

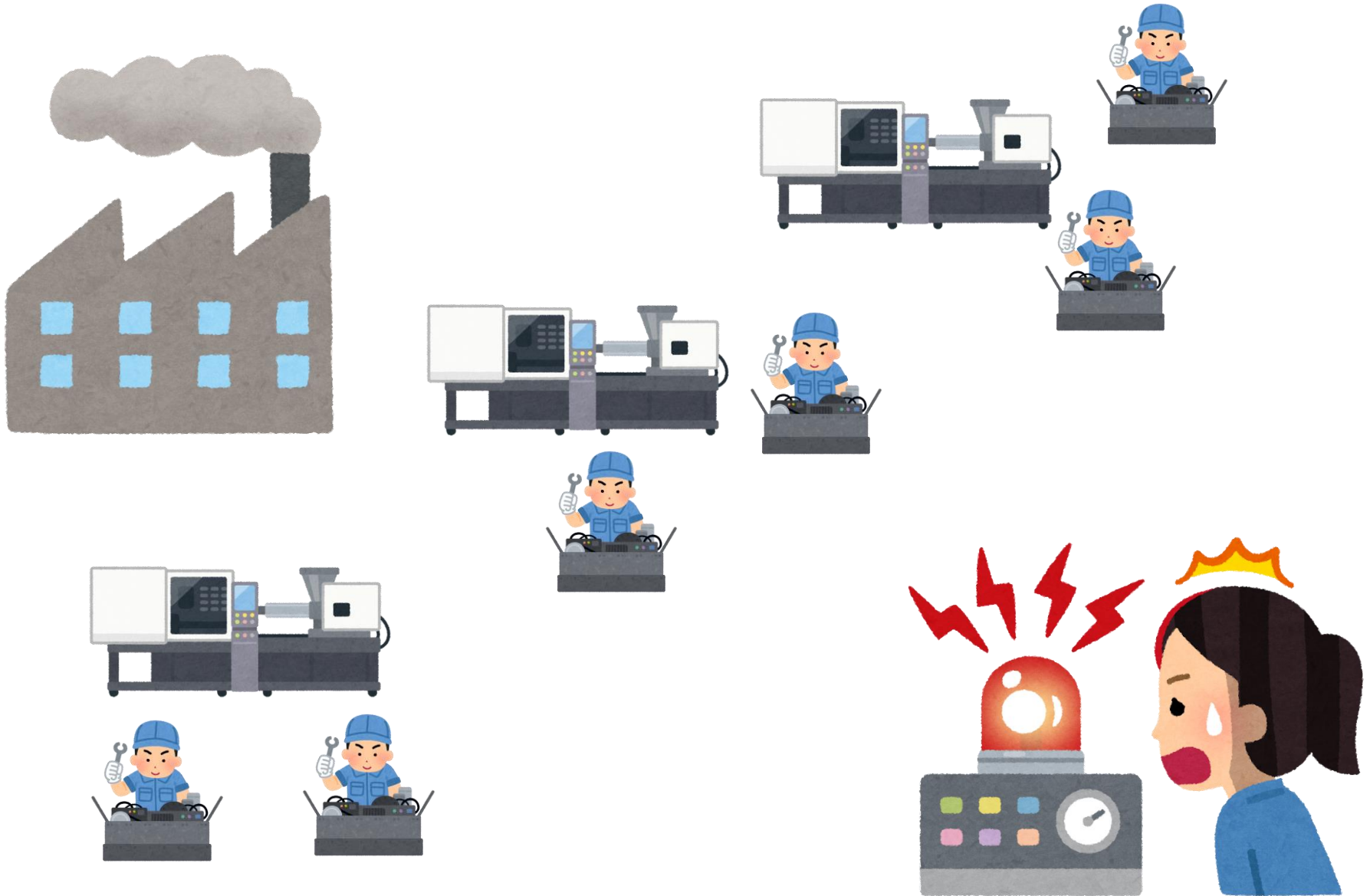
# Node-REDによる見える化の基礎

IoT

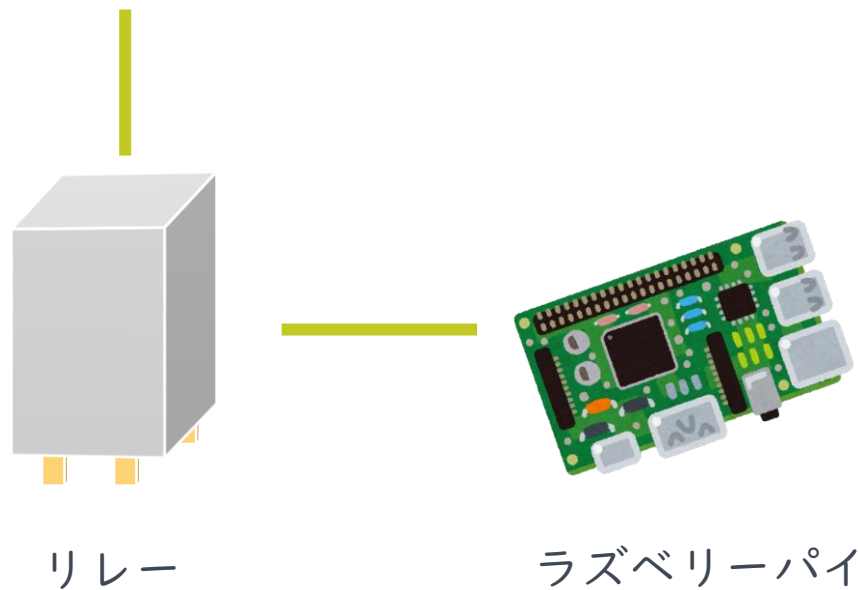
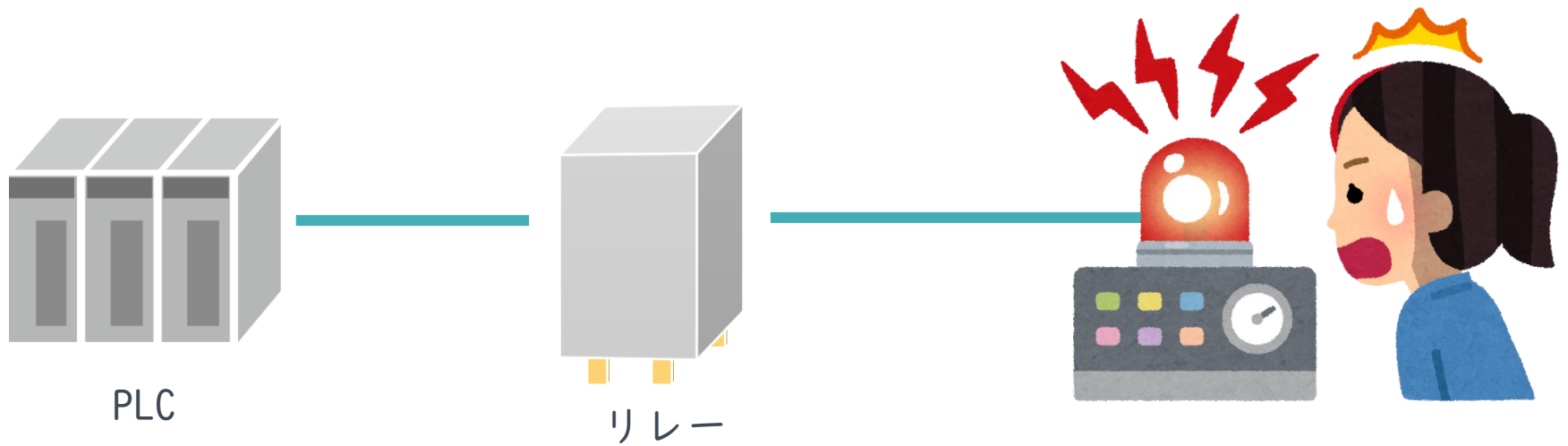
電気回路

