

AITCシニア技術者勉強会 —から始めるIoT 5周年 第3回 インターネット連携編

2021年1月22日

先端IT活用推進コンソーシアム
シニア技術者勉強会
近藤 繁延

講師紹介

●名前：

近藤 繁延 (Shigenobu. Kondo)

●SNS：

Twitter：[のぶ \(@mininobu\)](https://twitter.com/mininobu)

Facebook：[近藤 繁延 \(shigenobu.kondo\)](https://www.facebook.com/shigenobu.kondo)



●会社：

ウルシシステムズ株式会社



●好きなセンサー：

温湿度センサー、加速度センサー

勉強会スケジュール

1. 2020/11/20 Raspberry Piのセットアップしよう
2. 2020/12/18 Raspberry Piでセンサーを操作する
3. **2021/ 1/22 Raspberry Piでインターネット連携**
4. 2021/ 2下 Raspberry Piで画像認識
5. 2021/ 3下 アイデアソン&ハッカソン
6. 2021/ 4下 ハッカソン&成果発表会

第3回のゴール

- Raspberry PiとMQTTプロトコルを用いたデータ連携方法を習得する。
 1. 通信環境構築
 2. 基本的な使い方
 1. コマンドで利用
 2. スマホアプリで利用
 3. センサーとの連携
 4. インターネット上のサーバと連携

1. MQTTとは
 1. 全体像
 2. MQTT Brokerの種類
2. MQTT Broker「Mosquitto」とは
3. Mosquittoのセットアップ
4. 実践MQTT
 1. RaspberryPi編
 2. スマホ編
 3. センサー編
 4. サーバ連携編

参考

第2回の資料（センサーの取り扱い）

http://cloud.aitc.jp/20201218_RaspberryPi2/

MQTTとは

MQ Telemetry Transport(Message Queuing Telemetry Transport、略称MQTT)は、メッセージ指向ミドルウェアのアプリケーション層で使用される、TCP/IPによるPub/Sub型データ配信モデルの軽量なメッセージキュープロトコルである。

非力なデバイスやネットワークが不安定な場所でも動作しやすい様にメッセージ通信電文が軽量に設計されている事が特徴。

Pub/Sub型メッセージング・パターンには、メッセージブローカーが必要である。ブローカーは、メッセージのTopicに基づいて、それを必要としているクライアントにメッセージ配信をしている。

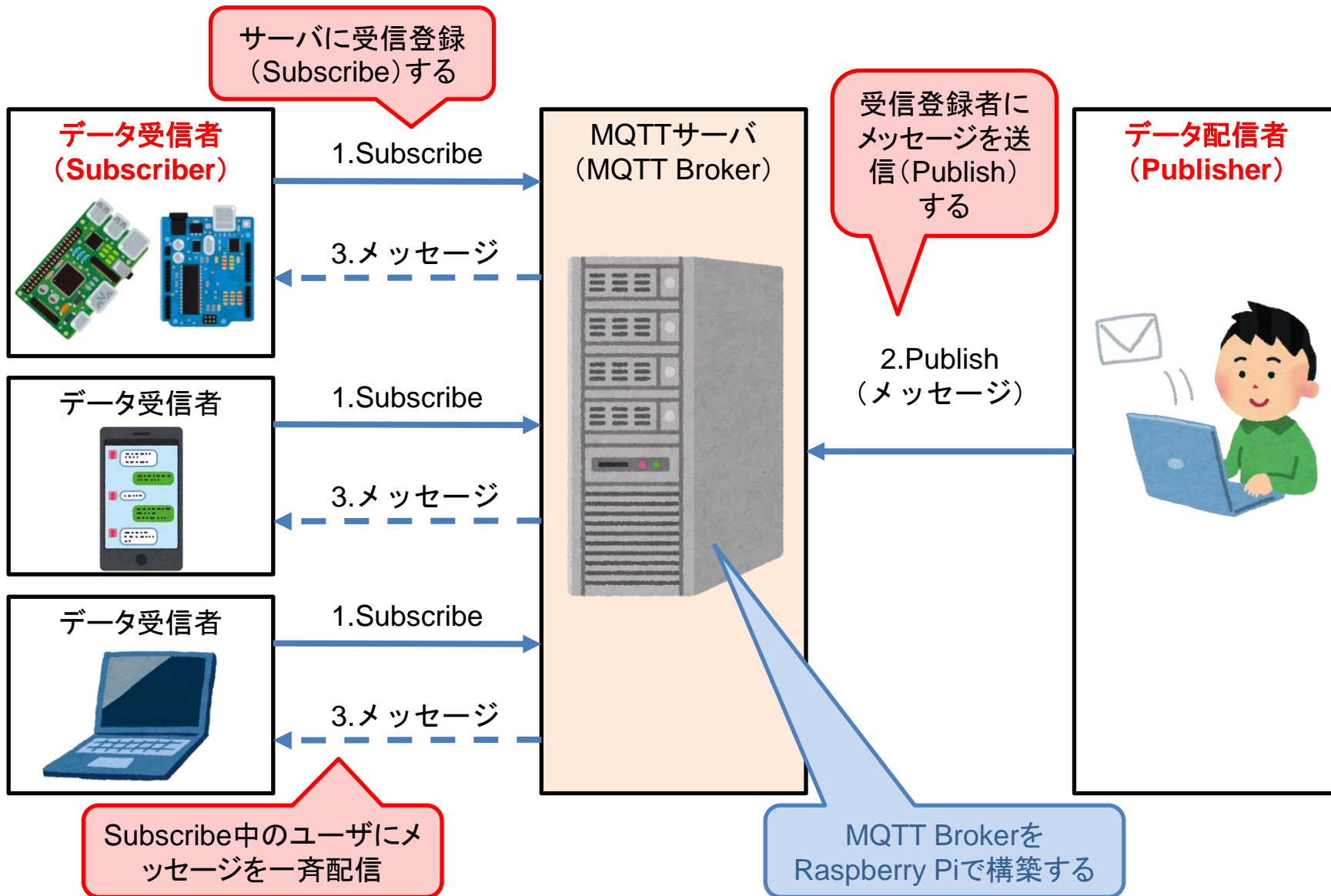
アンディー・スタンフォード・クラーク(英語版)とシーラスリンクソリューションのアーレンニッパは1999年に、プロトコルの最初のバージョンを執筆している。

