

エナジーハーベスティングに適した軽量通信プロトコルの 実装と評価

樽水 康平[†] 田中 和明[‡]

九州工業大学大学院情報工学府[†] 九州工業大学大学院情報工学研究院[‡]

1. 研究背景

近年、組み込み機器の機能増加や性能の向上に伴って、組み込み開発の規模が増加している。特に、組み込み製品開発におけるソフトウェア開発費の割合は年々増加しており、開発技術の向上や生産性の効率化が求められている。

また、組み込み分野では、モノのインターネットと呼ばれる IoT に注目が集まっている。近年では、通信モジュールの小型・省電力化によって、センサデバイスなどの組み込み機器がインターネットに接続することが可能となっている。しかし、IoT の状況下では、デバイスが高速で信頼性の高い回線に接続されるという保証はなく、不安定な通信回線や、低用量なバッテリーといった環境で動作しなければならない場合がある。そのため、不安定な環境でも利用することが可能な通信方法が求められている。

本研究では、IoT の実現に適した MQTT プロトコルによる通信を、エナジーハーベスティング技術を使ったデバイス上で実現し、mruby のような生産性の高い言語を用いて開発が行えるようにすることを目的とする。そのために、不安定な環境での利用に適している BLE (Bluetooth Low Energy) で送信されたデータを受信し、MQTT の形式に変更して送信するためのゲートウェイを実装し、その評価を行った。

2. システム構成

2.1. MQTT (MQ Telemetry Transport)

MQTT は、M2M/IoT のための接続プロトコルであり、単純で軽量な Publish-Subscribe 型のモデルとして設計されている。このモデルでは、メッセージの送信者 (Publisher) が特定の受信者 (Subscriber) を指定せずにメッセージを送信す

ることが可能である。MQTT では、メッセージにトピックを付与し、受信するメッセージの種類を指定することで、1対1だけでなく、1対多の通信を行うことが可能である。

また、HTTP のヘッダサイズが「50 バイト～」であるのに対して、MQTT は「2 バイト～」である。そのため、小さいデータを大量、かつ高頻度にやり取りするのに適している。

その他の特徴として、到達保証を指定する QoS. Publisher の接続が切れた際に、指定したメッセージを Subscriber に送信するための Will. 最後に送信されたメッセージを保存しておき、新たに接続された Subscriber に送信するための Retain などをオプションとして指定することが可能である。

2.2. mruby

mruby とは、オブジェクト指向スクリプト言語である Ruby をベースに、最小限の機能以外を取り除くことで、組み込みシステムに使用出来るようにするために軽量化された言語である。Ruby には、C 言語と比較して短いコードでの可読性の高い記述が可能というメリットがあり、mruby でも同様の記述が可能である。

3. 開発環境

3.1. Raspberry Pi 3

BLE で送信されたデータを MQTT の形式に変更して送信するゲートウェイとして動作させるデバイスには、Raspberry Pi 3 を利用した。

Raspberry Pi 3 はラズベリーパイ財団によって開発されたシングルボードコンピュータであり、標準で BLE 通信やインターネット接続が可能である。その他の仕様に関しては表 1 に示す。

表 1: Raspberry Pi 3 の仕様

CPU	ARM Cortex-A53(1.2GHz)
RAM	1 GB
OS	Raspbian(Linux 系の OS)

Implementation of lightweight communication protocol for energy harvesting and its evaluation

Kohei Tarumizu[†] Kazuaki Tanaka[‡]
Kyusyu Institute of Technology Graduate School of
Computer Science and Systems Engineering^{†‡}

3.2. MQTT ブローカー

ブローカーは、MQTT による通信を行う際に、クライアント間のメッセージを中継するサーバである。MQTT の実装には、オープンソースである Mosquitto を利用した。Mosquitto は、Windows や Linux など、様々なアーキテクチャ向けのバイナリが用意されており、今回は Linux 向けのバイナリを使用した。

3.3. エナジーハーベスティングデバイス

エナジーハーベスティング技術を使って動作しているデバイスとして、Cypress 社の S6SAE101A という PMIC を搭載したマザーボードを利用した。本デバイスは、200LX 以上の照度を得られる環境で、太陽電池によって収穫した電力で動作させることが可能である。また、温度・湿度のセンサや BLE 通信のモジュールが備わっているため、センサから取得した値を BLE によって送信することが可能である。

