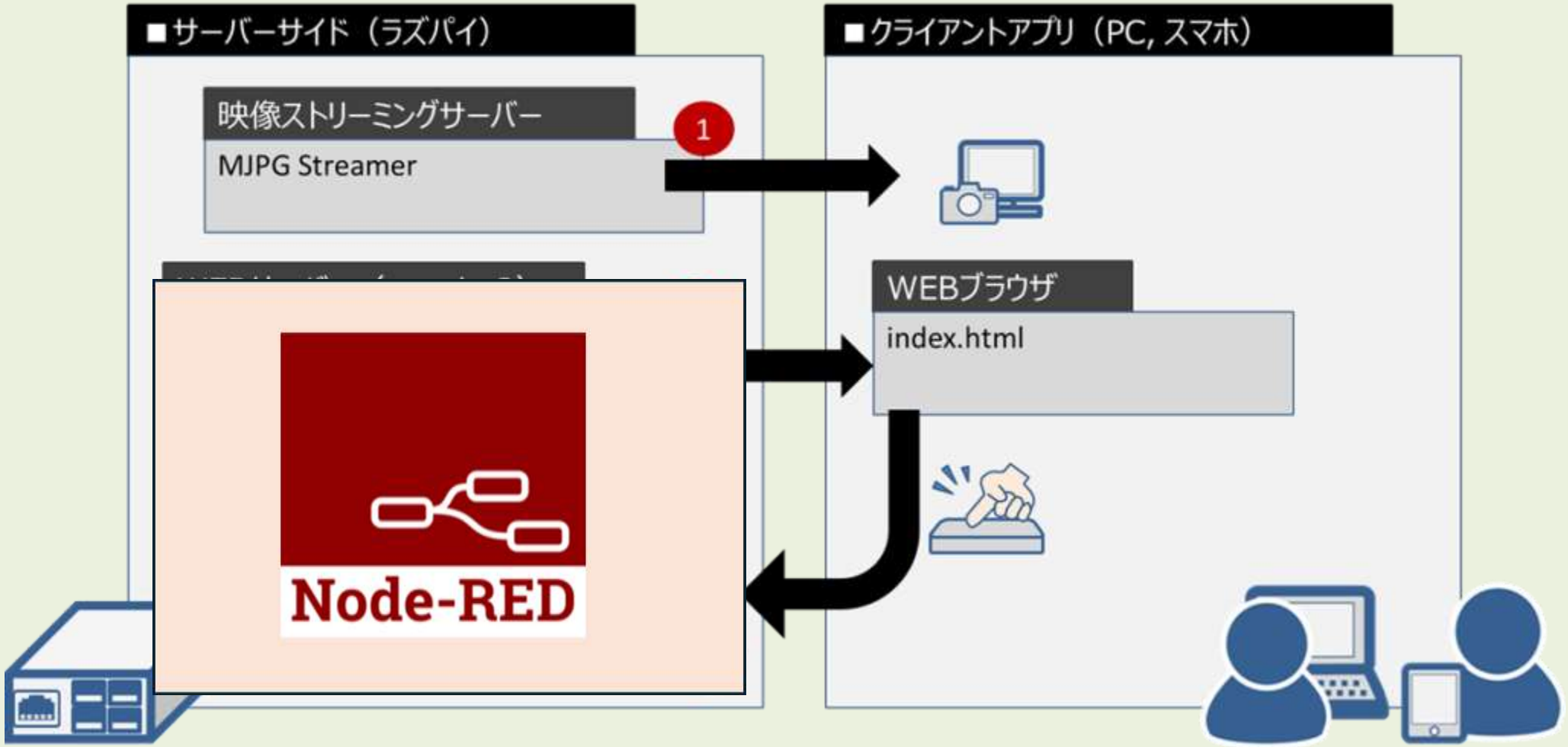
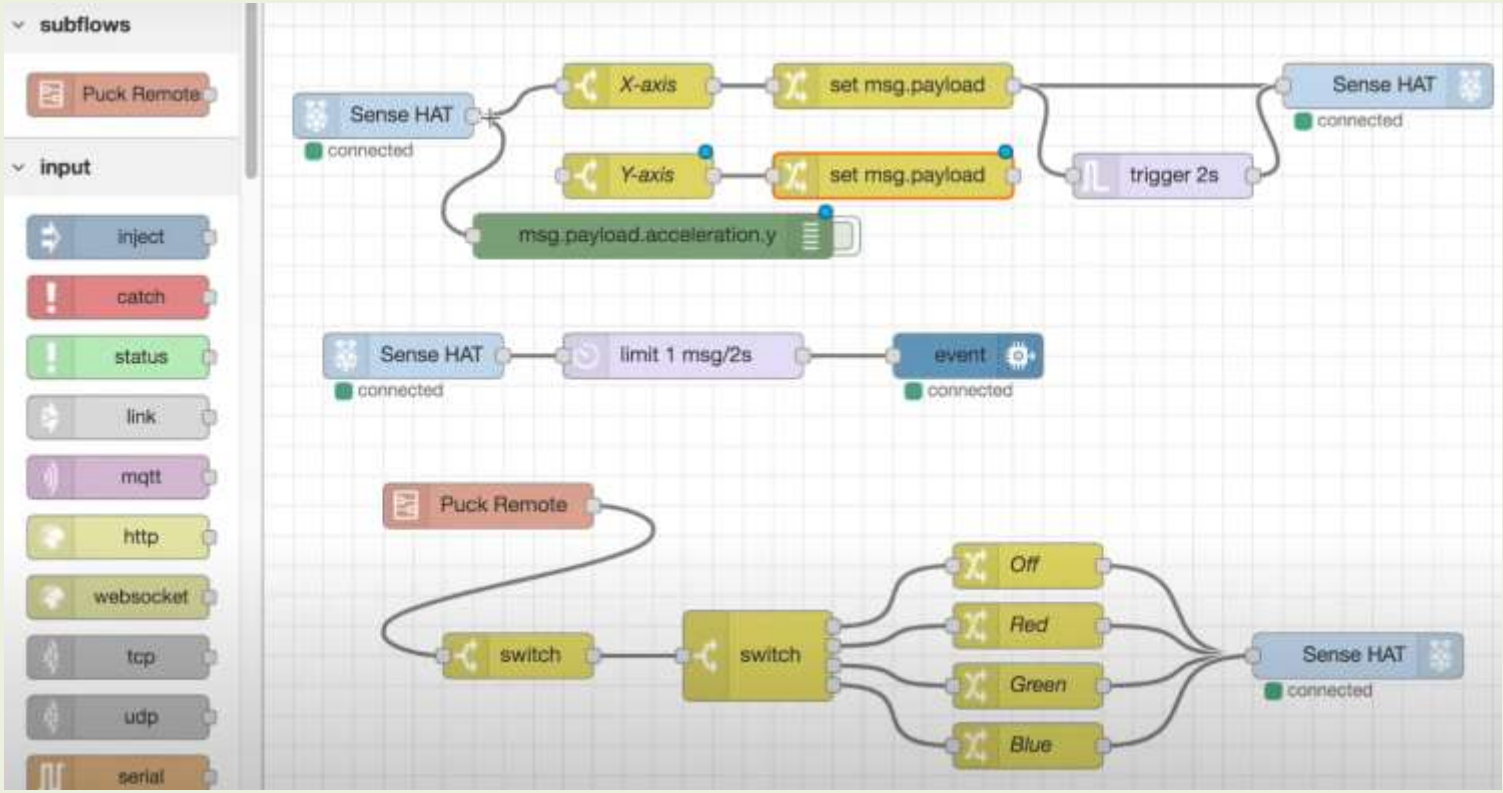
ラズタンク・操作ダッシュボード開発