Node-RED

# セミナーの目的



# セミナーの目的



見える化

ネットワーク



# セミナーの流れ

## 1. 電気回路の作成

回路図を見ながら結線をすることで、電気回路の基礎を学びます。

## 2. LEDの点灯

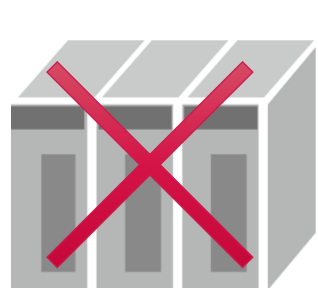
スイッチの状態の取得し、LEDの点灯をさせることで、Node-REDによるプログラミングの基礎を学びます。

## 3. スwitchの状態を見える化

スイッチの状態をブラウザ画面に表示させることで、Node-REDによる見える化の基礎を学びます。

# 1. 電気回路の作成

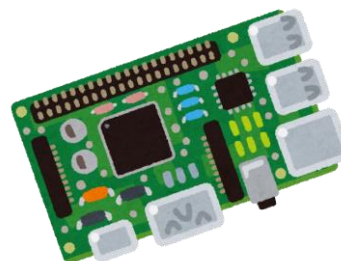
# 電気回路の概要



PLC



リレー



ラズベリーパイ



LED



指



スイッチ

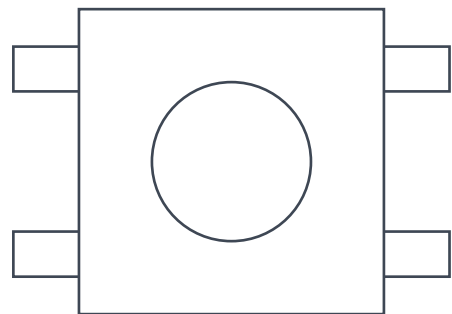
作成する電気回路

- ① スイッチの状態を取り込む
- ② LEDを点灯させる

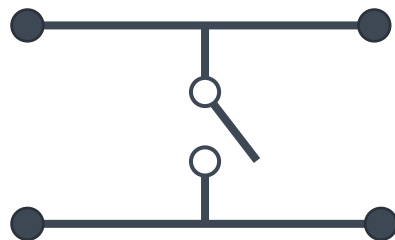
# 部品について



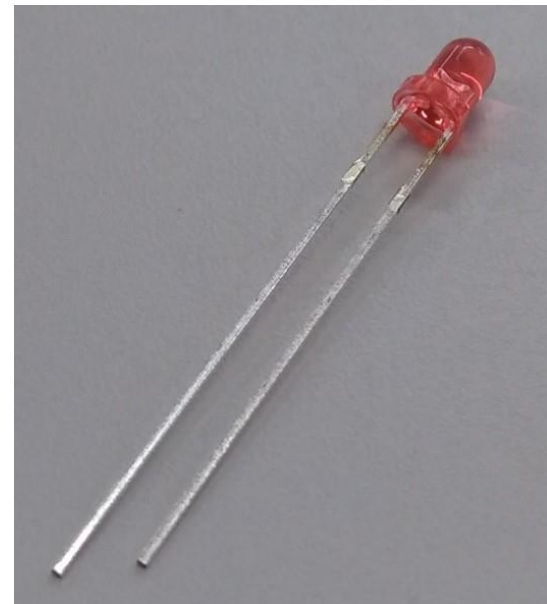
タクトスイッチ



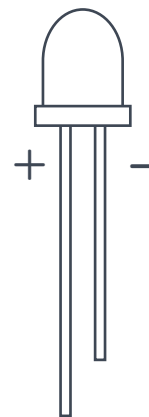
平面図



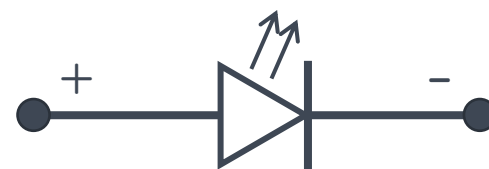
回路図



LED

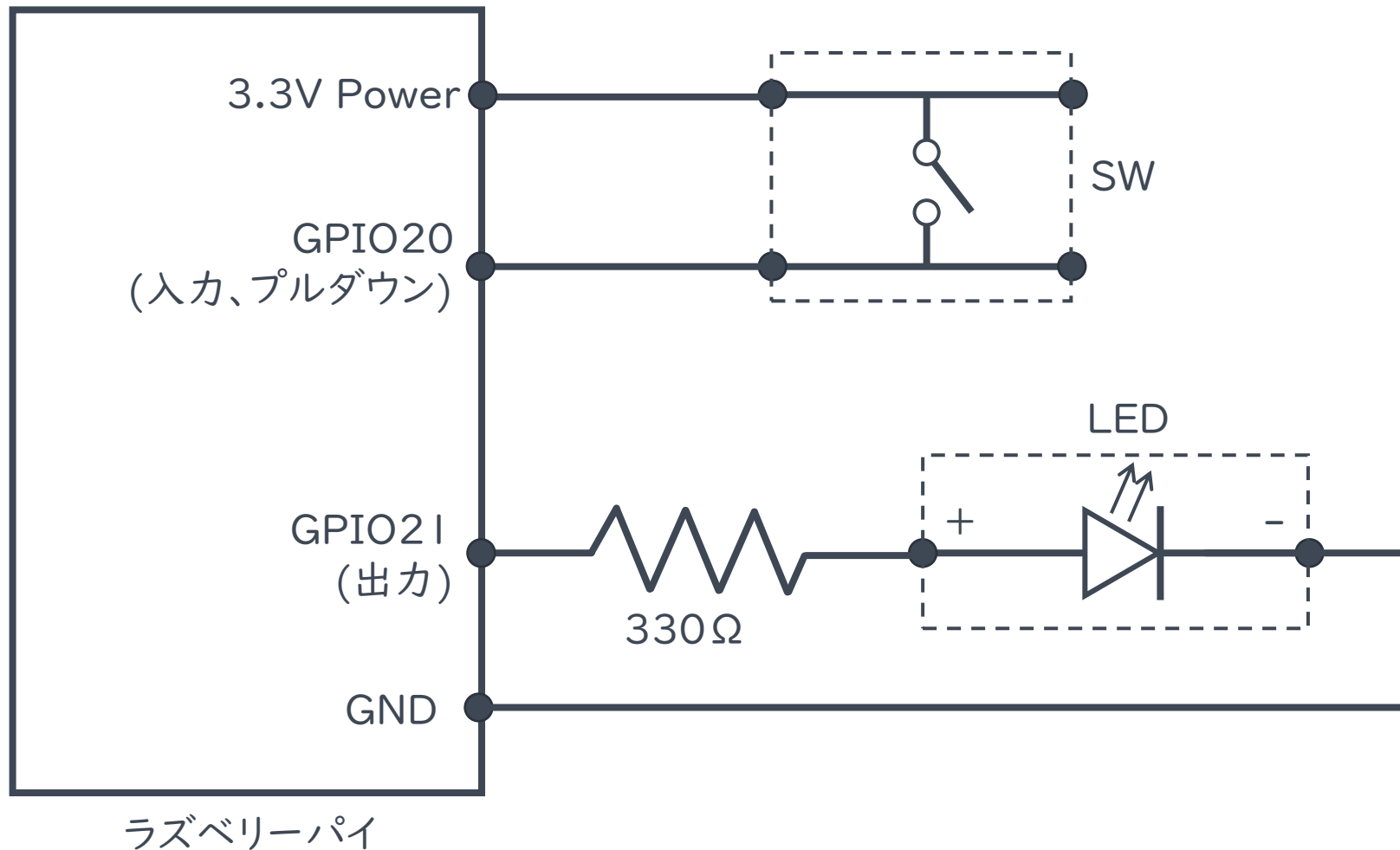


側面図



回路図

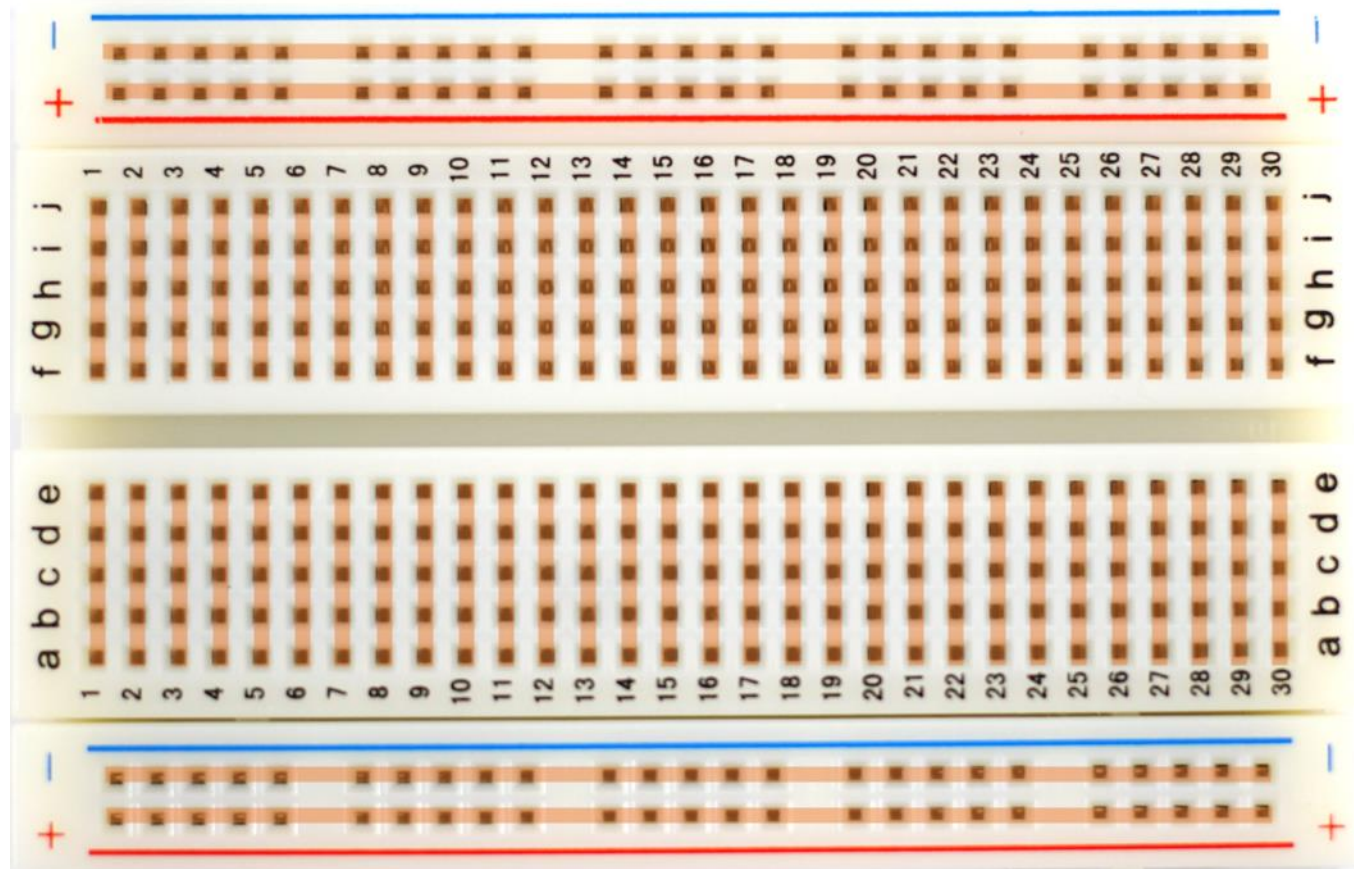