4. 研究内容

MQTT プロトコルによる通信を行う際は、TCP/IP によるネットワーク接続が必要となる。そのため、エナジーハーベスティングで動作しているデバイスのような、不安定な環境には適していない。そこで、エナジーハーベスティングで動作しているデバイスから BLE によって送信されたデータを受信し、MQTT の形式に変更して送信するゲートウェイの実装を行った。図 1 に、その構成を示す。

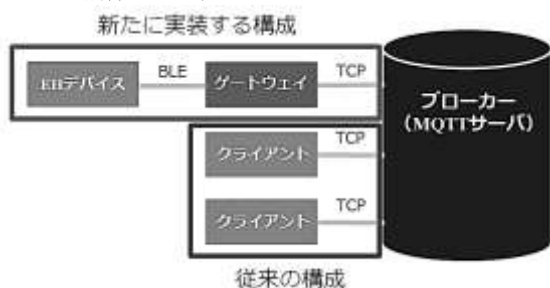


図 1: 本研究で想定するネットワーク構成

BLE で送信されるパケットには、温度センサから取得した値の他に、送信したデバイスやサーバを識別するための UUID という識別子が含まれている。今回実装したゲートウェイでは、送信元のデバイスの識別と、MQTT の形式に変更する際のトピックを指定するために UUID を利用している。

ゲートウェイの実装には、mruby で並行処理を行うための fiber というコルーチンを利用した。複数のコンテキストを切り替えながら動作させ

ることで、BLE で送信されたデータを受信する処理と、受信したデータを解析し、MQTT の形式に変更して送信する処理を、非同期な並行処理として実現している。

5. 研究結果

BLE によるデータを受信した際、UUID によるデバイスの識別を行い、指定したデバイスであった場合に、MQTT の形式に変更してデータを送信することが出来た。

また、通常の MQTT によるデータ送信と、今回実装したゲートウェイを経由した場合のデータ送信の実行時間を、データ量を変化させて測定した結果を図 2 に示す。それぞれの通信時間の差分は、最大でも 0.04[ms]で、全体の通信時間に対する割合はいずれも 1%未満であった。

今回実装したゲートウェイでは、この結果に BLE の通信時間が加わることになる。今回、BLE でデータを送信する際は、アドバタイズメントパケットを利用しており、データの総量は 45[byte] (14[byte] はヘッダとして使われている)で、通信時間は約 38[ms]であった。

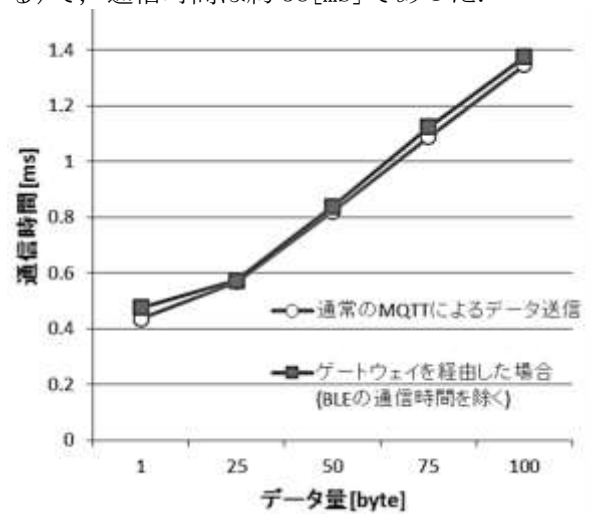


図 2: データ量毎の通信時間の変化

6. 結言

本研究では、エナジーハーベスティング技術で動作しているデバイスのような不安定な環境で、IoT の実現に適した MQTT による通信を行うことが出来た。

本研究のネットワーク構成では、インターネットに接続する必要があるのはゲートウェイのみである。そのため、エナジーハーベスティング技術で動作しているデバイスと併用することで、ネットワークや電源の配線が不要となり、IoT のような、デバイスを大量に設置したい状況での活用が期待できる。