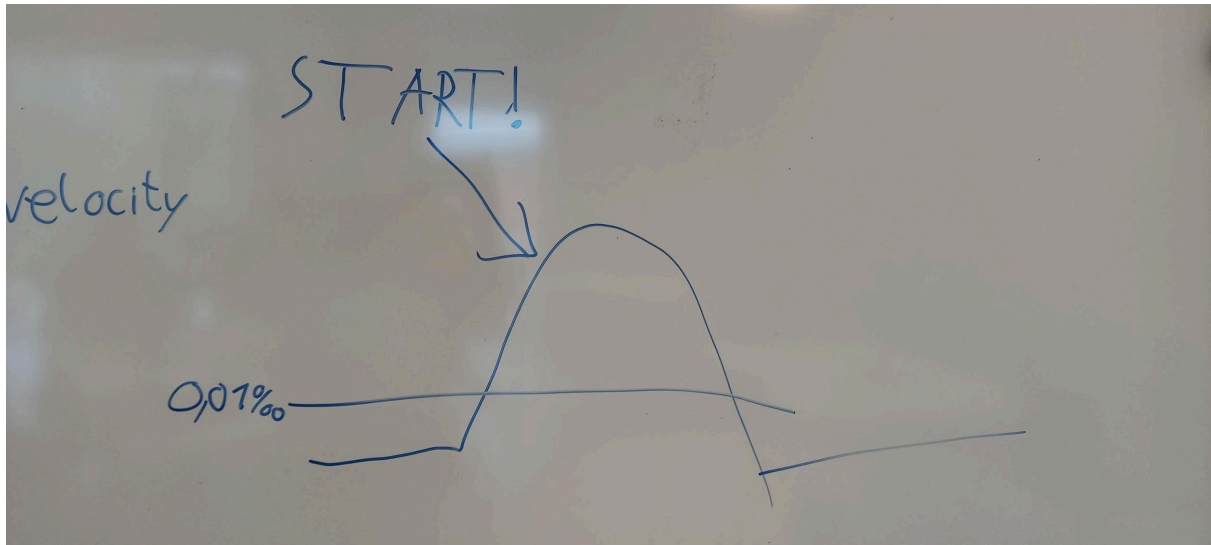
- Raspberry pi 5
- 4 han-hun ledninger
- MQ-3 sensor
- Pimoroni Explorer PHAT

Sådan skal du forbinde raspberry pi

1. Tilslut Pimoroni Explorer pHAT til GPIO-headeren på Raspberry Pi.
2. Forbind MQ3-gassensoren til Explorer pHAT via breadboard:
 - a. VCC fra MQ3 til 5V på Explorer pHAT.
 - b. GND fra MQ3 til GND på Explorer pHAT.
 - c. AO fra MQ3 til en analog indgang (A0) på Explorer pHAT.
 - d. DO fra MQ3 til digital indgang (Input 0) på Explorer pHAT.



Hvordan fungere det



Alkometeret måler altid med noget baggrundsstøj, der kan stige alt efter alkometerets temperatur. Så når den måle en promille over en vis grænse så begynder den at optage målingerne og når den målte promille går under grænsen igen så bliver den højeste promille målt sendt til databasen.

Python kode til raspberry pi

Software

- Raspberry Pi OS
- Python 3
- Libraries: gpiozero, explorerhat, time. Til visualisering kan matplotlib bruges

```
import explorerhat as eh
import time
import requests
import time

# Calibration constants (update with actual calibration data)
VOLTAGE_TO_BRAC_MULTIPLIER = 50 # Placeholder multiplier
BASELINE_VOLTAGE = 0.26 # Sensor baseline voltage in clean air (adjust as needed)

def calibrate_voltage_to_brac(voltage):
    """
    Convert voltage to Breath Alcohol Concentration (BrAC) in mg/L.
    Adjusts for baseline voltage.
    """
    adjusted_voltage = max(voltage - BASELINE_VOLTAGE, 0) # Remove baseline
    brac = adjusted_voltage * VOLTAGE_TO_BRAC_MULTIPLIER
    return brac

def voltage_to_bac(brac):
    """
    Convert Breath Alcohol Concentration (BrAC) in mg/L to BAC (‰ promille).
    """
    bac = (brac * 2100 / 1e6) * 10 # Convert to promille (‰)
    return bac

def analog_to_voltage(analog_value):
    """
    Convert analog value to voltage (5V reference).
    """
    return (analog_value * 5.0) / 1023
```

Følgende Python-script illustrerer, hvordan man indsamler data fra MQ3-sensoren og konverterer dette til brugbare aflæsninger:

```

def read_sensor():
    """continuously read from the MQ3 sensor, calculate BrAC and BAC, and display results"""
    try:
        cutOffLevel = 0.5
        maxRecorded = None
        api_url = "https://promillepartnerbackend.azurewebsites.net/api/piReading"
        timeNow = None
        while True:
            # Read analog value from ADC0 (connected to MQ3 OUT)
            analog_value = eh.analog.one.read()
            voltage = analog_to_voltage(analog_value)
            brac = calibrate_voltage_to_brac(voltage)
            bac = voltage_to_bac(brac)
            print(f"Analog Value: {analog_value:.6f} | BrAC: {brac:.2f} mg/L | Estimated BAC: {bac:.4f} ‰ promille")
            if(analog_value > cutOffLevel):
                if(maxRecorded == None):
                    maxRecorded = analog_value
                if(maxRecorded < analog_value):
                    maxRecorded = analog_value

            if(analog_value < cutOffLevel and maxRecorded != None):
                #send to database'
                maxRecorded = maxRecorded/3*500
                print(maxRecorded)
                timeNow = time.time()

                print(timeNow)
                data = {
                    'promille': maxRecorded,
                    'timestampMilliseconds': int(timeNow*1000)
                }
                response = requests.post(api_url, json=data)
                print(response)
                print(response.status_code)
                print(response.text)
                if(response != None):
                    time.sleep(60)
                    maxRecorded = None

            time.sleep(2)
    except KeyboardInterrupt:
        print("Exiting.")

if __name__ == "__main__":
    read_sensor()

```