# 回路図





# ブレッドボードについて

- ・ オレンジ色の部分が内部で繋がっています。
- ・ ジャンパーワイヤーや抵抗などを直接ブレッドボードに挿して使います。  
(はんだ付けは不要です)



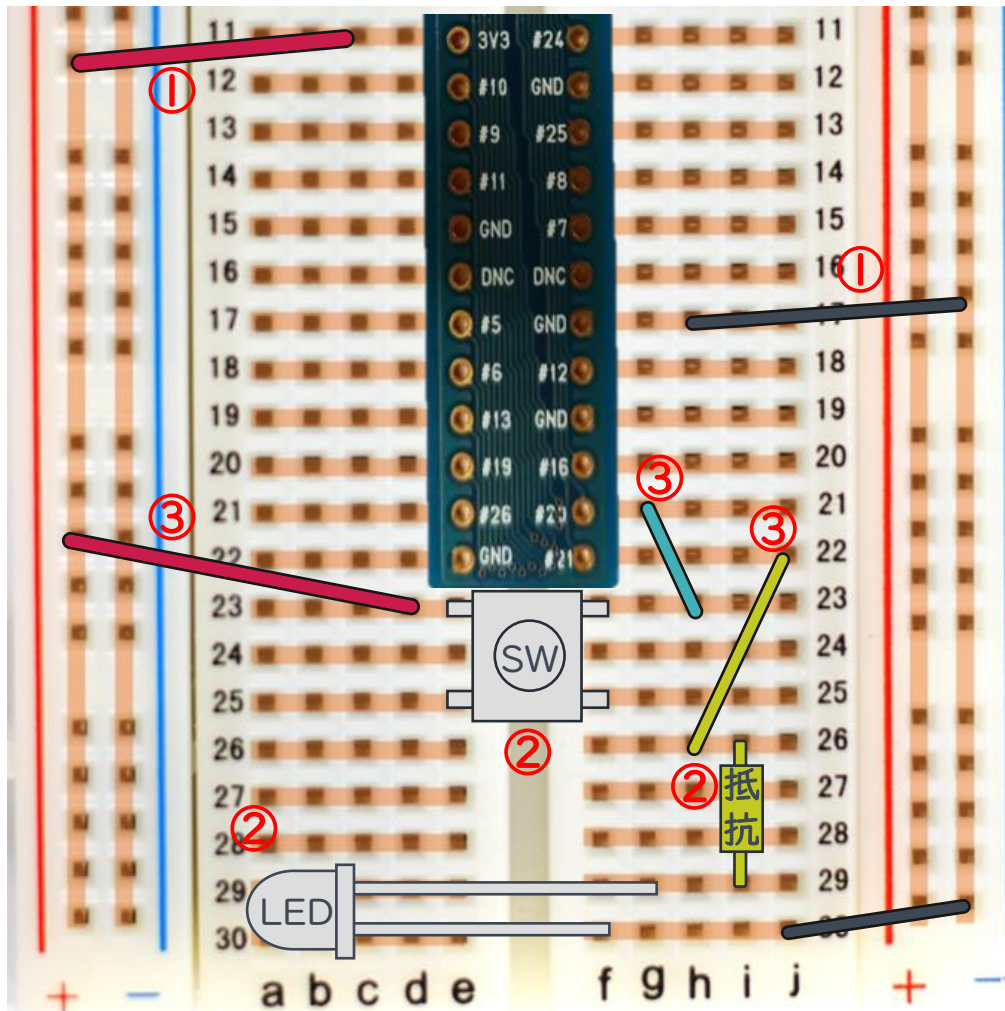
# ラズベリーパイ ピン配置

3.3V Power - 1 ○	○ 2 - 5V Power
SDA1 - GPIO02 - 3 ○	○ 4 - 5V Power
SCL1 - GPIO03 - 5 ○	○ 6 - Ground
GPIO04 - 7 ○	○ 8 - GPIO14 - TxD
Ground - 9 ○	○ 10 - GPIO15 - RxD
GPIO17 - 11 ○	○ 12 - GPIO18
GPIO27 - 13 ○	○ 14 - Ground
GPIO22 - 15 ○	○ 16 - GPIO23
3.3V Power - 17 ○	○ 18 - GPIO24
MOSI - GPIO10 - 19 ○	○ 20 - Ground
MISO - GPIO09 - 21 ○	○ 22 - GPIO25
SCLK - GPIO11 - 23 ○	○ 24 - GPIO8 - CE0
Ground - 25 ○	○ 26 - GPIO7 - CE1
SD - 27 ○	○ 28 - SC
GPIO05 - 29 ○	○ 30 - Ground
GPIO06 - 31 ○	○ 32 - GPIO12
GPIO13 - 33 ○	○ 34 - Ground
GPIO19 - 35 ○	○ 36 - GPIO16
GPIO26 - 37 ○	○ 38 - GPIO20
Ground - 39 ○	○ 40 - GPIO21

