

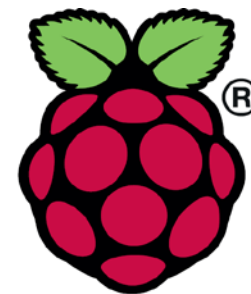
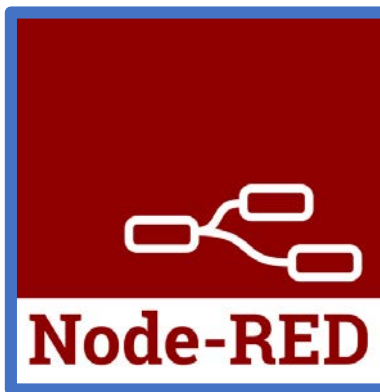
# はじめてのAWS IoT開発キット2プラス

～センサ、RaspberryPiを使い、実体験。開発短縮～

## Node RED編(抜粋版)



**AWS IoT**

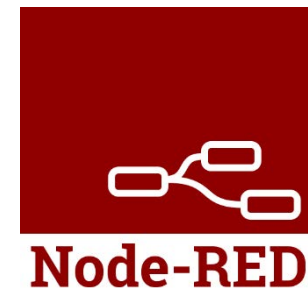


**Raspberry Pi**

スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

[sales@spectrum-tech.co.jp](mailto:sales@spectrum-tech.co.jp)



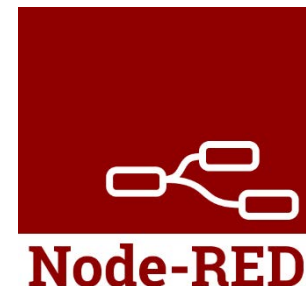
# AWS IoT開発キット2プラス 目次

| Node RED開発                   | ページ                       |
|------------------------------|---------------------------|
| ① Node RED概要                 | <a href="#"><u>3</u></a>  |
| ② Node RED起動                 | <a href="#"><u>5</u></a>  |
| ③ Node RED画面説明               | <a href="#"><u>7</u></a>  |
| ④ Node RED事例                 |                           |
| • Hello world                | <a href="#"><u>10</u></a> |
| • Google home送信              | <a href="#"><u>13</u></a> |
| • BMP280(温度、気圧): 添付センサの表示    | <a href="#"><u>16</u></a> |
| • CT2 AWS 作成: 電力計キットの表示      | <a href="#"><u>25</u></a> |
| ⑤ Node RED管理                 | <a href="#"><u>29</u></a> |
| • 表示                         |                           |
| • 読み込み: フローをインポートできます        | <a href="#"><u>30</u></a> |
| • 書き出し: フローをエクスポートできます       | <a href="#"><u>31</u></a> |
| • ノード検索: 設定したノードの検索          |                           |
| • ノード設定: 全てのノード一覧と設定ができます    | <a href="#"><u>32</u></a> |
| • フロー: 新規、修正が可能              |                           |
| • サブフロー: 新規など                |                           |
| • パレット管理: パレットの削除、追加可能       | <a href="#"><u>33</u></a> |
| • 設定: 表示、キーボード、パレットのユーザ設定    | <a href="#"><u>34</u></a> |
| 参考                           |                           |
| ① ダッシュボードの外部公開方法             | <a href="#"><u>36</u></a> |
| ② Google HomeからNode REDでの受信例 | <a href="#"><u>37</u></a> |

抜粋版のためページ  
は一致しません

## Node RED開発

### ①. Node RED概要



#### Node REDとは

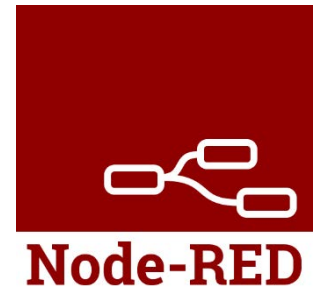
- Node-REDはフローベースプログラミング(flow-based programming)ツールであり、IBM Emerging Technology Services チームによって元は開発され、現在ではJS Foundationのひとつになっています。
- <https://nodered.jp/about/>
- 用途は、IoT開発ツール、可視化ツール(ダッシュボード)でJavaScriptが基本。
- パレット(ライブラリ)が豊富に用意されており、Raspberry Pi, Google homeなどとの連携も可能。特にIoT通信のMQTTの開発には最適
- 特徴は、プログラムの知識がなくても、フローを作成し、グラフを自由に作成できたり、各レガシー系システムとの連携なども可能。

## Node RED開発

### ①. Node RED概要

#### Node RED概要

- フローエディタとダッシュボードの二つに分類
- フローエディタは、ノードなどのアイコンを使ってプログラミングします
- ダッシュボードは、フローエディタで作ったものの実行結果をリアルタイム表示します。(ローカル又はHTTPSで、公開も可能)