ノード･レッド とは？



- IBMが開発
- ビジュアルプログラミング型の開発ツール
- WEBブラウザのエディタ
- IoT（ハードウェアデバイス ＋ API ＋ オンラインサービス を接続できる）
- JavaScript関数を作成できる

さっそく、みんなでやってみよう！

Node-RED を起動する

CoderDojo青梅のラズパイにはNode-REDがインストール済み

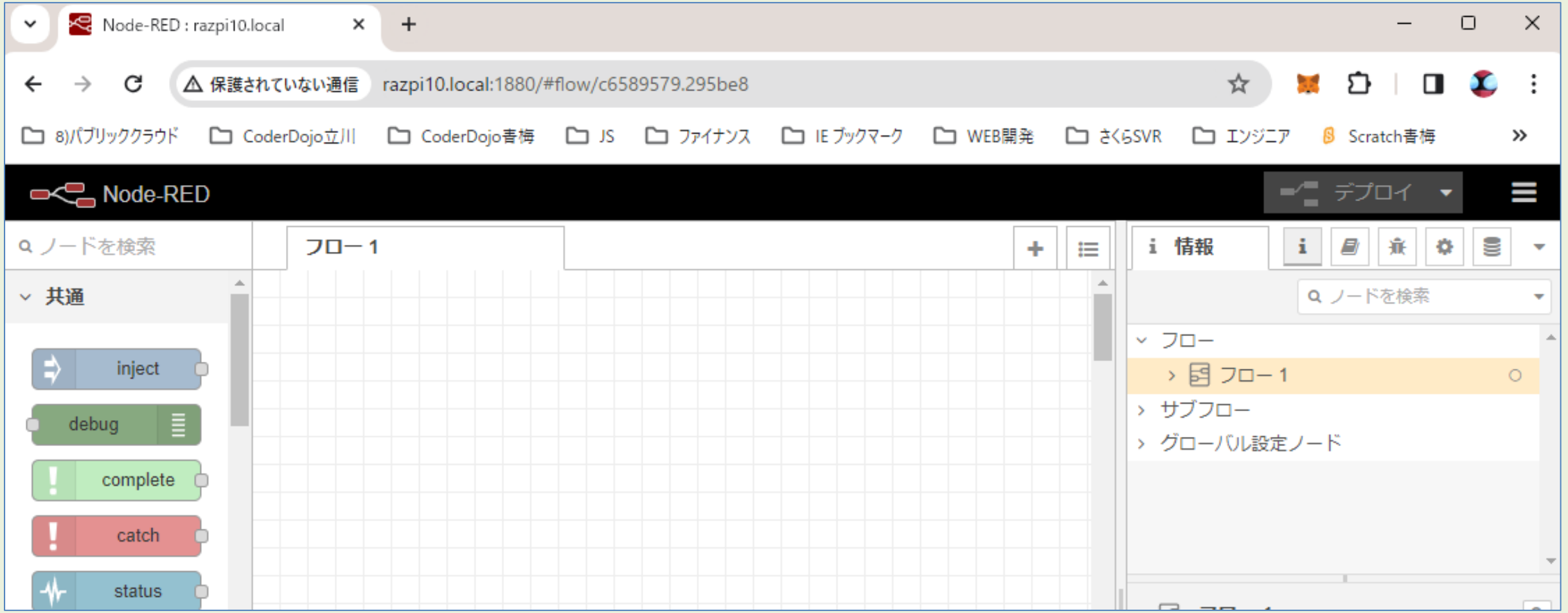
```
$ node-red-start
```

```
pi@razpi10:~ $ node-red-start

Start Node-RED

Once Node-RED has started, point a browser at http://192.168.1.63:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use   node-red-stop                         to stop Node-RED
Use   node-red-start                        to start Node-RED again
Use   node-red-log                          to view the recent log output
Use   sudo systemctl enable nodered.service  to autostart Node-RED at every boot
Use   sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
16 Jan 22:25:29 - [info]
Welcome to Node-RED
===================
16 Jan 22:25:29 - [info] Node-RED バージョン: v1.3.4
```

```
11 Feb 19:52:04 - [info] サーバは http://127.0.0.1:1880/ で実行中です
```