- Power  
3.3V/5V を常時出力するピン。  
電池での**プラス極**のようなもの。
- Ground  
基準電圧との差が**0V**であるピン。  
電池での**マイナス極**のようなもの。
- GPIO  
スイッチの状態取得やLEDの点灯など  
**汎用的な入出力**を行うためのピン。
- SDA/SCL  
**I2C**と呼ばれるシリアル通信用のピン。
- MOSI/MISO/SCLK/CE  
**SPI**と呼ばれるシリアル通信用のピン。

# スイッチとLEDの配線

スイッチとLEDをラズベリーパイに接続します。



- ① 3.3V電源とGroundを接続し
- ② スイッチ、LED、抵抗を取り付け
- ③ 残りの配線を接続します。

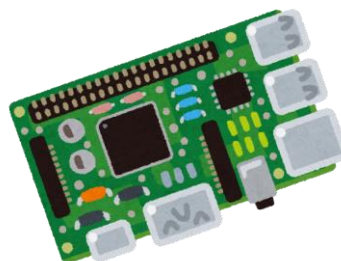
配線例

## 2. LEDの点灯

# 動作の流れ



スイッチON



ラズベリーパイ

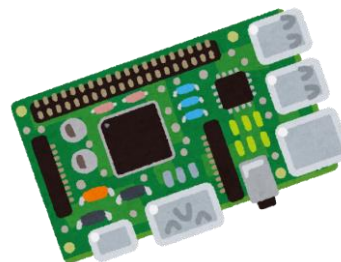


LED

Node-REDによるプログラミング



スイッチOFF



ラズベリーパイ

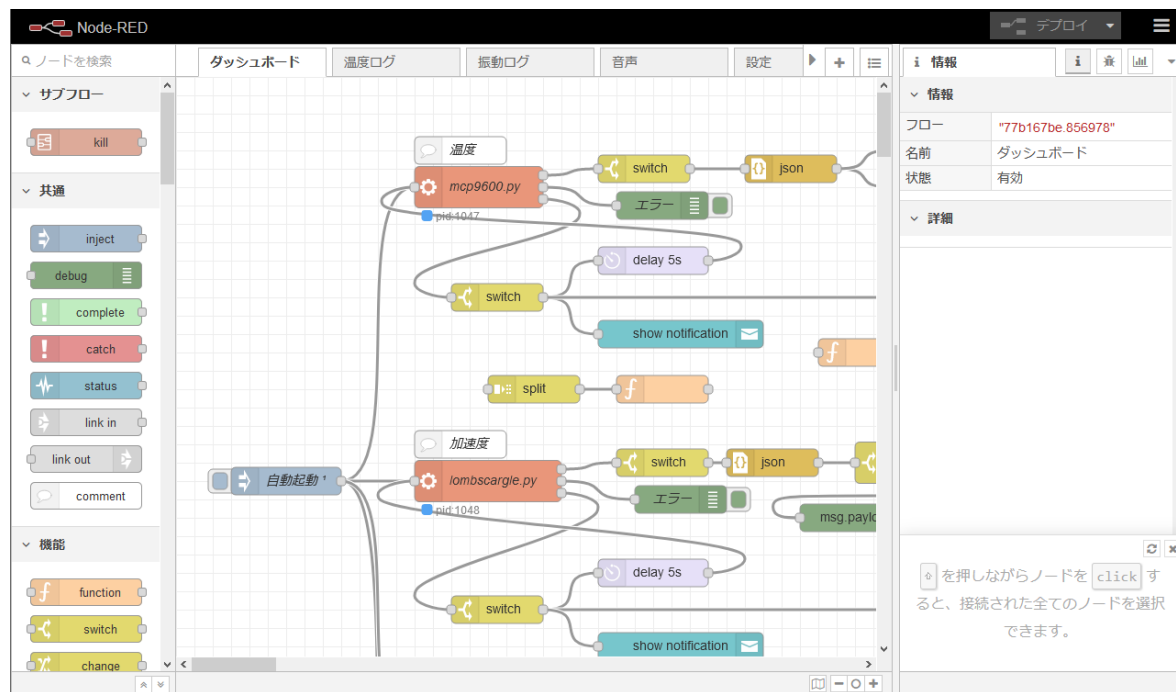


LED

# Node-RED とは？

Node-REDはフローベースのビジュアルプログラミングツールで、Webブラウザからフローを編集できる。ハードウェアデバイスやAPI、クラウドサービスなどを線で結んでいくため、IoTアプリ開発に使える。

もともとはIBMが開発したものだが、JS Foundationに寄贈されたため、今はOSSとなっている。



Node-REDによるプログラミングを行っていきます。

# Node-RED 開発画面を表示

Webブラウザを開いて以下のURLにアクセスしてください。

- Webブラウザ
  - Google Chrome
  - FireFox
- URL

`http://(IPアドレス):1880`  
(IPアドレスはラズベリーパイ毎に異なります。)
- アクセス後

Node-RED開発画面が表示されます。

# Node-RED 開発画面について

デプロイにより確定

ワークスペース

デプロイ

The screenshot displays the Node-RED development environment. On the left, a sidebar lists various node categories: 'サブフロー' (Subflows) with a 'kill' node, '共通' (Common) with 'inject', 'debug', 'complete', 'catch', and 'status' nodes, and '機能' (Functions) with 'function', 'switch', and 'change' nodes. The central workspace contains two flows. The top flow, titled '温度' (Temperature), starts with an 'mcp9600.py' node (pid:1047) connected to a 'switch' node. The bottom flow, titled '加速度' (Acceleration), starts with a 'lombscargle.py' node (pid:1048) connected to a 'switch' node. Both flows include 'switch' nodes, 'delay 5s' nodes, and 'show notification' nodes. The right sidebar shows the '情報' (Info) panel for the selected flow, displaying details such as 'フロー' (Flow) ID, '名前' (Name) 'ダッシュボード', and '状態' (Status) '有効'.