# Node RED開発

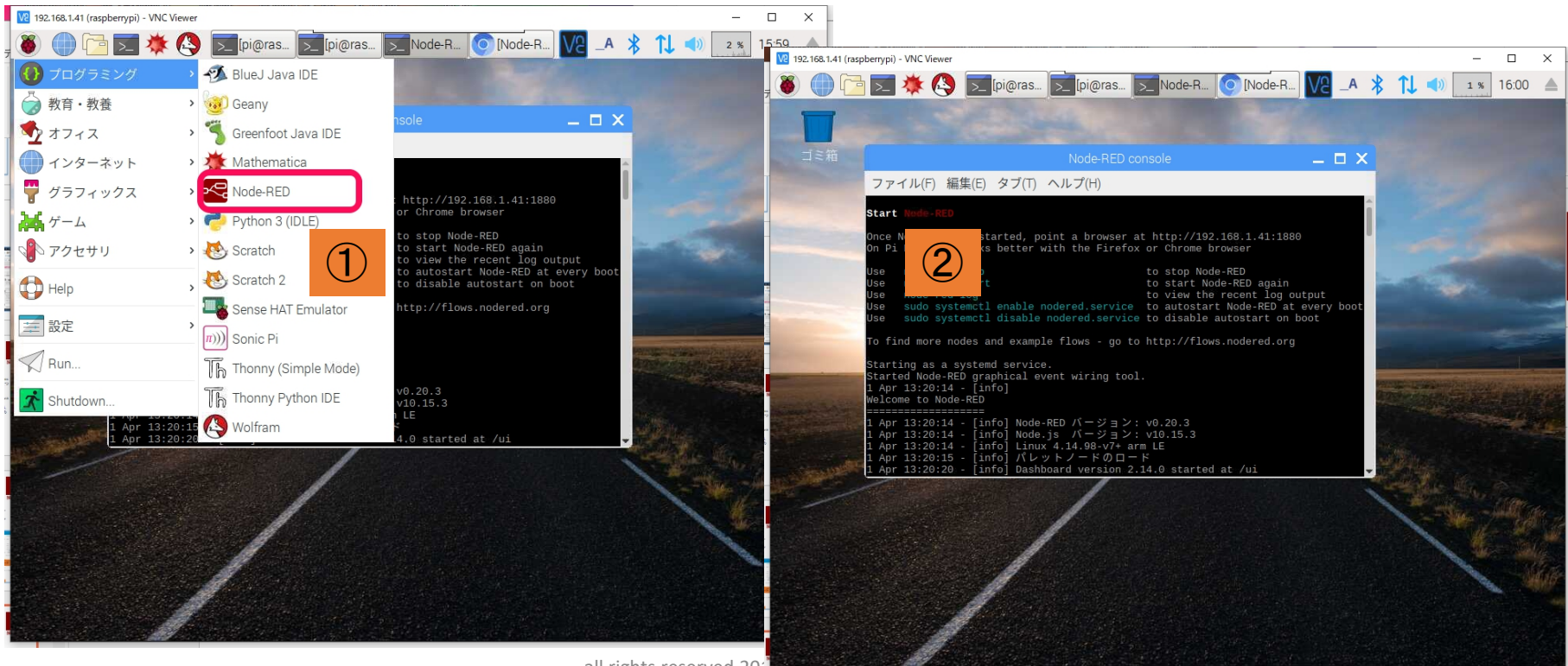
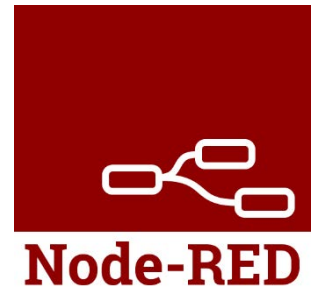
## ②. Node RED起動

### Node REDインストール

- 必要なものは、インストール済

### Node RED起動

- ① メニュー>プログラム>Node-RED
- ② コンソールが立ち上がり、自動起動します。





# Node RED開発

## ②. Node RED起動

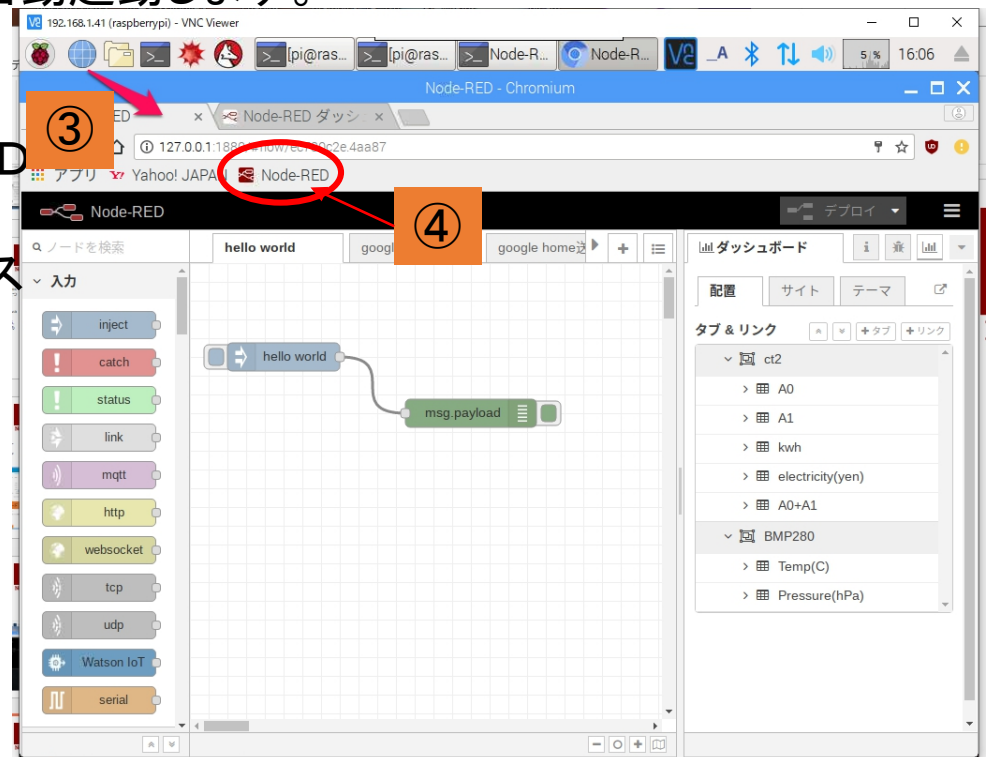
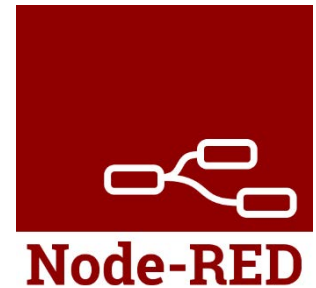
Node REDインストール

- 必要なものは、インストール済

Node RED起動

- ① メニュー＞プログラム＞Node-RED
- ② コンソールが立ち上がり、自動起動します。
- ③ Web画面立ち上げ
- ④ Chromeが立ち上がったら  
ショートカットのNode-RED  
を押します。又は