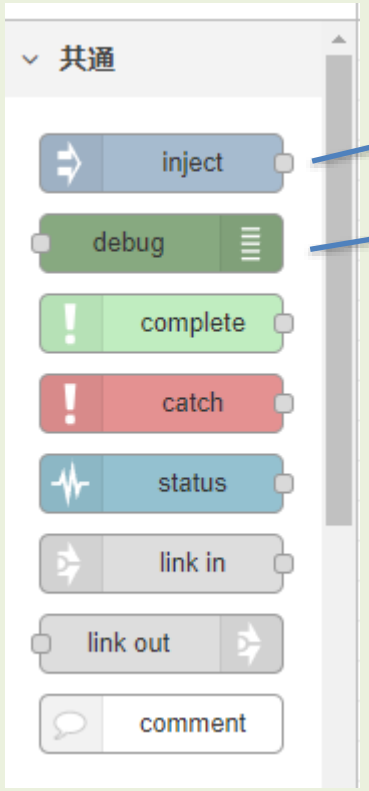
Node-RED にアクセス

同じネットワーク上のPCのWEBブラウザを起動
http://razpi01.local:1880/

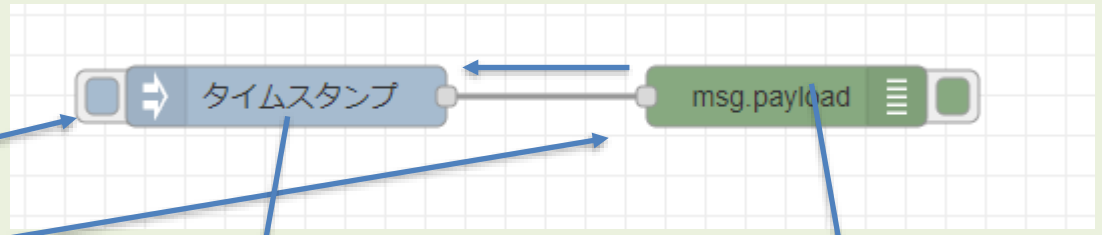


はじめての Node-RED プログラミング

はじめてのプログラミング 1



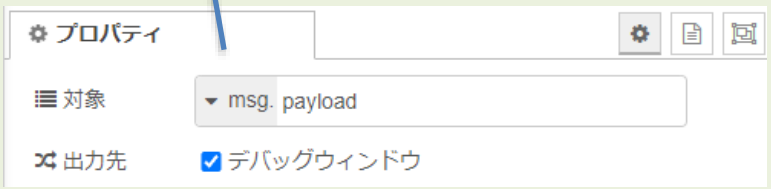
1



ダブルクリック

msg.payload = 日時

ダブルクリック



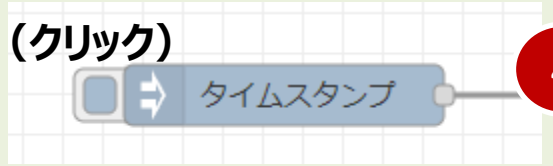
msg.payload をデバッグウィンドウに出力

3

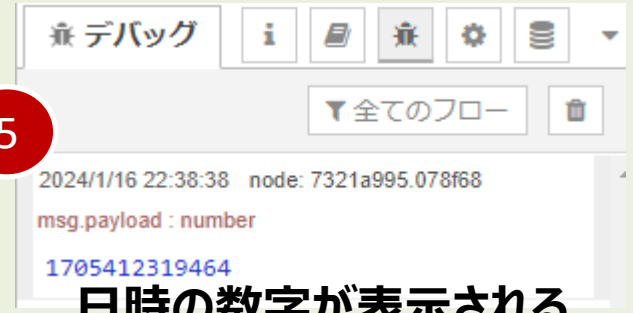
2

実行（クリック）

4



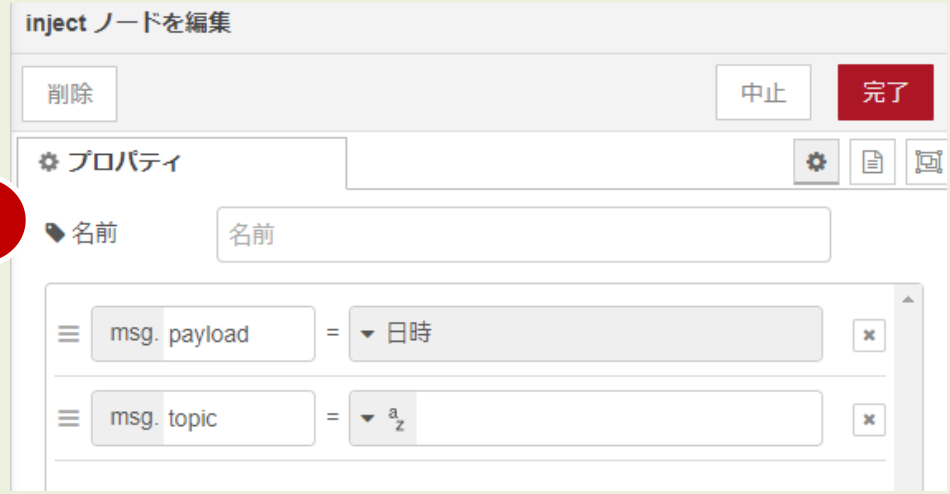
5



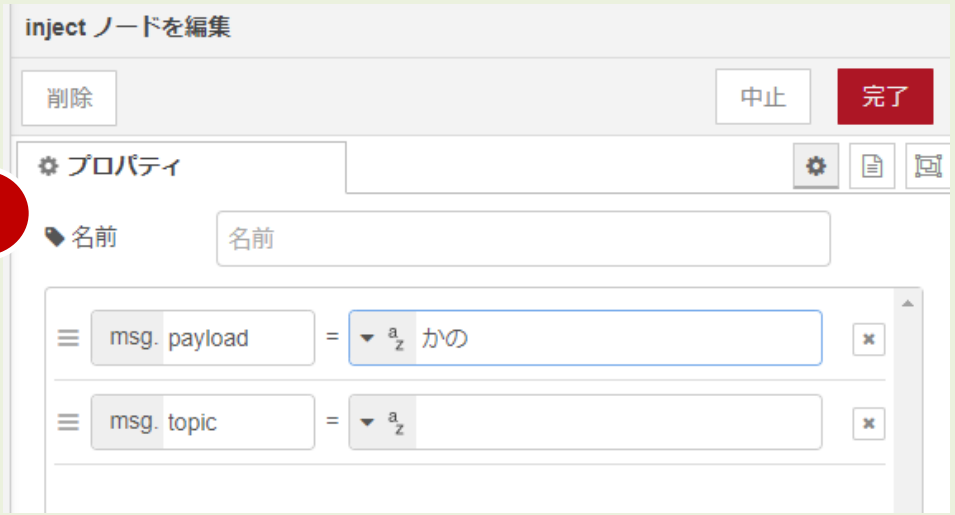
日時の数字が表示される

はじめてのプログラミング２（改造）

1

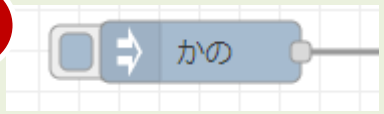


2



3

4

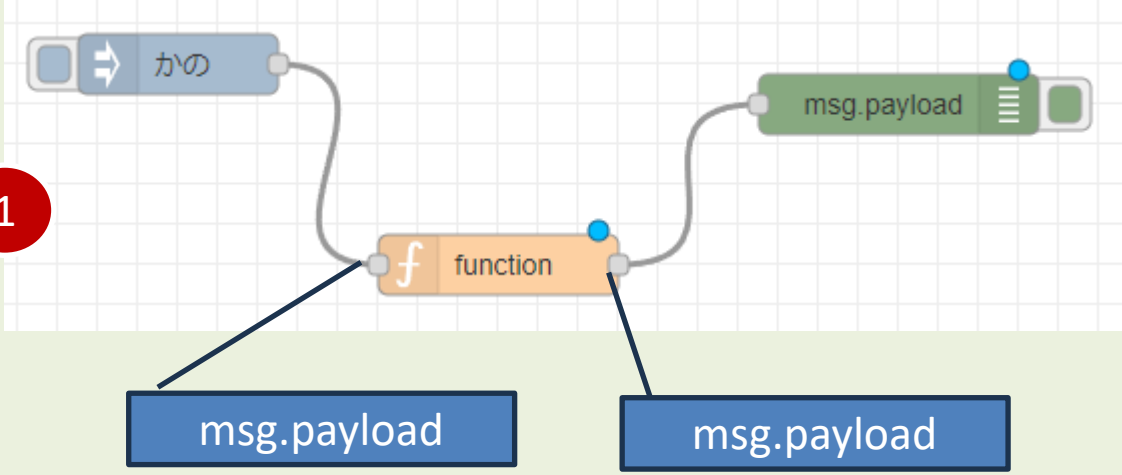


5

はじめてのプログラミング３（改造）

Functionノードに JavaScript コードを書く

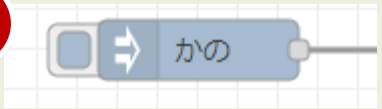
1



2

3

4

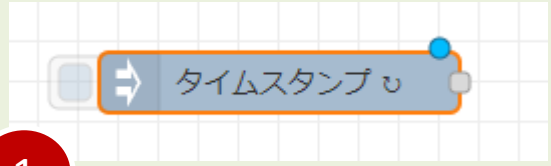


5

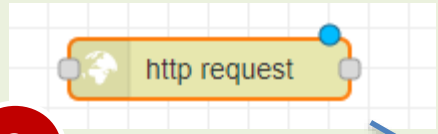
はじめての アプリ開発（防災アプリ）

はじめてのアプリ開発（防災アプリ）

Injectトリガー（ボタンを押す）



1



2