ドラッグ&ドロップ

用意された機能(ノード)



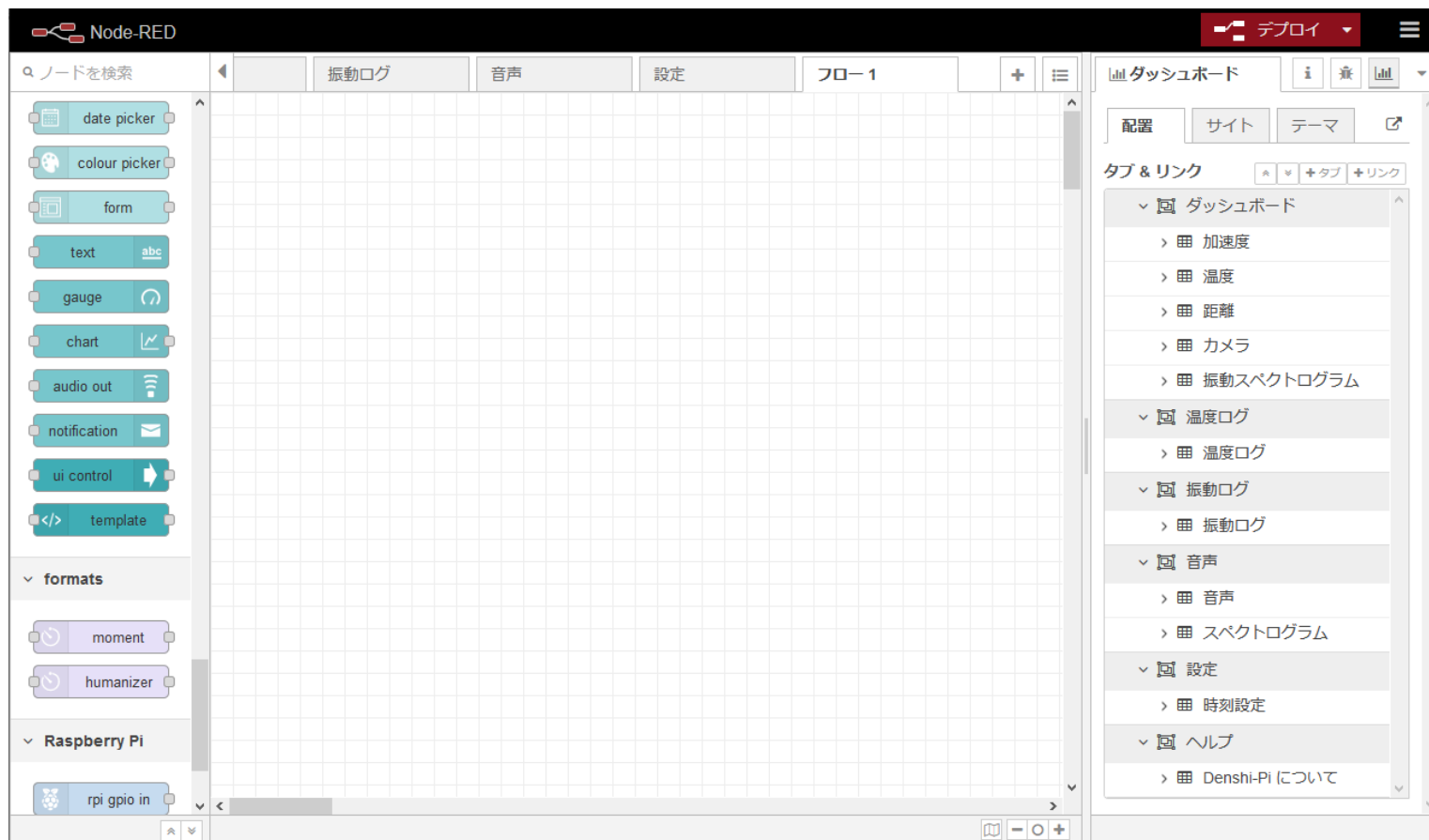
# Node-RED 新規ワークスペース①

セミナー用の新規ワークスペースを作成します。

The screenshot shows the Node-RED web interface. On the left, there is a sidebar with a search bar and several categories of nodes: サブフロー (kill), 共通 (inject, debug, complete, catch, status, link in, link out, comment), and 機能 (function, switch, change). The main workspace contains a flow diagram starting with a blue '自動起動' node. This node branches into two paths. The top path includes a red 'mcp9600.py' node (pid:1047) connected to a yellow 'switch' node, which then connects to a green 'エラー' node. The bottom path includes a red 'lombscargle.py' node (pid:1048) connected to a yellow 'switch' node, which then connects to a green 'エラー' node. Both 'エラー' nodes are connected to a purple 'delay 5s' node, which then connects to a blue 'show notification' node. A red dashed circle highlights a '+' button in the top right corner of the workspace, with a red callout box containing the Japanese text 'クリック' (Click) pointing to it. The right sidebar shows an '情報' (Info) panel with the text 'e.856978"ボード' and a '詳細' (Details) panel with the text '設定により、ヒントの表示/非表示を変更できます。' (Depending on the settings, you can change the display/non-display of hints.)

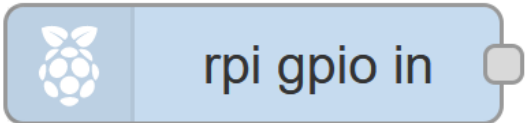
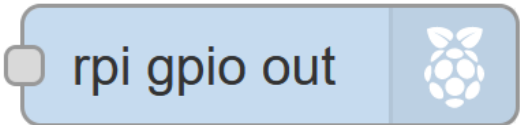
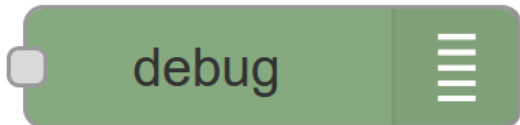
# Node-RED 新規ワークスペース②

