第3日午後

製造業ITマイスター指導者育成プログラム
研修テキスト　実習用教材（第3日）
高度IT実装技術の習得2
（IoTセンサー実装実習）

製造業ITマイスター研修教材一覧

日	テーマ		教材
1	製造業IT導入ワークショップ	午前	IoTとシステムの基礎
		午後	製造業IT導入ワークショップ
2	高度IT実装技術の習得１	午前	IoTによるシステム開発入門
		午後	高度IT実装技術の習得１（ラズパイ＋見える化実習）
3	高度IT実装技術の習得２	午前	IoTによる生産管理入門
		午後	高度IT実装技術の習得２（IoTセンサー実装実習）
4	システム構築技術の習得１	午前	IoTによる在庫管理入門
		午後	システム構築技術の習得１（業務システムの基本パターン）
5	システム構築技術の習得２	午前	IoTによるデータ分析入門
		午後	システム構築技術の習得２（データ分析）
6	PBL１（事例企業調査）	午前	事例企業調査
		午前	事例企業の課題モデル化実習
7	PBL２ （課題の設定と解決策の提案）	午後	システム構築の実際
		午後	システム構築実習（１）課題の設定と解決策の提案
8	高度IT実装技術の適用	午前	IT経営の実践方法
		午後	システム構築実習（２）高度IT実装技術の適用
9	システム構築技術の適用	午前	情報システムセキュリティ基礎
			知財とオープン＆クローズ戦略
		午後	システム構築実習（３）システム構築技術の適用
10	筆記試験および成果発表会	午前	個人と組織の発展に繋がるキャリアデザイン講座
			（筆記試験）
		午後	（成果発表会）

前半5日間の進め方

午後の実習

■1日目　実習のための環境設定 ⇨ 課題発見ワークショップ

■2日目　デバイス信号のイン/アウト ⇨ センサデータの見える化

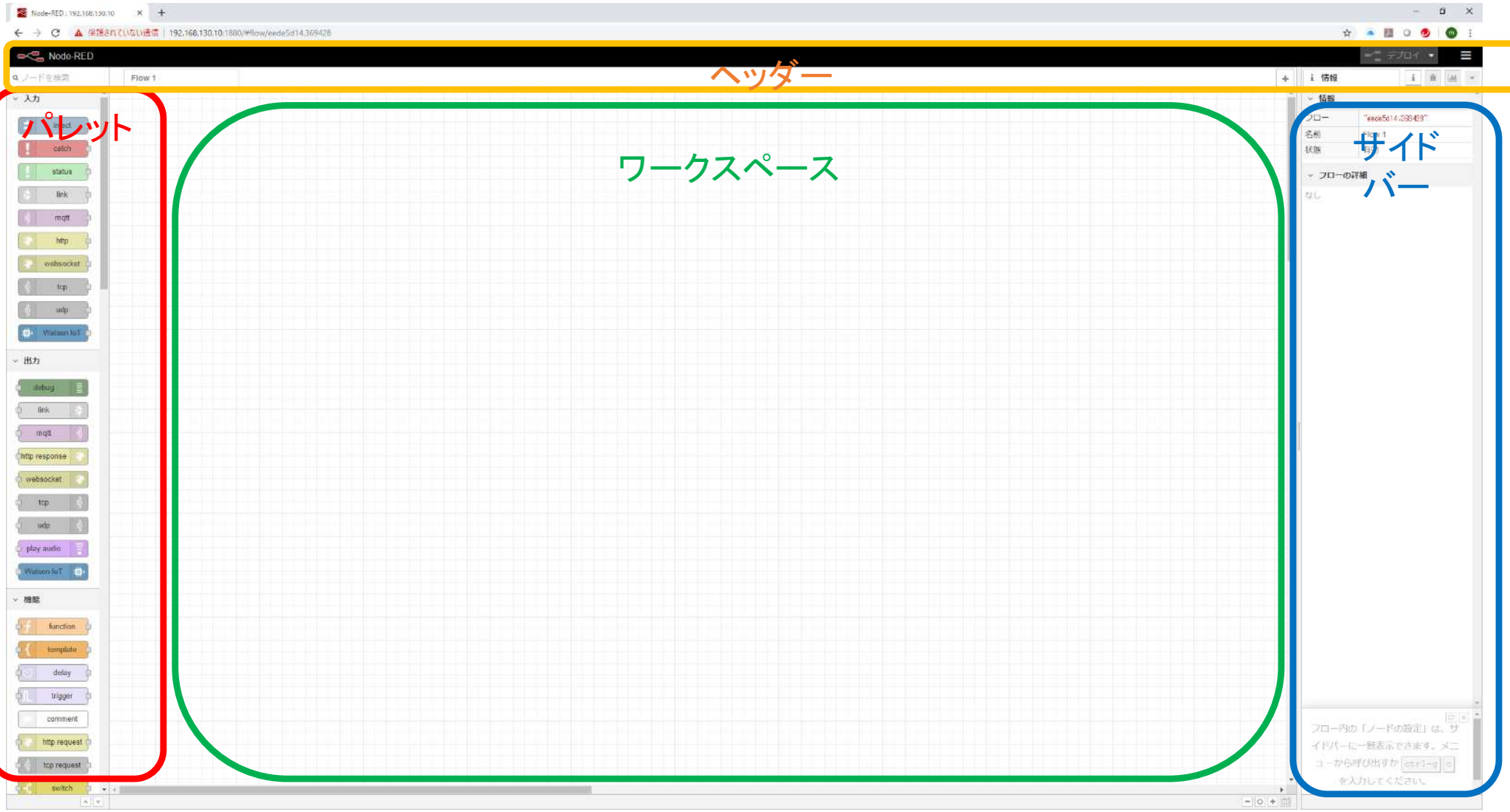
■3日目　メールとWebサーバ利活用 ⇨ 人感センサとカメラの利用

■4日目　業務システムの基本パターン ⇨ バーコードリーダとNFC

■5日目　データ分析続き ⇨ 工程進捗管理ボード

もくじ

Node-REDエディタ画面



ワークスペース

フロー

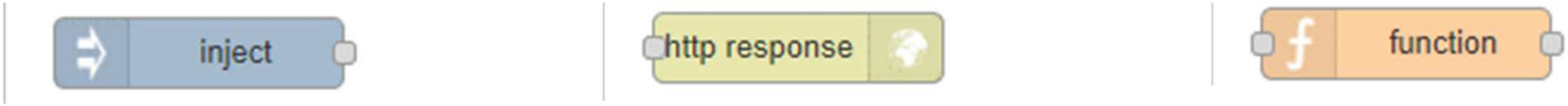
フローは、ワークスペースの1ページに相当するひと塊を指します。
フローの追加
新しいフローを追加するには右上の＋をクリックします。

フローの削除
フローのタブをダブルクリックしたら表示されるフロープロパティで
Deleteボタンをクリックします。

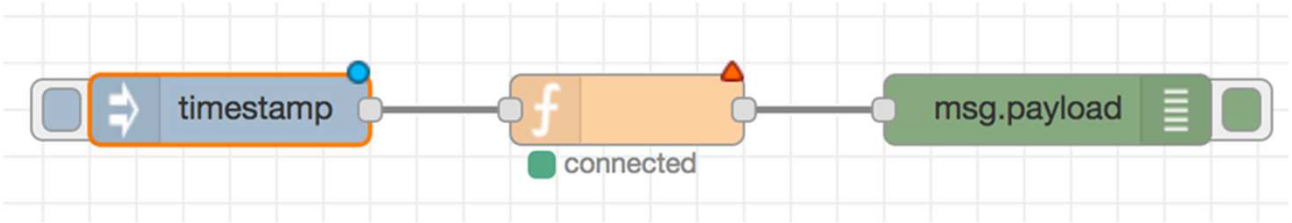
ノード

ノードは、アプリケーションを組み立てるためのソフトウエア部品です。
ノードのポートを介してワイヤーで結んで、アプリケーションを組み立てます。

ノードの例

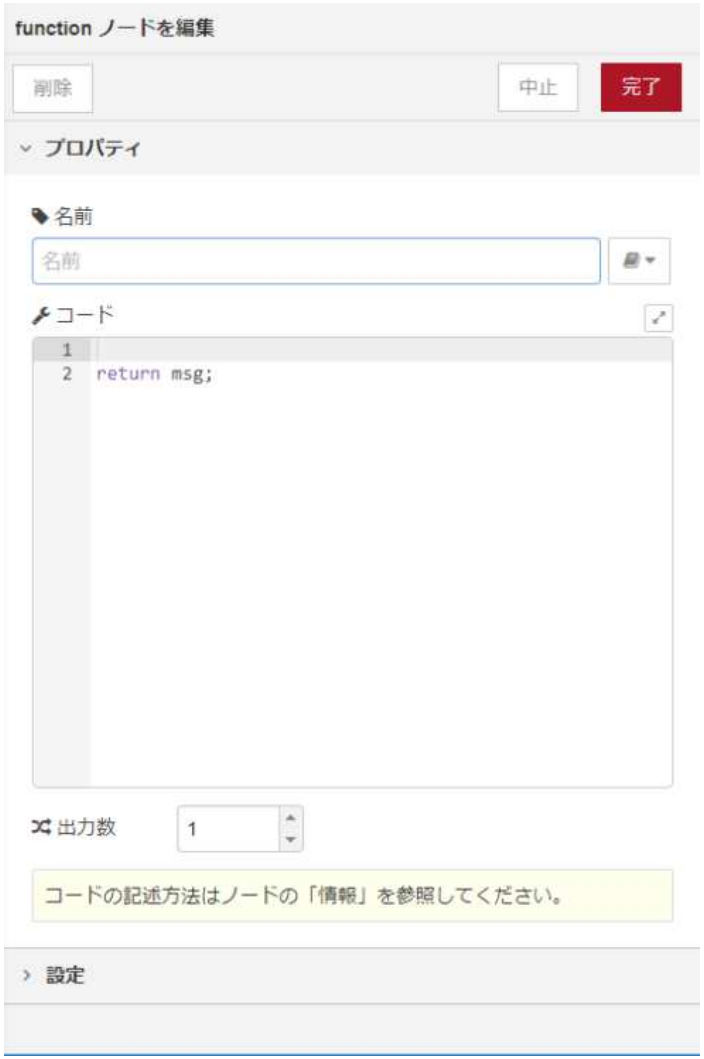


ノードにデプロイ（実行できるようにする）されていない変更箇所があると
ノードの上に青色の〇が表示されます。
ノードの設定にエラーがあった場合、赤い△が表示されます。



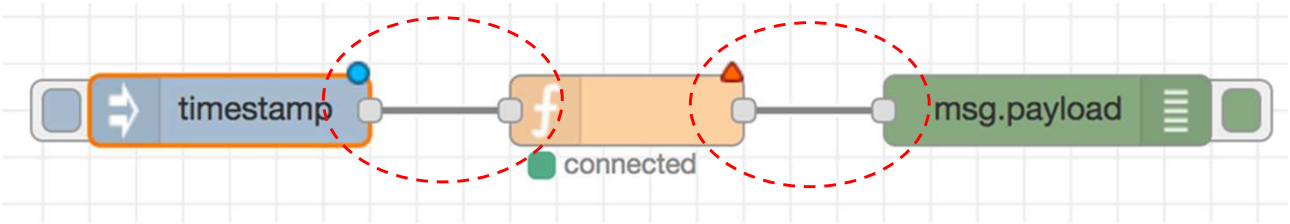
ノードでjavascriptを実行する

ノードの上でダブルクリックするとノードの設定を編集することができます。
Javascriptの処理プログラムを自前で記述することが可能です。



ワイヤー

ノード同士をつなぎ合わせるのがワイヤーです。
入力から出力までをつなぎ合わせて、必要なアプリケーションを組み立てます。



パレット

パレット

パレットには、インストール済みで利用可能なすべてのノードが表示されています。
ノードはカテゴリにまとめられています。

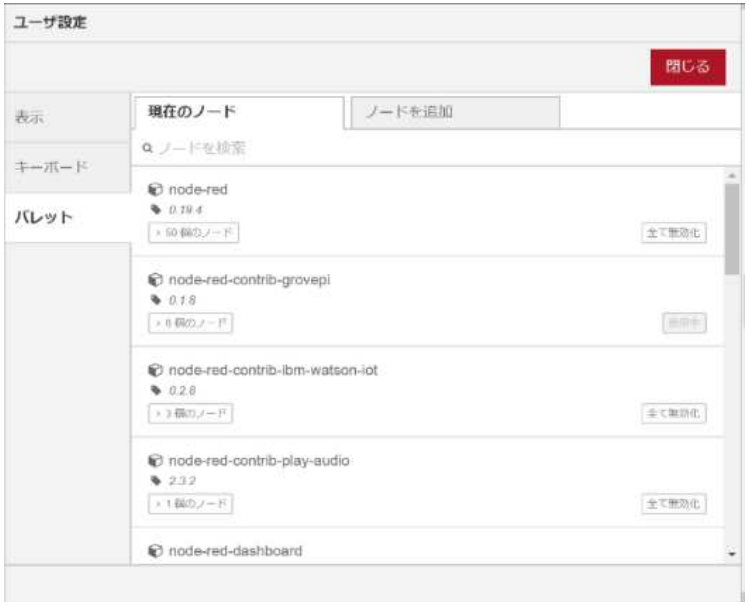
パレットマネージャ

パレットマネージャは、パレットに新しいノードをインストールするために利用できます。

デプロイボタンの右側の三本線アイコンをクリックし、パレットの管理をクリックします。

パレットマネージャダイアログが開きます。
現在のノードタブでは現在インストールされているノードが表示されます。

ノードの追加タブでは追加インストールしたいノードをキーワードで検索し、[ノードを追加]ボタンをクリックすると、インストールすることができます。



サイドバー

サイドバー

サイドバーには、エディタ内の便利ツールを提供するパネルが含まれています。

情報

サイドバーの初期状態またはタブの横の“i”と書かれた
アイコンボタンをクリックすると表示されます。

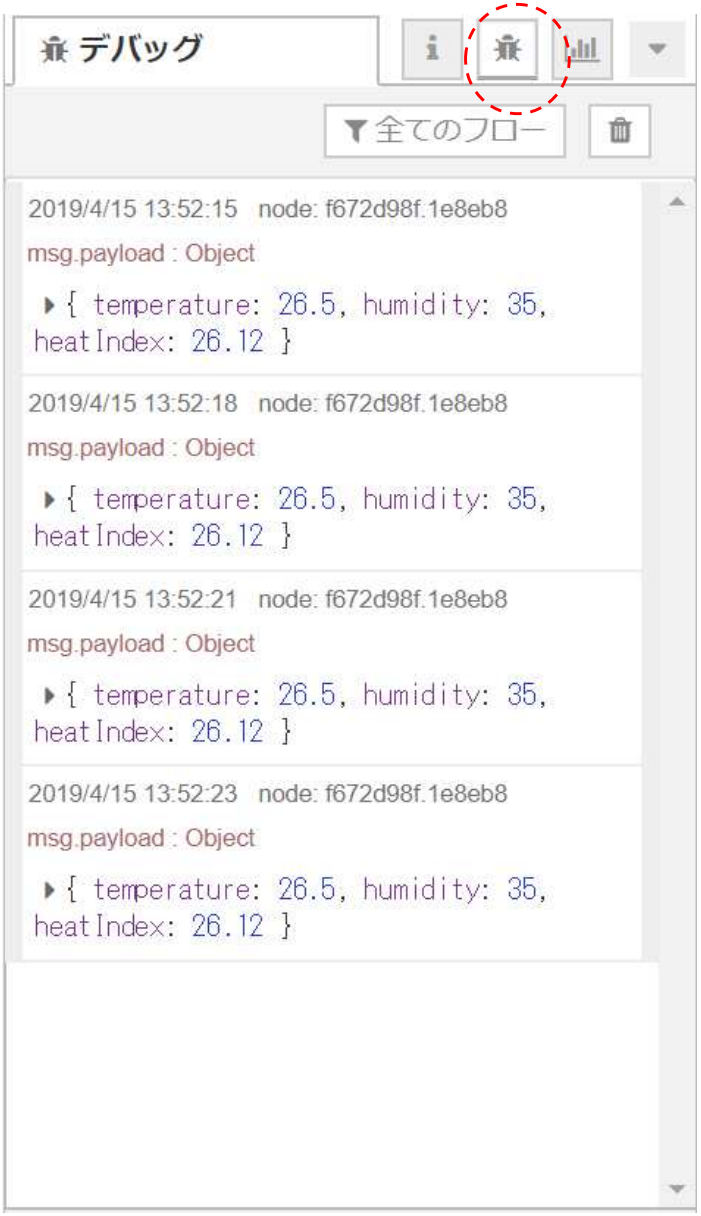
ノードを選択すると
ノードについてプロパティの概要、ノードのヘルプテキストを
表示します。

デバッグ

サイドバータブ右横の“虫の絵”が書かれた
アイコンボタンをクリックすると表示されます。

右図の「全てのフロー」と書かれているボタンを
クリックすると3つの選択肢を選ぶことができます。

・全てのフロー
　　全てのメッセージを表示します。
・選択したノード
　　全てのDebugノードの一覧から特定の
　　ノードを選択できます。
・現在のフロー
　　ワークスペースの現在のフローの
　　ノードからのメッセージを表示します。



Coreノード

Coreノード

- インジェクト
  エディタ内でこのノードをクリックすることで、手動でフローを始動できる

- デバッグ
  エディタ内サイドバーにメッセージを表示することができる

- ファンクション
  受け渡されるメッセージに対するjavaScriptコードの実行を可能にする

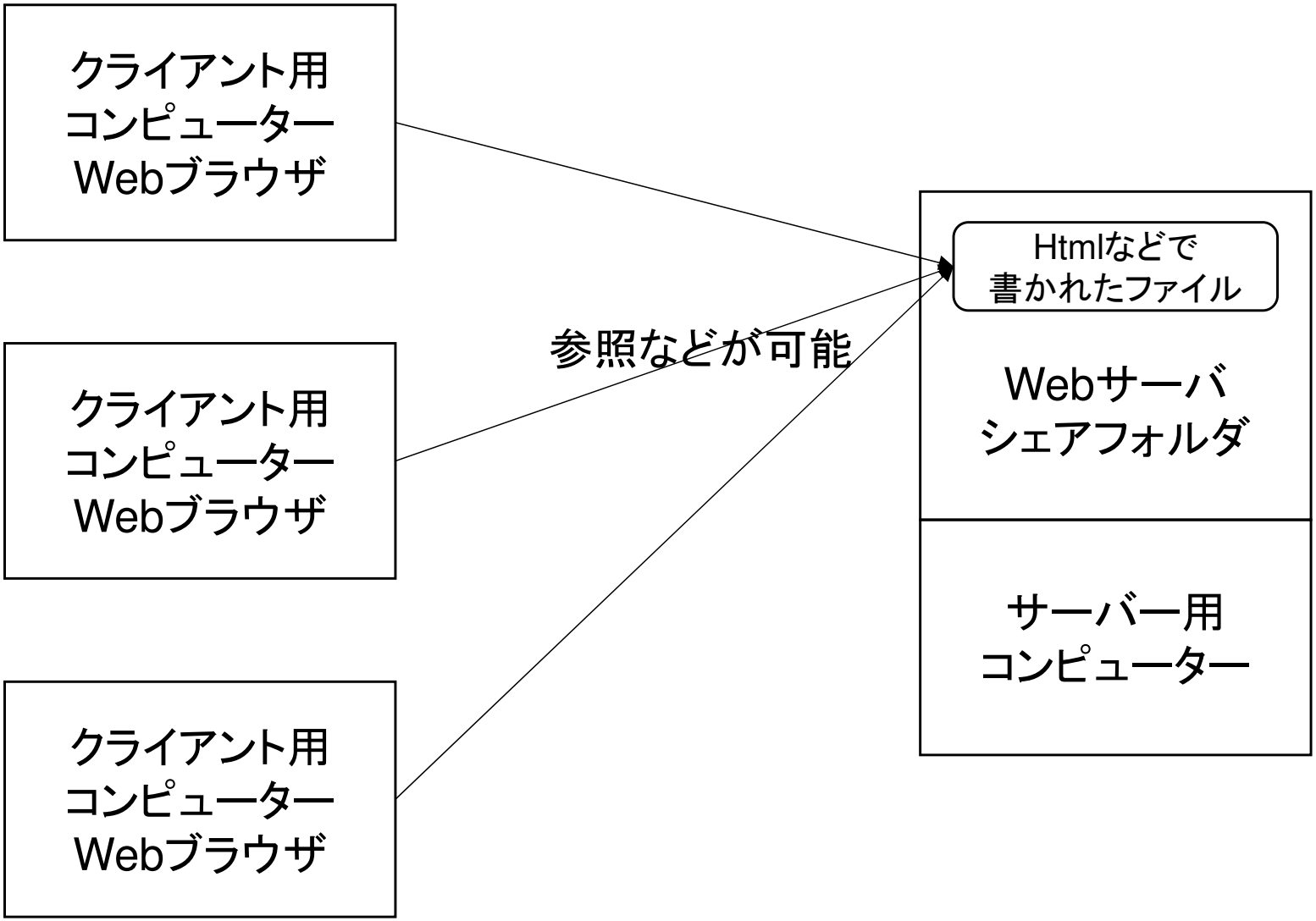
- チェンジ
  受け渡されたメッセージプロパティを変更したり、コンテキストプロパティを設定したりすることができる
  値の代入、値の置換、値の移動、値の削除など

