

---

## 안드로이드 간 비콘, MQTT 서버 통신 보고서

2021.05.17

---

작성자 조은희  
cho108@krri.re.kr

## 목 차

1. 실험 보고	3
1-1. 실험 환경	3
1-2. 실험 과정	4
1-3. 실험 결과	6

## 1. 실험 보고

### 1-1. 실험환경

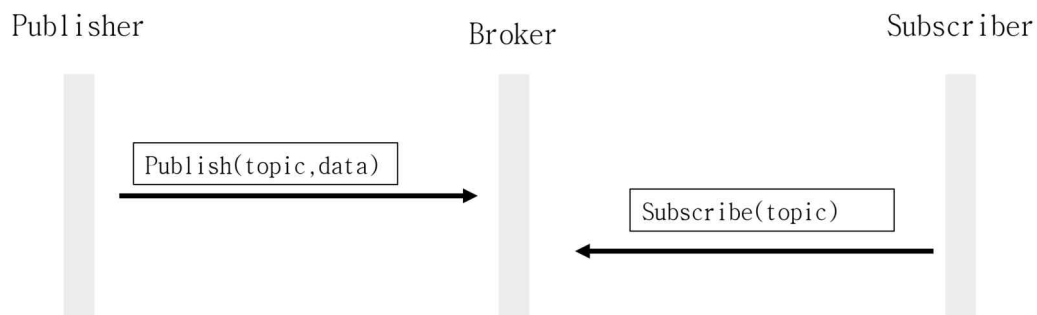
분류	종류	비고
스마트폰	삼성 갤럭시 S21+	안드로이드 버전: 11 (SDK 버전: 30)
스마트폰	삼성 갤럭시 S8	안드로이드 버전: 9.0 (SDK 버전: 28)
최소 sdk 버전	21	
타겟 sdk 버전	29	
MQTT 서버	MQTTBox 프로그램	서버 주소: 138.68.10.8

[표1 - 실험환경]

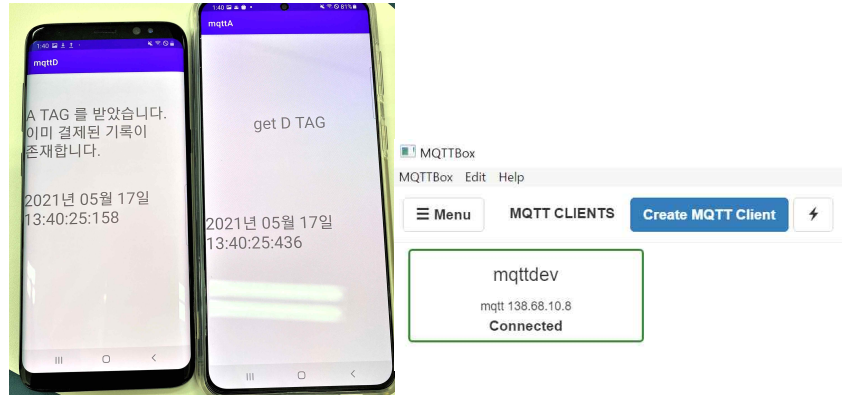
비콘의 송수신 통신에서 MQTT 서버를 추가하여 비콘에 대한 응답을 서버를 통해 수신하게 하는 실험이다. 안드로이드와 아두이노 어플리케이션은 기존과 같으나, MQTT 서버로 송신하고 수신하는 코드가 각각 추가되었다. MQTT 서버에 연결되어야 하기 때문에 두 스마트폰은 모두 데이터 통신이 가능한 상태여야 한다.

어플리케이션은 두 가지로 구현하였고 한 개는 지불 요금 정보를 전송하는 아두이노 역할의 어플리케이션, 다른 하나는 스마트폰 요금 결제 역할의 어플리케이션을 담당한다.

MQTT 서버란 퍼블리셔, 구독자, 브로커로 이루어져 있는데 퍼블리셔가 정해진 주제에 대한 데이터를 발행하고, 구독자는 이 주제를 구독하여 데이터를 받을 수 있도록 브로커가 연결시켜주는 서버이다. 단일 주제(topic)에 여러 구독자가 구독할 수 있기 때문에 1:N 통신 구축에도 매우 용이하다.

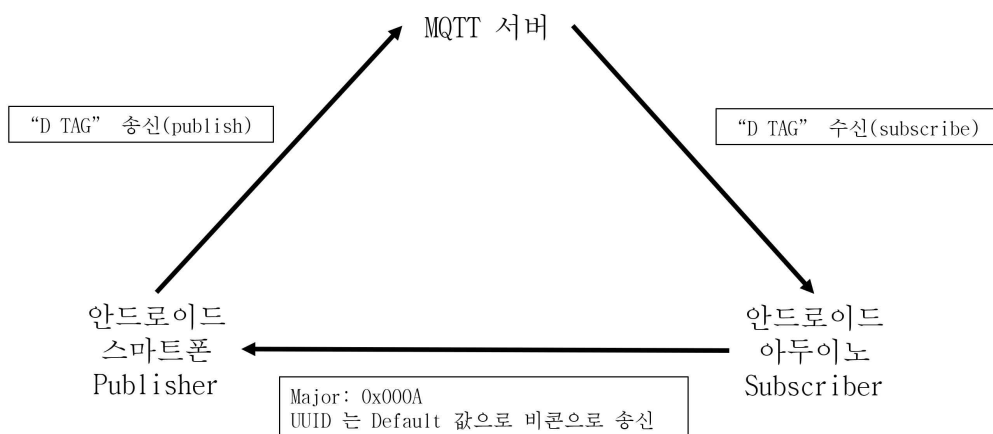


[그림 1 - MQTT 서버 구성도]



[그림 2 - 실제 구성한 안드로이드 어플리케이션과 MQTT서버]

## 1-2. 실험과정



[그림 3 - MQTT 서버를 사용한 구성도]

본 실험에서는 이미 요금이 결제되었다는 것을 가정하였다. MQTT 서버에서 퍼블리셔 역할은 아두이노 어플리케이션이, 구독자 역할은 스마트폰 어플리케이션이 담당한다.