http request でWEBページのファイル「地震データ」を取得する

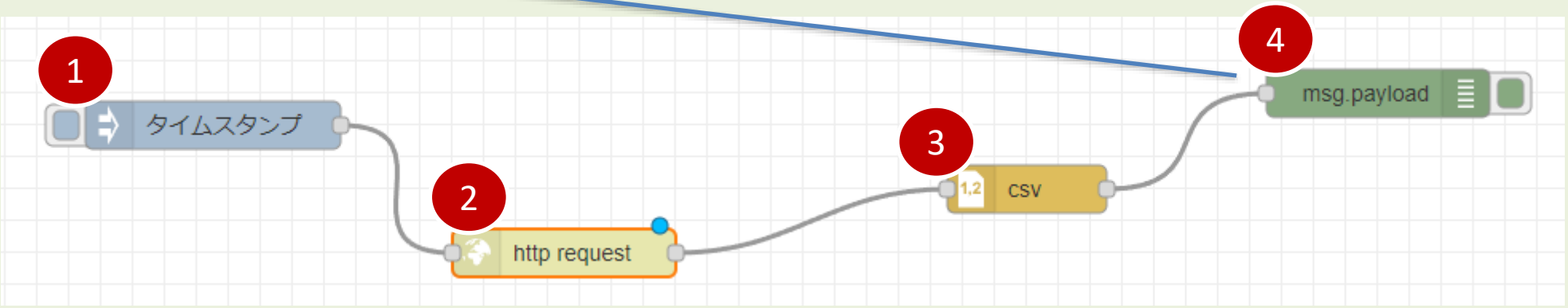
https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/significant_month.csv

はじめてのアプリ開発（防災アプリ）

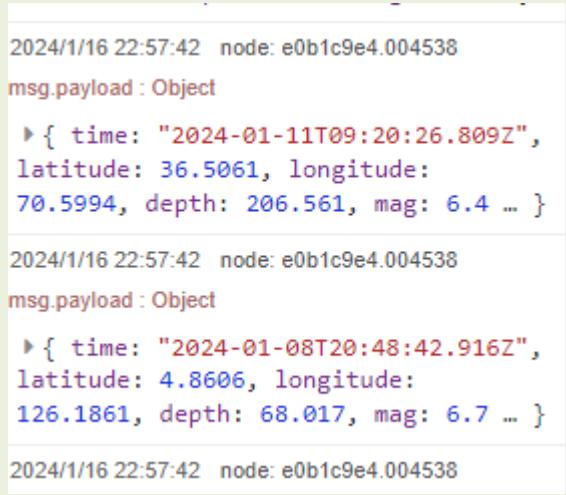
CSVノードを追加

はじめてのアプリ開発（防災アプリ）

debugノードを追加



1. Inject プログラムを実行する
2. WEBページからCSVファイルをダウンロード
3. CSVデータを整形
4. デバッグ画面に表示



はじめてのアプリ開発（防災アプリ）

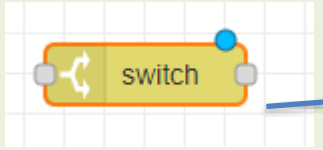
2024/1/16 23:11:39 node: e0b1c9e4.004538
msg.payload : Object
object
time: "2023-12-18T15:59:30.033Z"
latitude: 35.7377
longitude: 102.8059
depth: 10
mag: 5.9
magType: "mww"
nst: 157
gap: 12
dmin: 0.755
rms: 0.79
net: "us"
id: "us7000ljvg"
updated: "2024-01-14T15:06:24.649Z"
place: "39 km WNW of Linxia Chengguanzhen, China"
type: "earthquake"
horizontalError: 5.43
depthError: 1.757

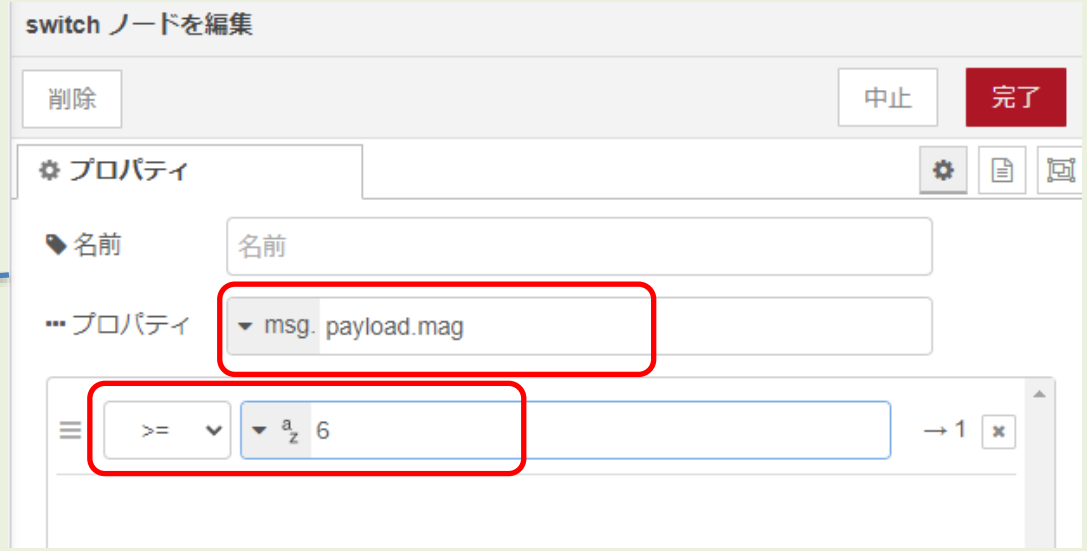
2023年12月18日　15時59分
緯度 latitude
経度 longitude
深さ depth
マグニチュード mag