<http://127.0.0.1:1880>にアクセス  
これで立ち上げ完了

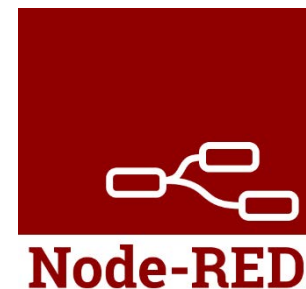
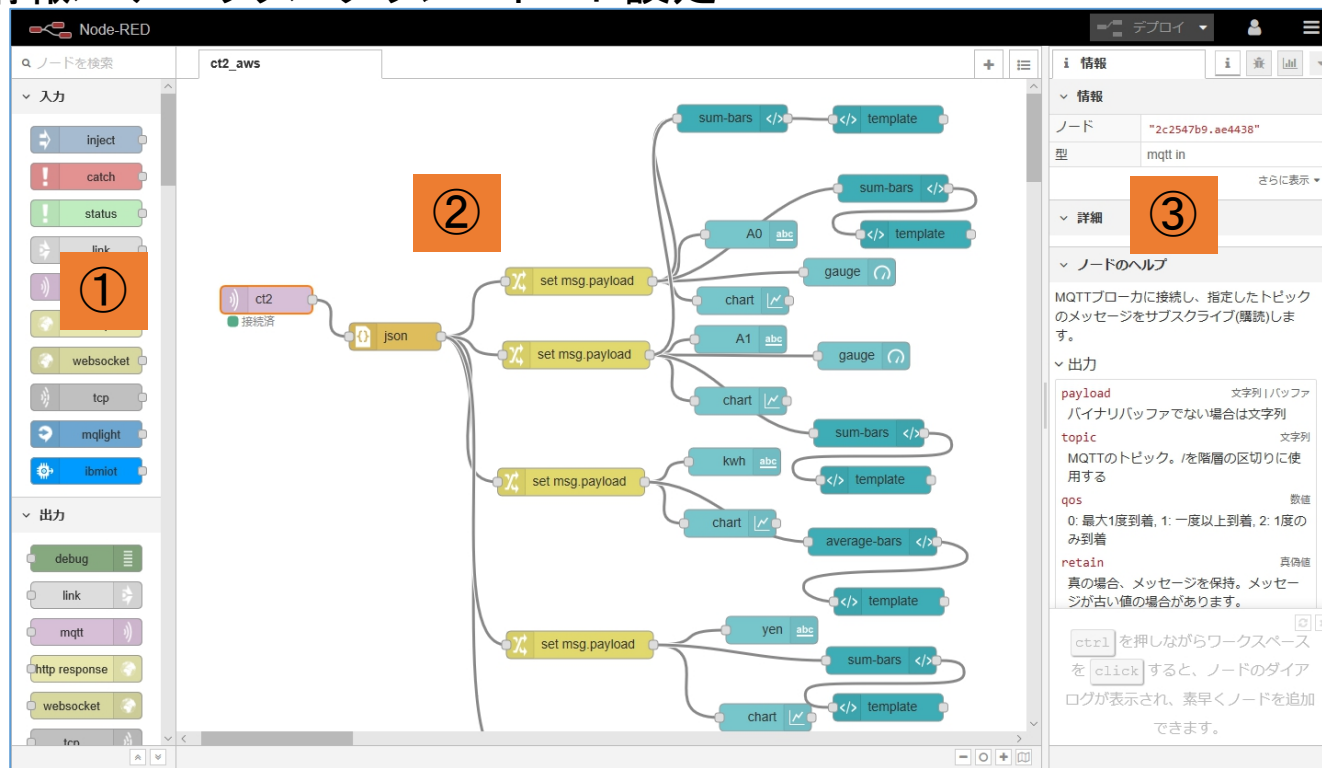


# Node RED開発

## ③. Node RED画面説明

### Node REDトップ画面

- ① パレット(ノード選択用アイコン):フローディタで作成するMQTTなどのノードなどのアイコンがあります。
- ② フローエディタ:作業領域です。ダブルクリックで設定画面になります
- ③ 情報/デバック/ダッシュボード設定