- スイッチ
  受け渡されたメッセージに対して、一連のルールで評価し、メッセージを異なるフローにルーティングすることができる
  Value rule：設定されたプロパティに対して評価をおこなう
  Sequence rule：Splitノードによって生成されるようなメッセージシーケンスを利用できる
  JSONata式：メッセージ全体に対して評価を行い、trueの値を返した場合に一致したとみなすことができる
  その他：どのルールのどれにも一致しなかった場合、一致するように利用できる

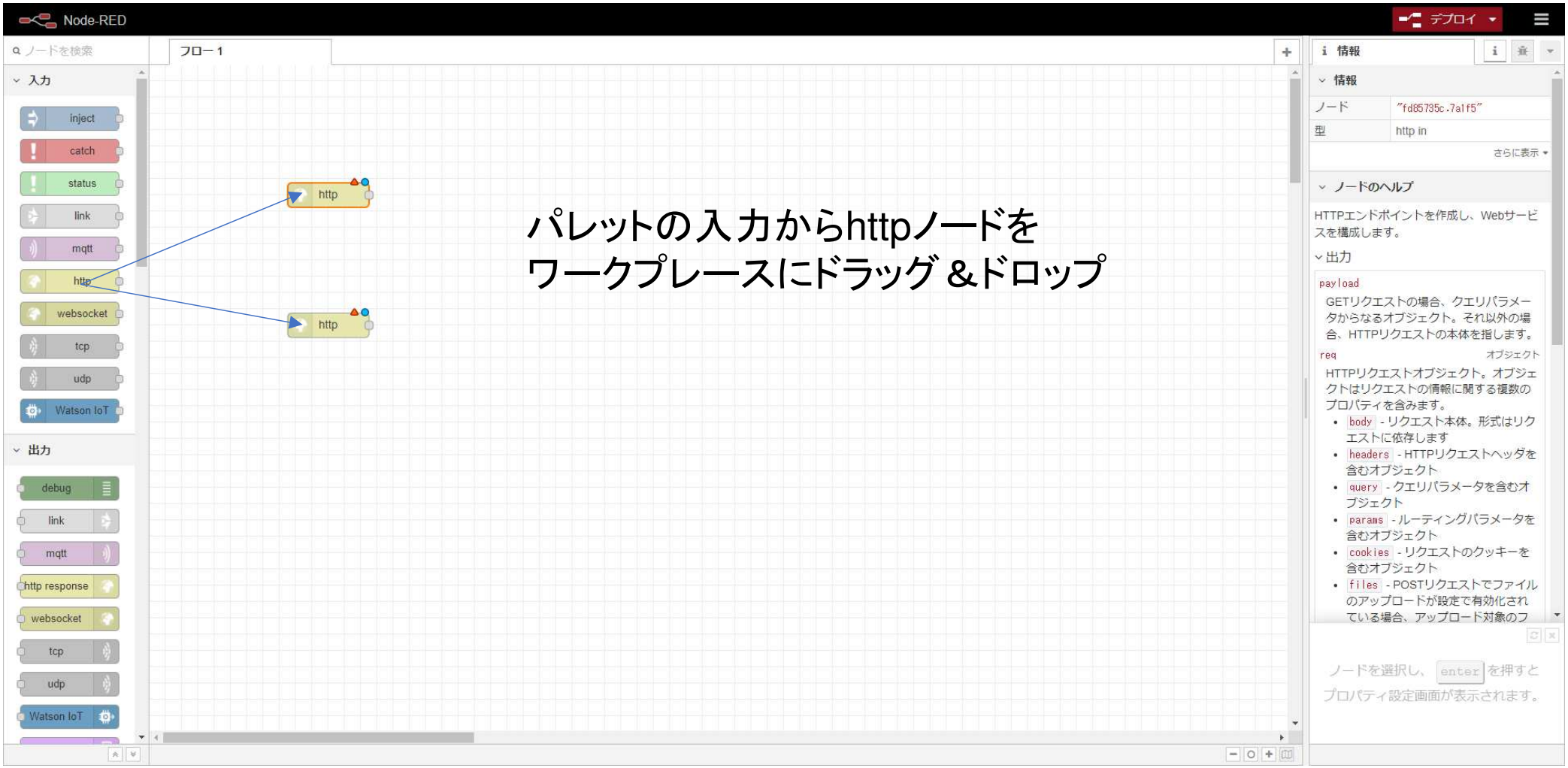
- テンプレート
  メッセージプロパティをテンプレートに設定することでテキストを生成することができる

簡易Webサーバーを作る

Webサーバーとは


クライアント用
コンピューター
Webブラウザ
クライアント用
コンピューター
Webブラウザ
クライアント用
コンピューター
Webブラウザ
参照などが可能
Htmlなどで
書かれたファイル
Webサーバ
シェアフォルダ
サーバー用
コンピューター

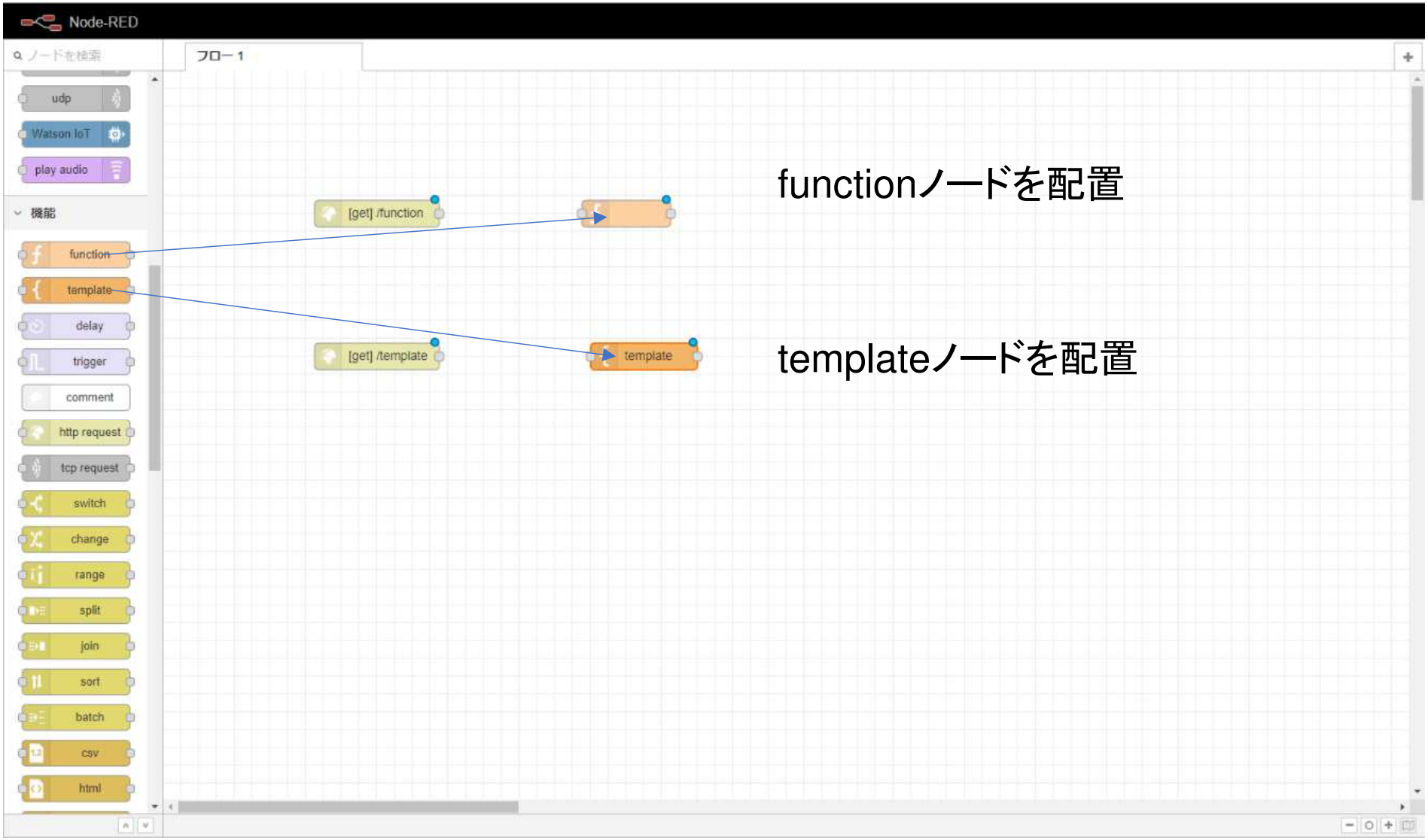
ノードの配置


Node-RED
デプロイ
ノードを検索
フロー 1
入力
inject
catch
status
link
mqtt
http
websocket
tcp
udp
Watson IoT
出力
debug
link
mqtt
http response
websocket
tcp
udp
Watson IoT
http
http
パレットの入力からhttpノードを
ワークプレースにドラッグ &ドロップ
情報
ノード
"fd85735c.7a1f5"
型
http in
さらに表示
ノードのヘルプ
HTTPエンドポイントを作成し、Webサービスを構成します。
出力
payload
GETリクエストの場合、クエリパラメータからなるオブジェクト。それ以外の場合、HTTPリクエストの本体を指します。
req
オブジェクト
HTTPリクエストオブジェクト。オブジェクトはリクエストの情報に関する複数のプロパティを含みます。
body - リクエスト本体。形式はリクエストに依存します
headers - HTTPリクエストヘッダを含むオブジェクト
query - クエリパラメータを含むオブジェクト
params - ルーティングパラメータを含むオブジェクト
cookies - リクエストのクッキーを含むオブジェクト
files - POSTリクエストでファイルのアップロードが設定で有効化されている場合、アップロード対象のフ
ノードを選択し、enterを押すとプロパティ設定画面が表示されます。

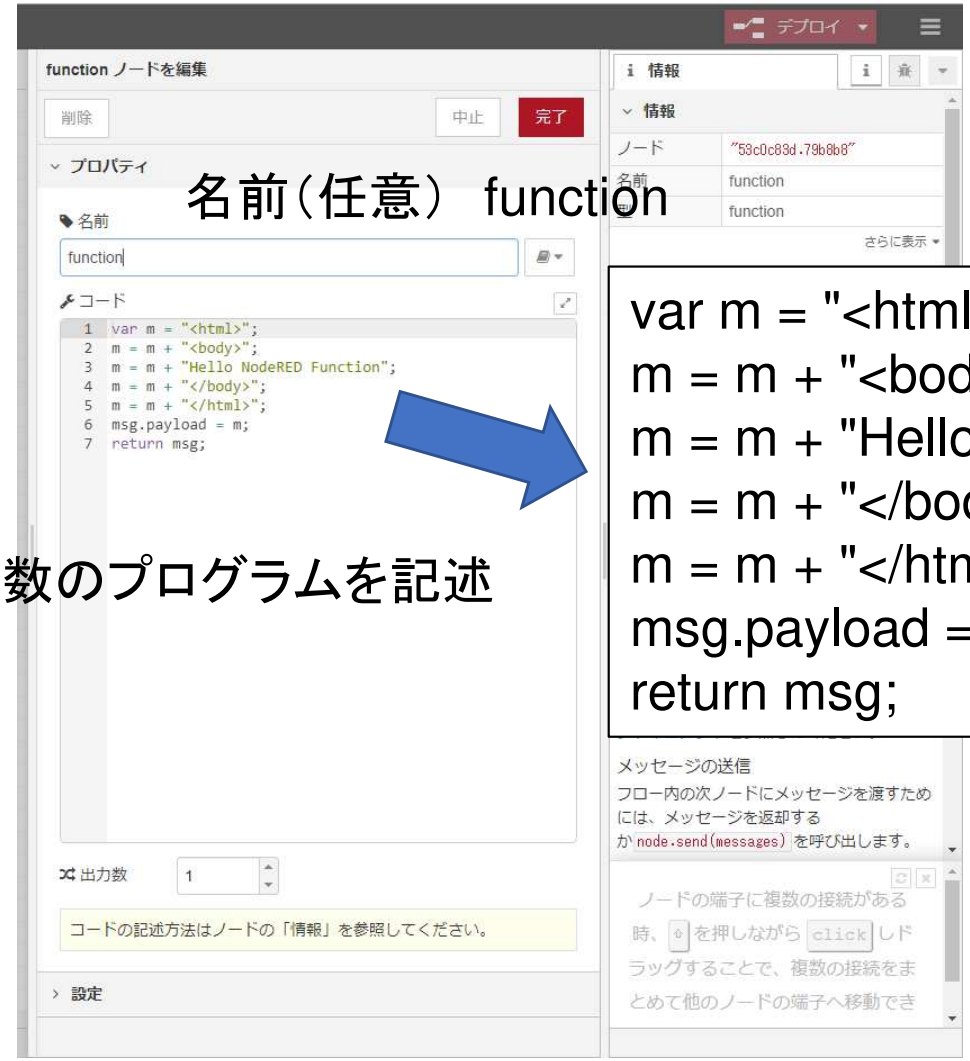
http入力ノードプロパティ入力

もう一つのノードも記入

機能ノードの配置


Node-RED
ノードを検索
フロー 1
udp
Watson IoT
play audio
機能
function
template
delay
trigger
comment
http request
tcp request
switch
change
range
split
join
sort
batch
csv
html
[get] /function
[get] /template
template
functionノードを配置
templateノードを配置

functionノードを編集

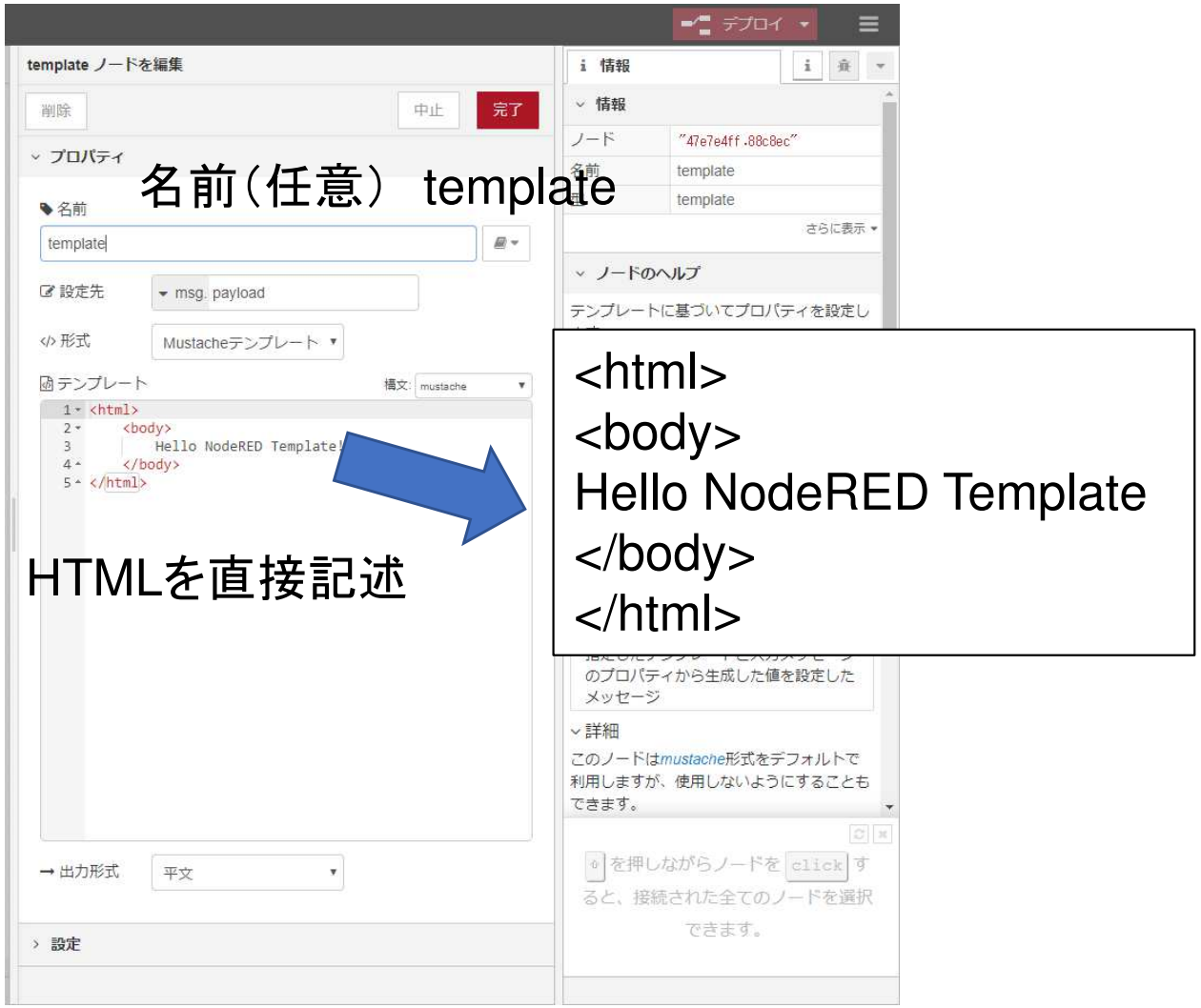


名前（任意） function

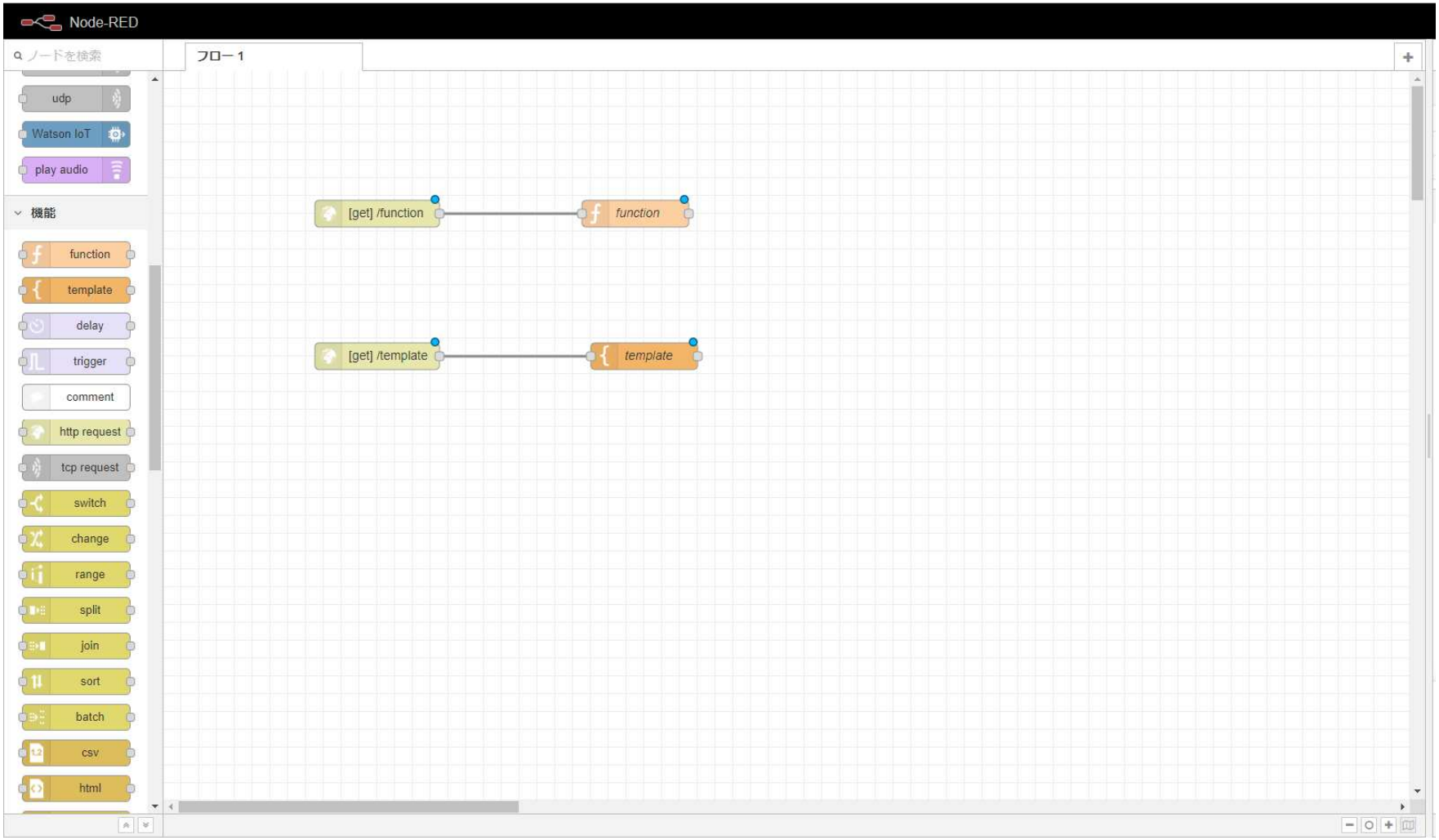
関数のプログラムを記述

```
var m = "<html>";
m = m + "<body>";
m = m + "Hello NodeRED Function";
m = m + "</body>";
m = m + "</html>";
msg.payload = m;
return msg;
```