セミナー用の新規ワークスペースが作成されました。



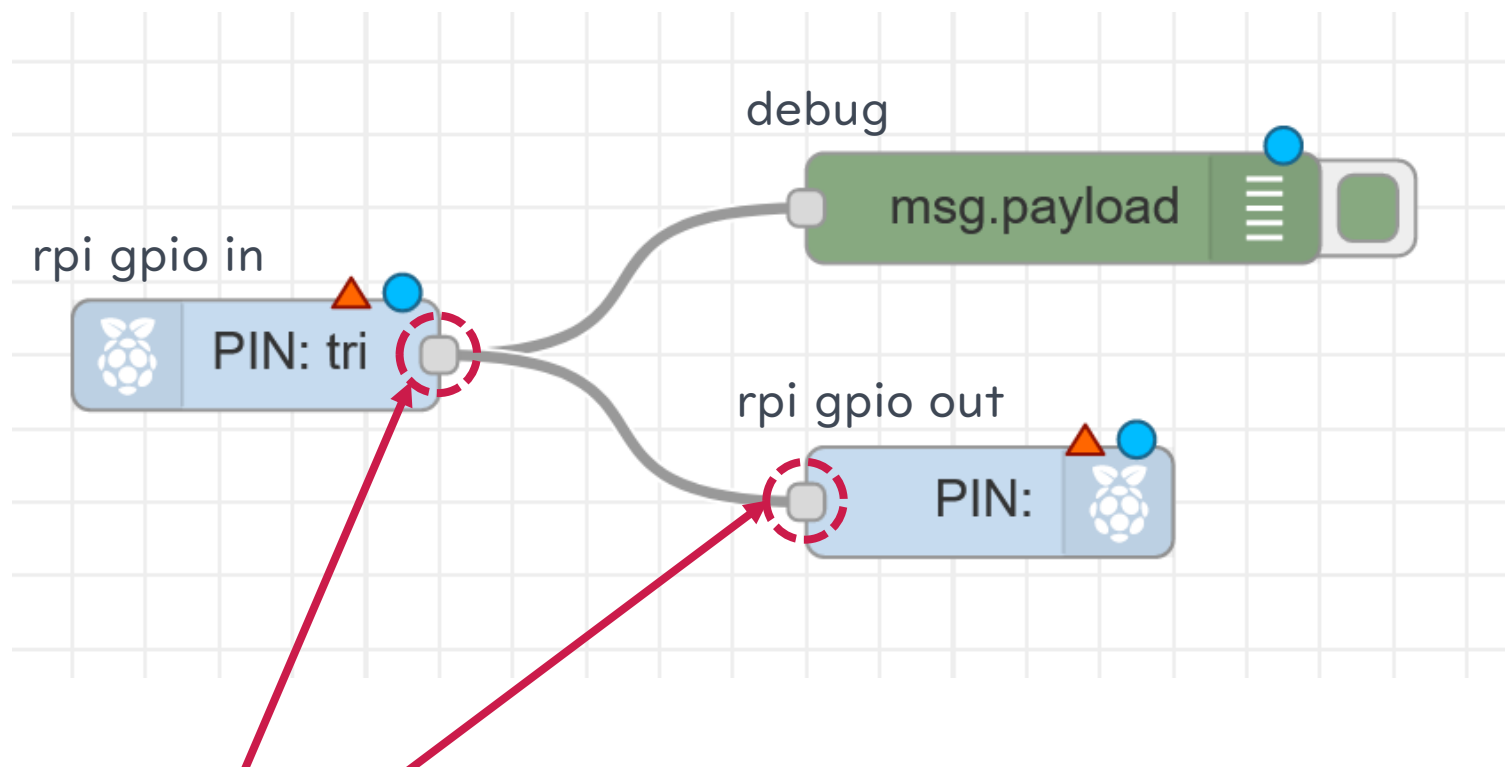
# Node-RED 使用するノード

各ノードをダブルクリックすることで詳細設定ができます。

項目	ノード	機能	説明
Raspberry Pi	 The image shows the 'rpi gpio in' node in Node-RED. It is a light blue rounded rectangle with a Raspberry Pi logo on the left and a small grey square on the right.	GPIO input	GPIOの入力用ノード。 スイッチの状態取得に使用。
Raspberry Pi	 The image shows the 'rpi gpio out' node in Node-RED. It is a light blue rounded rectangle with a small grey square on the left and a Raspberry Pi logo on the right.	GPIO output	GPIOの出力用ノード。 LEDのON/OFFに使用。
共通	 The image shows the 'debug' node in Node-RED. It is a dark green rounded rectangle with a small grey square on the left and a hamburger menu icon on the right.	デバッグ	デバッグ用のノード。 フローを流れるデータの表示に使用。

# Node-RED ノードの配線

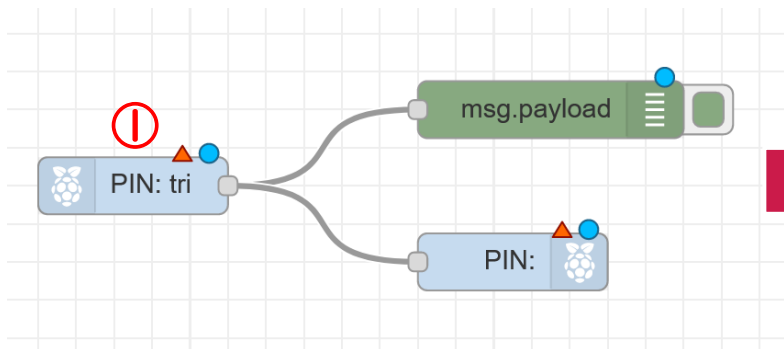
ノードをワークスペースに配置し、ノードの間を線(フロー)で接続します。  
データはノードの右から出力され、左から入力されます。



端子間をドラッグ&ドロップで接続

# Node-RED ノードの設定①

スイッチの状態を取得する設定を行います。



The screenshot shows the configuration window for the 'rpi-gpio in' node. The title bar says 'rpi-gpio in ノードを編集'. There are buttons for '削除', '中止', and '完了'. Below is a 'プロパティ' section with a 'Pin' dropdown. A table of GPIO pins is shown with radio buttons for selection. The pin '38 - GPIO20' is selected and has a red circle with the number 2 around it. At the bottom, there is a 'Resistor' dropdown set to 'pulldown' (with a red circle 3), a 'Debounce' field set to '25 ms', and a checkbox 'Read initial state of pin on deploy/restart?' (with a red circle 4).

3.3V Power - 1		2 - 5V Power	
SDA1 - GPIO02 - 3	<input type="radio"/>	4 - 5V Power	<input type="radio"/>
SCL1 - GPIO03 - 5	<input type="radio"/>	6 - Ground	<input type="radio"/>
GPIO04 - 7	<input type="radio"/>	8 - GPIO14 - TxD	<input type="radio"/>
Ground - 9	<input type="radio"/>	10 - GPIO15 - RxD	<input type="radio"/>
GPIO17 - 11	<input type="radio"/>	12 - GPIO18	<input type="radio"/>
GPIO27 - 13	<input type="radio"/>	14 - Ground	<input type="radio"/>
GPIO22 - 15	<input type="radio"/>	16 - GPIO23	<input type="radio"/>
3.3V Power - 17	<input type="radio"/>	18 - GPIO24	<input type="radio"/>
MOSI - GPIO10 - 19	<input type="radio"/>	20 - Ground	<input type="radio"/>
MISO - GPIO09 - 21	<input type="radio"/>	22 - GPIO25	<input type="radio"/>
SCLK - GPIO11 - 23	<input type="radio"/>	24 - GPIO8 - CE0	<input type="radio"/>
Ground - 25	<input type="radio"/>	26 - GPIO7 - CE1	<input type="radio"/>
SD - 27	<input type="radio"/>	28 - SC	<input type="radio"/>
GPIO05 - 29	<input type="radio"/>	30 - Ground	<input type="radio"/>
GPIO06 - 31	<input type="radio"/>	32 - GPIO12	<input type="radio"/>
GPIO13 - 33	<input type="radio"/>	34 - Ground	<input type="radio"/>
GPIO19 - 35	<input type="radio"/>	36 - GPIO16	<input type="radio"/>
GPIO26 - 37	<input type="radio"/>	38 - GPIO20	<input checked="" type="radio"/>
Ground - 39	<input type="radio"/>	40 - GPIO21	<input type="radio"/>

