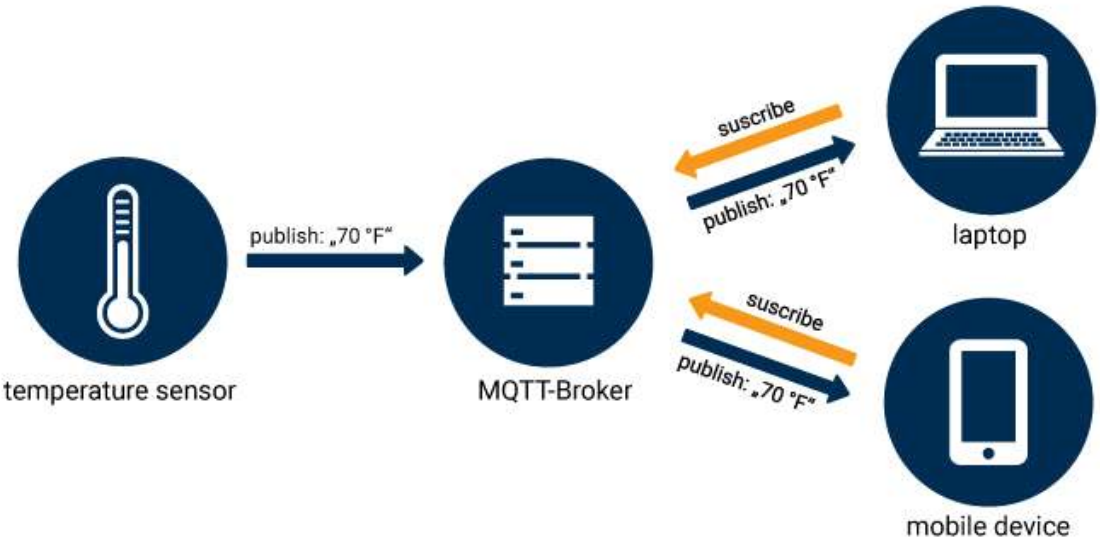
[MQTT : Message Queuing Telemetry Transport]



Message Queuing Telemetry Transport : 메시지 큐잉 텔레메트리 트랜스포트

- ISO 표준 발행-구독 방식 경량 메시징 프로토콜
- 사물통신, 사물인터넷과 같이 대역폭이 제한된 통신 환경에 최적화하여 개발된 푸시 기술 기반의 경량 메시지 전송 프로토콜
- TCP/IP 프로토콜 위에서 사용 <- TCP/IP 프로토콜 스택에서 작동
- 모바일 기기나 낮은 대역폭의 소형 디바이스들에 최적화
- 느리고 품질이 낮은 네트워크에서도 메시지를 안정적으로 전송할 수 있도록 설계됨
- 저전력 설계
- 가장 작은 메시지는 2byte까지 가능
- 세 가지의 QoS 레벨 제공

MQTT 자체는 메시지를 어떻게 보낼 것인지를 정의하는 규약일 뿐이다. 따라서 실제 “MQTT”를 동작시키기 위해서는 서버 역할을 해주는 장치와 프로그램이 필요한데 이를 MQTT 브로커(Broker)라고 한다. MQTT 브로커는 각종 장치들(MQTT Client)이 보내주는 메시지를 수집하고 이걸 다시 필요한 장치들에게 전송해주는 중계서버 역할을 한다.

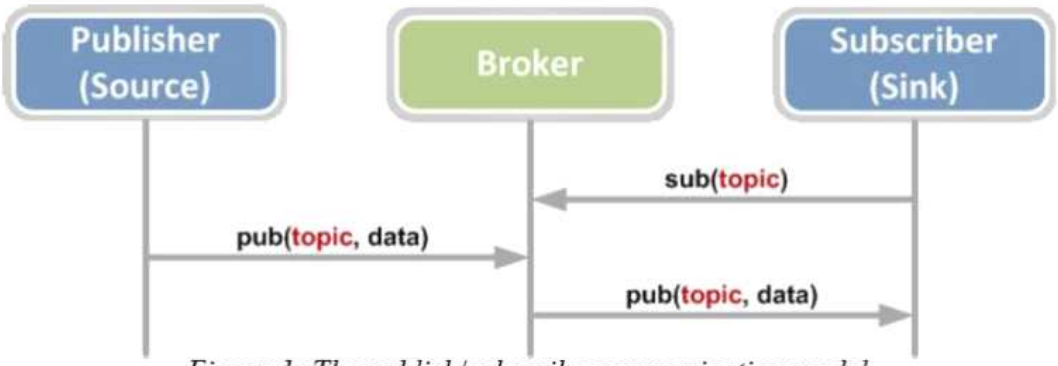
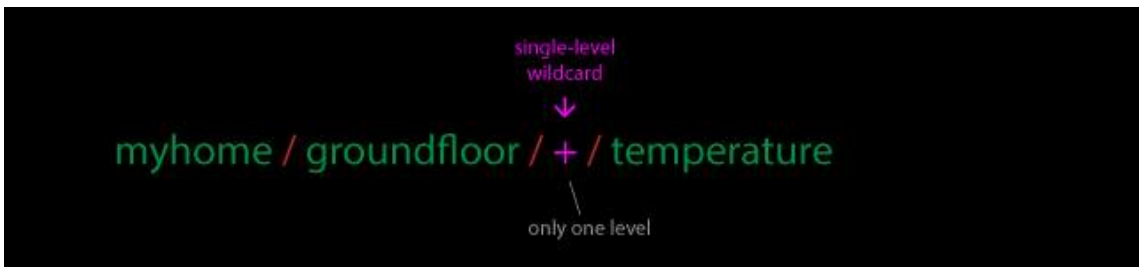