templateノードを編集



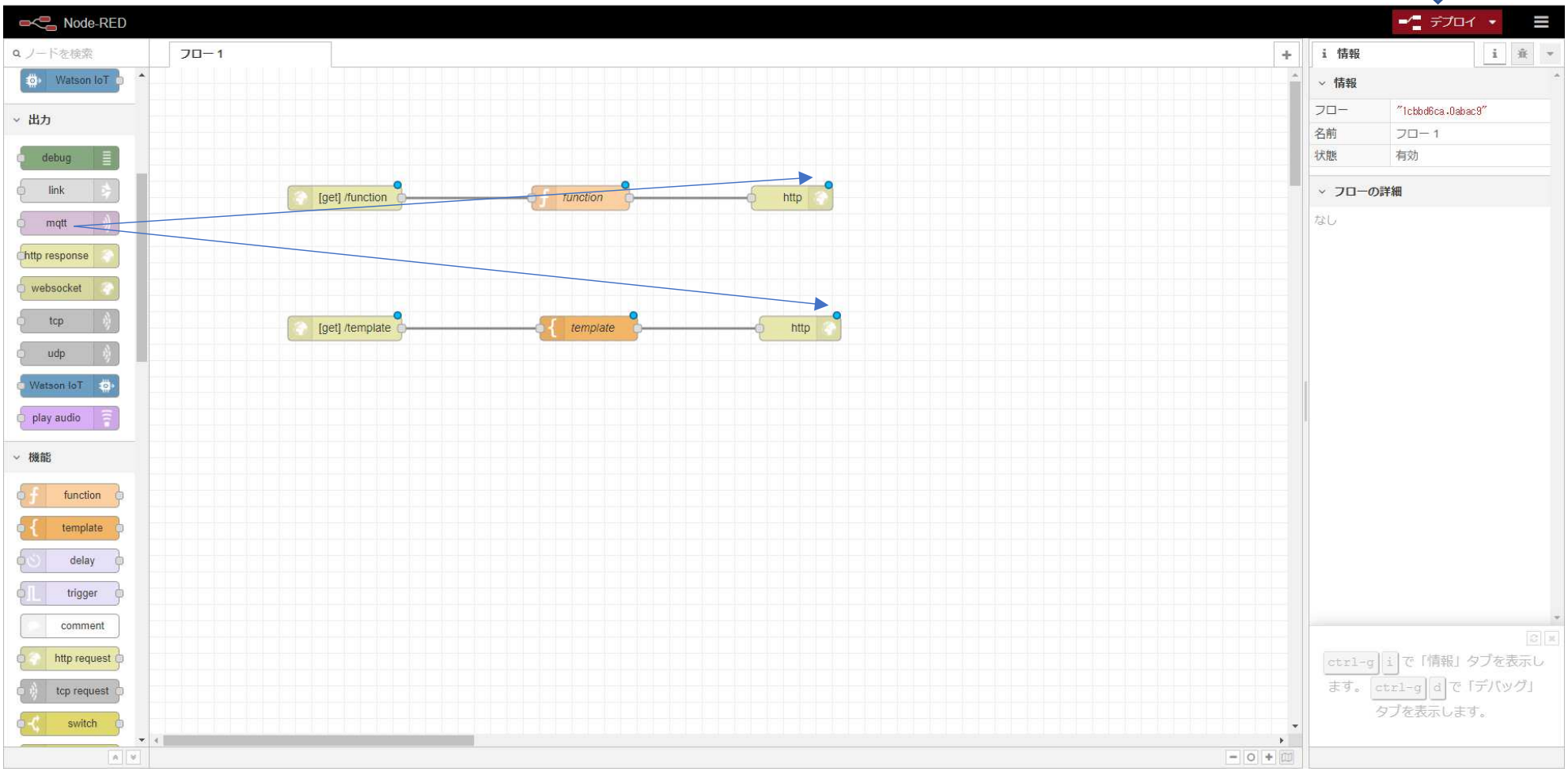
入力ノードと機能ノードをつなぐ



出力ノードhttp responseを配置

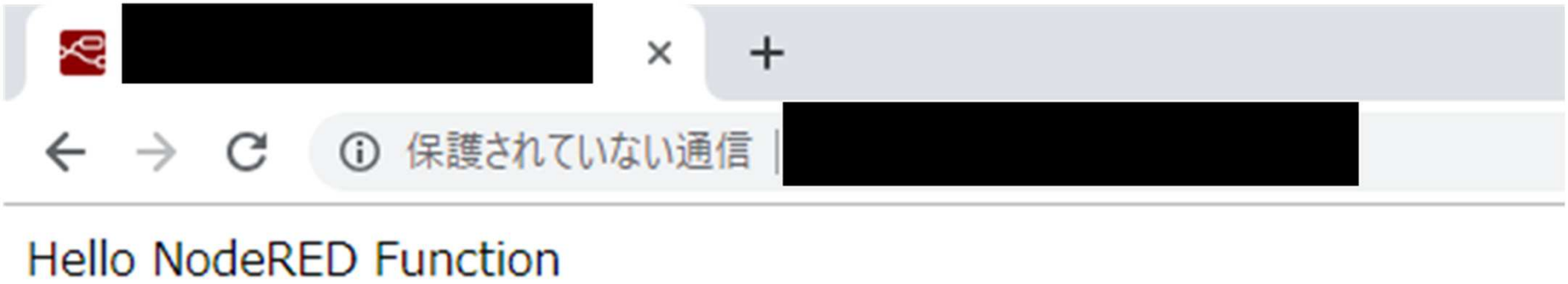
デプロイボタンをクリック

機能ノードと出力ノードをつないでデプロイすれば完成。

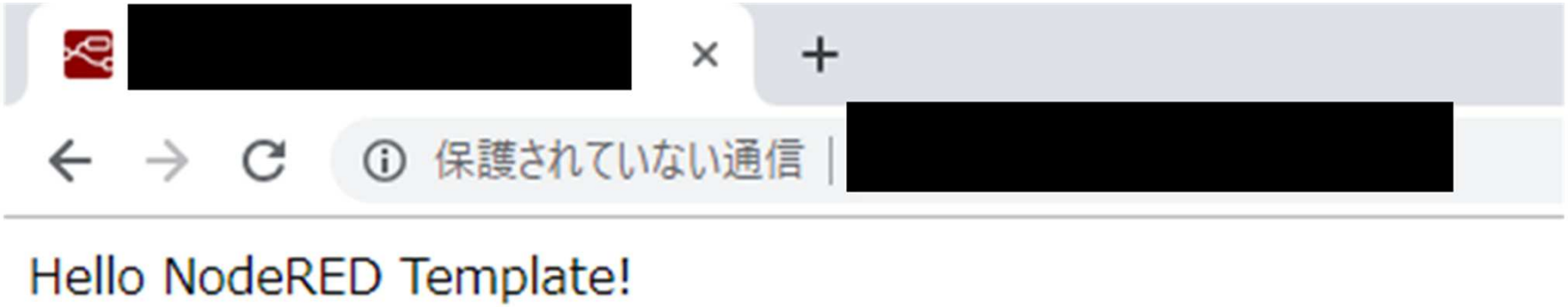


PCからWebページを開く

/function の表示



/template の表示



Internet explore の場合は
http://を必ず先頭に入れてください。

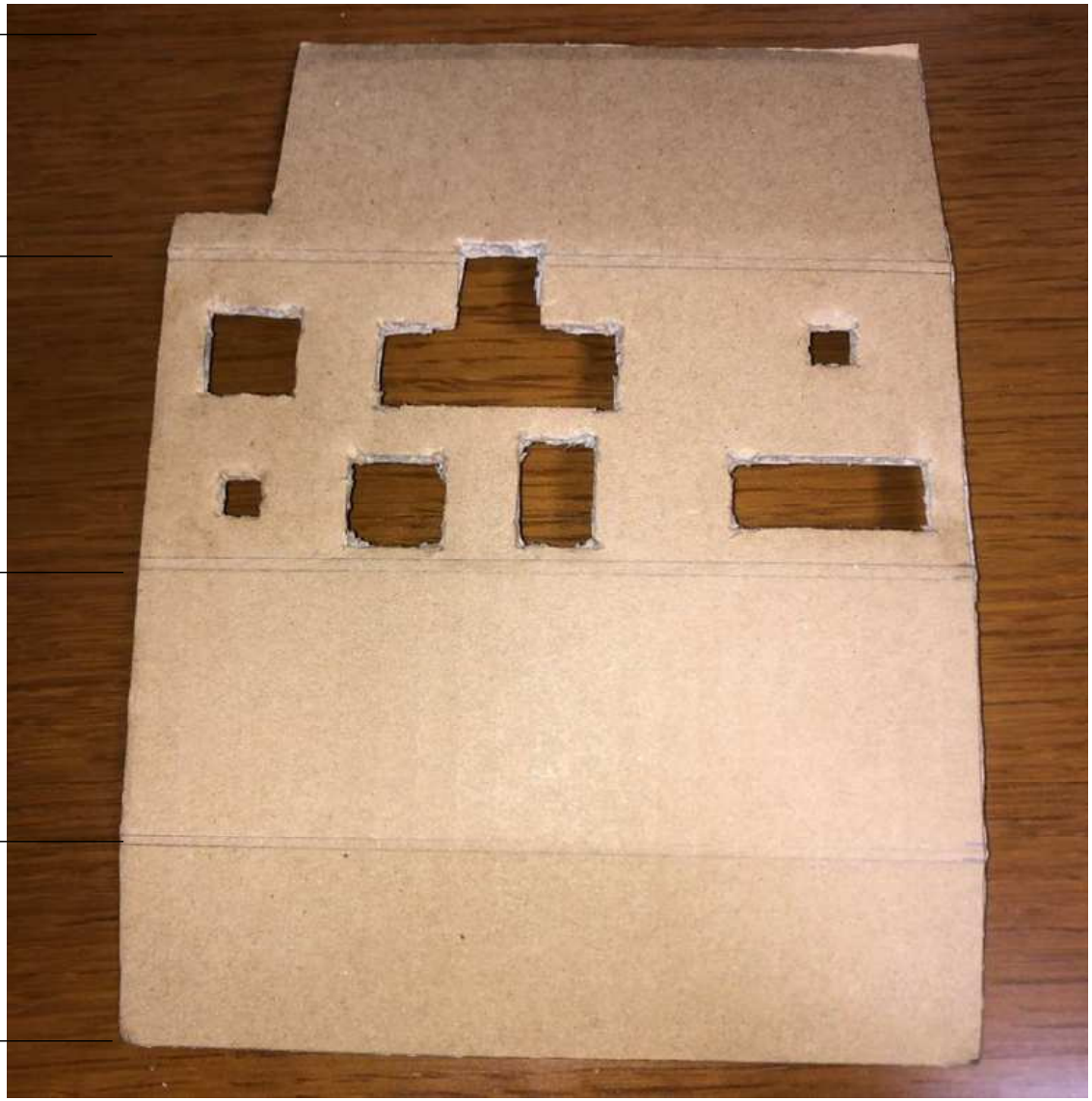
functionとtemplateの違い

- function
  - javascriptのプログラムを動かすノード
  - 事例のプログラムはhtml文を作っている

- template
  - templateに書いた文をそのまま渡す

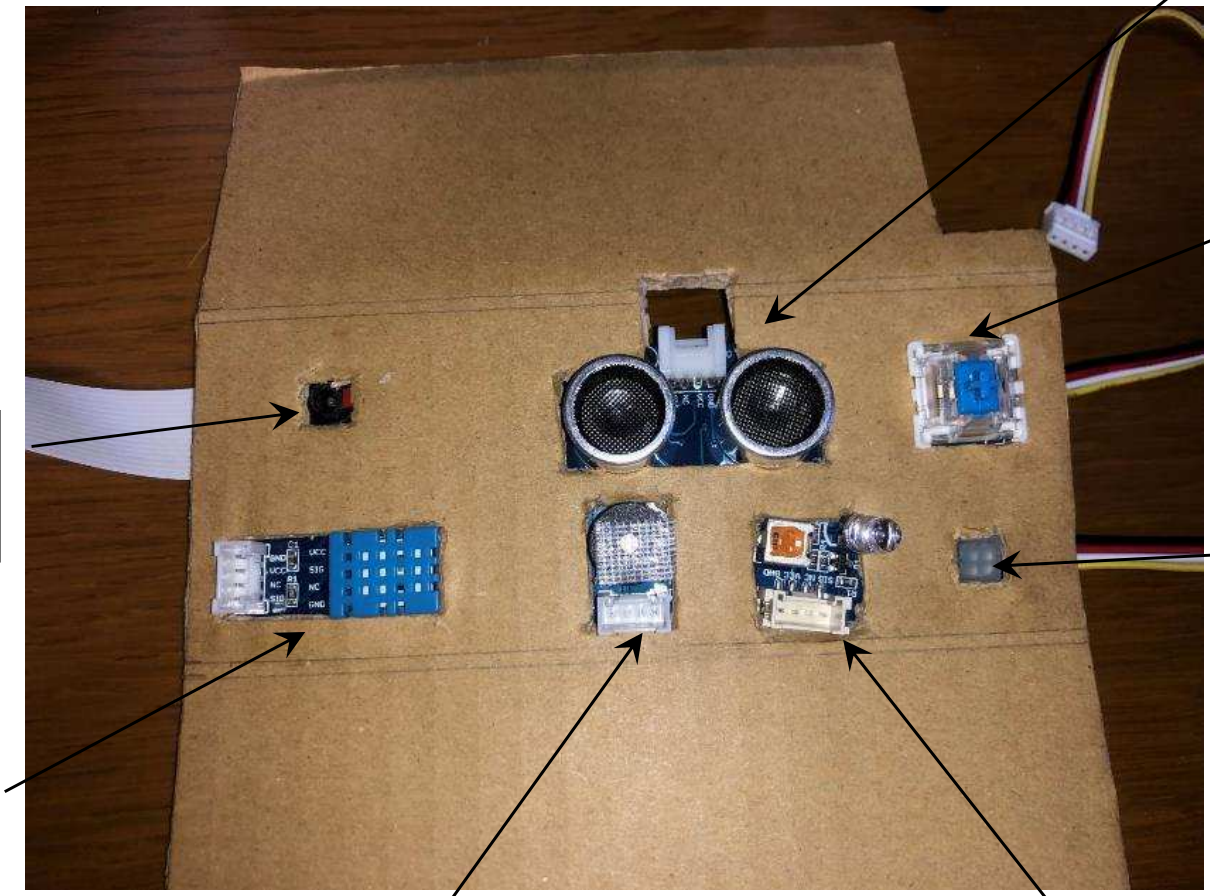
センサーボックスを組み立てる

内側に丸まるように折り曲げる



部品を穴に差し込む

丸まる内側から外側に向けて、穴あき面に部品を挿し込んでいきます。



超音波距離センサ

キースイッチ

カメラ

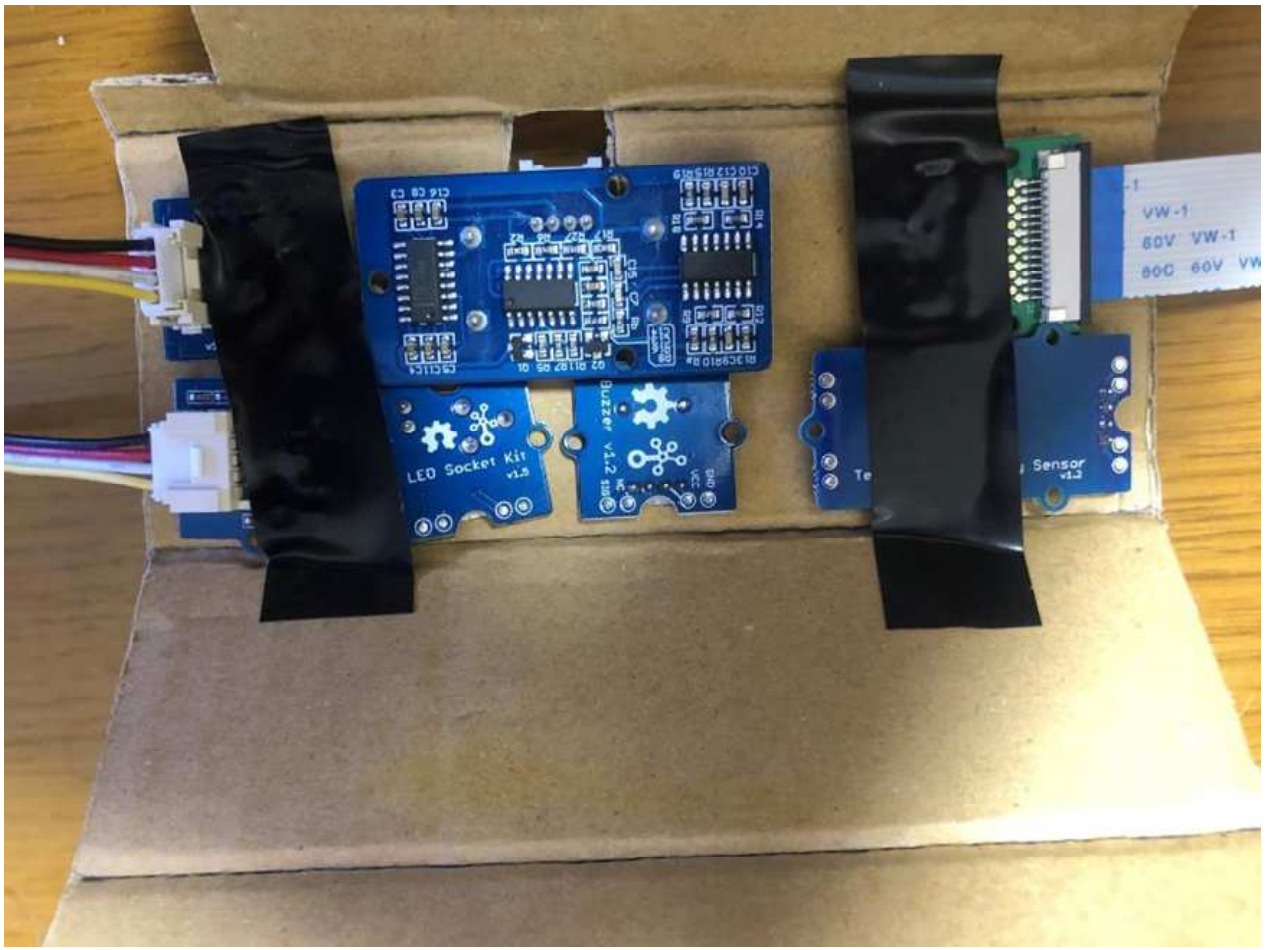
赤外線センサ

温湿度センサ

ブザー

LED

部品をテープで固定する



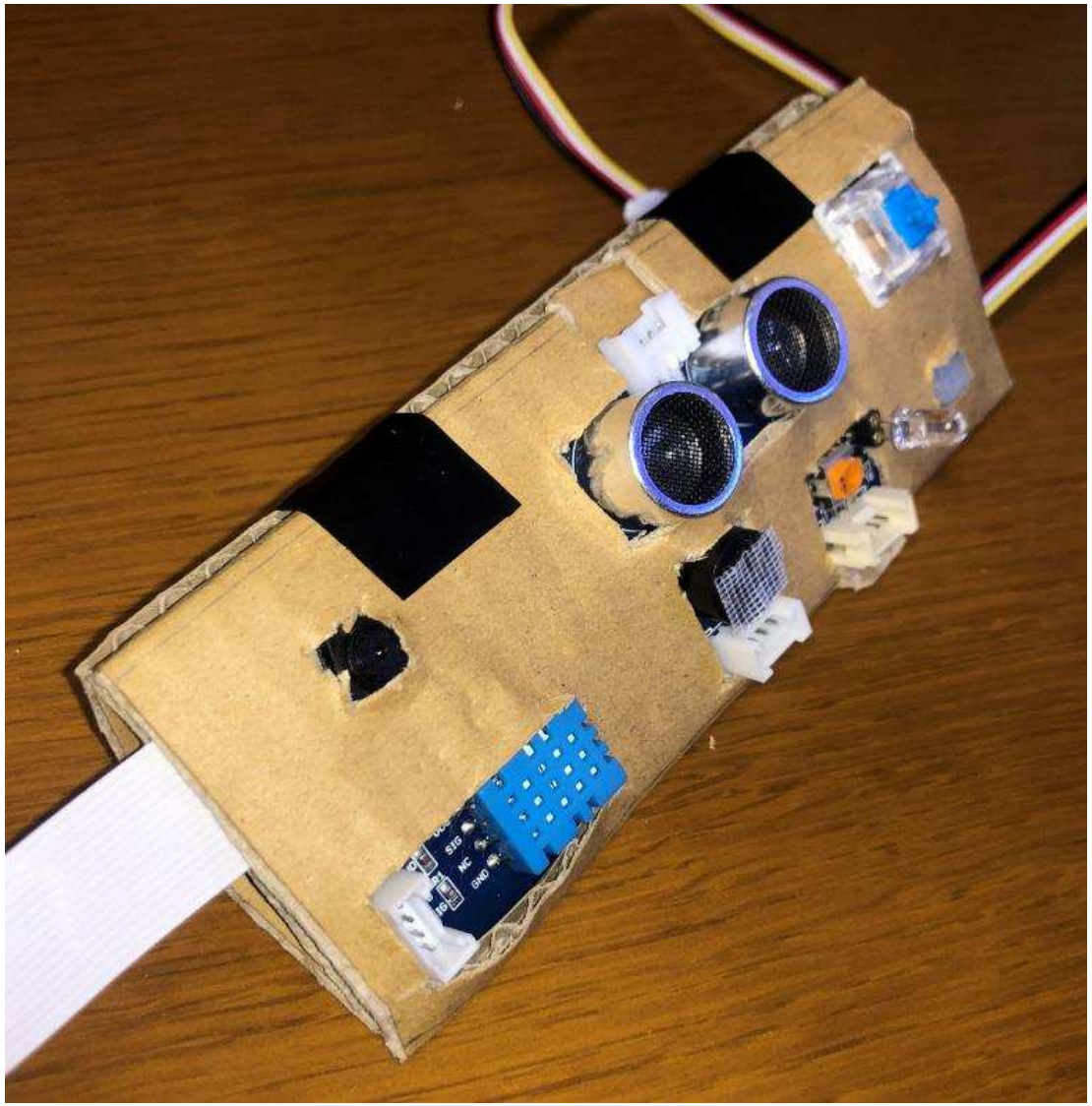
両脇のセンサーやカメラをテープで固定します。

上面の受け側面を折り曲げて固定

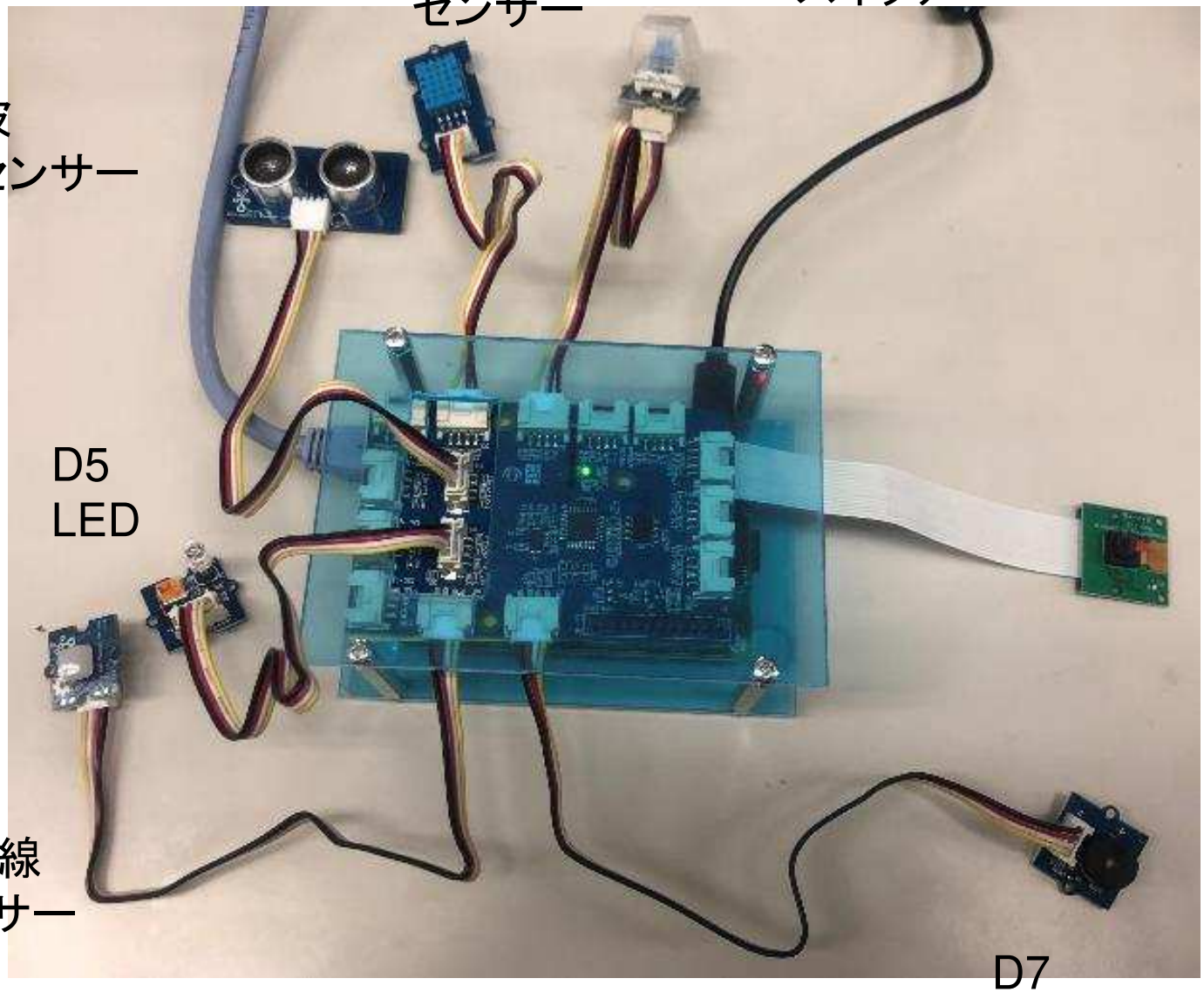
ぎゅっと押し付けてテープで固定します。

センサーボックスの完成

横に倒した三角柱の様な形に折り曲げてテープで固定します。



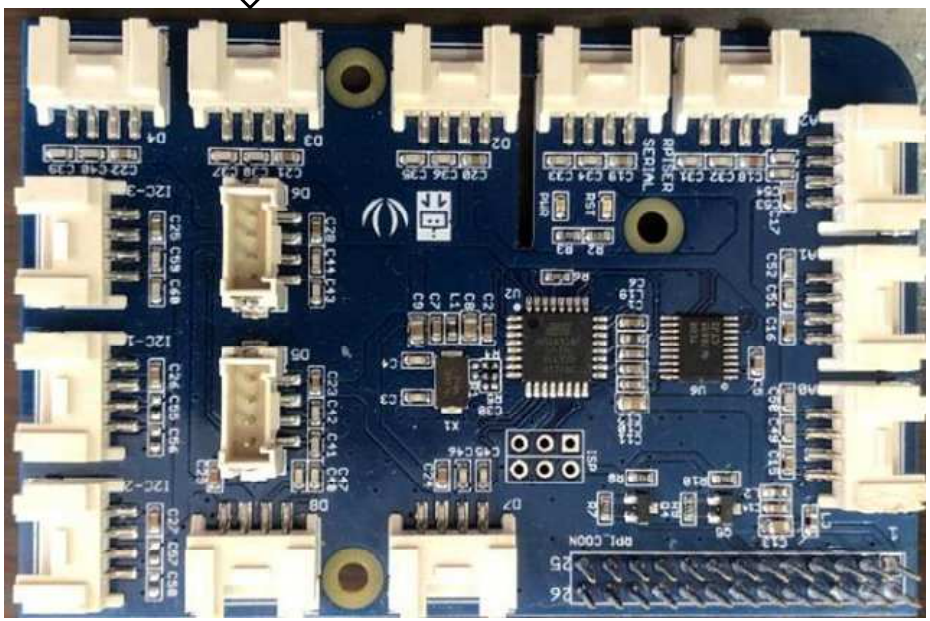
接続するポート


D3
温湿度
センサー
D2
押しボタン
スイッチ
D6
超音波
距離センサー
D5
LED
D8
赤外線
センサー
D7
ブザー

もくじ

Grove Temp&Humiの取付

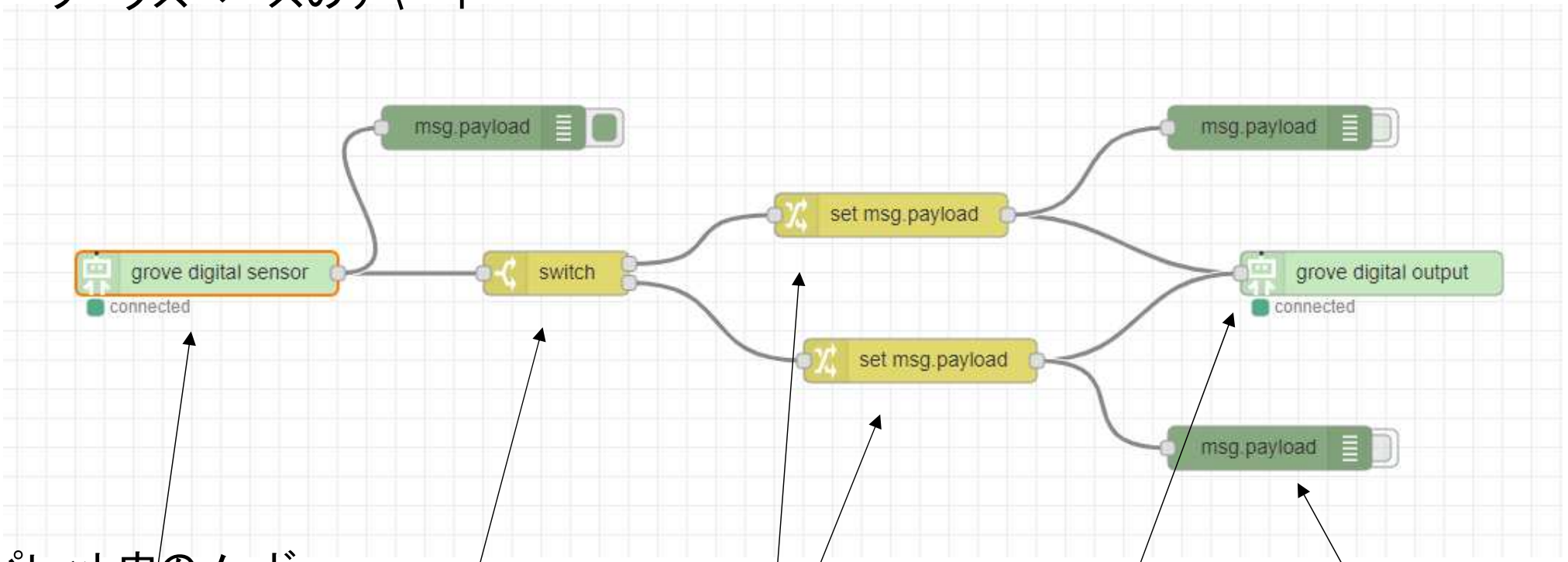
Grove Temp&Humi
