**MQTT** of Message Queueing Telemetry Transport

Message Queueing Telemetry Transport (MQTT) is een machine-tot-machine (M2M) data transfer protocol dat in een hoog tempo het leidende messaging-protocol voor het Industriële Internet der Dingen (IoT) aan het worden is.

Hoewel er momenteel een aantal concurrerende IoT technologieën en protocollen in het spel zijn, zijn de extreem lichtgewicht overhead (2-byte header), Publish/subscribe model en de bidirectionele mogelijkheden van MQTT bij uitstek geschikt om te voldoen aan de eisen van industriële besturingssystemen.

## Waarom is MQTT Gemaakt?

MQTT is gemaakt met het doel om data te verzamelen van zoveel mogelijk apparaten en om deze data vervolgens te  
transporteren naar een IT infrastructuur. Het is lichtgewicht en dus ideaal voor het monitoren op afstand, vooral bij M2M verbindingen waar een kleine footprint code vereist is of waar de netwerkbrandbreedte beperkt is.

Message Queueing Telemetry Transport werd in 1999 uitgevonden door Dr. Andy Stanford-Clark en Arlen Nipper.

**Wat zijn de voordelen van MQTT?**

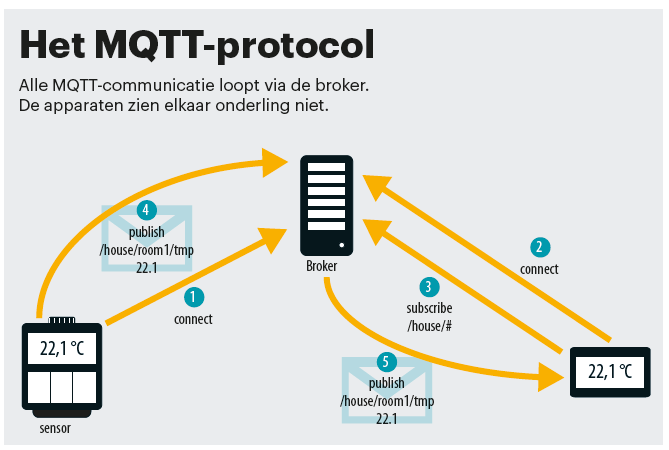
Het MQTT protocol maakt het mogelijk voor jouw systeem om toegang te  
krijgen tot IoT data. MQTT brengt vele krachtige voordelen naar uw proces:

* Distribueer informatie efficiënter
* Verhoog de schaalbaarheid
* Verminder drastisch het verbruik van netwerkbrandbreedte
* Verminder update snelheden naar secondes
* Zeer geschikt voor detectie en controle op afstand
* Maximaliseer de beschikbare bandbreedte
* Extreem lichtgewicht overhead
* Zeer veilig met permissie gebaseerde veiligheid
* Gebruikt door de olie en gas industrie, Amazon, Facebook…
* Bespaart ontwikkelingstijd
* Publish/subscribe protocollen verzamelen meer data met minder bandbreedte ten opzichte van polling protocollen.

## De BROKER staat centraal bij MQTT

Als een sensor, bijvoorbeeld een microcontroller met een temperatuursensor, wil communiceren via MQTT, moet hij eerst verbinding maken met de broker. Bij het MQTT-protocol is poort 1883 gereserveerd voor onversleutelde en poort 8883 voor versleutelde communicatie. MQTT is, anders dan bijvoorbeeld HTTP, een statusbehoudend protocol. Een verbinding kan dus ook blijven bestaan als er geen gegevens worden verzonden.

**Hoe werkt MQTT?**



**CLIENT**

**BROKER**

**PUBLISH**

**PAYLOAD**

**TOPIC**

**SUBSCRIBE**

**/**

**# alle berichten op de lagere niveaus**

**+ alle berichten op 1 niveau**

**Retain**

**QoS ( Quality of service )**

**0 geen bevestiging/check aankomst**

**1 bericht komt minstens 1 keer aan**

**2 bericht komt exact 1 keer aan**

**Het principe is zeer eenvoudig:**

PUBLISH data (PAYLOAD) met een bepaald onderwerp ( TOPIC) naar de server ( BROKER ).

Vervolgens zal de BROKER deze data forwarden naar alle clients die zijn ingeschreven ( SUBCRIBE ) voor dit onderwerp ( TOPIC).

Een client kan SUBSCRIB’en voor meerdere TOPIC’s en PUBLISH’en naar meerdere TOPIC’s.

Hoe kunnen we dit simuleren en testen?

BROKER >> op <http://www.mqtt-dashboard.com/>

CLIENT >> via <http://www.hivemq.com/demos/websocket-client/> ( poort is hier 8000 ipv 8883 omdat het een websocket verbinding betreft)

STAP 1 : Connect de client met de broker ( knop CONNECT)

STAP 2 : PUBLISH berichten met de juist topic

STAP 3: SUBCRIBE voor berichten die je wil volgen

Er bestaan natuurlijk ook programma’s voor PC, Smart Device etc.

Voor PC >> MQTTFX download [MQTT.fx Version 1.7.1](http://www.jensd.de/apps/mqttfx/1.7.1) -windows  op <http://www.jensd.de/apps/mqttfx/1.7.1/>

Pas Broker adres aan indien nodig ( broker.mqttdashboard.com en de poort 1883 )

Voor SMART DEVICE:

Android >> MQTT Dashboard <http://allbestapps.net/android/iot-mqtt-dashboard>

IoS >> MQTTool <https://apps.apple.com/us/app/mqttool/id1085976398> ( of zoek naar MQTT Terminal )

MIT-APP INV

Voor Chrome

MQTT lens >> <https://chrome.google.com/webstore/detail/mqttlens/hemojaaeigabkbcookmlgmdigohjobjm?hl=nl>

OEF ESP32+RP + Browser :

Ontwerp een MQTT system waarmee je een led en/of beeper bij elke cursist kan laten beepen.

Elke cursist kan ook jouw led en beeper activeren.

De leraar kan alle berichten volgen…

Topic bv CVOFOCUS/VOORNAAMACHTERNAAM/RP/BEEP met payload 10/50/250

zal de beeper verbonden met RP 10 keer laten beepen met ontime = 50 en offtime = 250 ms

Topic bv CVOFOCUS/VOORNAAMACHTERNAAM/ESP/LED met paytload 10/50/250

zal de led verbonden met ESP32 10 keer laten blinken met ontime = 50 en offtime = 250 ms

Opmerking je kan alle berichten volgen in de browser ( eventueel 2-de window met extra connectie)