google map
検索: 35.7377, 102.8059

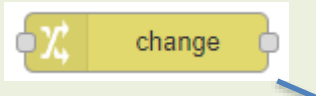
はじめてのアプリ開発（防災アプリ）

switchノードを追加

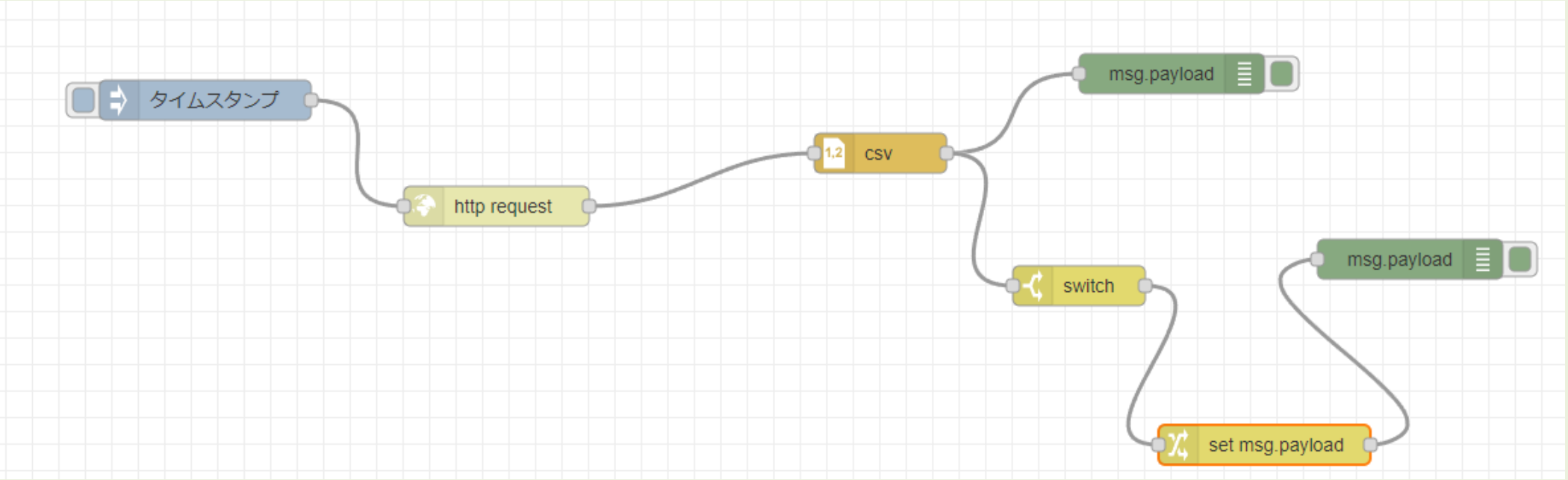


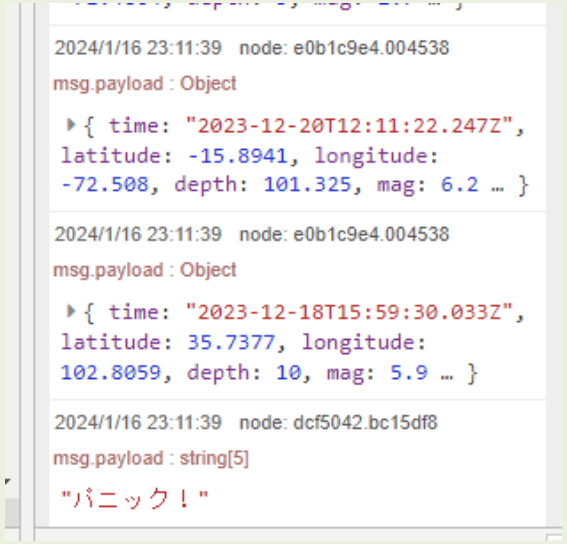


changeノードを追加

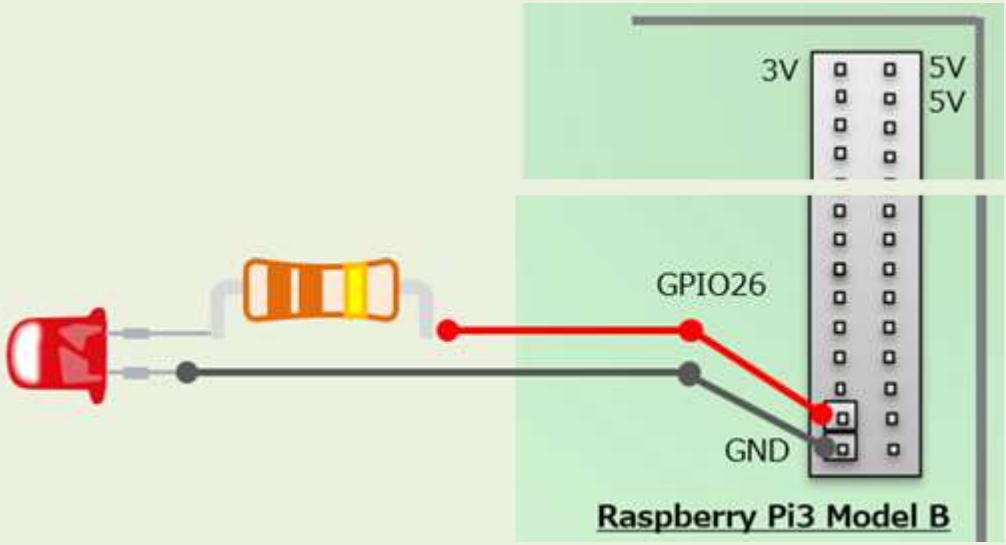


はじめてのアプリ開発（防災アプリ）

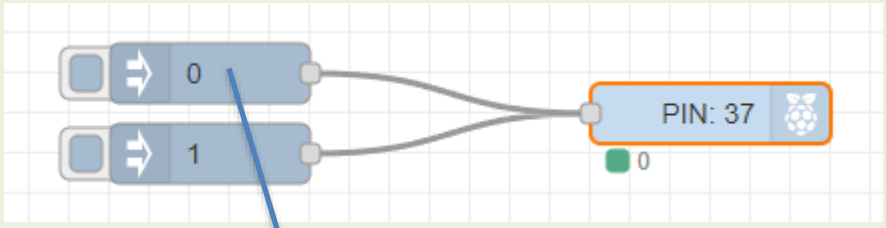


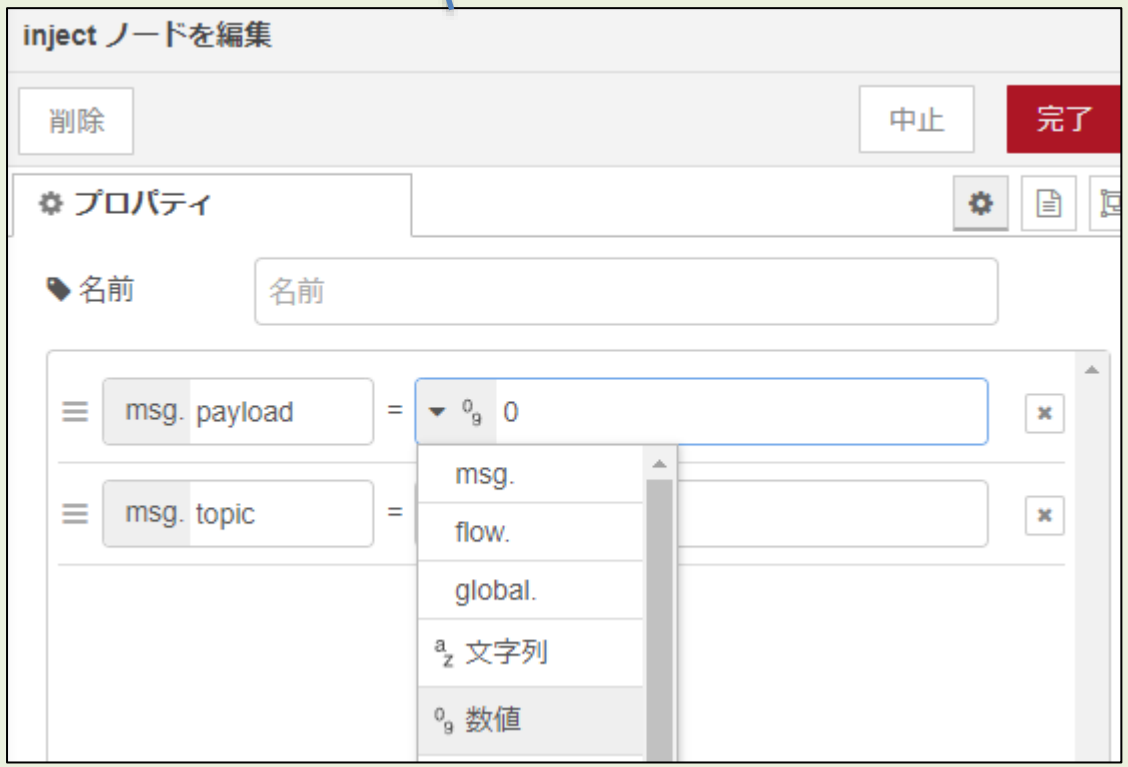


はじめての Node-RED 電子工作



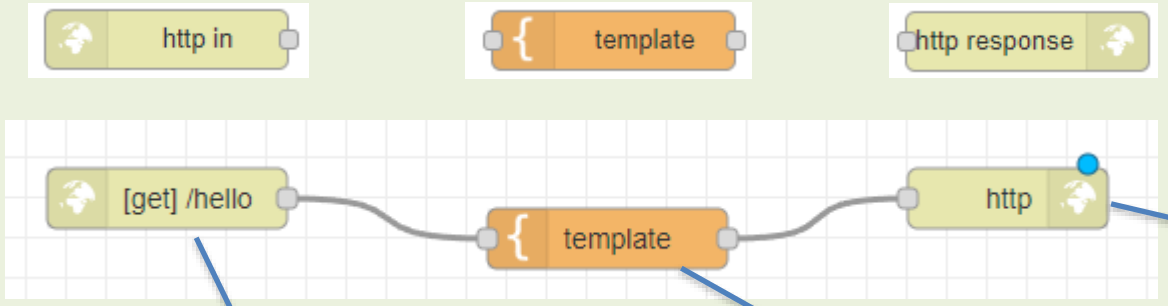
はじめての Node-RED 電子工作