Digital port 3 に接続する



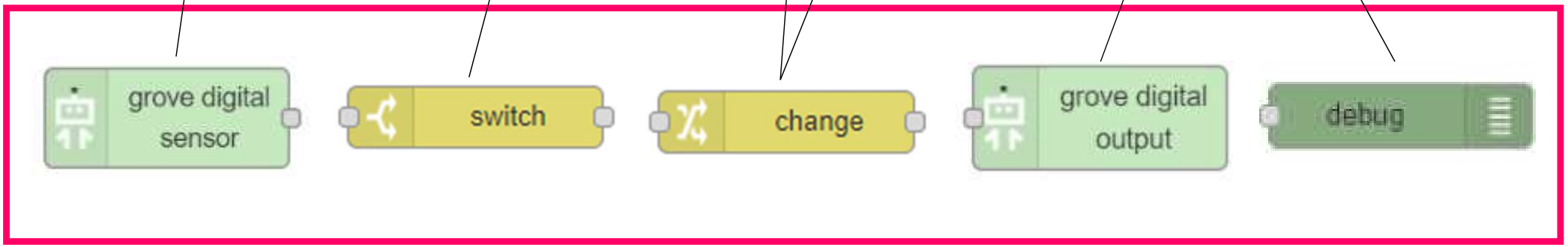
Grove Buzzer
Digital port 7に
接続する

フローを作ってみる

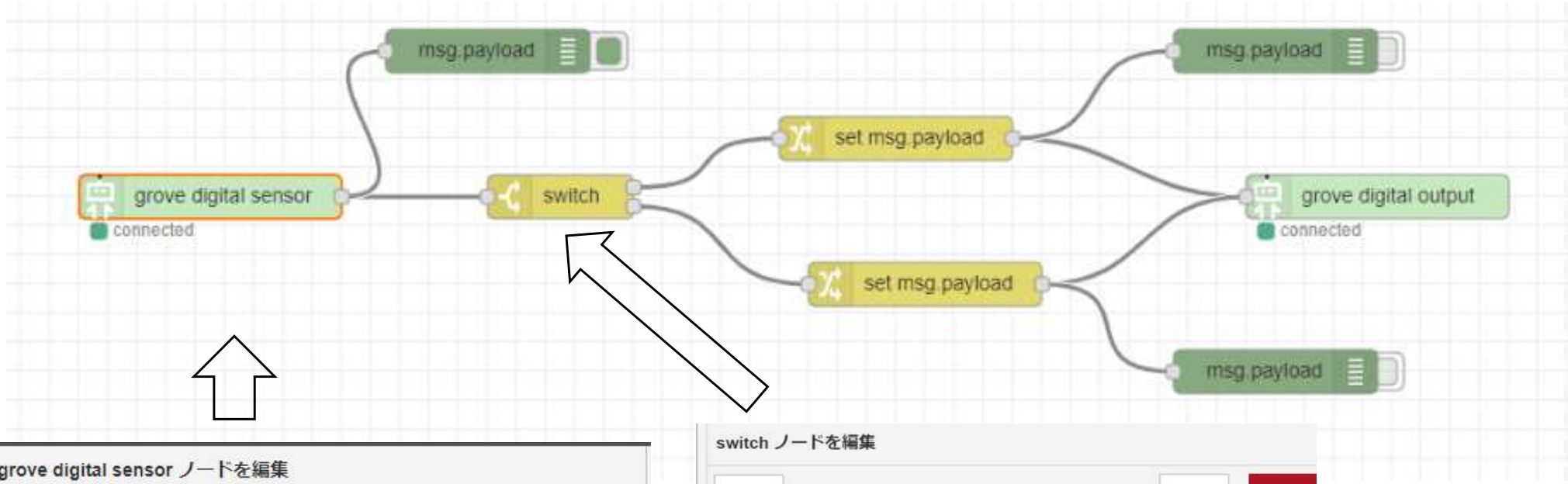
ワークスペースのチャート

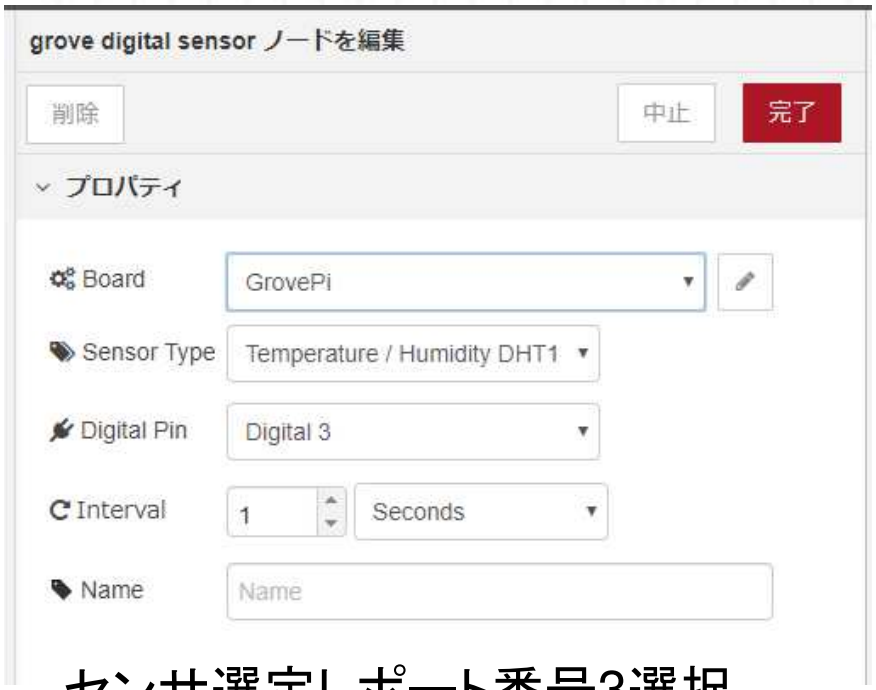


パレット内のノード



プロパティを設定する

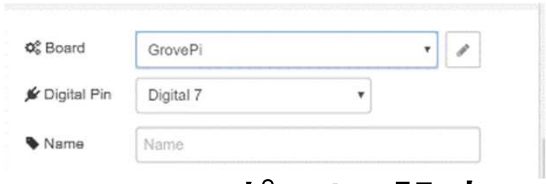




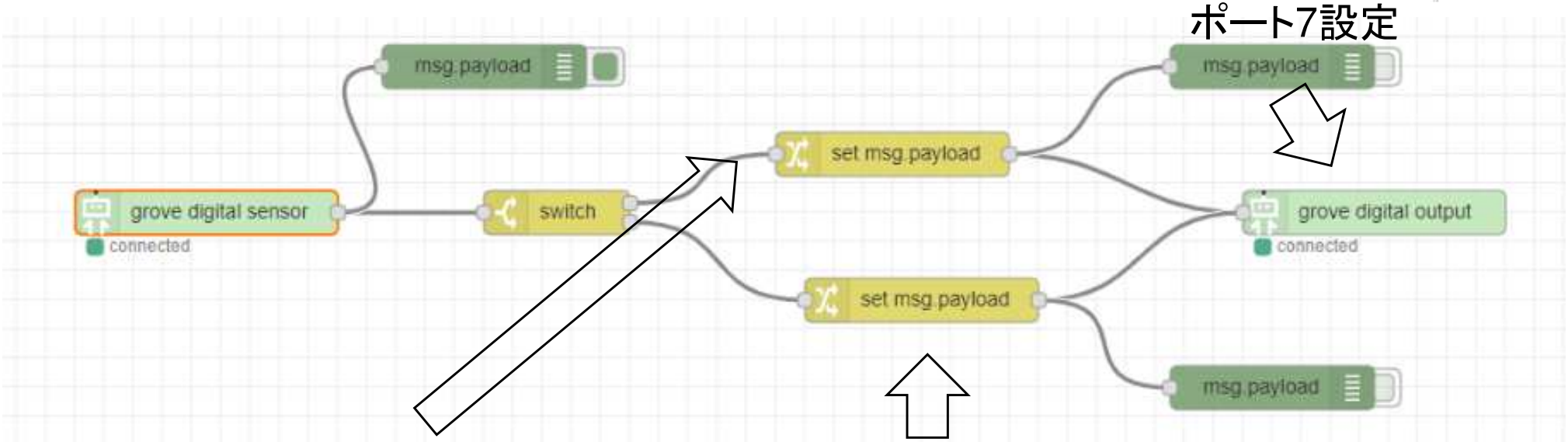
センサ選定しポート番号3選択

プロパティ
msg.payload.temperature
条件
>=　数値：40　→1
その他　→2

プロパティを設定する



ポート7設定



ルール
値の代入　msg.payload
対象の値　真疑　true

ルール
値の代入　msg.payload
対象の値　真疑　false

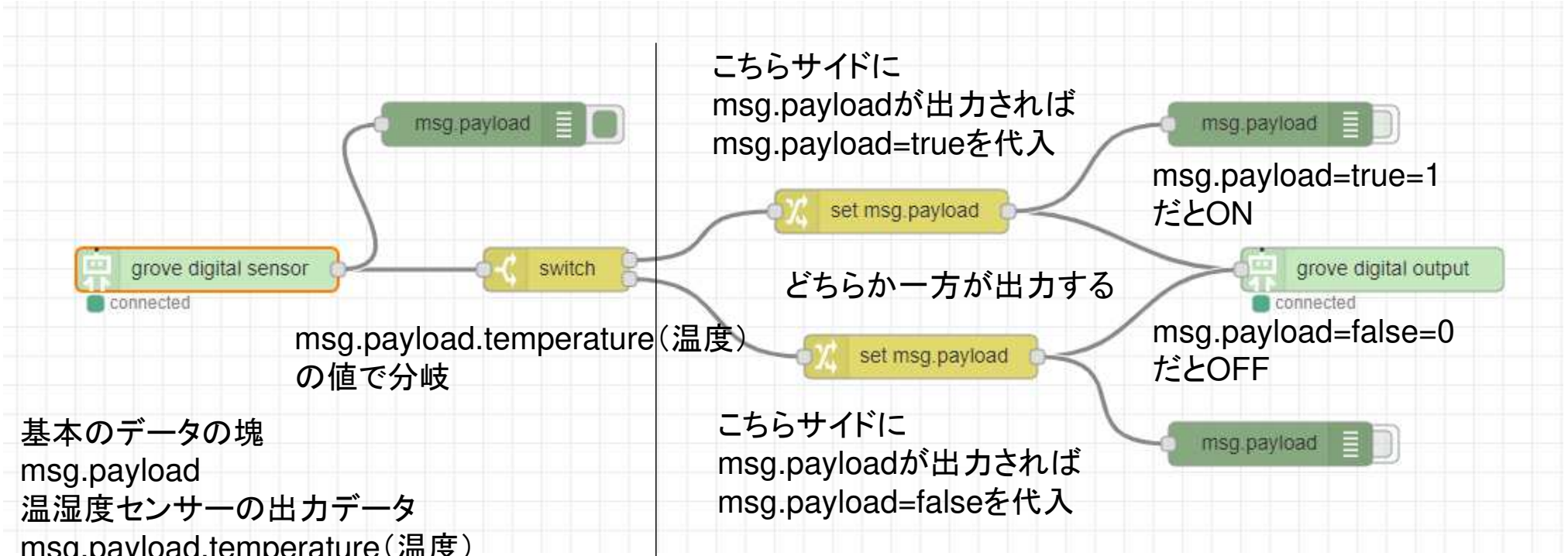
Node-RED+GrovePi使用での注意

GrovePiのポートにセンサーを沢山付けた場合、動作異常を起こす可能性があります。
今回の実習で使ったセンサーを全て接続して、この操作を行ったところ、測定値が異常になり、動作異常をおこしました。

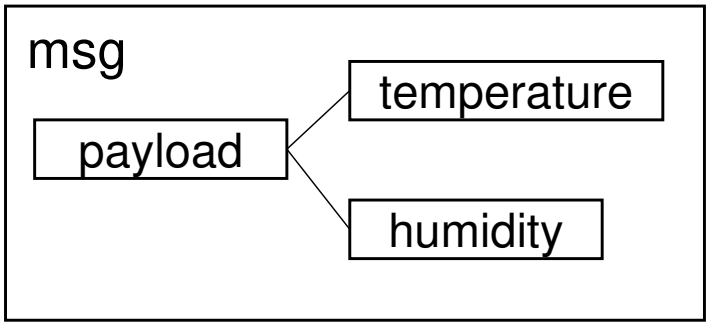
必ず起きるわけではありませんが、実際に使用する場合はその様な異常がない様にしておく必要があります。
以下の様な対策を実施するよう心掛けてください。

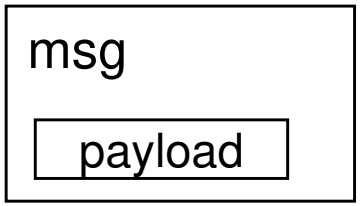
・センサーを接続しすぎない
・測定値が一瞬異常になっても異常反応させない
・測定異常値の形式が分かった場合、その様な形式をフィルタリングする

ノード間のデータの流れ



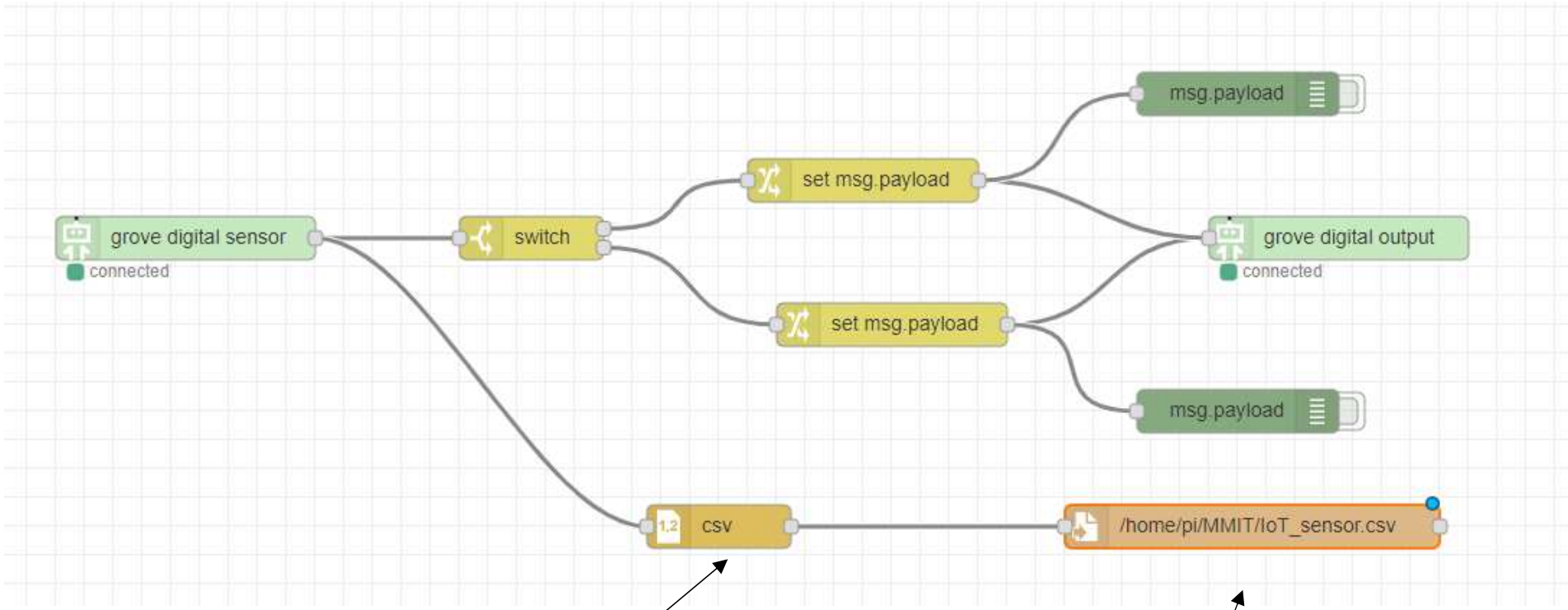
基本のデータの塊
msg.payload
温湿度センサーの出力データ
msg.payload.temperature（温度）
msg.payload.humidity（湿度）





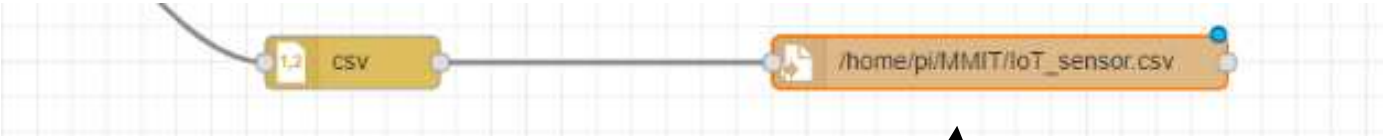
もくじ

CSV出力用フローを追加する