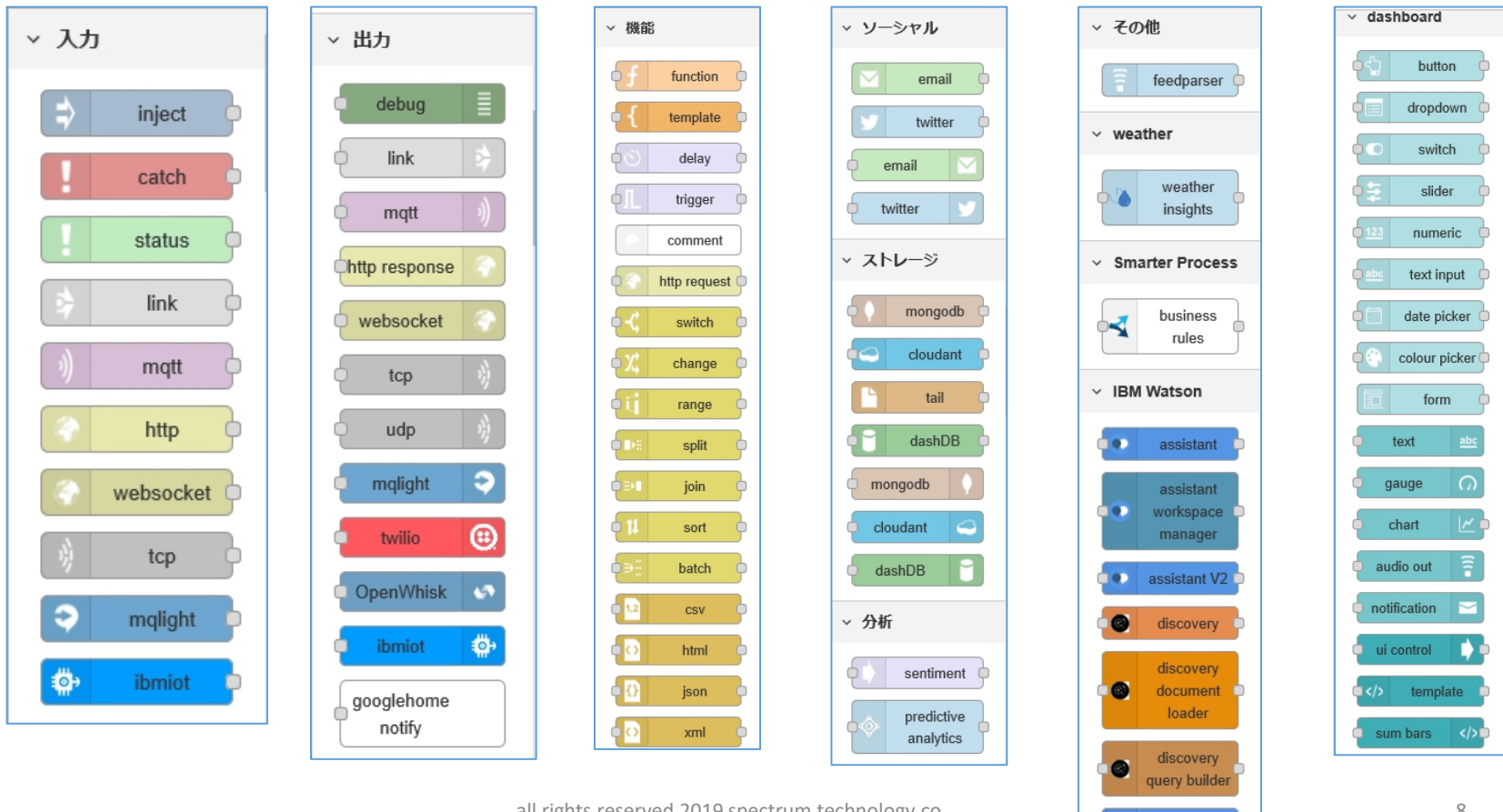
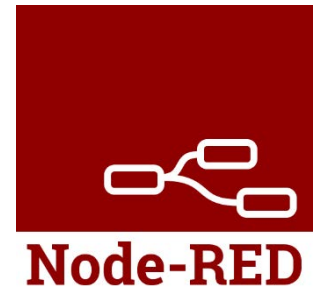


## Node RED開発

### ③. Node RED画面説明

Node REDトップ画面

① パレット: 入出力、機能、ダッシュボードなどから選択して設定。



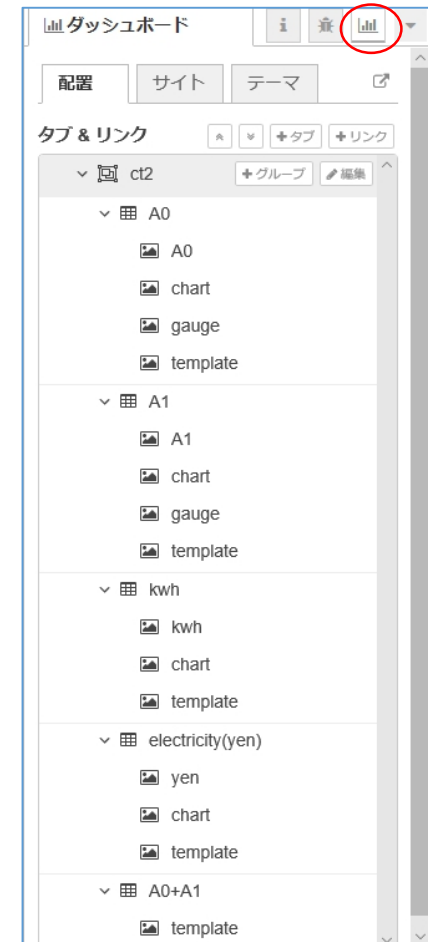
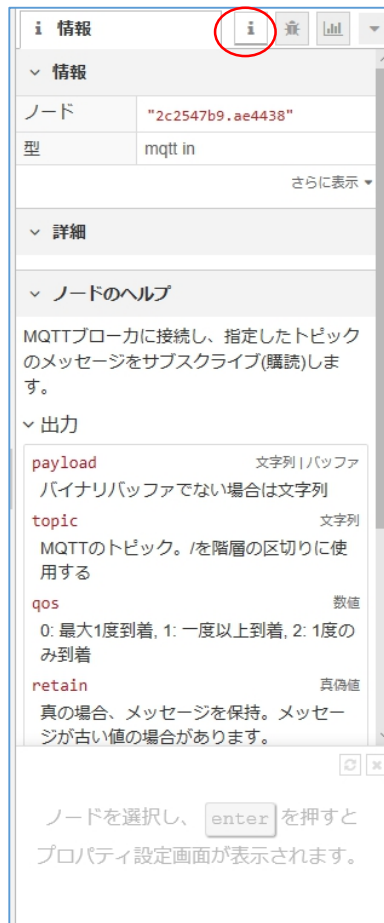
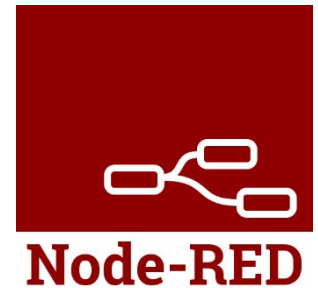