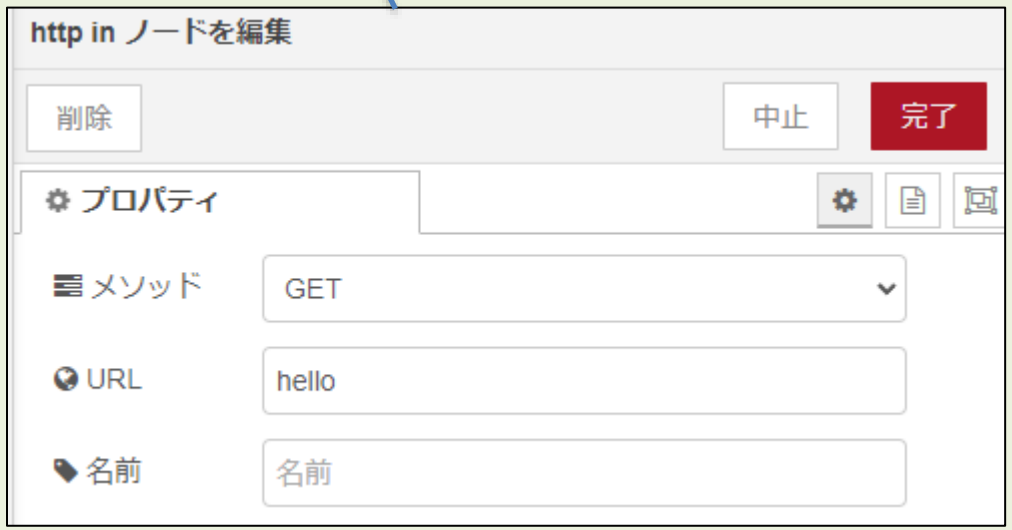




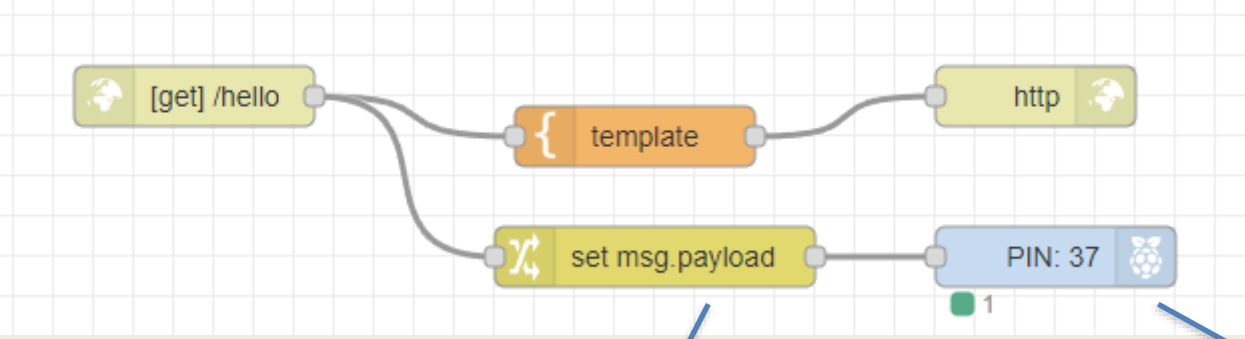
WEBページ（HTTPエンドポイント作成）

WEBページ（HTTPエンドポイント作成）

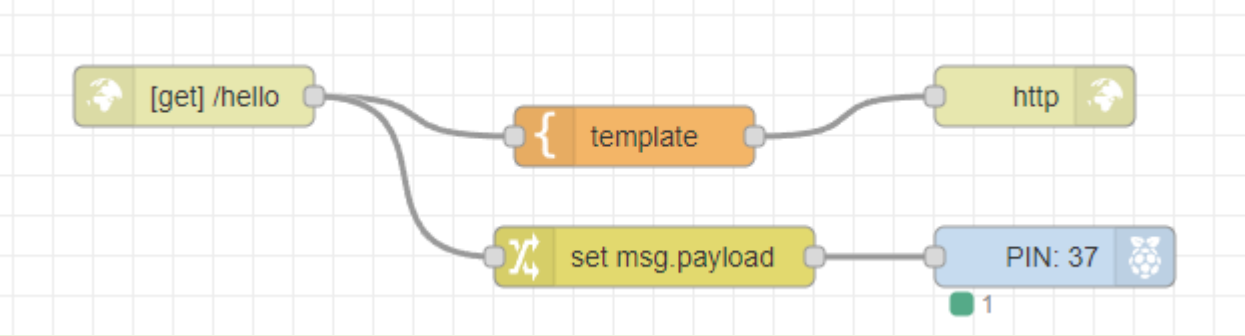


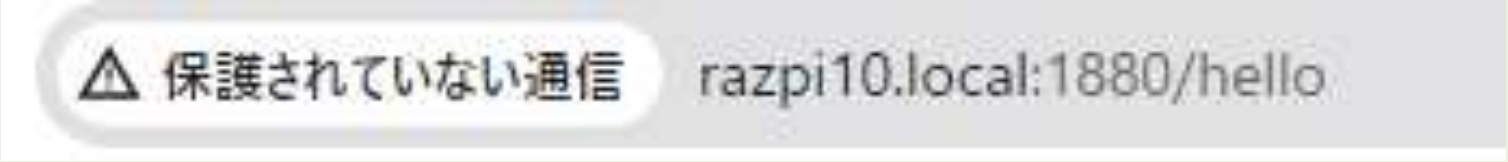


WEBページ（HTTPエンドポイント作成）



WEBページ（HTTPエンドポイント作成）

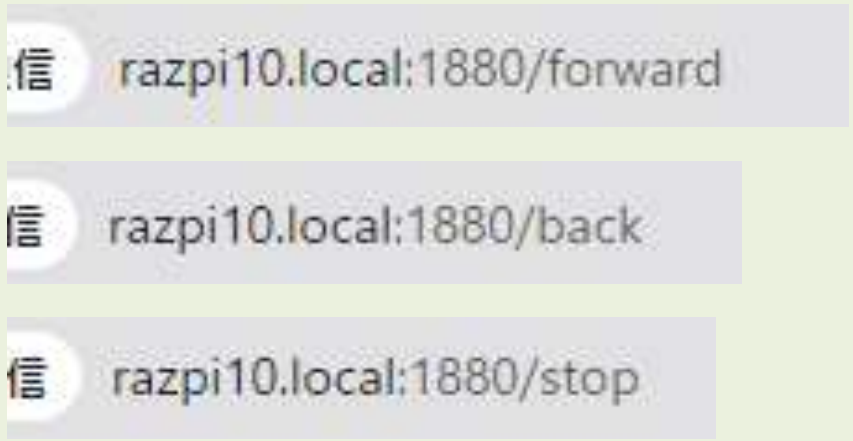




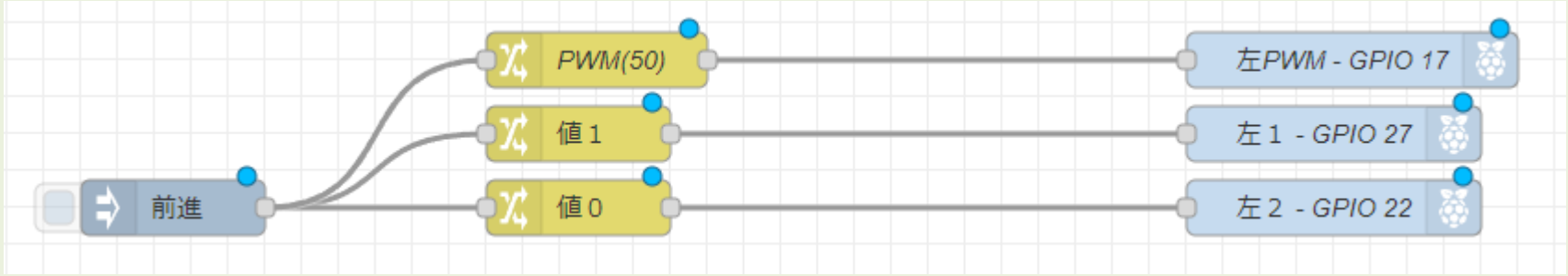
Hello World!

WEBブラウザに「Hello World!」が表示される。
同時にGPIO37に接続したLEDが点灯する。

ラズタンクのWEB-APIを作る！