パレット内のノード

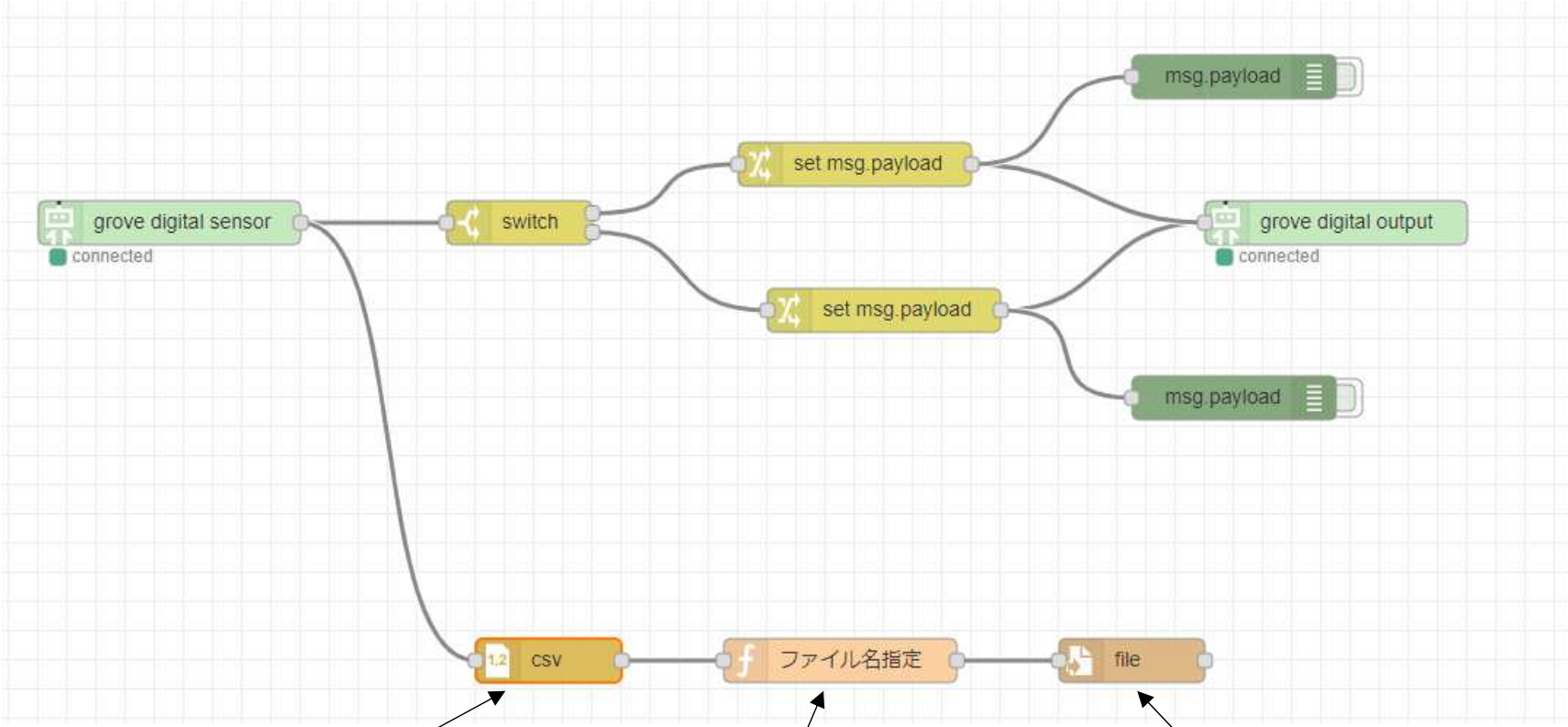
プロパティを設定する



特に修正なし。

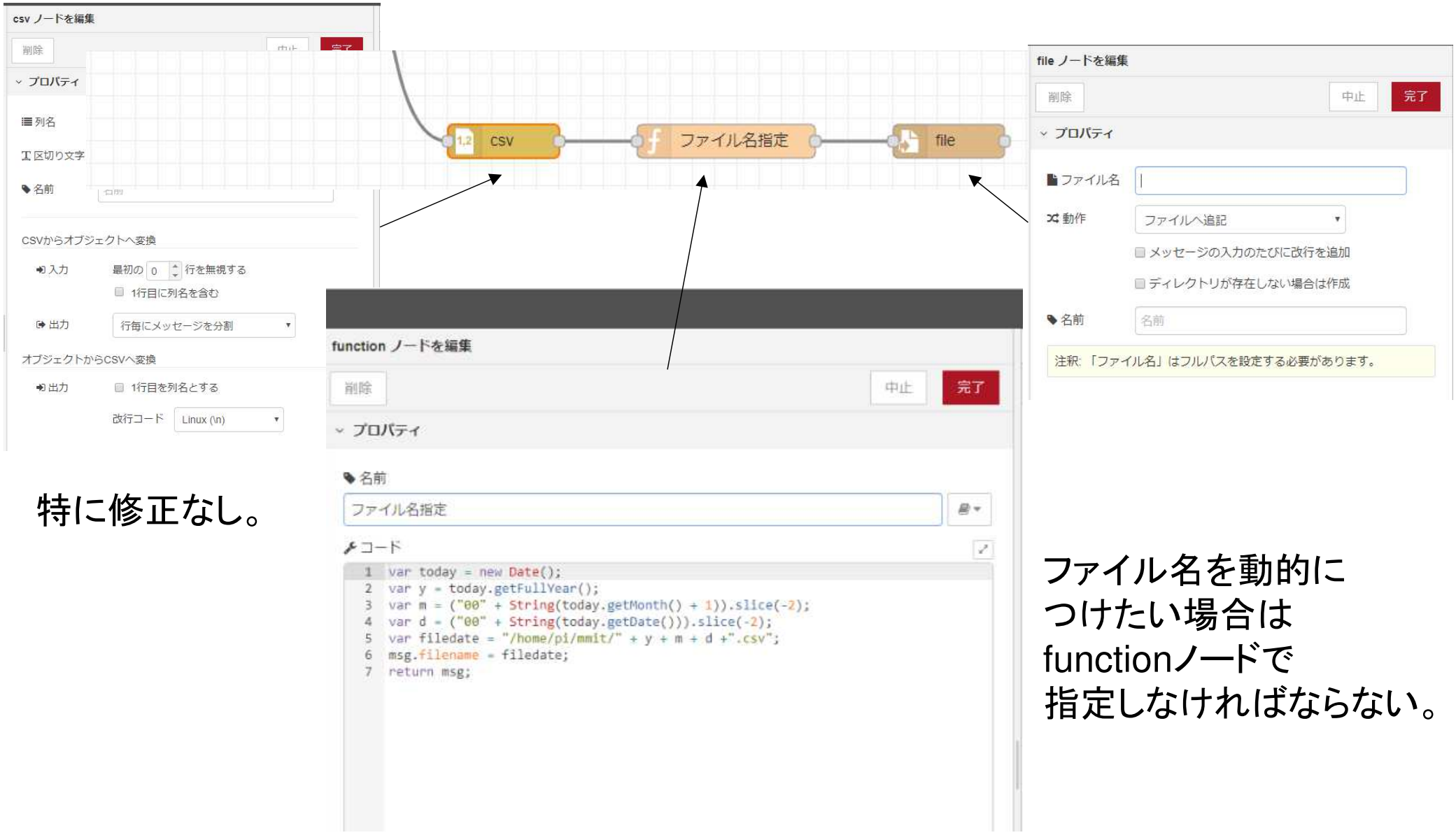
出力したいファイル名を
ディレクトリから入力する。

ファイル名を動的につけたい場合



パレット内のノード

プロパティを設定する



特に修正なし。

ファイル名を動的に
つけたい場合は
functionノードで
指定しなければならない。

ファイル名指定プログラム

```
1  var today = new Date();
2  var y = today.getFullYear();
3  var m = ("00" + String(today.getMonth() + 1)).slice(-2);
4  var d = ("00" + String(today.getDate())).slice(-2);
5  var filedate = "/home/pi/mmit/" + y + m + d +".csv";
6  msg.filename = filedate;
7  return msg;
```

注意

- Node-REDには2千に近いノードが公開されています
- オープンソースのため全てが正しく動作する保証はありません
- 想定通りに動かない場合は、functionノードなどでjavascriptを使って一から作成する場合もあります
- 今回のcsvノードも一例で、一行目にタイトル行を設定すると正常に動きません

CSVで1行目にタイトル行を入れる方法

タイトル行を出力するフローとデータを出力するフローを別にする

※ファイル出力の1行目処理時

[csv]ノード

- 「列名」にカンマ区切りで列名を指定。
- 「オブジェクトからCSVへ変換」の「出力」にて
- 「1行目を列名とする」にチェックを入れる。
- Windowsでcsvを使用する前提なら「改行コード」は”Windows”にした方が良いでしょう。

[file]ノード

- 「動作」にて”ファイルを上書き”を指定。

※ファイル出力の2行目以降処理時

[csv]ノード

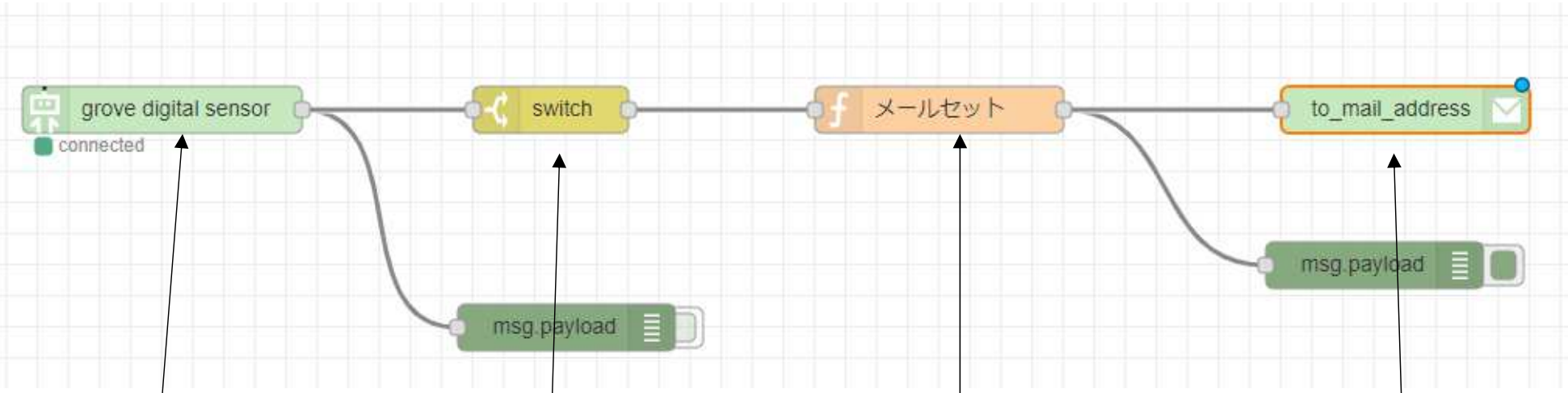
- 「列名」（1行目時と同様）
- 「オブジェクトからCSVへ変換」の「出力」にて
- 「1行目を列名とする」にチェックを入れない。
- 「改行コード」（1行目時と同様）