## Node RED開発

### ③. Node RED画面説明

#### Node REDトップ画面

#### ③ 情報／デバック／ダッシュボード設定



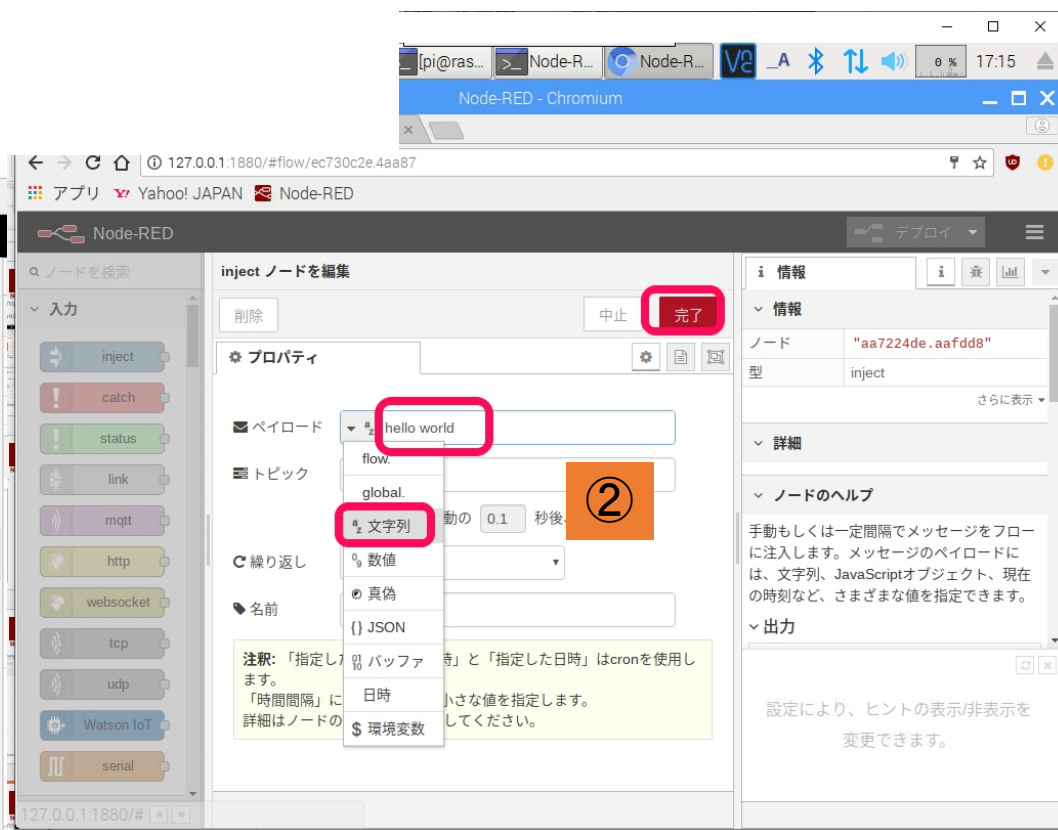
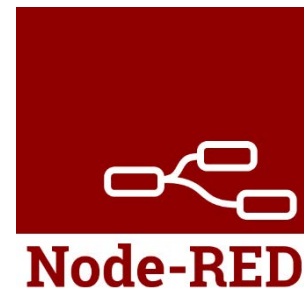
## Node RED開発

### ④. Node RED事例

Hello world 作成

- ① 入力>injectをドラッグ、ダブルクリック
- ② 文字列>Hello worldと入力し完了
- ③ 出力>debug
- ④ 確認のみ完了
- ⑤ デイプロイ: 完成

Node RED内でメッセージ  
の送受信を行います

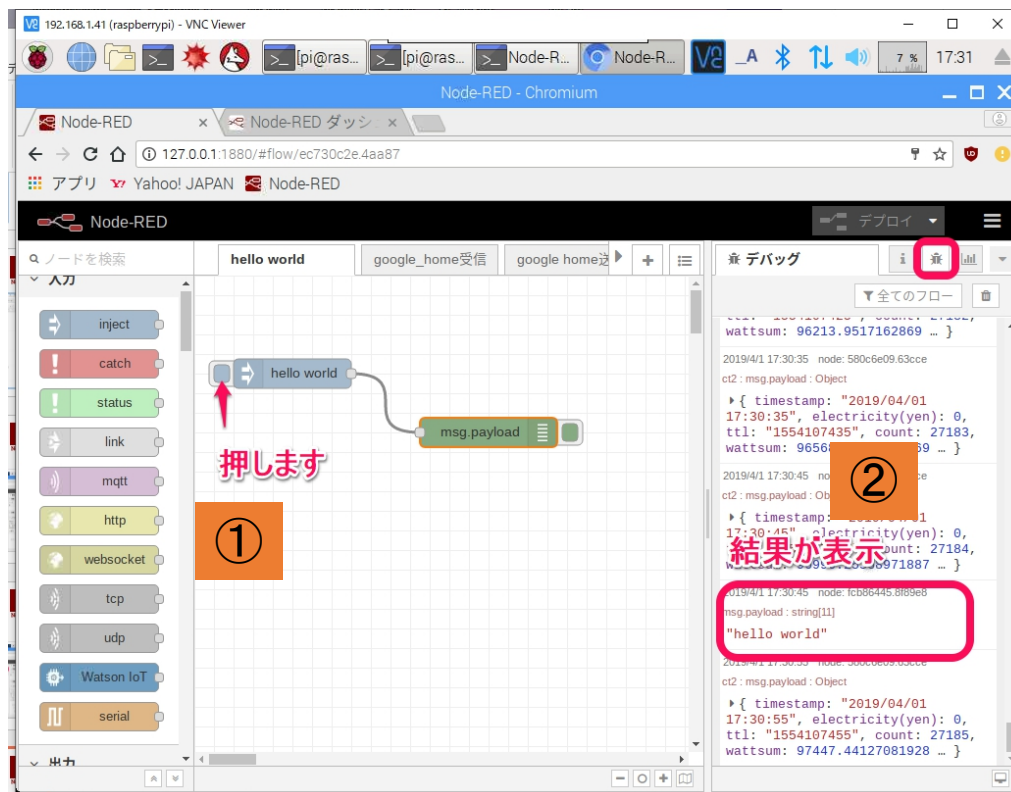
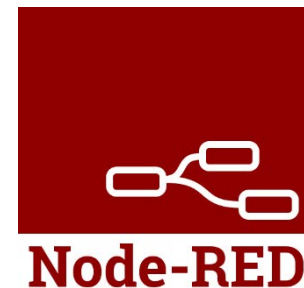


