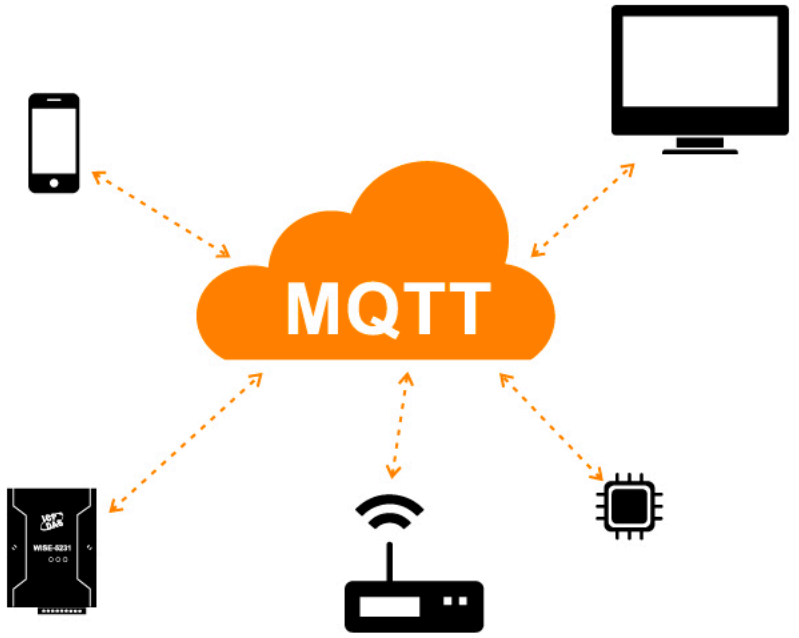
Internet of Things

MQTT

Message Queuing Telemetry Transport

Overview

- One of the most commonly used protocols in IoT projects
- Designed as a lightweight messaging protocol that uses publish/subscribe operations to exchange data between clients and the server



Why MQTT ?

- It’s a lightweight protocol. So, it’s easy to implement in software and fast in data transmission.
- It’s based on a messaging technique.
- Minimized data packets. Hence, low network usage.
- Low power usage. As a result, it saves the connected device’s battery.
- It’s real time! That’s is specifically what makes it perfect for IoT applications.

Why not HTTP

- HTTP is slower
  - because it uses bigger data packets to communicate with the server
- HTTP requires more overhead
  - HTTP request opens and closes the connection at each request, while MQTT stays online to make the channel always open between the broker “server” and clients
- HTTP consumes more power
  - since it takes a longer time and more data packets, therefore it uses much power

MQTT Components

- Broker
  - the server that handles the data transmission between the clients
- Topic
  - the place a device want to put or retrieve a message to/from
- Message
  - the data that a device receives “when subscribing” from a topic or send “when publishing” to a topic
- Publish
  - the process a device does to send its message to the broker
- Subscribe
  - where a device does to retrieve a message from the broker

How MQTT works

- It uses client-server implementation
- Server is called as broker and clients simply connect to the server
- Operations
  - Publish
    - Is performed to send a message to the broker
    - Any client send publish a message with required contents
    - A message is published for a required topic (subject)
  - Subscribe
    - Is performed to receive a message from the broker
    - Anyone interested in the messages can subscribe to a topic
    - Messages will be delivered to all the subscribed clients

Example

- There is Temperature sensor connected to NodeMCU wants to send its readings
- On the other side, a phone/desktop application wants to receive this temperature value
- Therefore, 2 things will happen
  - The device defines the topic it wants to publish on (”temperature”)
  - The phone/desktop application subscribes to the topic “temperature”. Then, it receives the message that the device has published, which is the temperature value.

Brokers libraries

- Mosquitto
- ThingMQ
- ThingStudio
- MQTT.io
- Heroku
- CloudMQTT