



## 図1 実験…実際にメッセージを配信してみる

ラズベリー・パイ4を2台用意して、1台をメッセージの送信の役割、もう1台を受信の役割としてMQTTクライアント・プログラムを用意する. 実際にMQTTブローカを介してメッセージを送受信してみる

## こんな実験…ラズパイ同士で メッセージを送受信してみる

今回は、2台のラズベリー・パイ4を使って、図1 に示すようなメッセージ配信の実験をします。

2台のうちの1台をメッセージ送信の役割、もう片方 を受信の役割として扱います.メッセージが配信され る様子をWiresharkでモニタして、どのようなMQTT パケットがやりとりされているかを確認します.

MQTTクライアント・プログラムは,第3回(本誌 2021年9月号)と同じように,Node.jsとnpmパッケー ジのmqtt.jsを使って実装します.これら実行環境の インストール方法は,次のウェブ・ページを参照して ください.

https://interface.cqpub.co.jp/pi\_
mqtt/

MQTTブローカは, test.mosquitto.orgを利用しま す. test.mosquitto.orgは無料で公開されていて,利 用手続きは特にありません.



**図2 MQTTメッセージ配信実験の全体構成** pub.jsが送信側、sub.jsが受信側のクライアント・プログラム

## ● 全体構成

図2に示すのは、実験の全体構成です. 各種バー ジョンは次の通りです.

- OS : Raspberry Pi OS/2020-05-07
- Node.js : 14.15.1
- mqtt.js : 4.2.8
- Wireshark : 3.4.7

ステップ1:Wiresharkを用意する

● 最新版のソースコードを入手してビルドする

今回の実験では、最新版のWiresharkを使いたいので、ソースコードからビルドします。

ラズベリー・パイのターミナルから、次のコマンド でWiresharkをビルドします.ビルドには約1時間掛 かります.

sudo apt install cmake flex bison libgcrypt20-dev libssh-dev libpcapdev libsystemd-dev libc-ares-dev qtbase5-dev qttools5-dev qtmultimedia 5-dev

cd \$HOME -

git clone https://github.com/wire

	第1回	HTTPとここが違う! MQTTがIoTに向く理由 (2021年5月号)
194	第2回	接続と切断を制御するパケット(2021年7月号)
	第3回	ラズパイをクライアントにしてウェブ上の公開ブローカに接続する(2021年9月号)

**Interface** 2021年11月号