wikipediaより

端的に言うと・・・

**IoT機器のような低スペック機器でも利用できる
データ連携用のプロトコル**です。

MQTTの全体像



MQTT Brokerの種類

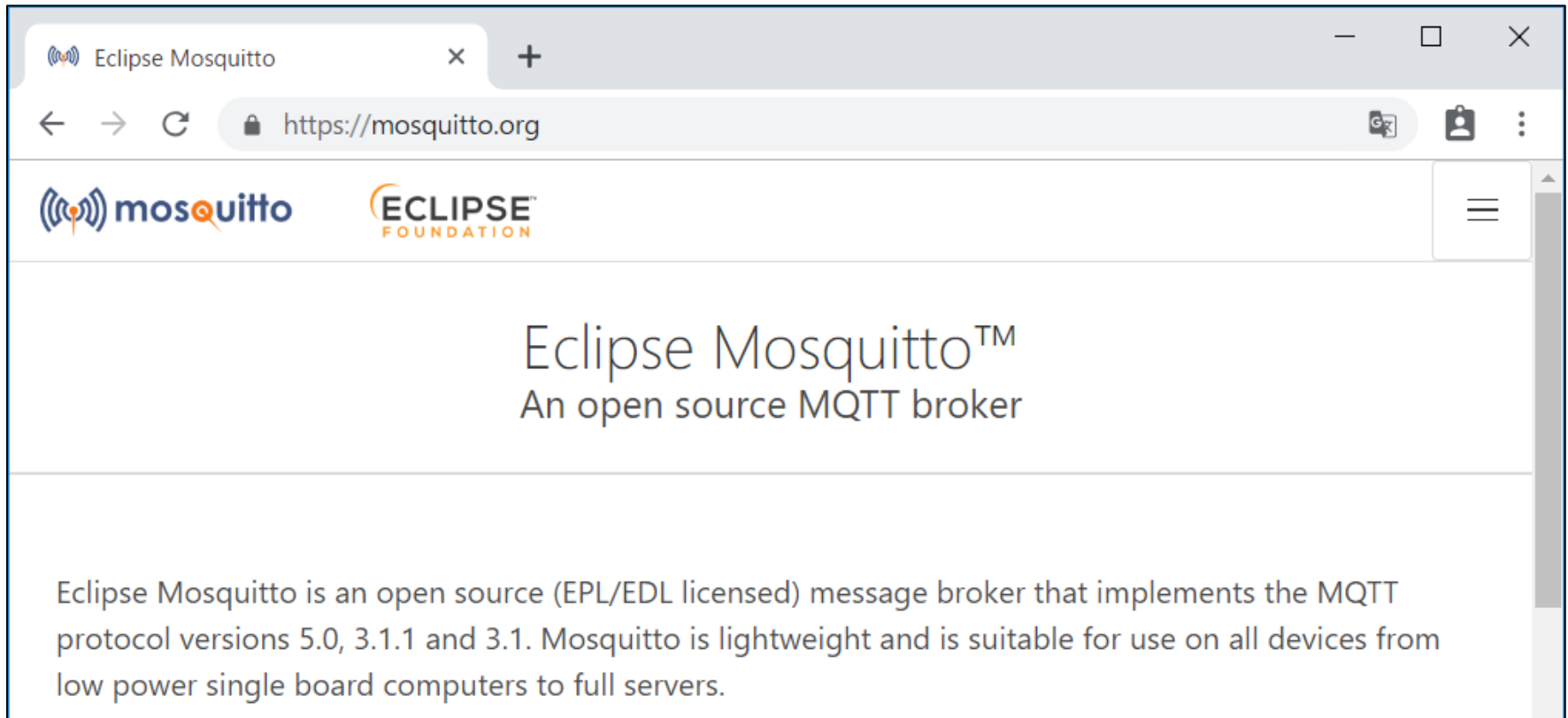
- MQTT Brokerは様々なソフトウェア、サービスで提供されています。

#	提供形式	ソフトウェア／サービス名	提供元
1	ソフトウェア	Mosquitto	Eclipse Foundation
2		ActiveMQ (旧名 : Apollo)	Apache Foundation
3		RabbitMQ	Pivotal
4		Mosca	githubで公開
5		EMQ X Broker	githubで公開
6		VerneMQ	githubで公開
7	サービス	AWS IoT	Amazon Web Services
8		Cloud IoT Core Cloud Pub/Sub	Google
9		Azure IoT Hub	Microsoft

Mosquittoとは

- MosquittoはMQTTのリファレンス実装です。MQTTの全仕様を網羅しています。

<https://mosquitto.org/>



Mosquitto をセットアップ

- Raspberry Piにsshログイン後、aptコマンドでインストールします。
- 各コマンドのバージョン確認、サービスが「active状態」であれば成功です。

```
# Mosquittoのインストール
```

```
sudo apt install mosquitto mosquitto-clients
```

```
# インストール確認(バージョンを確認)
```

```
mosquitto --help
```

```
mosquitto_sub --help
```

```
mosquitto_pub --help
```

```
# mosquittoサービスのステータス確認
```

