

MQTT 通信 / HTTP 通信

宮田 賢一

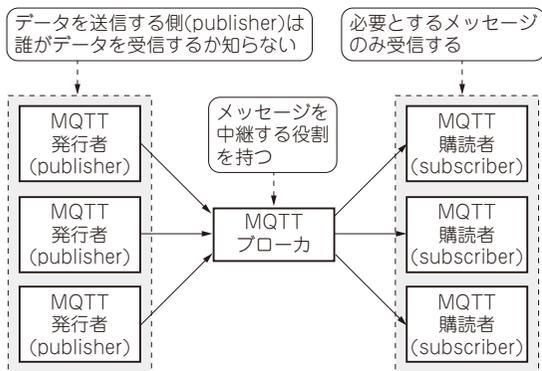


図1 MQTTによる通信方式

本章では、インターネットを介した通信プロトコルの活用方法を説明します。外付けモジュールなしに直接インターネットに接続できるラズベリー・パイ Pico W (以降、Pico W) と ESP32 を使います。

MQTT を使って通信する

● MQTT は軽量かつシンプルなデータ転送プロトコル

MQTT とは、軽量・シンプルなデータ転送プロトコルです。軽量というのは送りたいデータ本体以外に必要なメッセージ・ヘッダが最小2バイトと小さいことを意味していて、通信状況が十分とは言えない場所に設置されたセンサから情報をインターネットに送信するような使い方での効果を発揮します。

MQTT は 発行 者 / 購 読 者 モデル (publisher/subscriber モデル) を採用しています。発行者が送信したデータは MQTT プロセッサが受け取った後、MQTT プロセッサから購読者にデータが配信されるという仕組みです (図1)。この方式のポイントは、発行側は誰が受信するかを気にする必要がなく、購読者側も直接発行者を特定せずにデータを受け取れるということです。

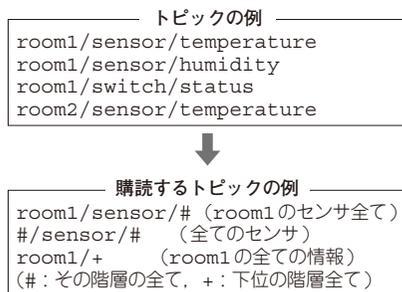


図2 MQTTのトピックの例

▶ MQTT のトピックとメッセージ

MQTT でやりとりされるデータは、トピックとメッセージからなります。トピックはメッセージに対するタグのようなものです。購読者側は自分自身を MQTT プロセッサに登録するときに、受信したいトピックを指定します。図2にトピックの例を示します。発行側はメッセージのトピックを細かく指定するのに対して、購読者側はワイルドカード (# や +) を使って特定の分類に属するメッセージのみ受け取ることができます。

● ラズベリー・パイ Pico W と ESP32 で MQTT を使う

MicroPython で MQTT を使うには、最初に `umqtt.simple` をインストールする必要があります。MicroPython では Python の `pip` モジュールと同じように、ネットワーク上のモジュールを検索してインストールしてくれる専用のモジュール `mip` が用意されているので、これを活用します。`mip` は Pico W または ESP32 がネットワークに接続している状態で使えます。

インストールの様子を図3に示します。これによりプログラム格納用のフラッシュ・メモリ上に、`lib/umqtt.simple` というフォルダと、関連する MicroPython ファイルが格納されるので、これ以降は `mip` を実行しなくても `umqtt.simple` モジュールを使えるようになります。