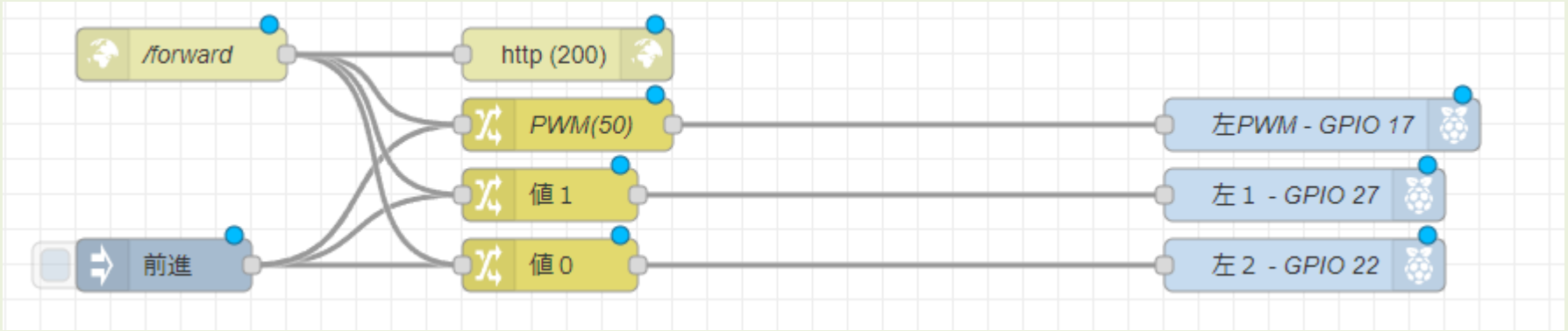


左側キャタピラー：前進駆動




```
# 初期設定(GPIO番号)
LeftPWM = 17
LeftIN1 = 27
LeftIN2 = 22

# 前進コマンド
def moveForward()
    GPIO.PWM( LeftPWM, 50 )
    GPIO.output( LeftIN1, 1 )
    GPIO.output( LeftIN2, 0 )
```

左側キャタピラー：前進駆動（WEB-API）




```
# 初期設定(GPIO番号)
LeftPWM = 17
LeftIN1 = 27
LeftIN2 = 22

# 前進コマンド
def moveForward()
    GPIO.PWM( LeftPWM, 50 )
    GPIO.output( LeftIN1, 1 )
    GPIO.output( LeftIN2, 0 )
```