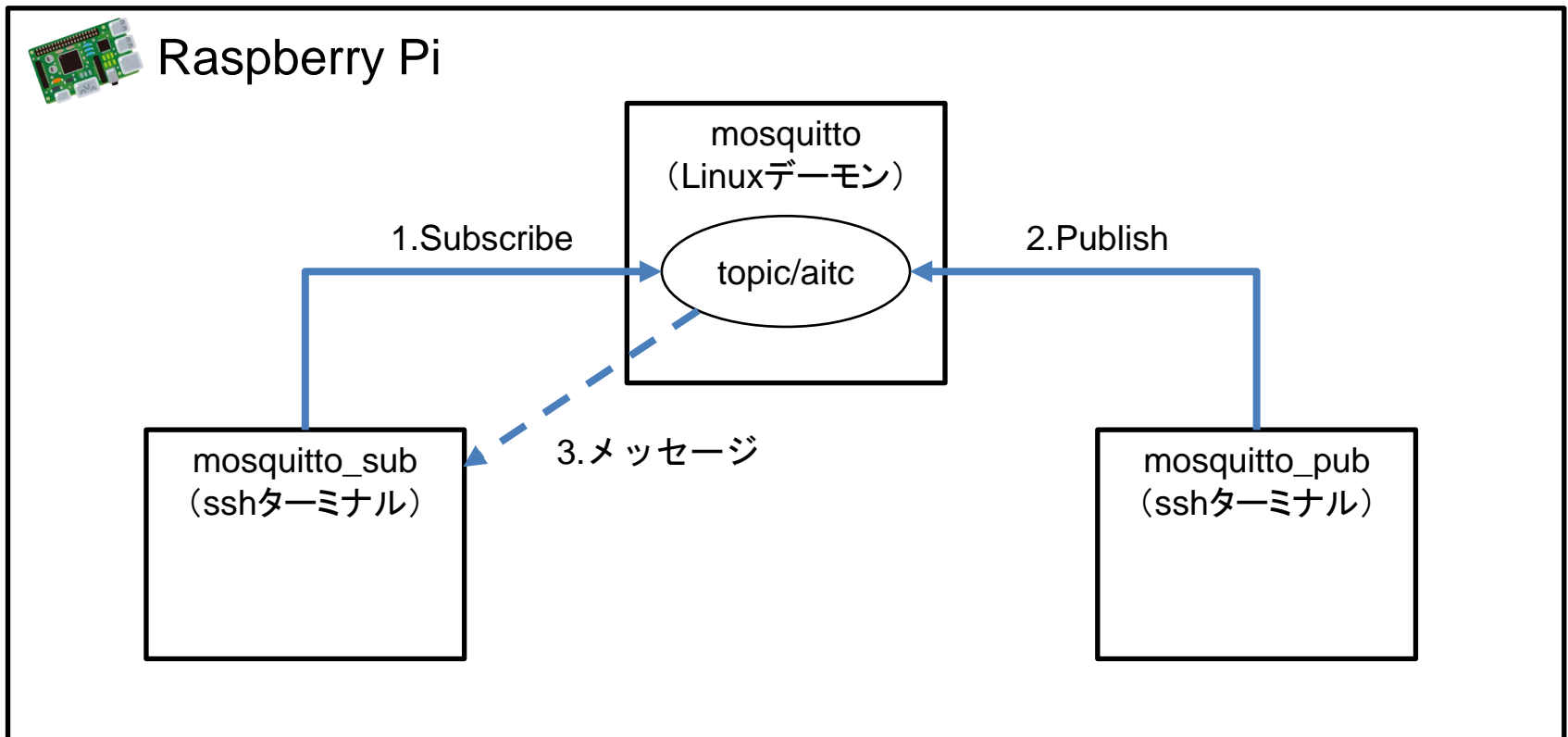
```
systemctl status mosquitto
```

Mosquittoのコマンド

#	コマンド	内容
1	mosquitto	MQTT Brokerサービス。
2	mosquitto_sub	Subscribeコマンド。 任意のトピックを監視し、メッセージを取得する。
3	mosquitto_pub	Publishコマンド。 任意のトピックに、メッセージを送信する。

実践MQTT (RaspberryPi編)

- Raspberry Piにsshターミナル2つでログインし、それぞれをPublisher、Subscriberとしてメッセージの送受信をします。



実践MQTT (RaspberryPi編)

- mosquitto_subでトピックの監視を開始、その後mosquitto_pubでメッセージを送信すると、mosquitto_sub側に表示されます。

```
$ mosquitto_sub -d -t topic/aitc -h 127.0.0.1
```

受信側

```
Client mosqsub/13382-raspberry sending CONNECT
```

```
Client mosqsub/13382-raspberry received CONNACK
```

```
Client mosqsub/13382-raspberry sending SUBSCRIBE (Mid: 1, Topic: topic/aitc, QoS: 0)
```

```
Client mosqsub/13382-raspberry received SUBACK
```

```
Subscribed (mid: 1): 0
```

```
Client mosqsub/13382-raspberry received PUBLISH (d0, q0, r0, m0, 'topic/aitc', ... (15 bytes))
```

```
hello mosquitto
```

メッセージを受信

```
$ mosquitto_pub -d -t topic/aitc -h 127.0.0.1 -m "hello mosquitto"
```

送信側

```
Client mosqpub/13385-raspberry sending CONNECT
```

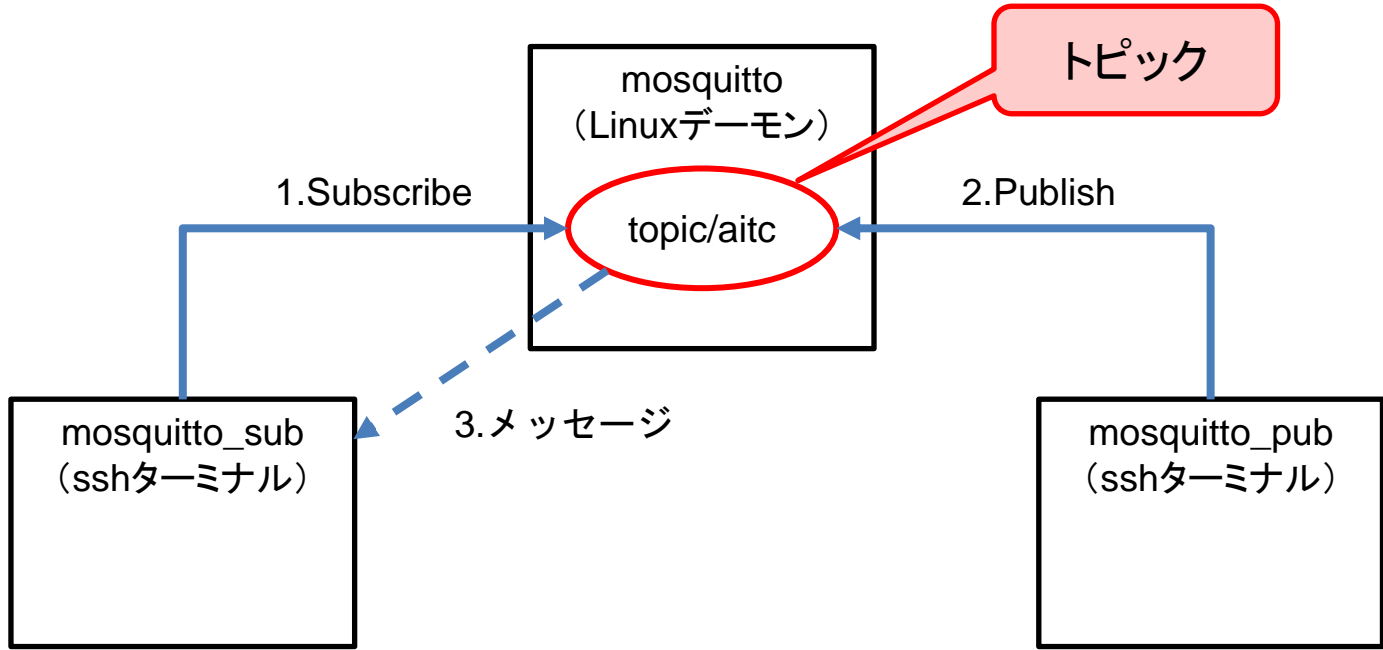
```
Client mosqpub/13385-raspberry received CONNACK
```

```
Client mosqpub/13385-raspberry sending PUBLISH (d0, q0, r0, m1, 'topic/aitc', ... (15 bytes))
```

```
Client mosqpub/13385-raspberry sending DISCONNECT
```