① ノードをダブルクリックする。

② 38-GPIO20を選択する。

③ pulldownを選択する。

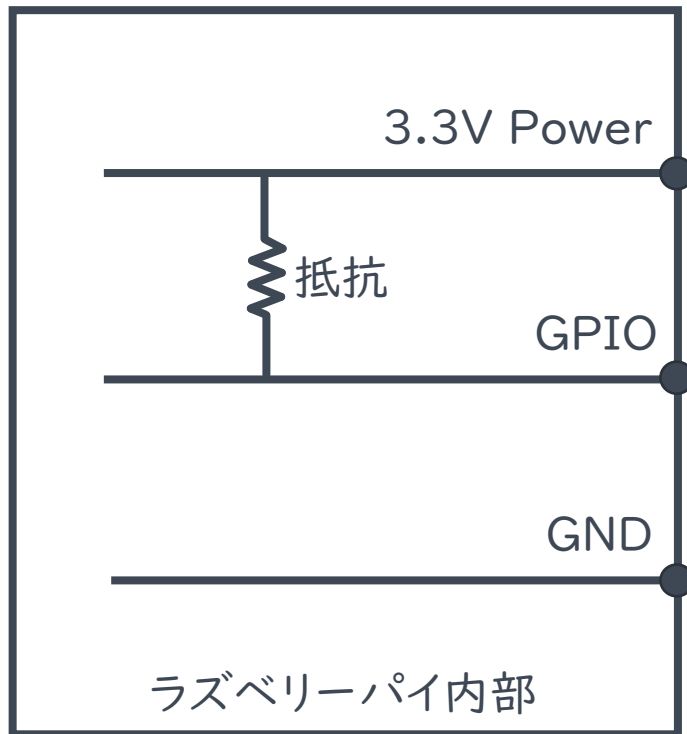
④ チェックを入れる。

⑤ 完了をクリックする。

# プルアップ/ダウンについて

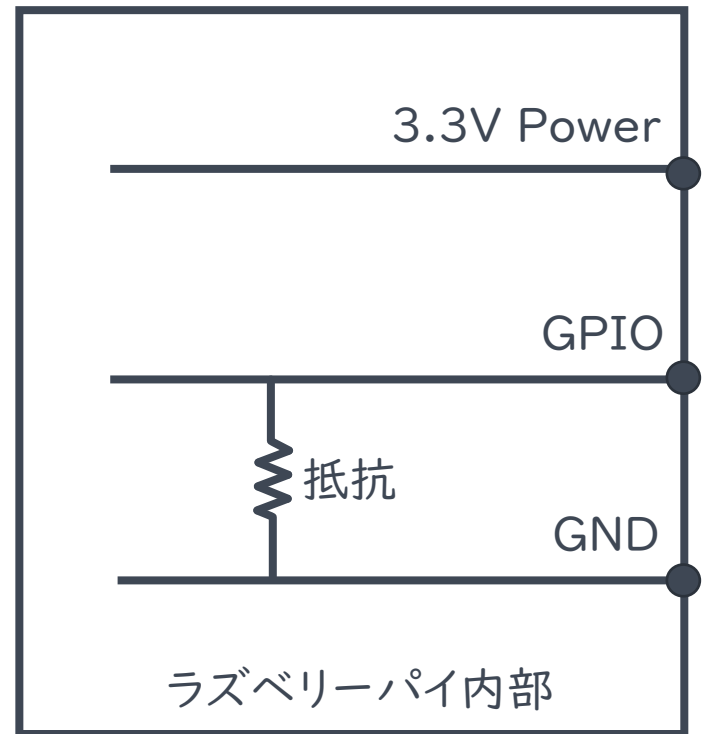
プルアップ/ダウンについて説明します。

## プルアップ



内部でPowerと接続

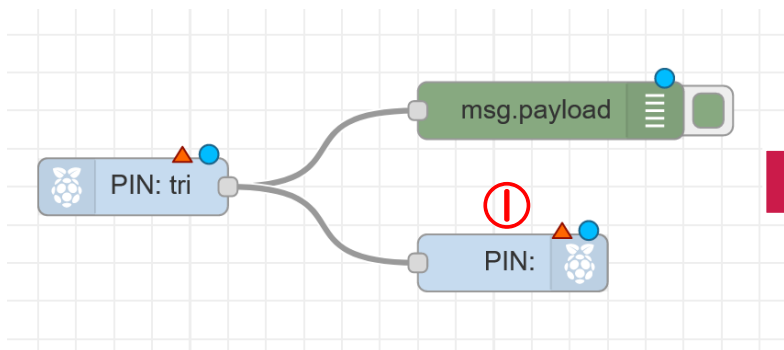
## プルダウン



内部でGNDと接続

# Node-RED ノードの設定②

LEDを点灯させる設定を行います。



① ノードをダブルクリックする。

② 40-PIO21を選択する。

③ Digital outputを選択する。

④ 完了をクリックする。

The screenshot shows the 'rpi-gpio out ノードを編集' dialog box. At the top right, there is a red circle with the number 4 over the '完了' (Done) button. Below the title bar, there are buttons for '削除' (Delete), '中止' (Cancel), and '完了' (Done). A 'プロパティ' (Properties) section is visible. Under the 'Pin' section, there is a table of pins with radio buttons next to them. The pin '40 - GPIO21' is selected, indicated by a red circle with the number 2. Below the table, the 'Type' dropdown is set to 'Digital output', indicated by a red circle with the number 3. At the bottom, there is a checkbox for 'Initialise pin state?'.

3.3V Power - 1		2 - 5V Power	
<input type="radio"/> SDA1 - GPIO02 - 3	<input type="radio"/>	<input type="radio"/> 4 - 5V Power	<input type="radio"/>
<input type="radio"/> SCL1 - GPIO03 - 5	<input type="radio"/>	<input type="radio"/> 6 - Ground	<input type="radio"/>
<input type="radio"/> GPIO04 - 7	<input type="radio"/>	<input type="radio"/> 8 - GPIO14 - TxD	<input type="radio"/>
<input type="radio"/> Ground - 9	<input type="radio"/>	<input type="radio"/> 10 - GPIO15 - RxD	<input type="radio"/>
<input type="radio"/> GPIO17 - 11	<input type="radio"/>	<input type="radio"/> 12 - GPIO18	<input type="radio"/>
<input type="radio"/> GPIO27 - 13	<input type="radio"/>	<input type="radio"/> 14 - Ground	<input type="radio"/>
<input type="radio"/> GPIO22 - 15	<input type="radio"/>	<input type="radio"/> 16 - GPIO23	<input type="radio"/>
<input type="radio"/> 3.3V Power - 17	<input type="radio"/>	<input type="radio"/> 18 - GPIO24	<input type="radio"/>
<input type="radio"/> MOSI - GPIO10 - 19	<input type="radio"/>	<input type="radio"/> 20 - Ground	<input type="radio"/>
<input type="radio"/> MISO - GPIO09 - 21	<input type="radio"/>	<input type="radio"/> 22 - GPIO25	<input type="radio"/>
<input type="radio"/> SCLK - GPIO11 - 23	<input type="radio"/>	<input type="radio"/> 24 - GPIO8 - CE0	<input type="radio"/>
<input type="radio"/> Ground - 25	<input type="radio"/>	<input type="radio"/> 26 - GPIO7 - CE1	<input type="radio"/>
<input type="radio"/> SD - 27	<input type="radio"/>	<input type="radio"/> 28 - SC	<input type="radio"/>
<input type="radio"/> GPIO05 - 29	<input type="radio"/>	<input type="radio"/> 30 - Ground	<input type="radio"/>
<input type="radio"/> GPIO06 - 31	<input type="radio"/>	<input type="radio"/> 32 - GPIO12	<input type="radio"/>
<input type="radio"/> GPIO13 - 33	<input type="radio"/>	<input type="radio"/> 34 - Ground	<input type="radio"/>
<input type="radio"/> GPIO19 - 35	<input type="radio"/>	<input type="radio"/> 36 - GPIO16	<input type="radio"/>
<input type="radio"/> GPIO26 - 37	<input type="radio"/>	<input type="radio"/> 38 - GPIO20	<input type="radio"/>
<input type="radio"/> Ground - 39	<input type="radio"/>	<input checked="" type="radio"/> 40 - GPIO21	<input type="radio"/>