아두이노 어플리케이션은 A 태그를 붙여 지불 가격에 대한 정보를 비콘으로 공중에 전송한다. 공중전송된 A 태그를 스캔한 스마트폰 어플리케이션은 비콘 스캔을 중지하고 “D TAG” 문자열을 MQTT 서버에 전송한다.

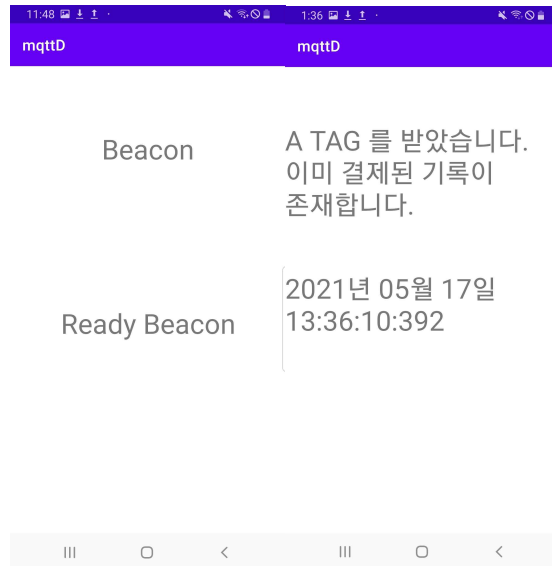
```
try{
    mqttAndroidClient.publish(TOPIC, "D TAG".getBytes(), qos: 0 , retained: false );
    Log.d(TAG2, msg: "*****mqtt send*****");
}
```

[그림 4 - D TAG를 MQTT 서버에 전송하는 코드]

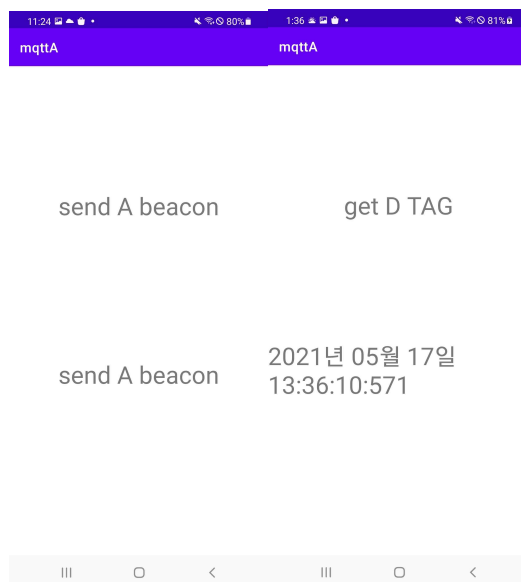
“D TAG”는 MQTT 서버로부터 아두이노 어플리케이션이 수신한다. 아두이노 어플리케이션은 수신받은 데이터에 “D TAG”가 있는지 확인하고 현재 시간을 출력한다.

```
try{
    mqttAndroidClient.subscribe(TOPIC, qos: 0);
}
```

[그림 5 - MQTT 서버로부터 데이터를 수신하는 코드]



[그림 4 - 안드로이드(기본 화면 / A TAG 비콘을 수신한 화면)]



[그림 5 - 아두이노(기본 화면 / MQTT 서버로부터 D TAG를 받은 화면)]

[그림 4]와 [그림 5]의 시간을 비교해보면 각각 10.392 초와 10.571 초로 차이는 0.179초이다. A 태그를 인식한 순간부터 D 태그를 인식하기까지 최대 약 0.2초 정도의 시간이 걸리는 것을 알 수 있다.

#### 1-4. 실험 결과

스마트 발판	스마트폰 단말기	통신 방법	스캔 시간
아두이노	스마트폰 어플리케이션	비콘 송수신 모드 연속변경	약 12 초
아두이노	스마트폰 어플리케이션	블루투스 모듈 2개 사용 비콘 송수신	약 3초
스마트폰 어플리케이션	스마트폰 어플리케이션	비콘 송수신	약 1초
스마트폰 어플리케이션	스마트폰 어플리케이션	비콘 송수신+MQTT	약 0.2초

기존의 안드로이드 비콘 송수신 통신의 스캔 시간 간격은 약 1초였지만 MQTT서버를 사용하여 스캔 시간 간격을 0.2초 수준으로 상당히 감소시켰다. publish 코드를 단 한번만 실행시켜도 서버를 통해 잃어버리는 패킷 없이 받을 수 있어 상당히 안정적이다. 하지만 MQTT서버를 사용할 경우 안드로이드가 모두 데이터 통신이 가능한 상태여야 하는 단점이 있다. 또한 본 실험에서 스마트 발판 역할로 스마트폰 어플리케이션을 사용하였는데, 이를 아두이노로 변경하게 될 경우 스캔 시간 변동이 얼마나 있을지 실험하여 알아볼 필요가 있다.