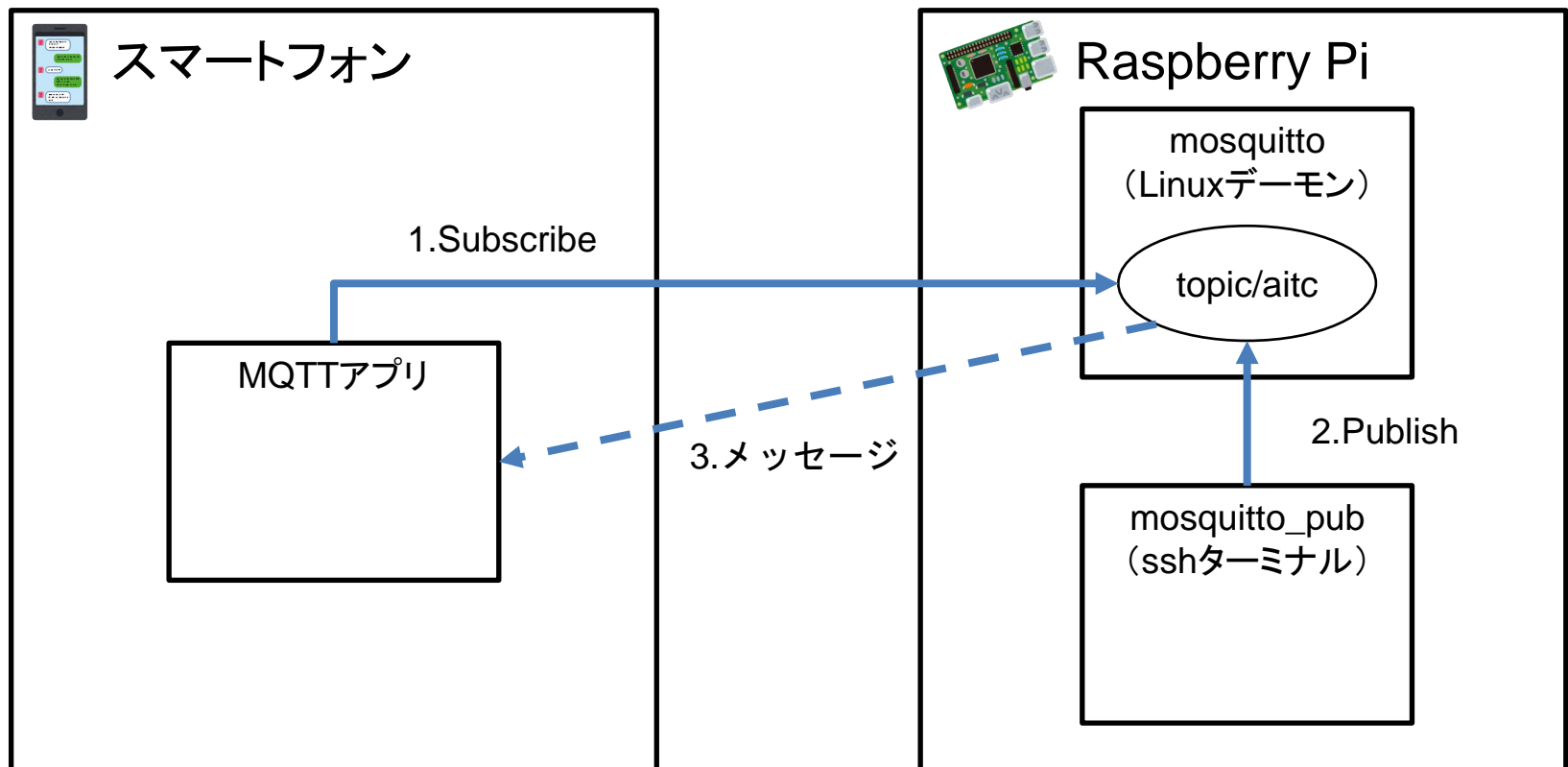
補足：トピック

- メッセージをやり取りするにあたって、エンドポイントとなるものがトピックです。
- 任意のトピックを決めることで、混線することなくメッセージの送受信ができます。



実践MQTT（スマホ編）

- スマホのMQTTアプリでSubscribeし、Raspberry PiからPublishしたメッセージを受信します。



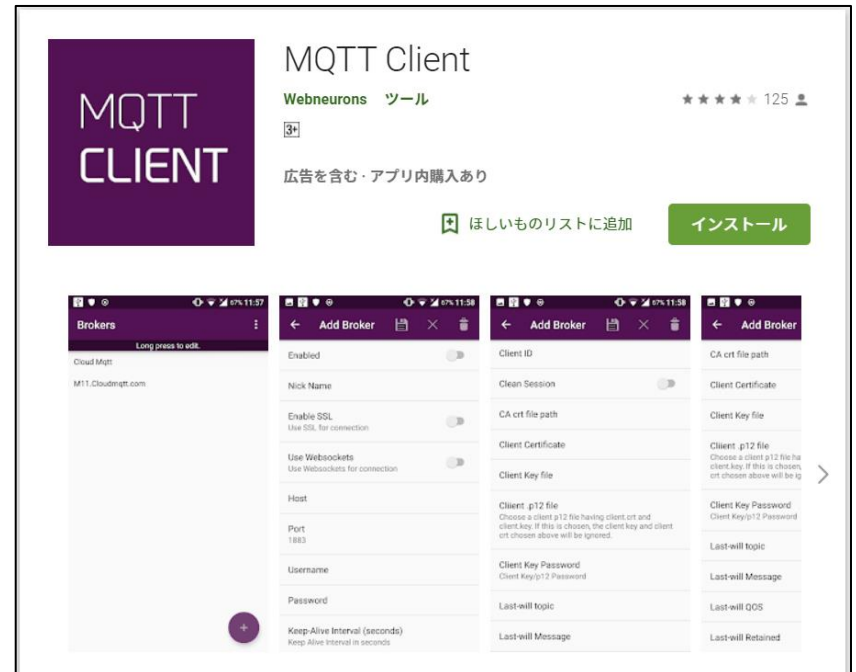
実践MQTT (スマホ編)

