

Node-REDによるIoTクラウド基盤を用いた室内環境の計測

常三島技術部門
情報システムグループ

辻 明典(TSUJI Akinori)

1. はじめに

インターネット上に公開するアプリケーションの開発には、サーバやネットワーク等のインフラ整備、プログラム開発環境の構築、複数のプログラミング言語の習得をはじめ、様々な準備と時間、コストが必要である。

本研究では、Node-REDによるIoTクラウド向けの開発環境の構築を目的とする。Node-REDの導入によって、特別なソフトウェアを必要とせず、ウェブブラウザさえあれば、いつでもどこでも開発ができる環境を整備できる。また、ノーコード／ローコード開発対応のため、非エンジニアや専門家でなくとも開発に参加できる。本稿では、構築したNode-REDによるIoTクラウド基盤の詳細について述べた後、実証実験として、室内環境を計測するアプリケーション開発を行ったので報告する。

2. ノーコード／ローコード開発

ノーコード／ローコード開発は、プログラマや専門家でなくとも「プログラムすることなく」アプリケーションを作成できる開発手法である。プログラム言語の習得には、コンピュータの仕組みやアーキテクチャ、プログラムの文法をはじめとした基礎からの学習が必要不可欠である。一方、ノーコード／ローコード開発は、画面上の操作のみでアプリケーション開発が行えるため、比較的容易に開

発を始められるのが特徴である。ノーコード／ローコード開発の利点は、完成された機能モジュールを組み合わせてプログラムを作成するため、開発や検証に係る工数を大幅に削減でき、新規サービスやアイデアの実証、アプリケーションの公開を短期間のうちに立ち上げられる。また、アプリケーション開発をオンライン上で行えるため、開発者からの視点に限らず、一般ユーザからの要望も取り入れることができ、開発したサービスのユーザエクスペリエンス(UX)の向上も期待できる。

2.1 Node-RED

Node-REDは、ノードと呼ばれる機能モジュールを接続することでプログラムなしでアプリケーション開発ができるオープンソースのノーコード／ローコード開発環境である^[1]。図1にIoTシステムにおけるNode-REDの対応範囲を示す。Node-REDでは、多数あるノードを画面上に並べ、ノード間を繋ぐことでプログラムに相当するフローを作成し、ウェブ上で動作するアプリケーションを開発できる。Node-REDを用いることで、IoTデバイスの運用管理、データ処理や可視化、システム間連携を、画面上の操作とノードのプロパティ設定のみで実装できる。また、ダッシュボードによる画面レイアウト機能により、フローからグラフや表、ボタン等を自動で生成できる。

3. IoTクラウド基盤の構築

Node-REDによるノーコード／ローコード開発環境の構築を目的に、IoTクラウド基盤の構築を行った。図2に、構築したIoTシステムの構成を示す。システムは、IoTデバイス、IoTゲートウェイ、Node-REDが稼働するクラウドサーバ、システムを利用するユーザによって構成される。それぞれの詳細を次に示す。

・IoTデバイス

一般的なIoTデバイスは、計測対象のセンシ

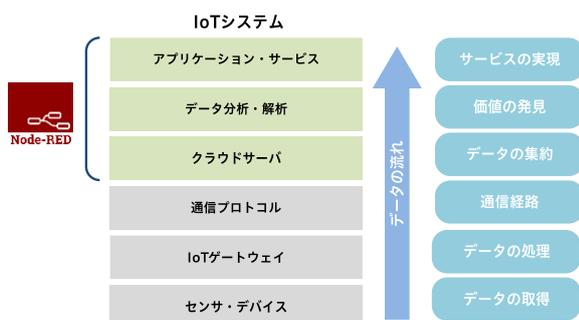


図1 IoTシステムにおけるNode-REDの対応範囲

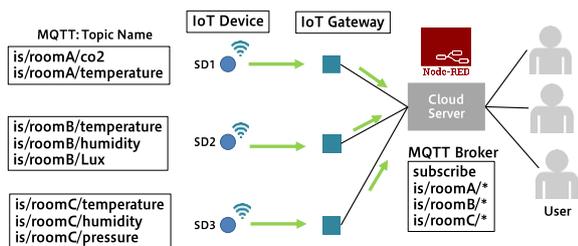


図2 構築したIoTシステムの構成

ング機能やアクチュエータ機能を有し、決められた通信規格に従ってIoTゲートウェイと通信するデバイスである。市販の計測機器(電源、テスタ、オシロスコープ等)やセンサに通信機能を搭載することでIoTデバイスの一つとして扱うこともできる。

・IoTゲートウェイ

IoTゲートウェイは、IoTデバイスをインターネットに接続するためのネットワーク機器である。IoTデバイスには、様々な通信プロトコル(シリアル通信(RS-232C, RS-485), Bluetooth, LoRa, Sigfox等)があるが、インターネットに接続するには、LANや無線Wi-Fi等のインターネット規格へのプロトコル変換が必要となる。汎用のIoTゲートウェイとしては、有線ルータや無線アクセスポイント等が用いられる。