[file]ノード

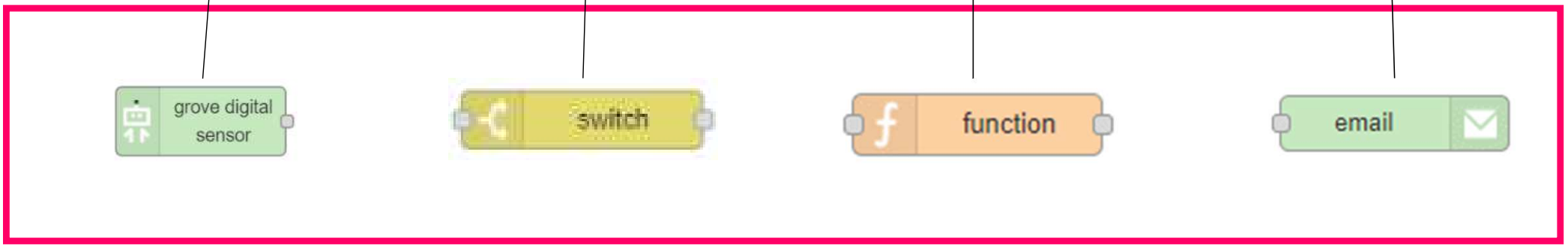
- 「動作」にて”ファイルへ追記”を指定。

もくじ

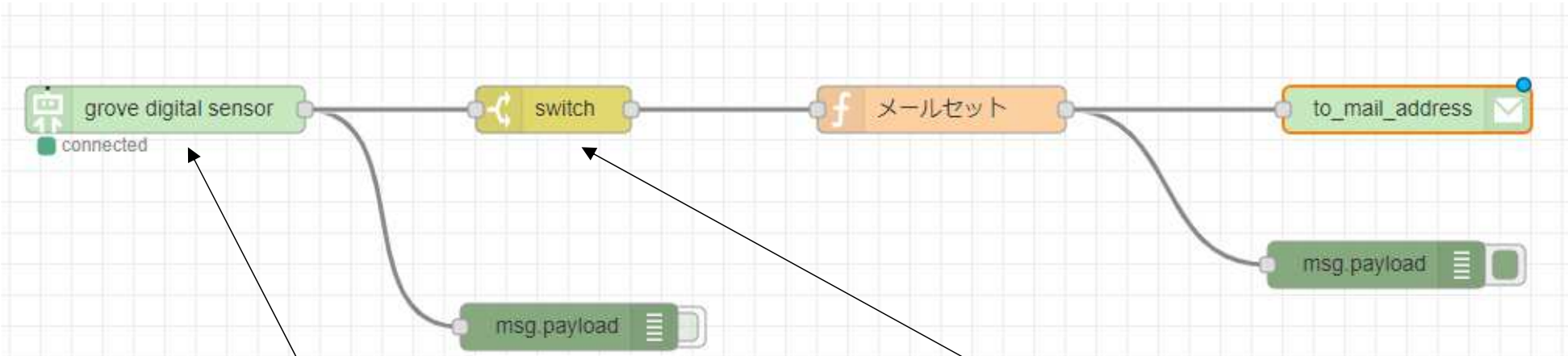
ボタンを押したらメール送信



パレット内のノード



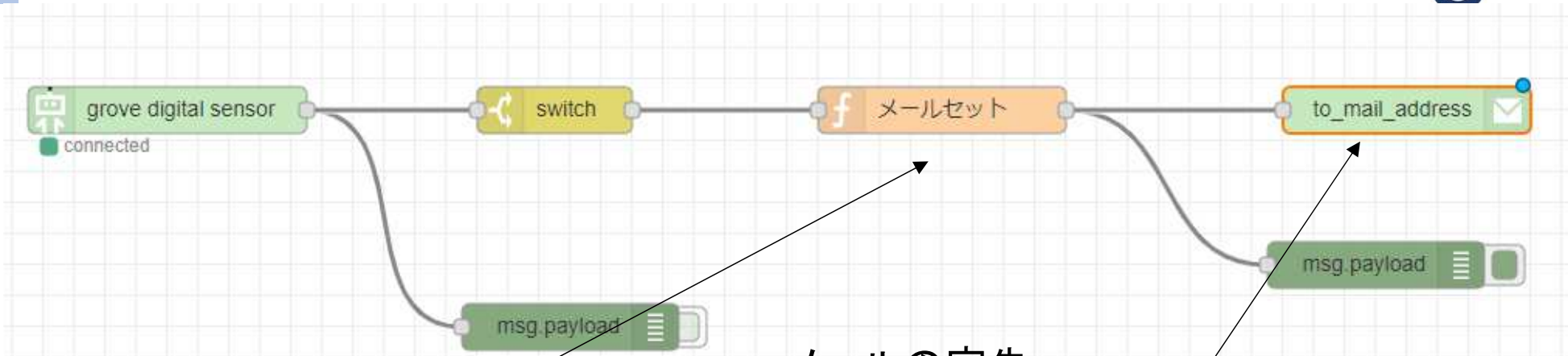
プロパティを設定する



ポート2の押しボタンスイッチに設定。

Trueが送られてきた時だけメッセージを送る様に設定。

プロパティを設定する



メール送信に必要なメッセージ
データを設定する。

メールの宛先
Gmail用ユーザID
パスワードを設定。

msg.payload
msg.topic

前半5日間の進め方

午後の実習

■1日目　実習のための環境設定 ⇒ 課題発見ワークショップ

■2日目　デバイス信号のイン/アウト ⇒ センサデータの見える化

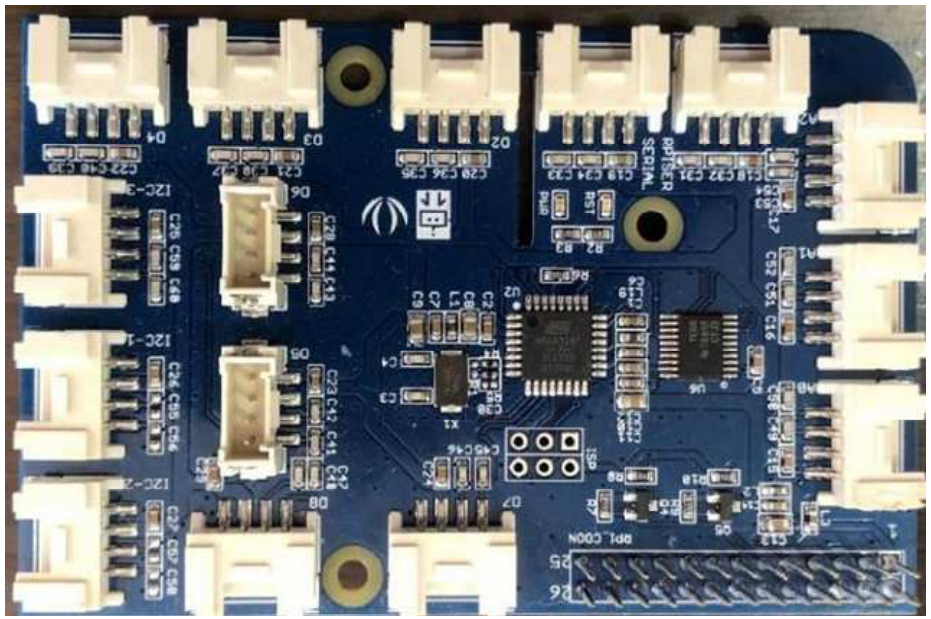
■3日目　メールとWebサーバ利活用 ⇒ 人感センサとカメラの利用

■4日目　業務システムの基本パターン ⇒ バーコードリーダとNFC

■5日目　データ分析続き ⇒ 工程進捗管理ボード

もくじ

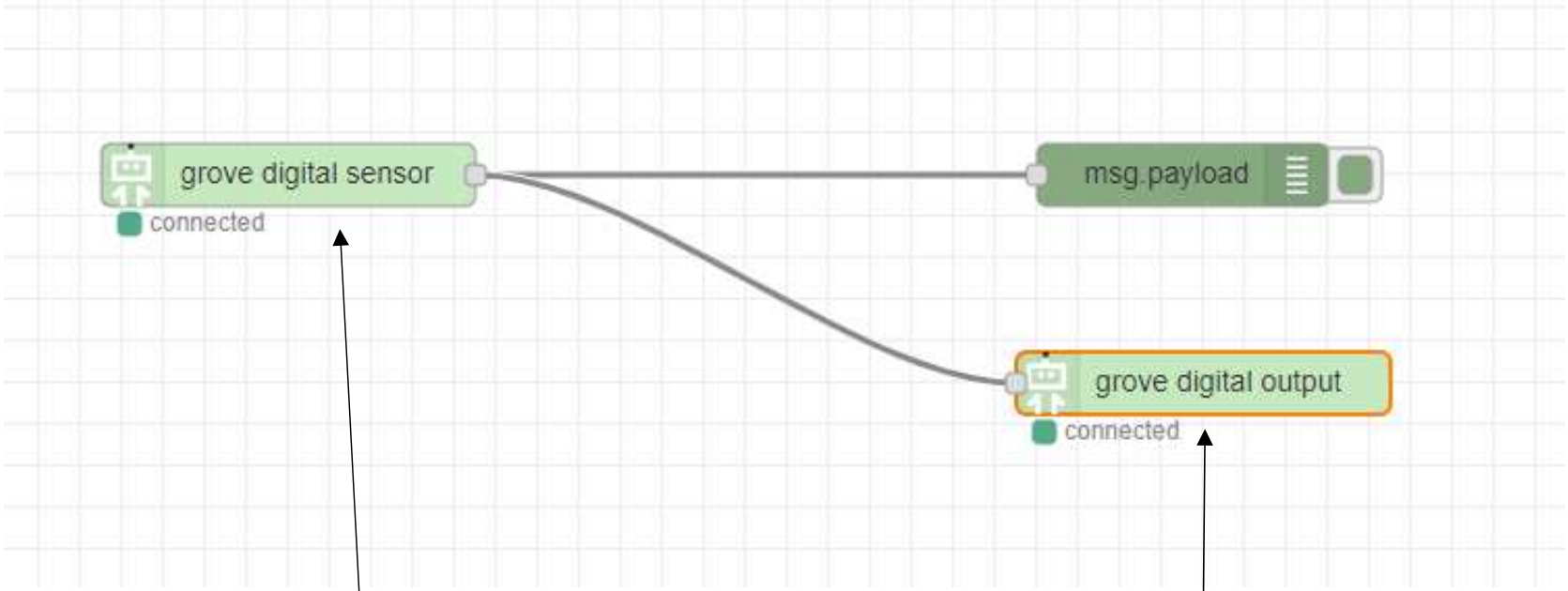
Grove Mini PIR motion の取付



Digital port 8 に接続する

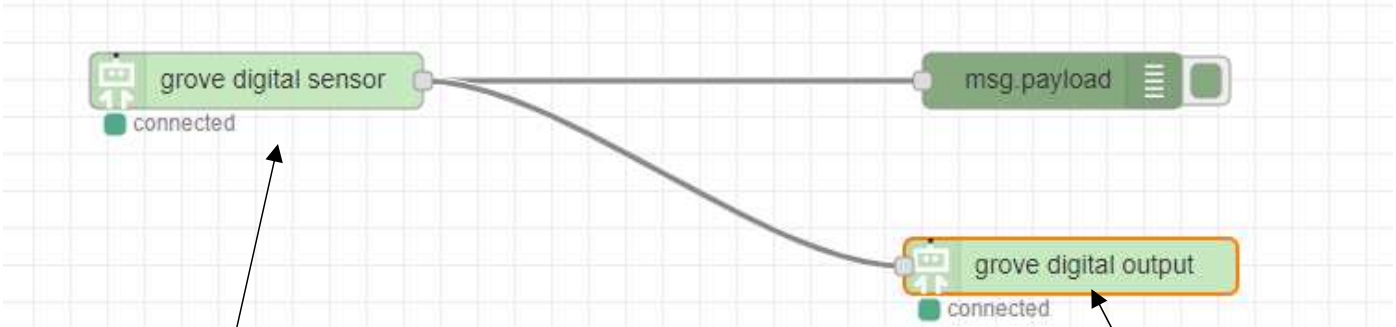
Buttonスイッチと同じように使えます。
センサーの前の人が動くとONします。

赤外線センサーが反応しLED点灯

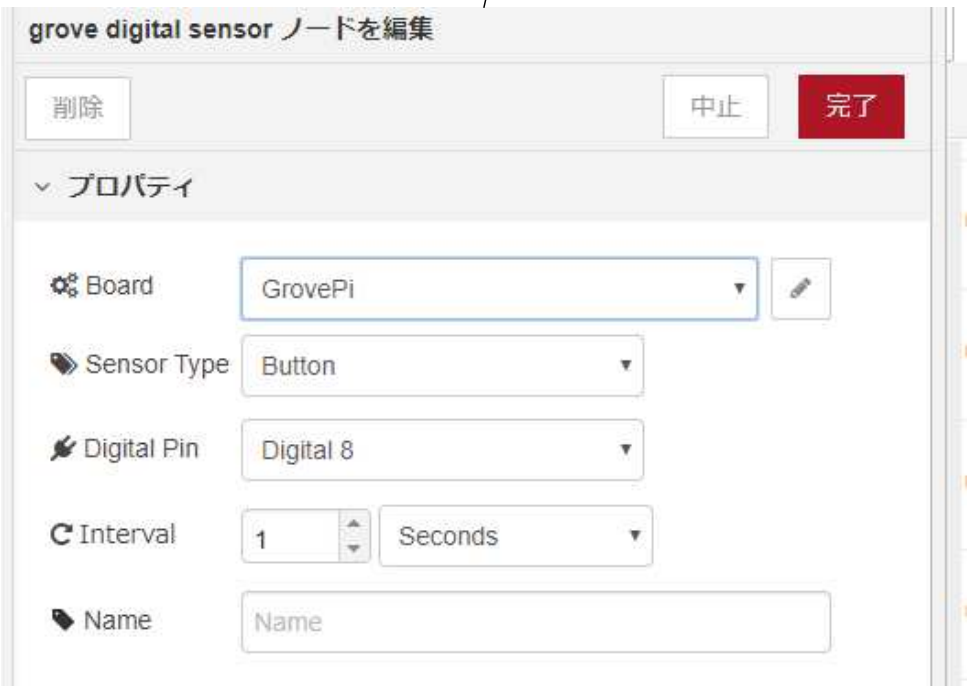


パレット内のノード

赤外線センサーが反応しLED点灯



赤外線センサーの接続された
ポート8に設定。

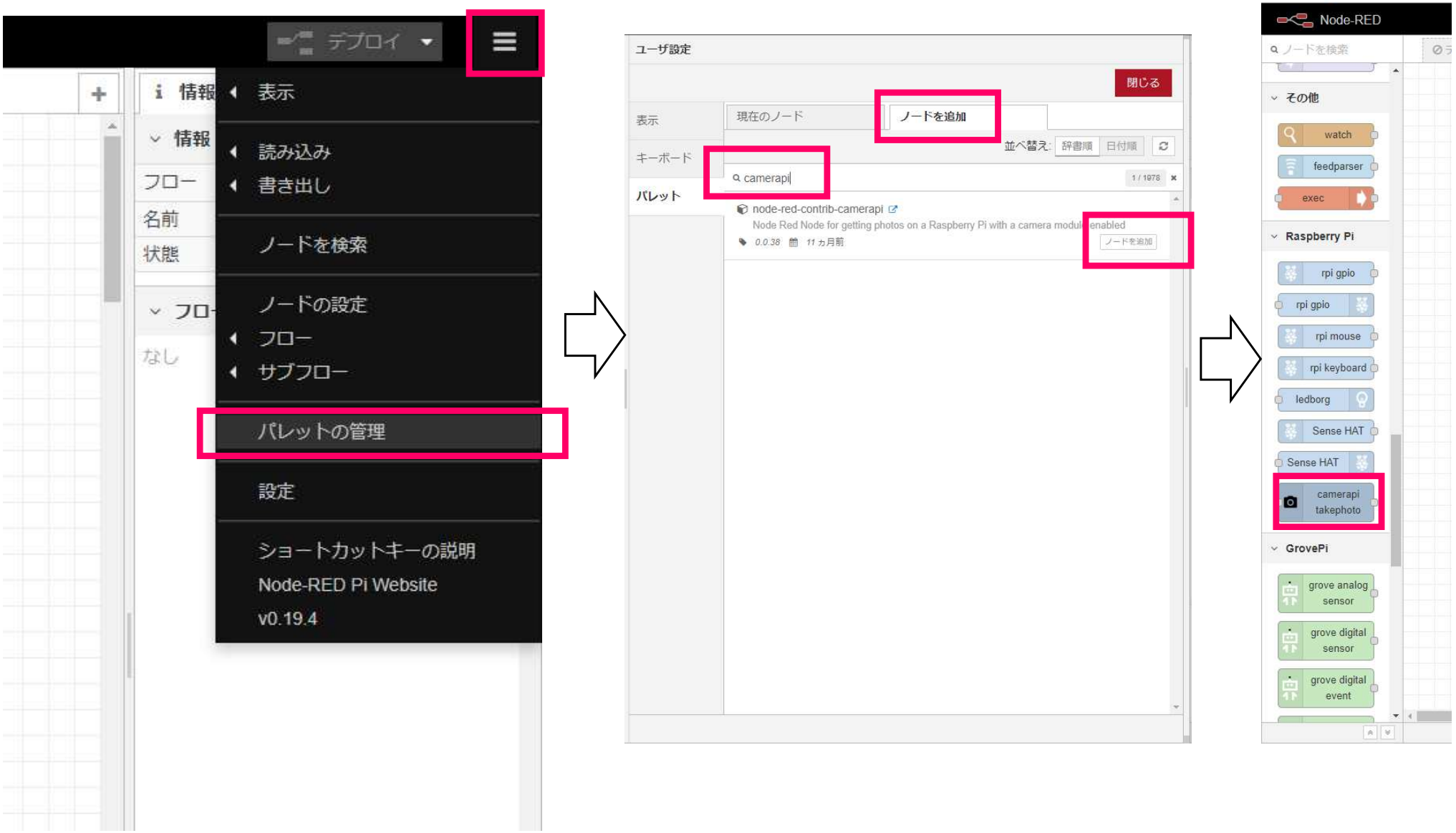


LEDの接続された
ポート5に設定。

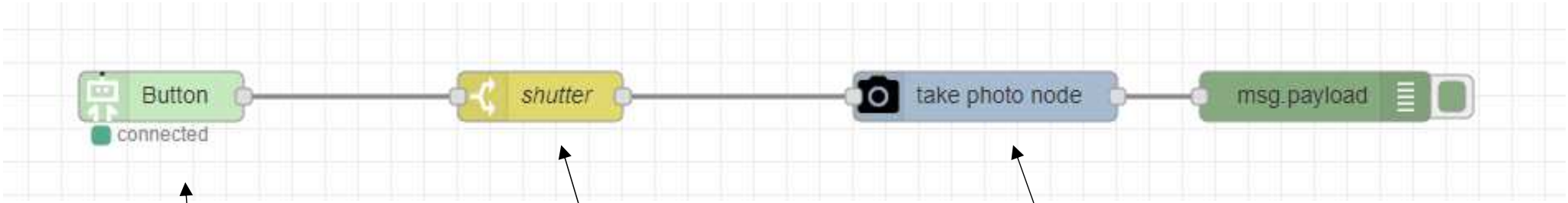
もくじ

camerapiノードのインストール

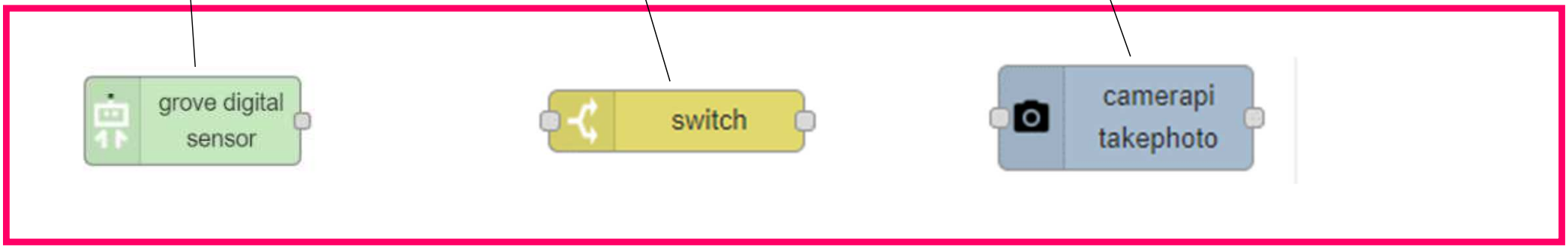
サイドバーメニュー＞パレットの管理＞ノードを追加＞camerapiで検索＞インストール＞追加