## Node RED開発

### ④. Node RED事例

Hello world 試験

- ① Injectの左を押します
- ② デバッグ画面に結果が表示:完了



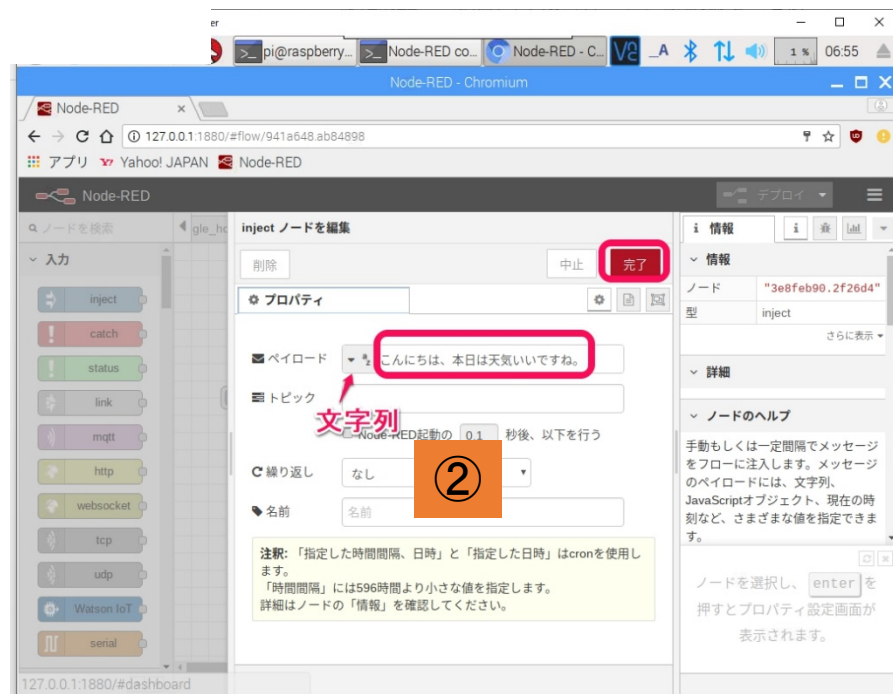
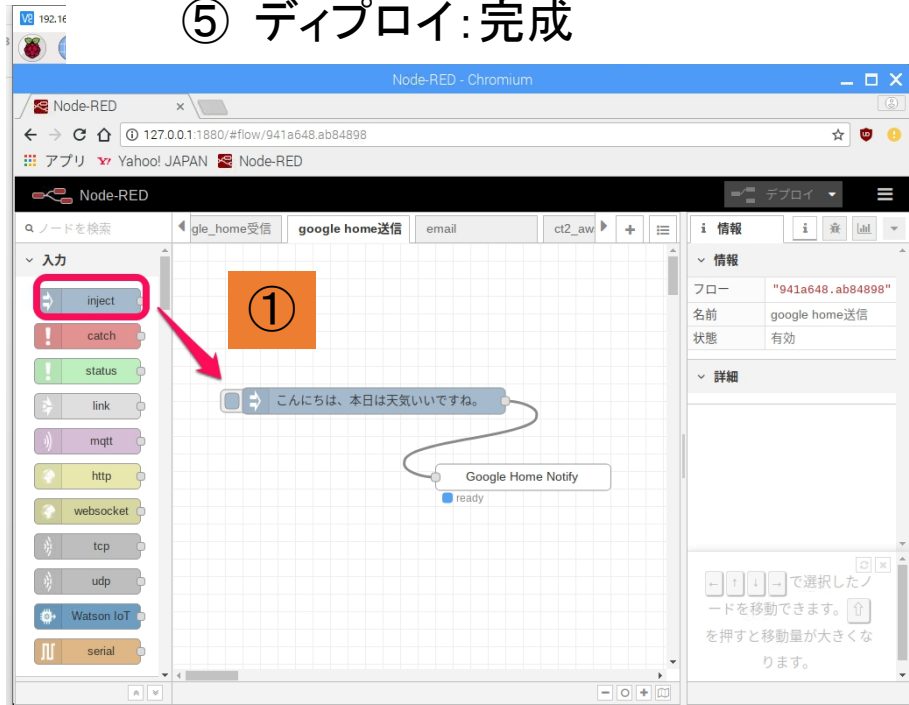
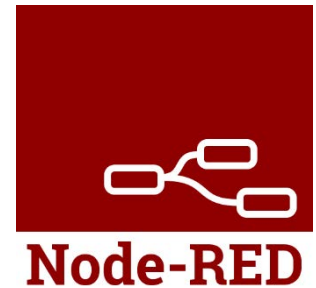
## Node RED開発

### ④. Node RED事例

Google home送信 作成

- ① 入力>injectをドラッグ、ダブルクリック
- ② 文字列>“こんにちは”などと入力し完了
- ③ 出力>Google home notifyをドラッグ、ダブルクリック
- ④ IPアドレス入力>言語選択>完了
- ⑤ デイプロイ:完成