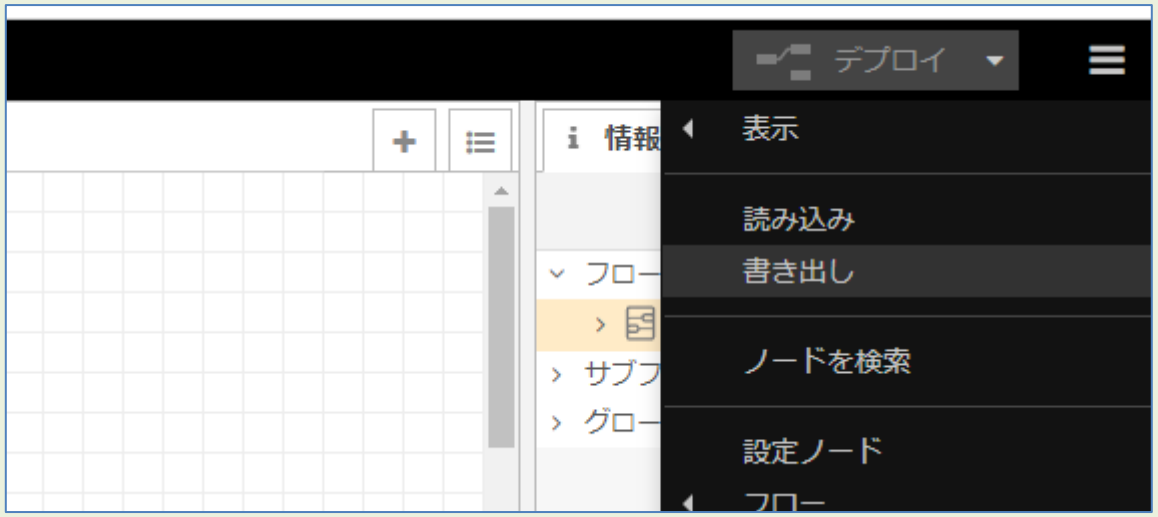
フローの保存先

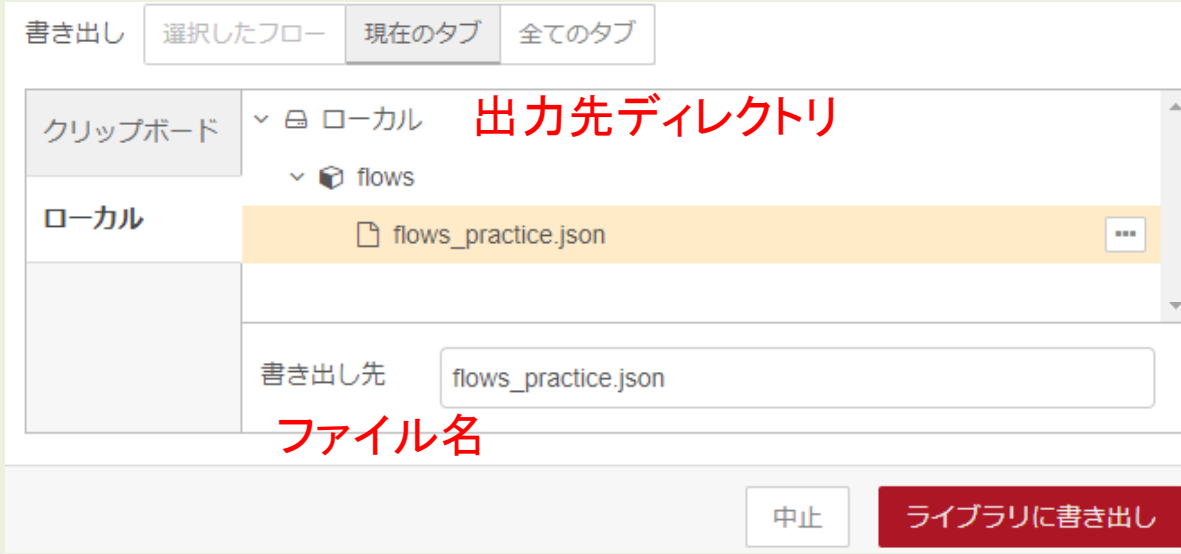
ラズパイ側にファイルを保存するには

デプロイすると作成されるファイル

```
$ cd .node-red
$ ls –l
-rw-r--r-- 1 pi pi  8539  1月 17 23:22 flows_razpi10.json
```

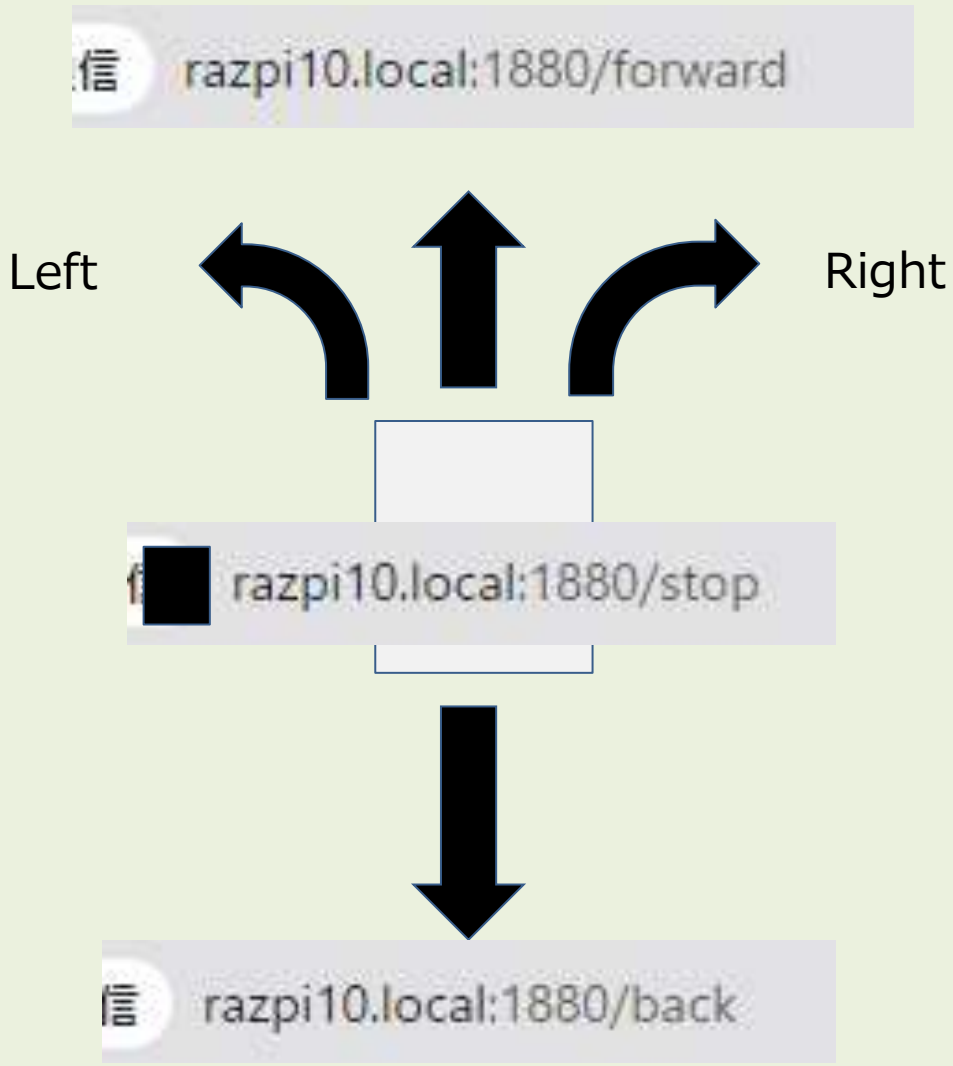
ファイルに出力する（書き出し）






```
[{"id":"3227a95.bfdda56","type":"tab","label":"フロー 1","disabled":false,"info":""},↓
{"id":"438bd2d8.1b960c","type":"inject","z":"3227a95.bfdda56","name":"","props":[{"p":"payload"},↓
{"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"前進","p
{"id":"4e59432c.56ec4c","type":"inject","z":"3227a95.bfdda56","name":"","props":[{"p":"payload"},{"p":"topic",
```

ここから先


razpi10.local:1880/forward
Left
Right
razpi10.local:1880/stop
razpi10.local:1880/back