- 今回使用するMQTTアプリをインストールしてします。

iOSの方 MQTTTool



Androidの方 MQTT Client



実践MQTT (iOS編)

- MQTTToolを起動し、Mosquittoへの接続設定、Subscribeの設定をします。

接続設定

15:35

MQTTTool

Host: 192.168.128.202

Port: 1883 Clean Session:

Client Id: MQTTTool-528452877

Leave blank for unauthenticated access

Username:

Password:

Connect Save Password

Status: Disconnected Disconnect_Requested

Connect Subscribe Publish Stats About

接続先を設定する
※Hostは外部から接続できるIP

設定後
タップ

Subscribe設定

15:36

MQTTTool

Topic: topic/aipc

QOS: 0 1 2

Subscribe Clear

Status: Unsubscribed

Connect Subscribe Publish Stats About

監視するトピックを設定

設定後
タップ

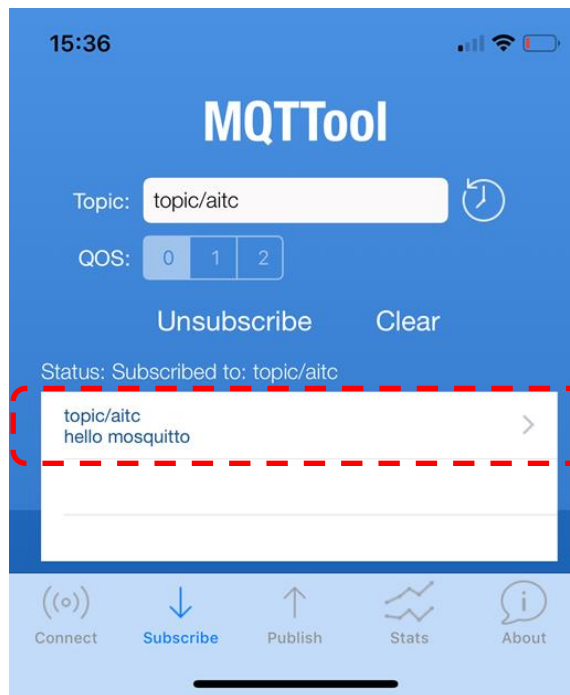


実践MQTT (iOS編)

- Raspberry Pi上からメッセージをPublishするとMQTTToolにメッセージが表示されます。

```
$ mosquitto_pub -d -t topic/aipc -h 127.0.0.1 -m "hello mosquitto"
```

Raspberry Pi



Raspberry Piから送信した
メッセージがMQTTToolに
表示された

実践MQTT (Android編)

- MQTT Clientを起動し、Mosquittoへの接続設定、Subscribeの設定をします。

接続設定

The first screenshot shows the 'Brokers' list with a '+' button at the bottom right. A red callout bubble points to this button with the text '追加ボタンをタップ'.

The second screenshot shows the 'Add Broker' screen with fields for 'Enabled', 'Nick Name', 'Host', 'Port', 'Username', and 'Password'. A red callout bubble points to the 'Host' field with the text '接続先を設定する'.

The third screenshot shows the 'Matt test - Edit' screen with fields for 'Client ID', 'Clean Session', 'CA crt file path', 'Client Certificate', 'Client Key file', 'Client .p12 file', 'Client Key Password', 'Last-will topic', 'Last-will Message', and 'Last-will QOS'. A red callout bubble points to the 'Client ID' field with the text 'Client ID設定'. Another red callout bubble points to the 'Client ID' field with the text '設定後タップ'.

Subscribe設定

The screenshot shows the 'Matt test - received messages' screen. At the top, it says 'Connected to tcp://192.168.128.202:1883'. Below that, the topic 'topic/aitc' is shown with 'No Message Received'. A red callout bubble points to the topic field with the text '監視するトピックを設定'.

Below the topic field, there is a 'SUBSCRIBE' button. A red callout bubble points to this button with the text '設定後タップ'.

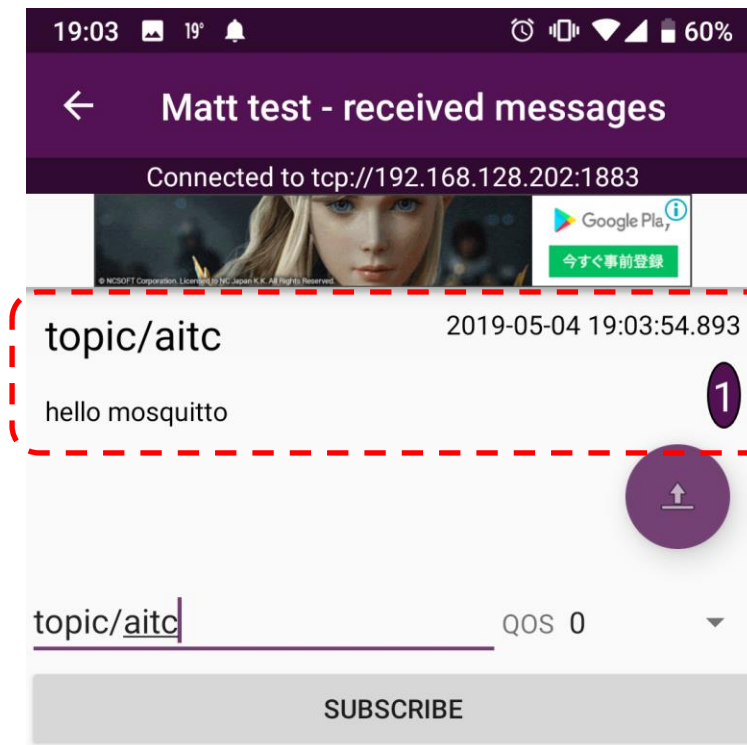
A keyboard is visible at the bottom of the screen.