Node REDから、Google homeにメッセージの送信します



## Node RED開発

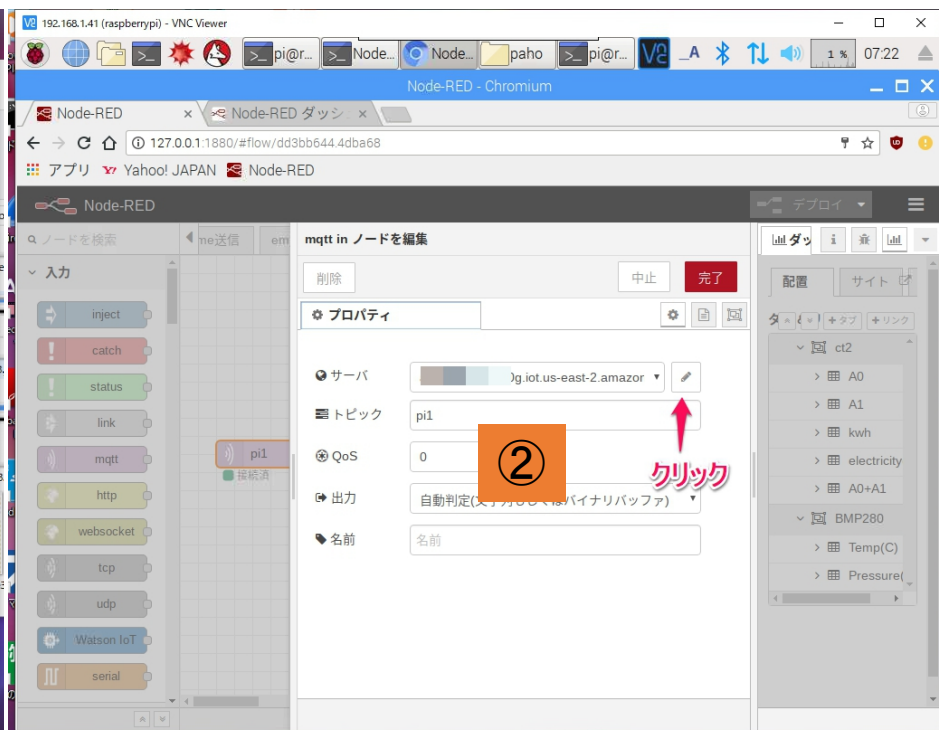
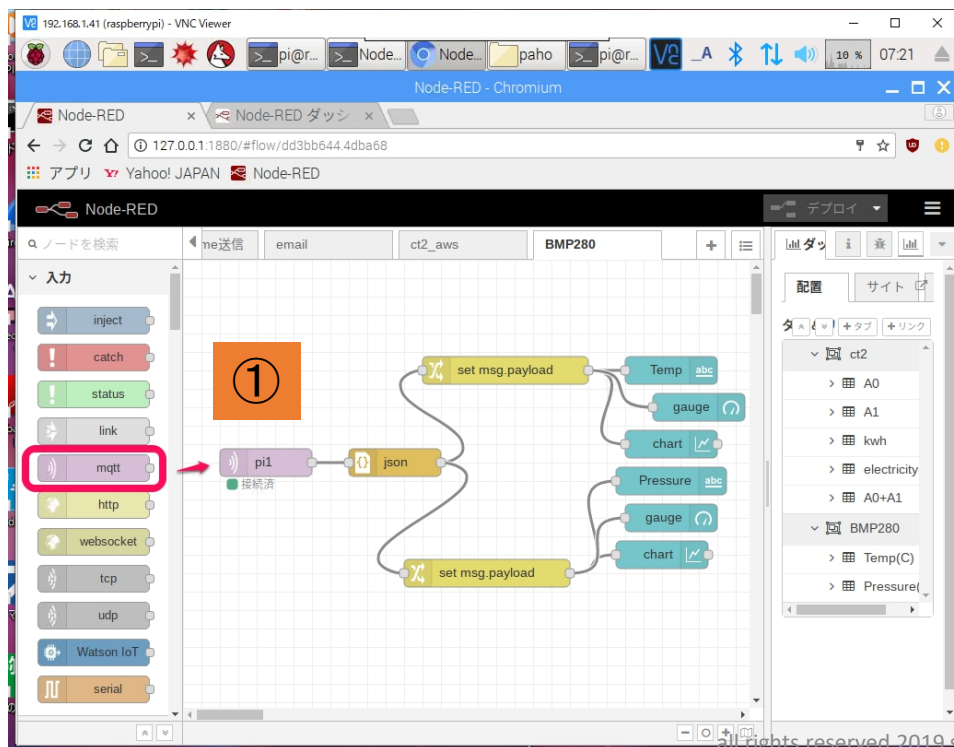
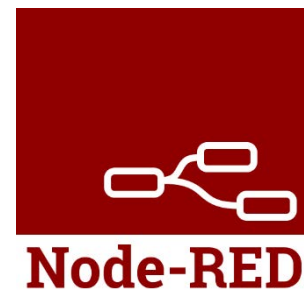
### ④. Node RED事例

BMP280(温度、気圧) 作成

- ① 入力>MQTTをドラッグ、ダブルクリック
- ② サーバ設定

AWS IoT開発

⑥センサ情報リアルタイム表示のNode REDでの表示





## Node RED開発

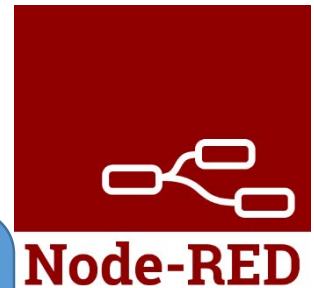
### ④. Node RED事例

BMP280(温度、気圧) 表示

①⑦ Dashboard>配置確認

①⑧ Dashboard表示

AWS IoTリアルタイム表示を行うためには、pythonのpublishを行ってください。  
pi@raspberrypi:~/Documents/paho  
\$ **python bmp280\_mqtt\_publish\_json.py**



The left screenshot shows the Node-RED editor interface. A flow is visible with a 'pi1' node connected to a 'json' node, which then connects to 'set msg.payload' nodes. The right sidebar shows the 'Dashboard' tab selected, with a 'BMP280' widget expanded, showing 'Temp(C)' and 'Pressure(hPa)' options. A red arrow points from the 'BMP280' widget to the 'Temp(C)' option, with a red box around it and the number 17. Another red arrow points from the 'Temp(C)' option to the 'Pressure(hPa)' option, with a red box around it and the number 18.