Figure 1: The publish/subscribe communication model

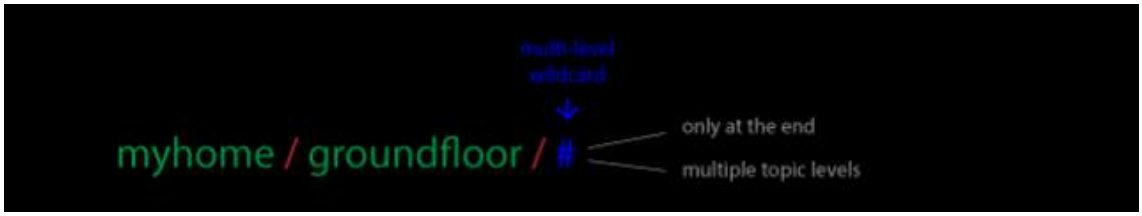
MQTT 시스템에 참여하는 MQTT 클라이언트는 메시지 발행(publish, 트윗에 해당), 메시지 구독(subscribe, follow에 해당) 두 가지 동작을 할 수 있다. MQTT 클라이언트가 메시지를 특정 채널(Topic, 토픽)에 발행하면 이 채널을 구독한 모든 클라이언트에게 메시지가 전달된다. 중간에서 메시지를 수집, 재분해 하는 작업은 MQTT 브로커가 담당한다.

토픽 (Topic)

메시지를 발행/구독(pub/sub) 하는 행위는 채널 단위로 일어난다.
이를 MQTT에서는 토픽(Topic)이라고 한다.
토픽은 슬래시(/)로 구분된 계층구조를 갖는다.



메시지를 구독/발행 할 때 여러개의 토픽을 한번에 지정할 수 있도록 wild card 문자를 지원한다. [+] 문자는 단 1개의(1 레벨) 토픽을 임의의 토픽으로 대체하는 와일드카드 문자이다.



반면 [#] 문자는 여러 레벨의 토픽을 대체할 수 있다. [myhome/#] 처럼 사용하면 myhome 의 하위 토픽들 모두를 지칭하게 된다. [#] 문자는 토픽 문자열 끝에만 사용할 수 있으며 하위 토픽 모두를 지칭한다. (2단계 이상 하위 토픽도 포함)

[$] 문자로 시작하는 토픽은 시스템에 의해 사용되는 특수한 토픽을 의미한다. 이 토픽들은 [#] 문자열로 지정해도 포함되지 않는 토픽이 됩니다. 주로 [$SYS/] 로 시작하는 토픽들이 브로커의 내부 메시지를 위해 사용된다. (이에 대한 명확한 표준은 없는 상태인 듯)

QoS (Quality of Service)

MQTT는 시스템에 참여하는 장치들의 처리 능력, 네트워크 대역폭, 메시지 오버헤드 등 주변상황에 맞게 시스템이 동작할 수 있도록 3단계 QoS (Quality of Service) 를 제공한다.

0 : 메시지는 한번만 전달하며, 전달여부를 확인하지 않는다. Fire and Forget 타입이다.
1 : 메시지는 반드시 한번 이상 전달된다. 하지만 메시지의 핸드셰이킹 과정을 엄밀하게 추적하지 않기 때문에, 중복 전송될 수도 있다.
2 : 메시지는 한번만 전달된다. 메시지의 핸드셰이킹 과정을 추적한다. 높은 품질을 보장하지만 성능의 희생이 따른다.

0에 가까울수록 메시지 처리에 대한 부하가 적은 대신 메시지 손실 위험이 높아집니다. 2에 가까울수록 메시지 손실 위험은 줄어들지만 메시지 처리 부하가 급격히 늘어납니다.

(보통 0~1 정도의 QoS를 사용하면서 메시지 손실 등의 위험은 상위 어플리케이션 차원에서 관리하도록 하는 듯.)

Arduino IDE ESP TEST 예제
1. HIVE MQ 브로커 사용예제
2. 모스키토 브로커 사용예제