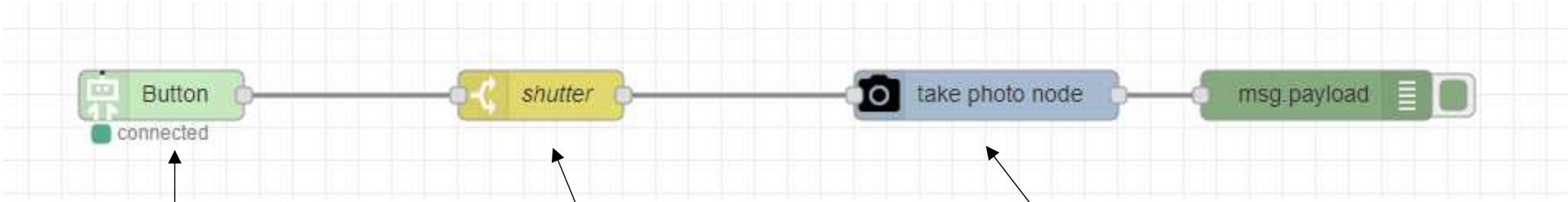
ボタンを押したら写真を撮る



パレット内のノード



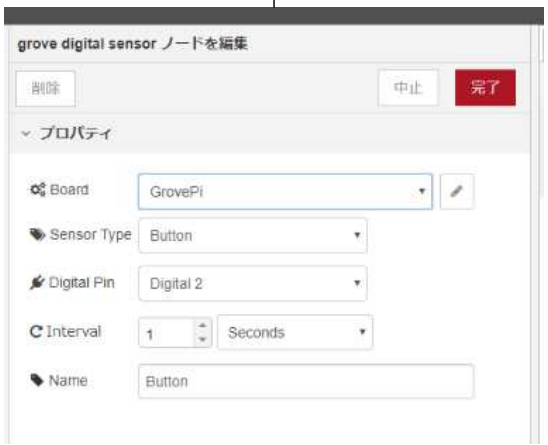
ボタンを押したら写真を撮る

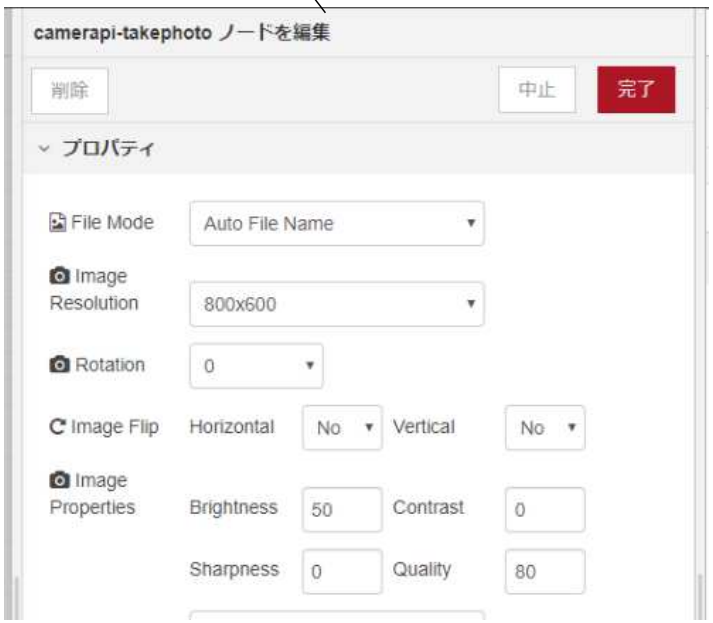


押しボタンスイッチが接続されたポート2に設定。

Buttonノードからtrueデータが渡された時にのみ出力するように設定。

写真データが重くならないように800x600に設定。
デフォルトのままだと/home/pi/Picturesにファイル名自動設定で写真を保存。

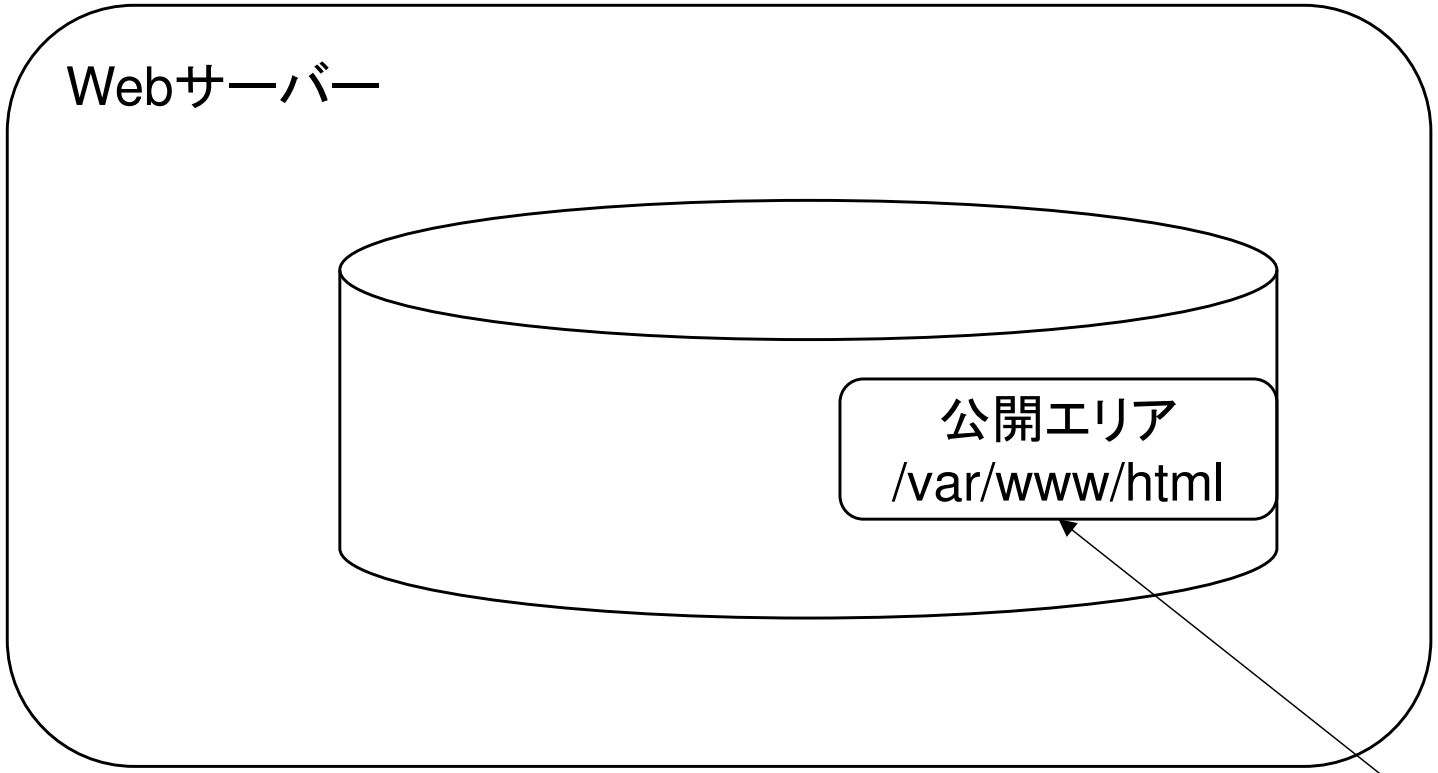




もくじ

Webサーバーのしくみ（超簡略）

Raspbian（ラズパイのOS）にはApache2というWebサーバーが
インストールされています



htmlファイル：
Webコンテンツの
基本となるファイル

cssファイル：
htmlファイルの表
示形式を共通で設
定することができる
ファイル

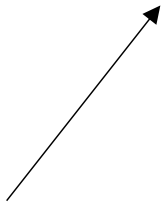
この中のファイルをブラウザで開く

シェアディレクトリの所有権を変える

- 以下のコマンドをコンソールから入力する
  /var/www/htmlの所有者をpi ユーザーに変更する。
  $ sudo chown –R pi /var/www/html

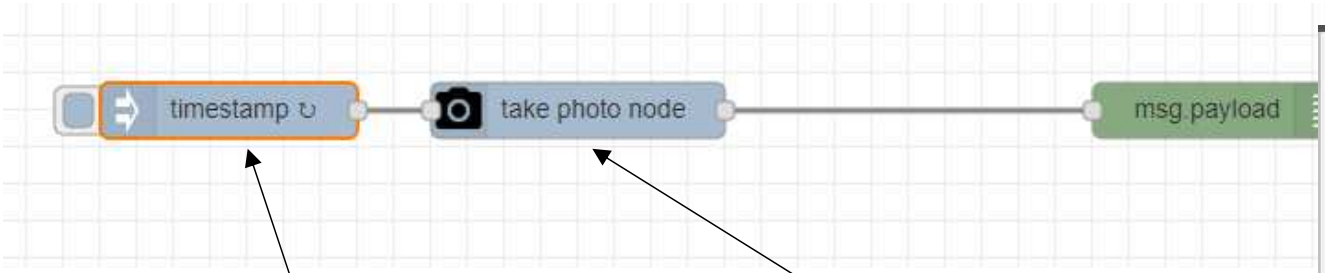
photo.html

```
<html>
<head>
<meta http-equiv="refresh" content="5">
<meta http-equiv="Pragma" content="no-
cache">
<meta http-equiv="Cache-Control" content="no-
cache">
</head>
<body>
hello world!<br>
<img src="photo.jpg?"/>
</body>
</html>
```



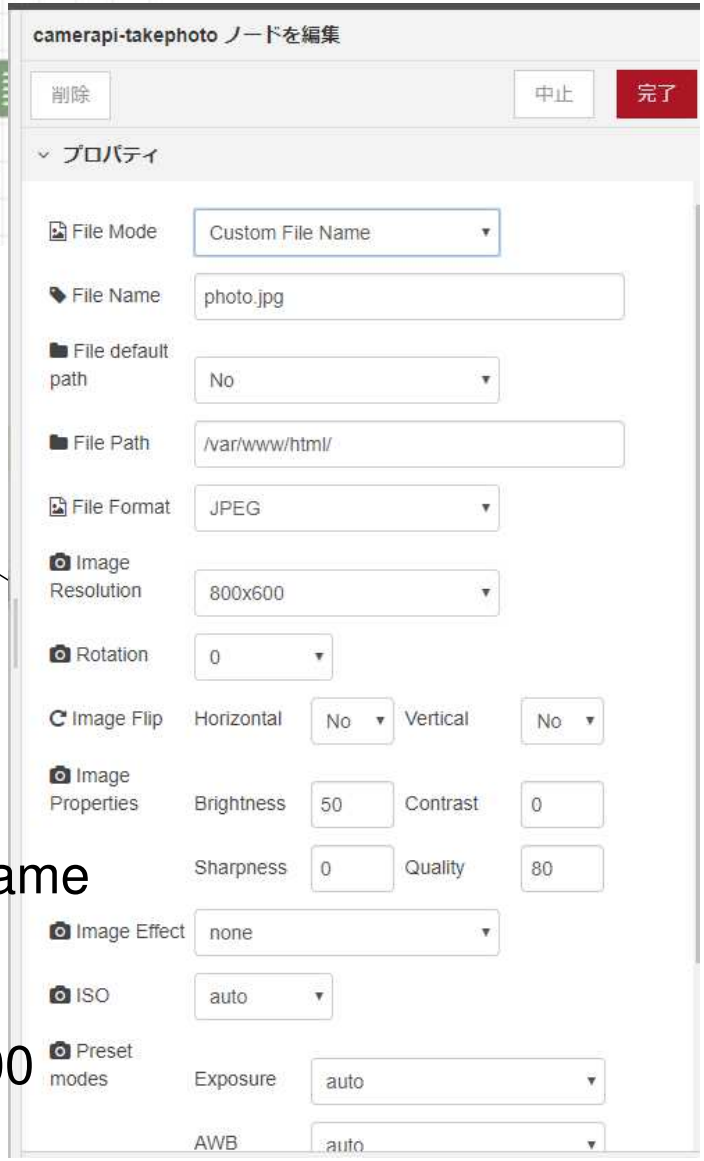
撮影した写真ファイルがこれで表示されます。

10秒間隔でカメラで撮像し更新する



タイムスタンプで
10秒間隔に
msg.payloadを出力

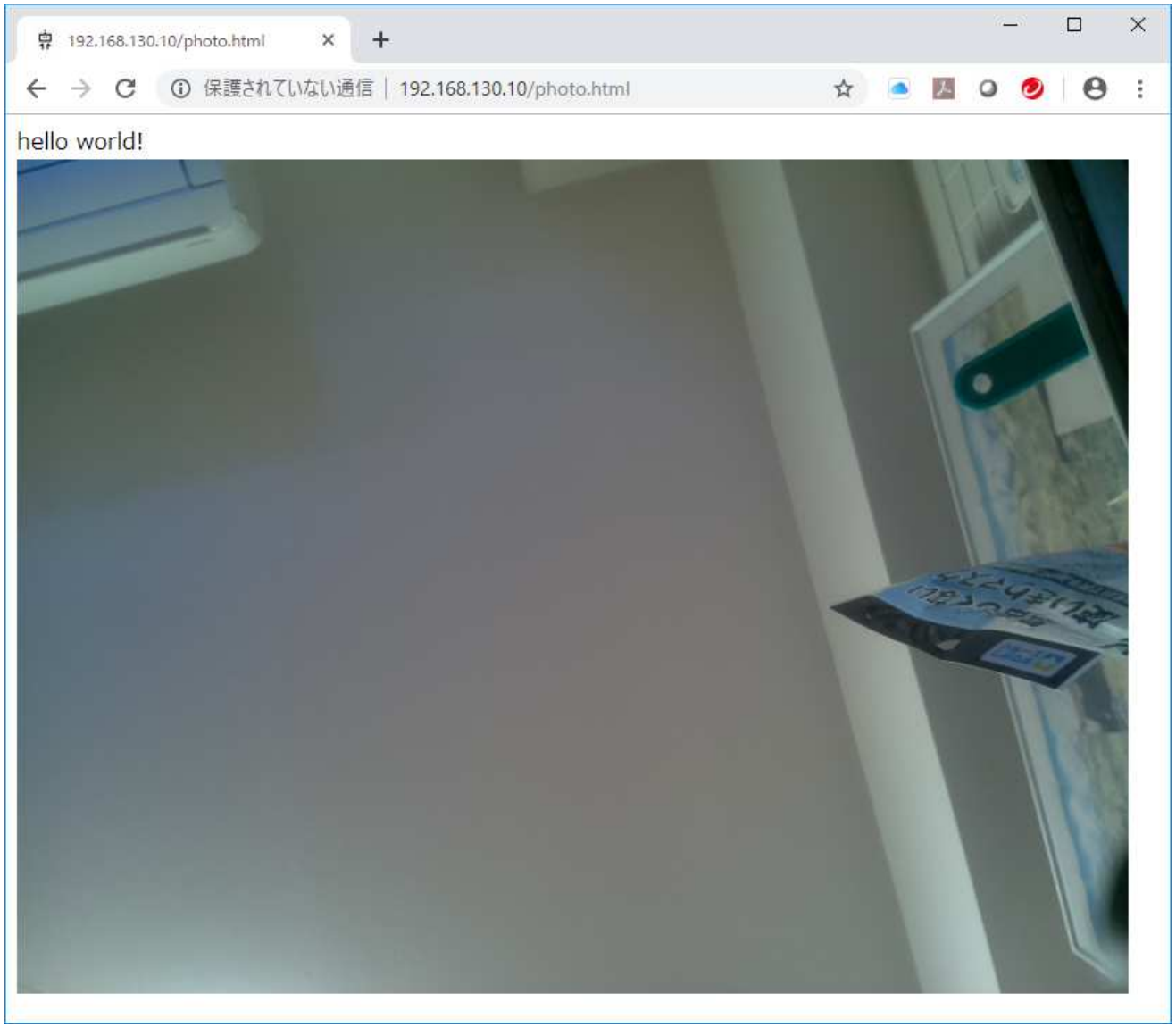
msg.payloadの出力を
受けて、写真を撮影



File Mode: Custom File Name
File Name: photo.jpg
File Path: /var/www/html/
Image Resolution: 800x600

PCのブラウザーで確認

URL [redacted]



もくじ

課題

赤外線センサーで人の動きを検出したら自動的に写真を撮影し、PCのWebブラウザで監視できるシステムを作成してください。

ヒント:

ボタンの代わりに

Grove-Mini PIR Motion Sensorを使う。

写真の保存先を/var/www/html/にする。

自宅や職場で
新たにラズパイやGrovePiを
使ってみるときの注意点

今入手できる最新の環境

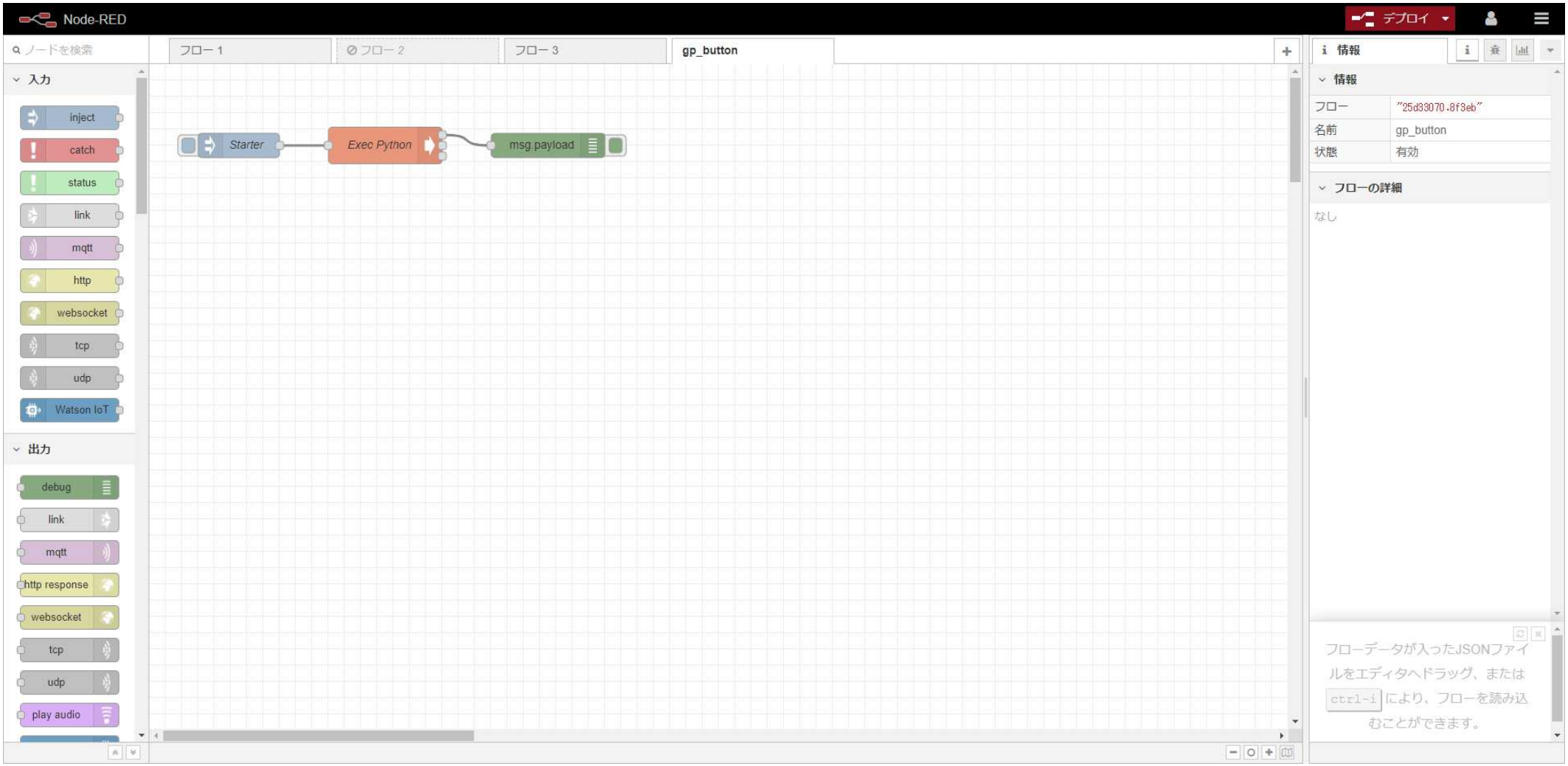
- ラズパイ
  - 最新のRaspbian Stretch with desktop and recommended softwareをダウンロード・インストール
    https://www.raspberrypi.org/downloads/raspbian/

