

## Bilag 1: Liste over vigtige MQTT-pakker

- CONNECT: Sendes når en klient anmoder om tilladelse til at forbinde til brokeren.
- CONNACK: Brokeren skal sende dette tilbage om den har godkendt klienten.
- SUBSCRIBE: Sendes til broker når en klient vil abonnere på et bestemt emne.
- SUBACK: Brokeren kan sende tilbage at den har modtaget abonnement besked.
- UNSUBSCRIBE: Sendes til broker hvis en klient vil fjerne et abonnement.
- UNSUBACK: Brokeren kan sende dette tilbage omkring modtagelse af abonnement fjernelse.
- PUBLISH: Sendes fra klient til broker og derefter fra broker til modtager klient
- PUBACK: Acknowledgement som kan sendes tilbage fra en PUBLISH besked
- DISCONNECT: Sendes til broker hvis klienten vil fjerne forbindelse til brokeren.
- PINGREQ: Sendes til broker for at holde styr på at forbindelsen til klienten stadig er til stede
- PINGRESP: Broker sender tilbage at den har modtaget et PINGREQ fra klienten.
- Andre: Findes flere, men kommer jeg ikke til at arbejde med i mit system.

## Bilag 2: QoS

MQTT-protokollen indeholder også en funktion kaldet Quality of Service. Dette er et flag på PUBLISH pakken der indikerer hvordan besked-proceduren skal forløbe og repræsenteres af en integer værdi.

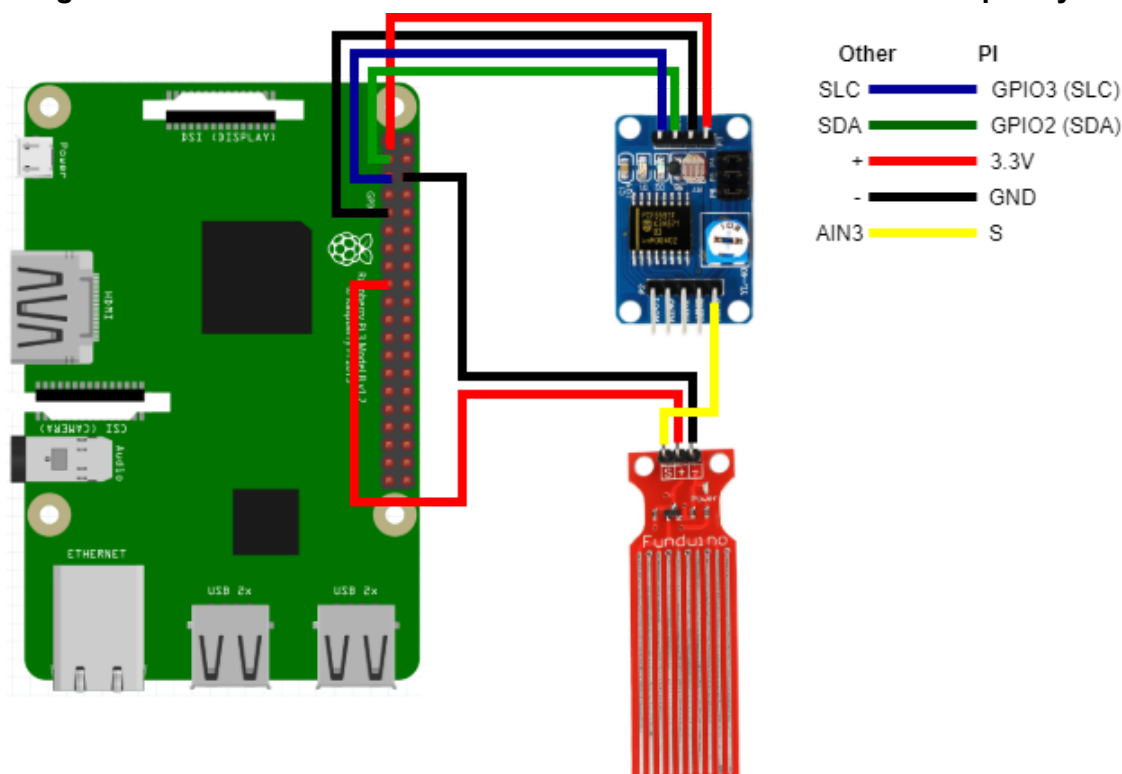
Der findes 3 muligheder for QoS:

- QoS = 0: Kaldet "fire and forget" hvor beskeden sendes en gang og bliver ikke tjekket hvorvidt den er blevet modtaget af enten broker eller modtager klient.
- QoS = 1: Beskeden bliver sendt mindst en gang indtil senderen modtager et acknowledgement fra brokeren.
- QoS = 2: Beskeden bliver sendt nøjagtig én gang hvor sender klient og modtager klient indgår i et to-vejs håndtryk for at sikre at modtageren har fået præcis en kopi af beskeden.

Disse funktioner kan være nyttige i forskellige sammenhænge afhængig af systemet. F.eks. I sammenhæng med MQTT-modellen på billede 1, kunne det være en god ting at forsikre sig at config\_change publiceringen kommer frem til alle sensorer og man kunne derfor indføre QoS 1 eller 2 for ekstra forsikring.

Jeg har valgt ikke at fokusere på QoS da jeg følte at tidsmæssigt ville jeg ikke kunne nå at implementere de forskellige QoS håndteringstyper. Det skal dog nævnes at lave en standard broker hvor man ikke tager hensyn til QoS vil automatisk falde ind i QoS = 0 da alle beskeder bare bliver sendt en enkelt gang uden nogen form for validering.

### Bilag 3: Større billede af Schematics for Funduino Water Sensor + Raspberry Pi + YL-40



### Bilag 4: Database + REST API

For at kunne involvere noget praktisk brug af broderen valgte jeg at lave en MySQL database + Restful API for at kunne trække data fra sensorerne ud og opbevare dem. Dette har jeg gjort ved at gemme information omkring sensoren; navnet på sensoren, typen (wifi eller zigbee i mit tilfælde) samt id. Dette id har jeg så brugt som foreign key i en anden tabel der holder alle målingerne.

Sensor	
PK	ID, INT, NOT NULL, AUTO_INCREMENT
	Name, NVARCHAR(100), NOT NULL
	Type, NVARCHAR(100), NOT NULL

Measurement	
PK	ID, INT, NOT NULL, AUTO_INCREMENT
FK	SensorId, INT, NOT NULL
	Date, DATETIME, NOT NULL, DEFAULT, NOW()
	Moisture, FLOAT, NULL
	Temperature, FLOAT, NULL
	Pressure, FLOAT, NULL

Denne tabel indeholder informationer omkring hvilken sensor den tilhører via SensorId. En dato for hvornår målingen blev foretaget og gemt i databasen, samt de tilhørende værdier for målingerne. Afhængig af sensoren kan der kun måles fugt eller ved andre kan der måles både fugt, temperatur og tryk.

Dataen bliver opsamlet af en simpel MQTT-klient der abonnerer på alle sensorer indenfor wifi eller zigbee-netværket med struktur: "wifi/sensor/#" og "zigbee2mqtt/#".

## Bilag 5: Hjemmeside

Hjemmesiden jeg har valgt at lave inkluderer en simpel brugergrænseflade hvorpå man kan se alle sensorerne der er gemt i databasen. Hvis siden opdateres vil der blive hentet de seneste værdier ud af databasen og blive vist under hver sensors navn. Som det kan ses har jeg flere forskellige typer af sensorer på netværket.

- Konke Sensor: Zigbee sensor der kan måle fugt og temperatur
- Xiaomi Agara: Zigbee sensor der kan måle fugt, temperatur og tryk
- Plant Sensor 1: Funduino Water sensor der kun måler fugt.
- SenseHat Sensor: Raspberry Pi + SenseHat der måler fugt, temperatur og tryk.