実践MQTT (Android編)

- Raspberry Pi上からメッセージをPublishするとMQTT Clientにメッセージが表示されます。

```
$ mosquitto_pub -d -t topic/aitc -h 127.0.0.1 -m "hello mosquitto"
```

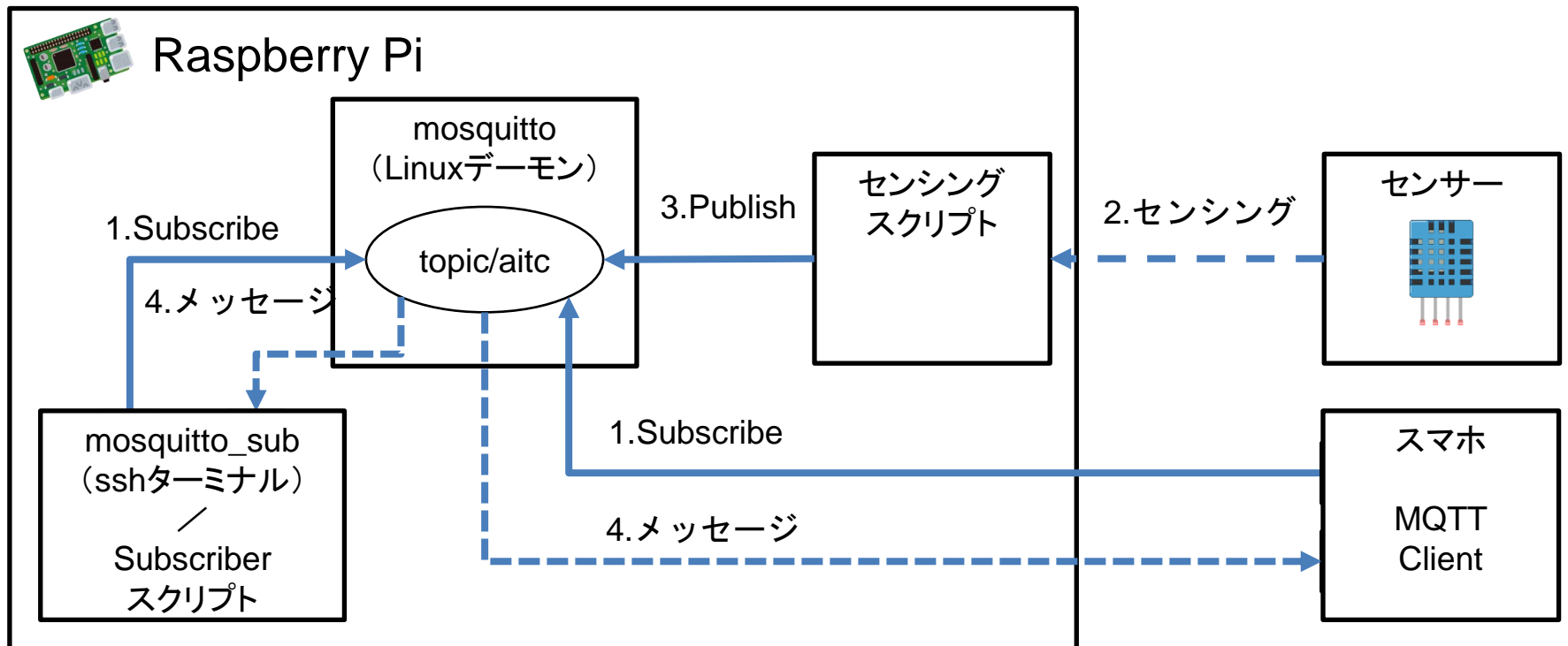
Raspberry Pi



Raspberry Piから送信したメッセージがMQTT Clientに表示された

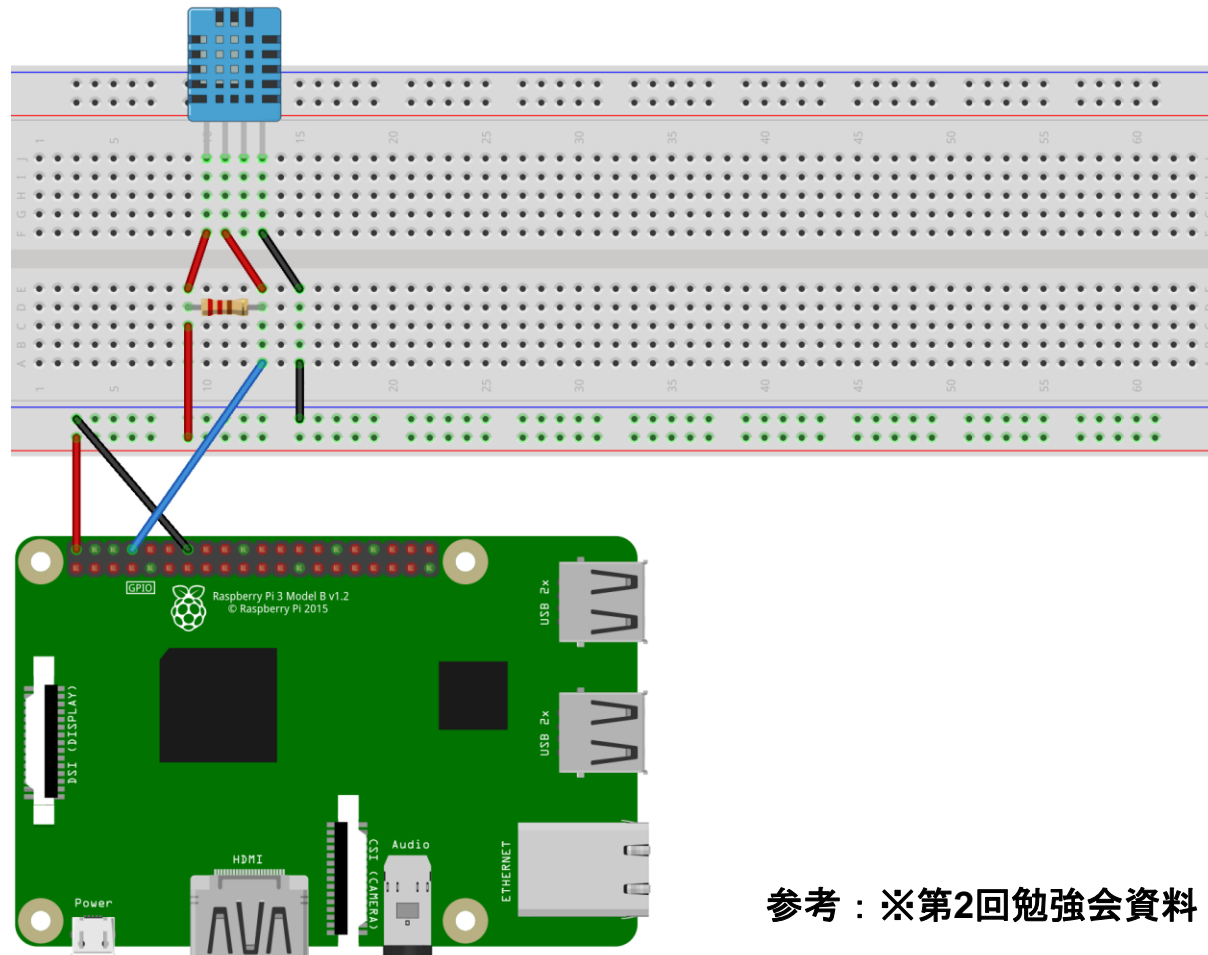
実践MQTT (センサー編)