・クラウドサーバ

クラウドサーバは、IoTデバイスで収集されたデータの入力、処理、可視化を行うサービスを提供するサーバである。また、他のクラウドサービスとのAPI連携も提供する。ここでは、Node-REDによるノーコード/ローコード開発環境を提供し、IoTデバイスの運用管理を統合的に行う。

・ユーザ(開発者・一般ユーザ)

ユーザは、クラウドサーバにアクセスして、Node-REDのフローの作成、アプリケーションの開発を行う。アプリケーションの開発には、特別なソフトウェアを必要とせず、ブラウザ上ですべての操作が行える。開発者だけでなく、一般ユーザとも共同で開発が行える。

3.1 Node-REDサービス

クラウドサーバ上にNode-REDを複数動作させるために、サーバのポート割り当て、及びサービスとして起動する設定を行った。

・サーバのポート割り当て

Node-REDは、アプリケーション毎にポートの割り当てが必要である。デフォルトの設定では1880番が割り当てられる。Node-REDの環境設定ファイルsettings.jsのポート番号を変更したファイルを用意することで、異なるポートを使用してNode-REDを起動できる。

・サービスとしての起動

Node-REDは、Node.jsをベースとした開発環境である。Node-REDをサービスとして管理するためにPM2をインストールした^[3]。

```
$ sudo npm install -g pm2
```

PM2を導入することで、Node-REDをサーバプロセスとして起動でき、プロセスの開始、停止、再起動をそれぞれ次のように行える。

```
$ sudo pm2 start/stop/restart <service name>
```

PM2により、サーバ起動時にNode-REDを起動するよう設定すると、システムサービスの一つとして起動できる。

4. IoTクラウド基盤に基づく室内環境の計測

Node-REDによるIoTクラウド基盤の実証実験のため、室内環境の計測を行うIoTシステムを構築した。図3に示すようにCO₂センサ、温湿度センサ、気圧センサ、照度センサを搭載したIoTデバイスを室内に設置して検証を行った。IoTデバイスは、各部屋の無線アクセスポイントに接続し、クラウドサーバとの通信を行った。計測データは、MQTTプロトコルに従って、センサ毎にトピック名を付与してクラウドサーバへデータ送信を行った。

4.1 室内環境計測用のフロー

Node-REDは、IoTデバイスから送信されたデータの入力処理、データの可視化を行う。次に、Node-RED上に開発したフローの詳細について示す。

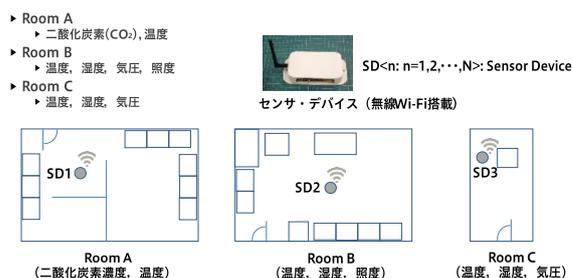


図3 室内へのIoTデバイスの設置

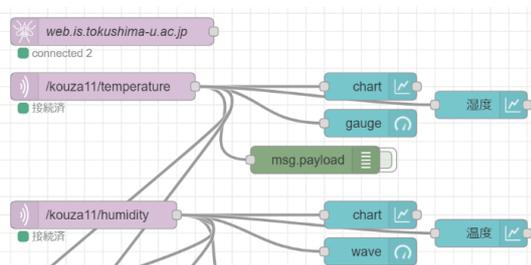
・時刻設定フロー

IoTデバイスで計測したデータの時刻設定を行うフローである。クラウドサーバはNTPによりシステムの時刻同期を行っており、Node-REDにデータが入力した時点の時刻(yyyy/mm/dd hh:mm:ss)をデータに付加する。



・MQTT入力フローとダッシュボードによるデータ可視化フロー

MQTTブローカは、node-red-contrib-aedesノードを追加するのみで実装できる。ここでは、MQTTブローカのホスト名とポート番号を設定する。IoTデバイスで計測したデータ(payload)は、センサ毎にMQTTのトピック名を付けてPublishされるので、MQTTブローカは、IoTデバイスから送られてくるデータのうちトピック名で指定したデータのみをSubscribeして受信する。トピック名を変更することで、センサごとにデータを受信したり、部屋ごとに受信したり変更できる。例えば、トピック名を*/temperatureとすると、全部屋の温度センサのデータを取得でき、/room/*とすると指定の部屋の全センサのデータを取得できる。MQTT入力のデータをダッシュボードによるチャートとゲージノードに接続することでデータの可視化ができる。ゲージはデータの瞬時値、チャートはリアルタイム更新する時系列グラフとして表示される。



・データベース管理フロー

計測したデータは、データ受信と同時にデータベースへの記録を行う。データベース管理は、Node-RED上で行うことができ、データベースの作成、参照、削除をフローによって実装できる。ここで、データベースにはSQLiteを用いた。SQLiteは、node-red-node-sqliteノード

の追加で使用できる。DB作成・参照・削除・テストなどはinjectボタンを押すことで実行できる。各動作の命令は、次のようにSQL構文をfunctionノードに記述する。