- GrovePi
  - ファームウェアバージョンv.1.3.0
    Node-REDのgrovepi digital sensor でbutton入力を検出できない（回避方法あり）

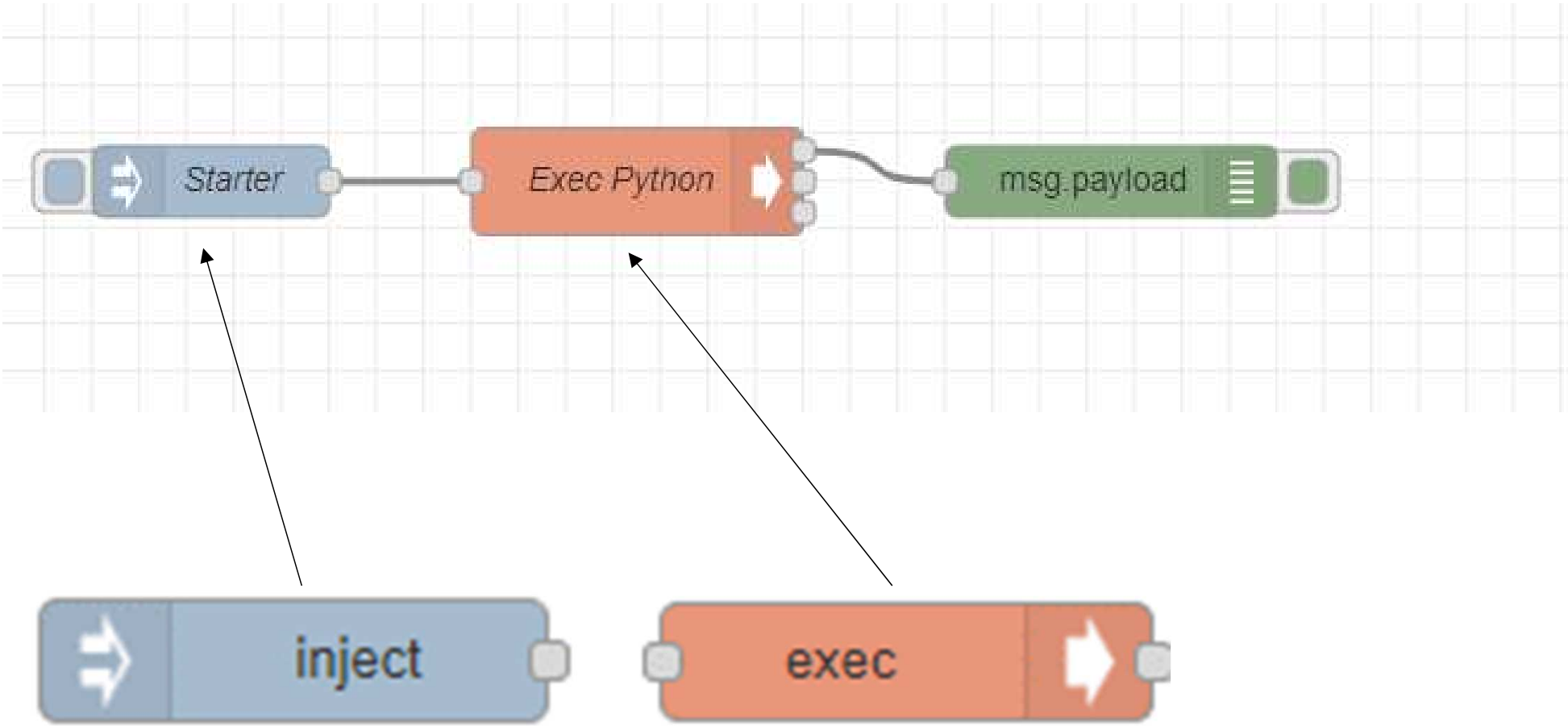
GrovePiライブラリインストール方法

- piユーザーホームディレクトリ（/home/pi）で次のコマンドを実行する
  $ curl -kL dexterindustries.com/update_grovepi | bash
- 念のためGrovePiのファームウェアバージョンを確認するために以下のコマンドを実行する
  $ cd /home/pi/Dexter/GrovePi/Software/Python
  $ python grove_firmware_version_check.py
- GrovePi has firmware version 1.3.0と表示されればOK

Button入力への対応例



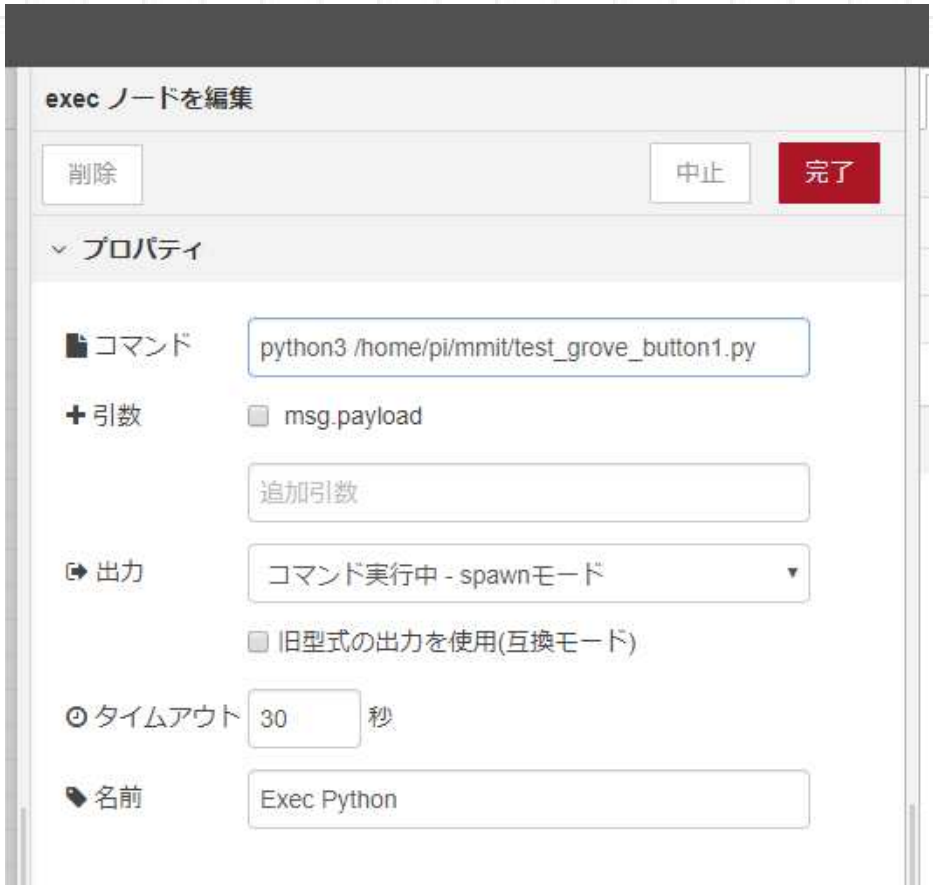
ノードを配置する



プロパティを設定

起動時にpythonプログラムを走らせる

python3でtest_grove_button1.pyを実行

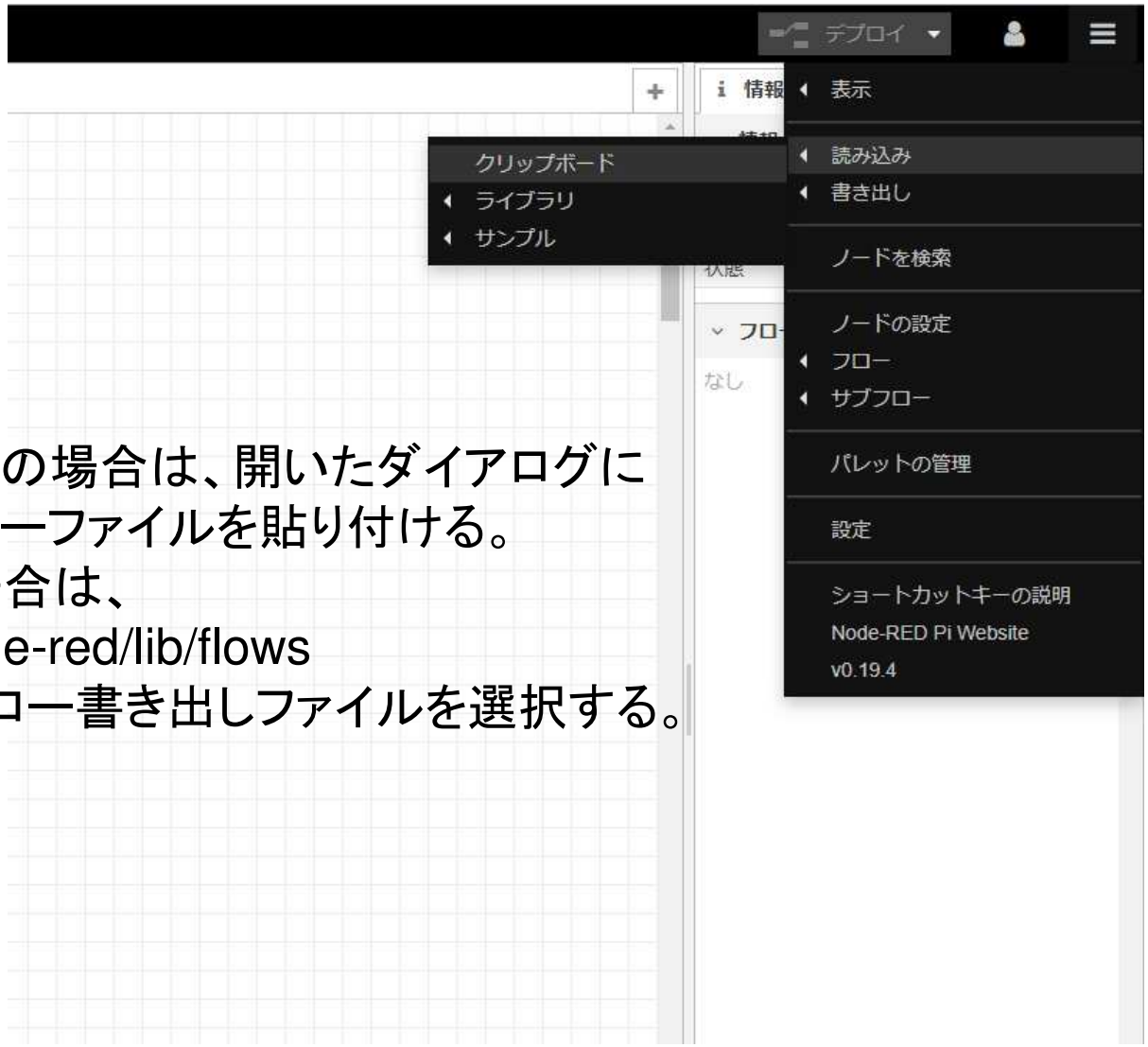


フローの出力

右上の三本線アイコンをクリック＞書き出し＞ライブラリかクリップボード

クリップボードでは、ここから
右クリック＞コピーで
Windowsの.txtにもできる。

作ったフローの読み込み



ダッシュボードの場合は、開いたダイアログに
コピーしたフローファイルを貼り付ける。
ライブラリの場合は、
/home/pi/.node-red/lib/flows
に置かれたフロー書き出しファイルを選択する。

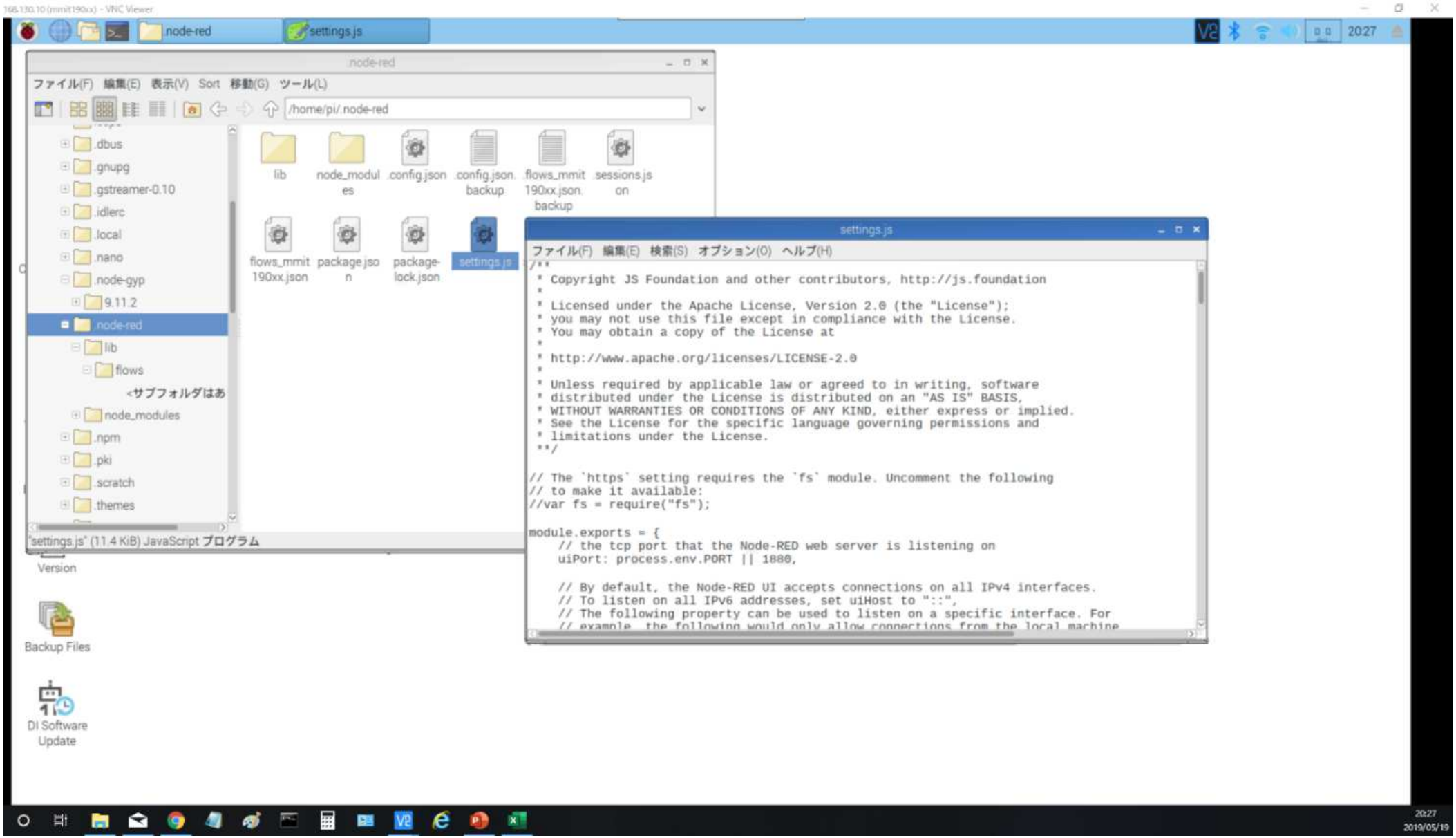
最新バージョンでの機能

- v0.20.5 ではJSONファイル単位でダウンロード（エクスポート）や 読み込み（インポート）する機能が追加されたので、貼り付け以外に選択肢ができました
- JSONでのダウンロードは、大規模なフローのエクスポートにおいて圧倒的に効率が良くなりました。（処理が早い）

Node-REDをwebサーバにする方法

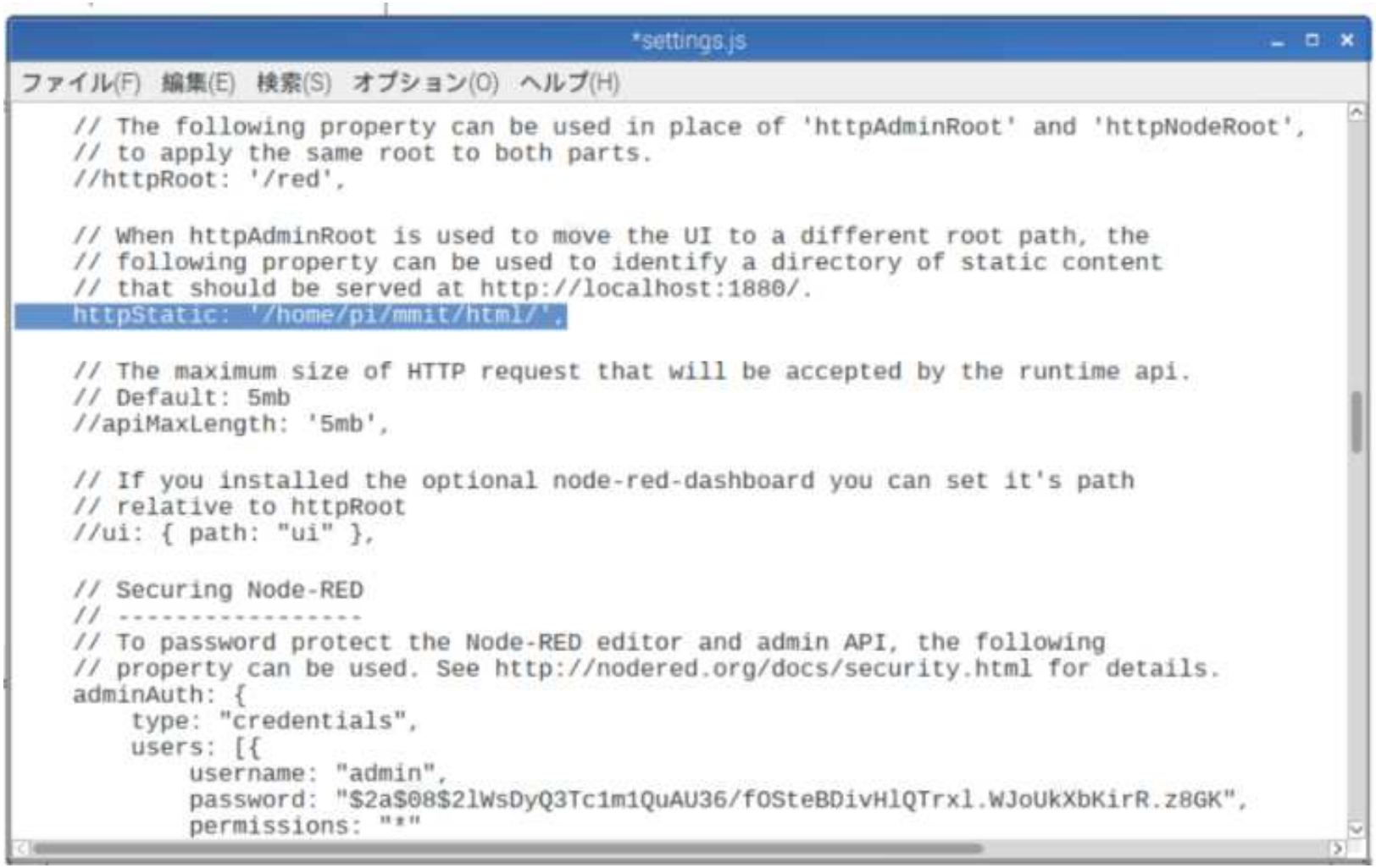
- シェアディレクトリを新規作成（例）
  $ mkdir /home/pi/mmit/html
- photo.htmlとstyle.cssを上記ディレクトリにコピー
- /home/pi/.node-red/setting.jsを変更

settings.jsを開く



シェアディレクトリの設定

反転部分を下図のように修正する。
Node-REDのWebサーバーのシェアフォルダが/home/pi/mmit/htmlになる。
ラズパイを再起動する。



本教材利用上の注意事項