- センサーから取得したデータをメッセージとして送受信します。



実践MQTT（温度センサー）

- 温度センサー（DHT11）から温湿度を取得します。



参考：※第2回勉強会資料

実践MQTT（温度センサー）

- Python3用のMQTTクライアントモジュールをインストールします。

```
$ sudo pip3 install paho-mqtt
```

Raspberry Pi

実践MQTT（温度センサー）

- DHT11のサンプルを修正し、温湿度取得時にメッセージを送信するように修正します。
 - example.pyをコピーして dht11_example.py作成
 - 20行目付近に赤字部分を追加

```
import paho.mqtt.client as mqtt
```

dht11_example.py

```
...
```

```
if result.is_valid():
```

```
    print("Last valid input: " + str(datetime.datetime.now()))
```

```
    print("Temperature: %d C" % result.temperature)
```

```
    print("Humidity: %d %% " % result.humidity)
```

```
    # 温湿度をMQTTにPublishする
```

```
    client = mqtt.Client()
```

```
    client.connect('127.0.0.1', 1883, keepalive=60)
```

```
    client.publish('topic/aitc', '時刻:' + str(datetime.datetime.now()) + ' 温度:' + str(result.temperature) + ' 湿度:' + str(result.humidity))
```

実践MQTT（温度センサー）

- dht11_example.pyを実行します。
 - 6秒毎に温度・湿度の数値が送信されます。

```
$ python3 dht11_example.py
```

送信側

```
$ mosquitto_sub -d -t topic/aitc -h 127.0.0.1
```

受信側

```
Client mosqsub/13382-raspberry sending CONNECT
```

```
Client mosqsub/13382-raspberry received CONNACK
```

```
Client mosqsub/13382-raspberry sending SUBSCRIBE (Mid: 1, Topic: topic/aitc, QoS: 0)
```

```
Client mosqsub/13382-raspberry received SUBACK
```

```
Subscribed (mid: 1): 0
```

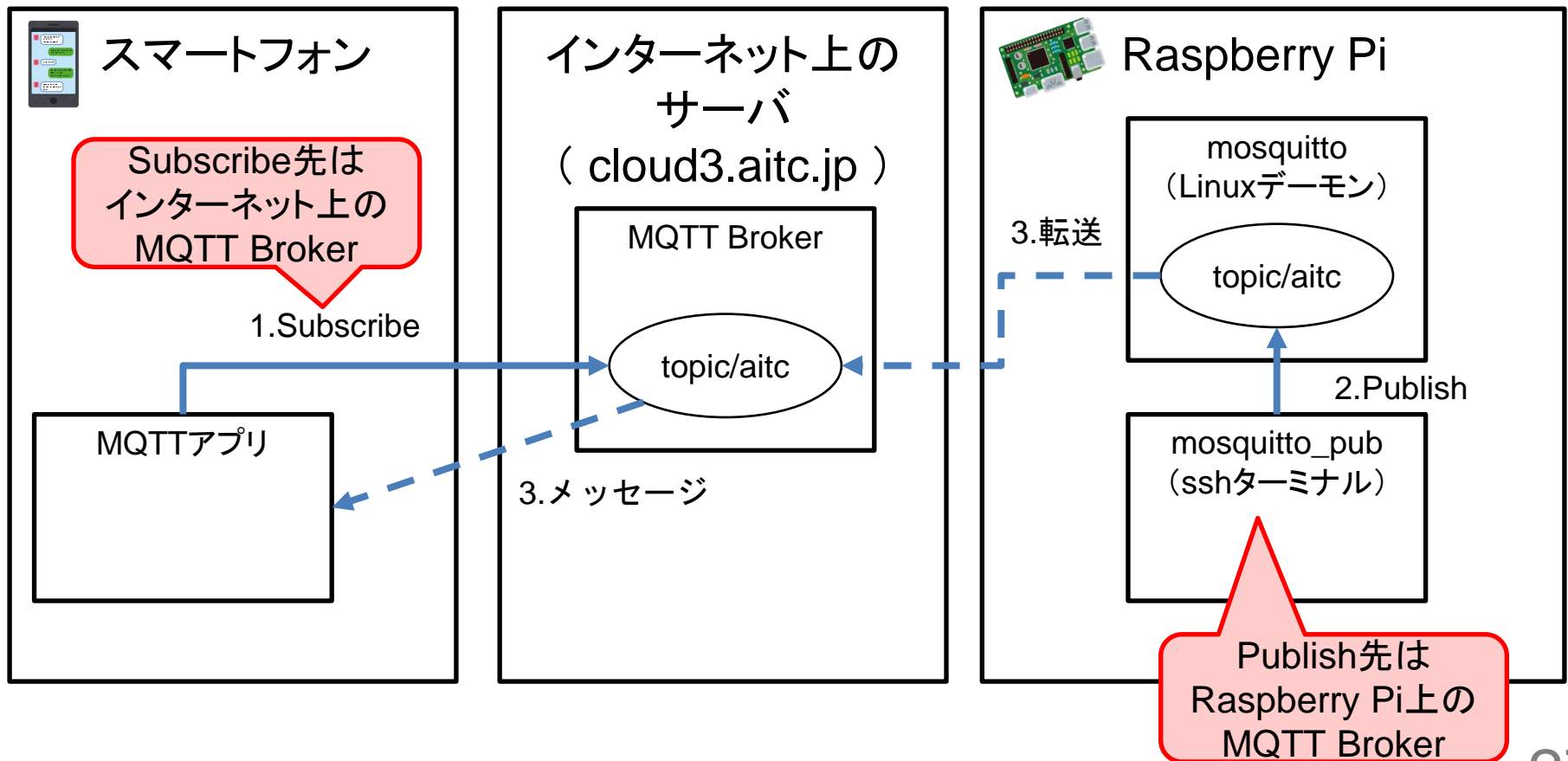
```
Client mosqsub/13382-raspberry received PUBLISH (d0, q0, r0, m0, 'topic/aitc', ... (15 bytes))
```

```
時刻:2019-05-08 17:14:10.597891 温度:26 湿度:40
```

AITC 実践MQTT (サーバ連携編)