<データベースの作成>

```
create table DB_NAME(time, room, sensor, ...);
```

<データベースの登録>

```
insert into DB_NAME VALUES("a", "b", ...);
```

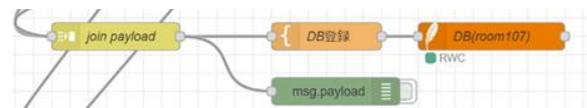
<データベースの参照>

```
select * from DB_NAME;
```



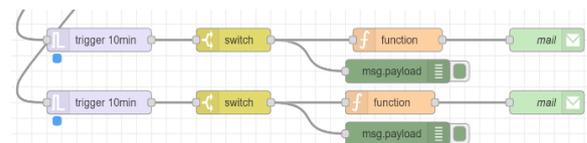
・データベース登録フロー

IoTデバイスで計測したデータをjoinノードで結合した後、データベースにまとめて登録を行う。データベースへの登録は、あらかじめ部屋毎にテーブルを作成しておき、テーブルに合わせたデータを挿入する。



・アラート送信フロー

計測したデータが設定した上限値・下限値を超過した場合に、アラート(警告)のメールを指定のアドレスに送信する。上限値・下限値は、switchノードのプロパティで設定できる。センサ毎にサンプリング時間が異なるため、triggerノードを用いて10分ごとに設定値の判定を行う。また、functionノードでアラートのメールの件名と本文を設定している。



5. Node-REDによるIoTシステムの検証結果

構築したNode-REDによるIoTクラウド基盤を用いて、ノーコードで室内環境の計測を行うシステムの検証を行った。図4は、IoTデバイスを設置したすべての部屋について、ダッシュボードノードを用いて、部屋の温度・湿



図4 全室内の計測データの表示例

度、CO2、照度等を、ゲージとグラフで表示した結果である。Node-REDでは前述のとおりノードを繋いでデプロイするだけで、画面のレイアウトや配置、グラフのサイズや表示位置等を気にせず、直ちにデータを可視化できた。さらに、HTMLやPHP等のウェブページ用のコードを記述する必要がなく、ノードのプロパティ項目で柔軟に設定変更できた。図5は、一部屋分の温湿度データを表示した結果である。ここでは、画面レイアウトをフレーム表示に設定し、現在の計測結果、データベースに記録した計測データの参照、計測データのCSVダウンロードをメニューとして追加した。図6は、データベースに記録した直近のデータをデータベースから読み出して表形式で数値表示し、さらにチャート表示として1日分のデータをグラフ表示した結果である。

実証実験を通じて、これら表示例のように、Node-RED上のノードの操作のみで、MQTTによるデータ入力、計測データの可視化、データの閾値処理、データベースへの記録、アラートの送信等を、非常に少ない工数で実現できることを確認した。Node-REDを使用せず開発を行った場合と比較すると、MQTTブローカMosquittoサーバの構築、ウェブサーバApacheの構築、メールサーバpostfixの構築、データの整形python、データの可視化Grafana、データベース管理Influx、データベースMySQL、ホームページの表示php、HTML等、複数のサーバを使用する必要があった^[3]。実際の開発工数を比較すると、システム全体で約60%の節約ができた。以上の検証結果より、Node-REDがIoTシステムの運用管理において有効であることを確認した。



図5 一部屋分の温度・湿度の表示例

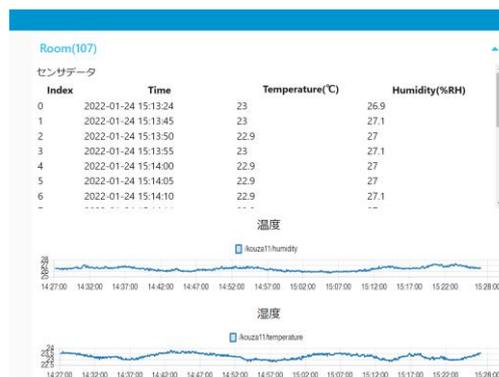


図6 データベース内の計測データの表形式表示とグラフ表示（一日分）

6. まとめ

本稿では、Node-REDによるノーコード/ローコード開発環境をクラウドサーバ上に構築し、室内環境を計測するIoTシステムの実証実験を行った。Node-REDはノーコード開発に注目が集まるが、フローによってサーバ構築やデータ処理、デバッグやプログラム検証等の複雑な処理を統合的に扱えるため作業効率上がる。結果として、IoTシステム開発に係る時間・コストを大幅に削減でき、さらにプログラム習得に係る時間も減らせる。課題としては、システムがブラックボックス化されるため、導入には作業内容を理解し、処理手順を最適化できる技術者が必要となることが挙げられる。

参考文献

- [1] Node-RED, <https://nodered.org/>
- [2] Node-RED FAQ, <https://nodered.jp/docs/faq/starting-node-red-on-boot>
- [3] 辻明典, 温湿度・空気質センサを用いた室内環境のIoTモニタリング, 徳島大学技術支援部技術報告, No.4, pp.10-13, 2021.