The right screenshot shows the Node-RED dashboard. It displays two charts and two gauges. The first chart is labeled 'Temp(C)' and shows a line graph of temperature over time. The second chart is labeled 'Pressure(hPa)' and shows a line graph of pressure over time. Below each chart is a gauge. The temperature gauge shows a value of 19.5, and the pressure gauge shows a value of 1,003.2. A red arrow points from the 'Temp(C)' chart to the 'Pressure(hPa)' chart, with a red box around it and the number 18.

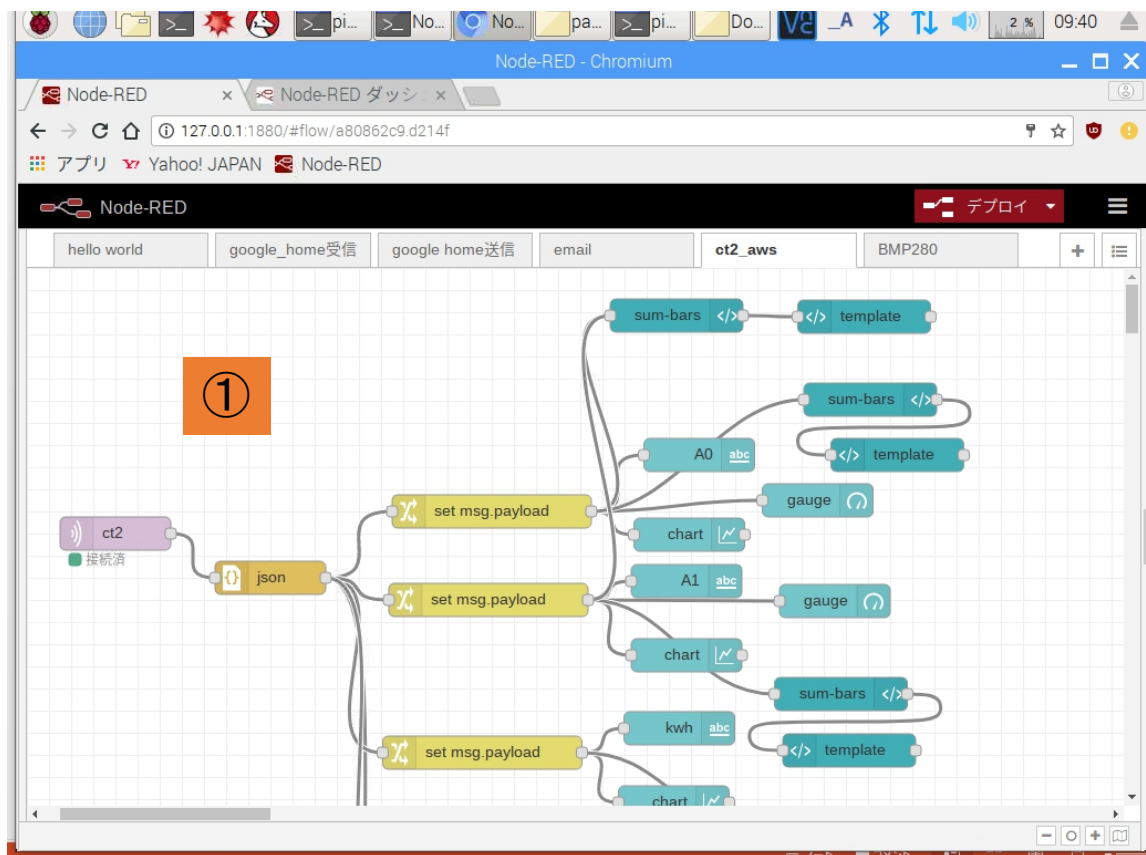
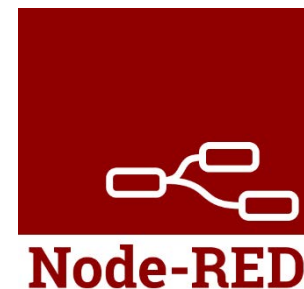
# Node RED開発

## ④. Node RED事例

CT2 AWS 作成

- ① フロー図: BMP280と同じ、一部グラフで積算、  
平均値表示を追加

AWS IoT開発  
電力計キットのNode RED  
での表示



サーバ接続先、証明書  
は、必ず各自の値に再  
設定してください

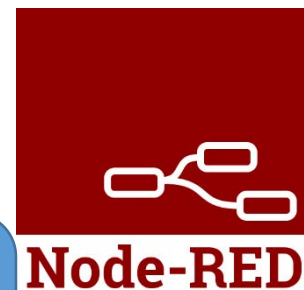
## Node RED開発

### ④. Node RED事例

CT2 AWS 作成

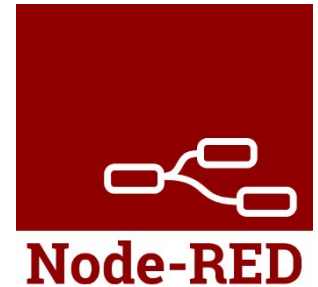
### ⑥ Dashboard表示

AWS IoTリアルタイム表示を行うためには、pythonのpublishを行ってください。  
pi@raspberrypi:~/Documents/ct \$ **python**  
**ct\_mqtt\_publish\_json.py**



## Node RED開発

### ⑤. Node RED管理



#### Node REDメニュー

- ① 表示: 画面の表示設定
- ② 読み込み: フローをインポートできます
- ③ 書き出し: フローをエクスポートできます
- ④ ノード検索: 設定したノードの検索
- ⑤ ノード設定: 全てのノード一覧と設定ができます
- ⑥ フロー: 新規、修正が可能
- ⑦ サブフロー: 新規など
- ⑧ パレット管理: パレットの削除、追加可能
- ⑨ 設定: 表示、キーボード、パレットのユーザ設定
- ⑩ ショートカットキー