先端IT活用推進コンソーシアム

- インターネット上のMQTT Brokerを中継してメッセージを受信します。
※ブリッジといえます。



実践MQTT（サーバ連携編）

- Mosquittoでブリッジをするには、ブリッジの設定を設定ファイルに記述します。

```
$ cd /etc/mosquitto/conf.d
```

設定ファイルの作成と反映

```
$ sudo vim airc-cloud3.conf
```

・・・設定ファイル作成

～ 設定記述後 ～

```
$ sudo systemctl restart mosquitto
```

・・・設定ファイルを反映

```
connection airc-cloud3.<名前<のアルファベット>
```

airc-cloud3.conf

```
address cloud3.airc.jp:1883
```

```
try_private false
```

```
topic # out 0
```

```
topic # in 0
```

例

airc-cloud3.shigenobu.kondo

実践MQTT（サーバ連携編）

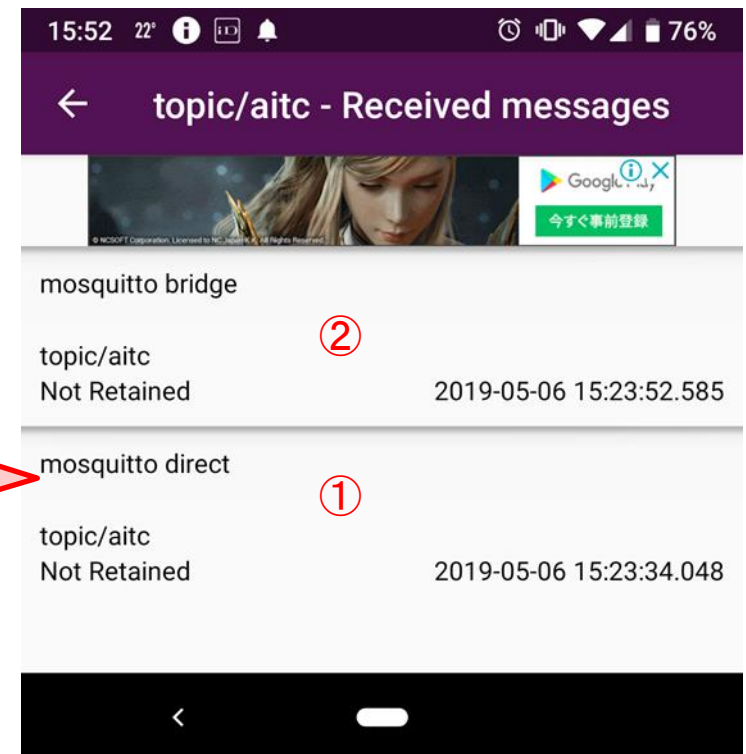
- Mosquittoでブリッジをするには、ブリッジの設定を設定ファイルに記述します。

```
$ mosquitto_pub -d -t topic/aitc -h cloud3.aitc.jp -m "mosquitto direct" ①
$ mosquitto_pub -d -t topic/aitc -h 127.0.0.1 -m "mosquitto bridge" ②
```

MQTTアプリの設定

Host : cloud3.aitc.jp
Port : 1833

mosquitto_pubの送信先が異なる2つのメッセージが、「cloud3.aitc.jp」から受信できている。

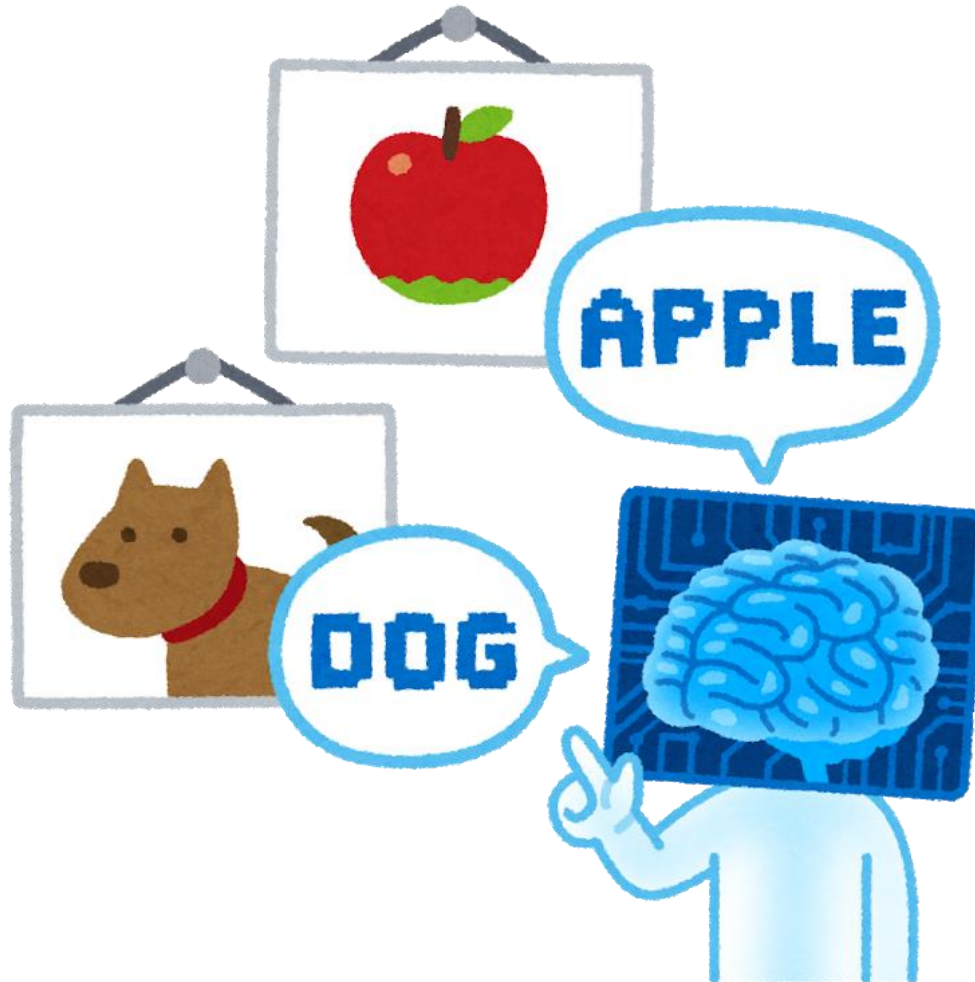


本日はここまで

質疑応答

次回予告

- Webカメラで撮影した画像を解析する方法を学びます。



アンケートのお願い

本日の勉強会についてご意見／ご感想を
ご記入ください。

<http://apps.aitc.jp/senior/3.html>