# Node-RED 動作確認①

デプロイにより設定を確定します。

クリック

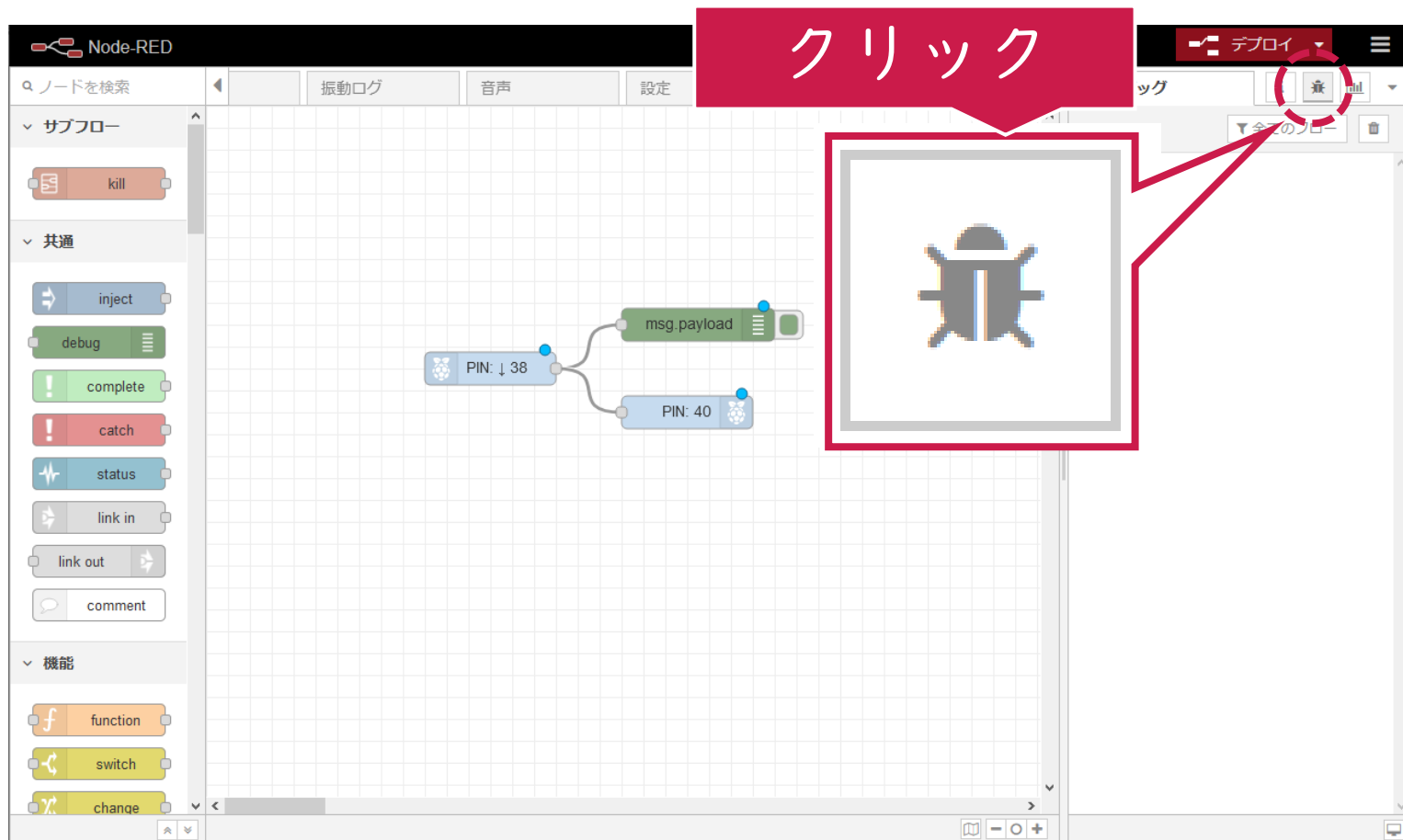
The screenshot shows the Node-RED web interface. The main workspace contains a flow with three nodes: a 'PIN: 38' node, a 'msg.payload' node, and a 'PIN: 40' node. The 'PIN: 38' node is connected to the 'msg.payload' node, which is then connected to the 'PIN: 40' node. The left sidebar shows the 'Common' (共通) category with nodes like 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link out', and 'comment'. The right sidebar shows the 'Dashboard' (ダッシュボード) configuration panel with various settings and links.

スイッチ押下によりLEDの点灯を確認してみましょう。



# Node-RED 動作確認②

debugログにより流れるデータを確認します。



スイッチ押下により流れるデータを確認してみましょう。

# Node-RED 動作確認結果

スイッチの押下によりdebugログに数字の0と1が交互に表示されます。

The screenshot displays the Node-RED interface with a workflow containing a switch node. The debug console on the right shows a sequence of messages where the payload alternates between 0 and 1. A red dashed box highlights the entire debug log area, and a red solid box highlights a specific message entry.

Timestamp	Node ID	msg.payload
2020/12/3 15:24:48	node: 47ae06b6.13af68	0
2020/12/3 15:28:22	node: 47ae06b6.13af68	1
2020/12/3 15:28:23	node: 47ae06b6.13af68	0
2020/12/3 15:28:24	node: 47ae06b6.13af68	1
2020/12/3 15:28:24	node: 47ae06b6.13af68	0

### 3. スイッチの状態を見える化

# 作成する機能

## 1. スイッチの状態を見える化

スイッチの状態を把握することができます。

例:装置が稼働しているか否かの状態を見える化する。

## 2. スイッチが押された回数を見える化

スイッチが押された回数を把握することができます。

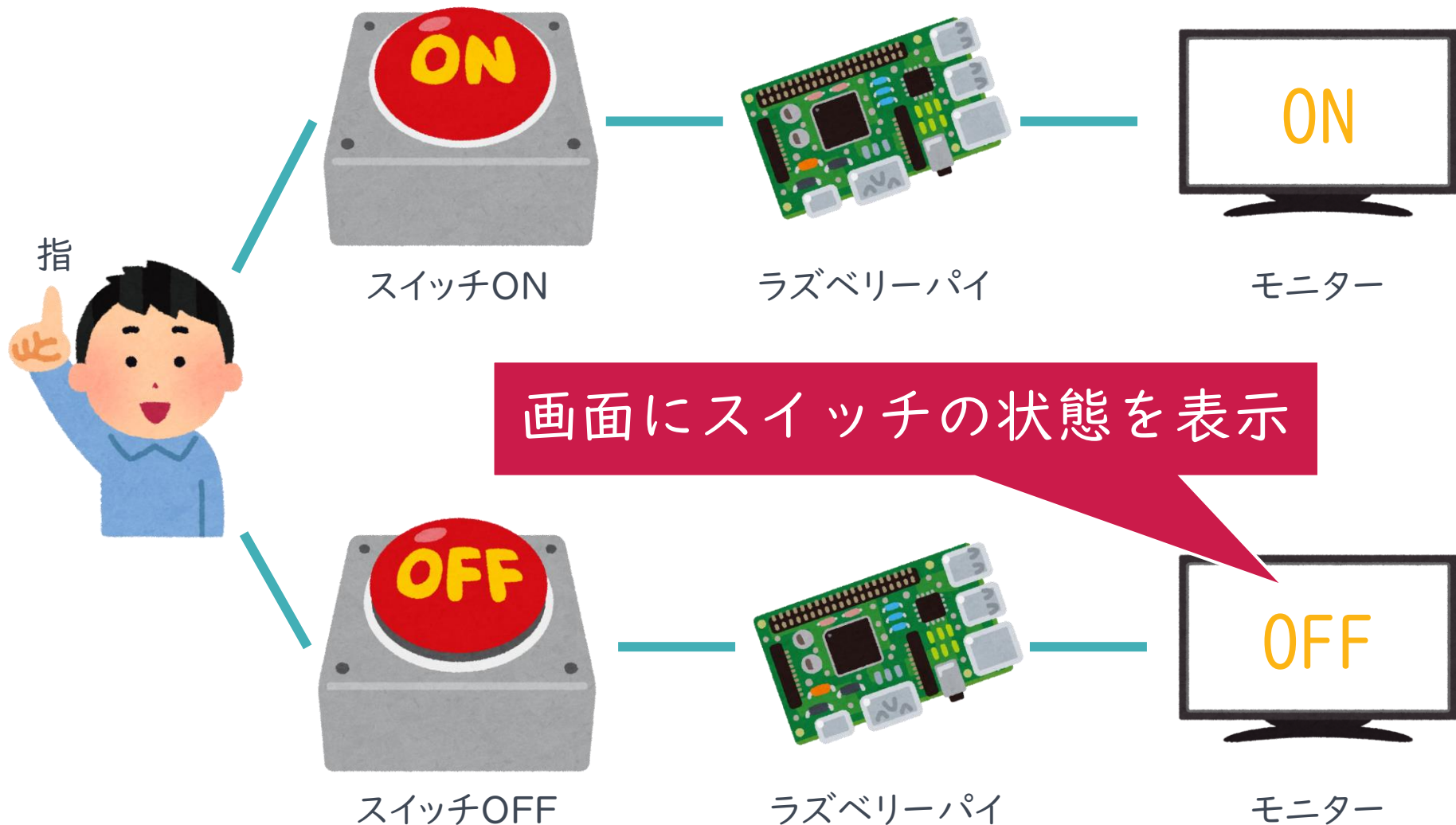
例:製品が何個製造されたかの状態を見える化する。

## 3. スイッチが押された回数をグラフ化

スイッチが押された回数の推移を把握することができます。

例:製品製造個数の時間経過による累積回数の変化を見える化する。

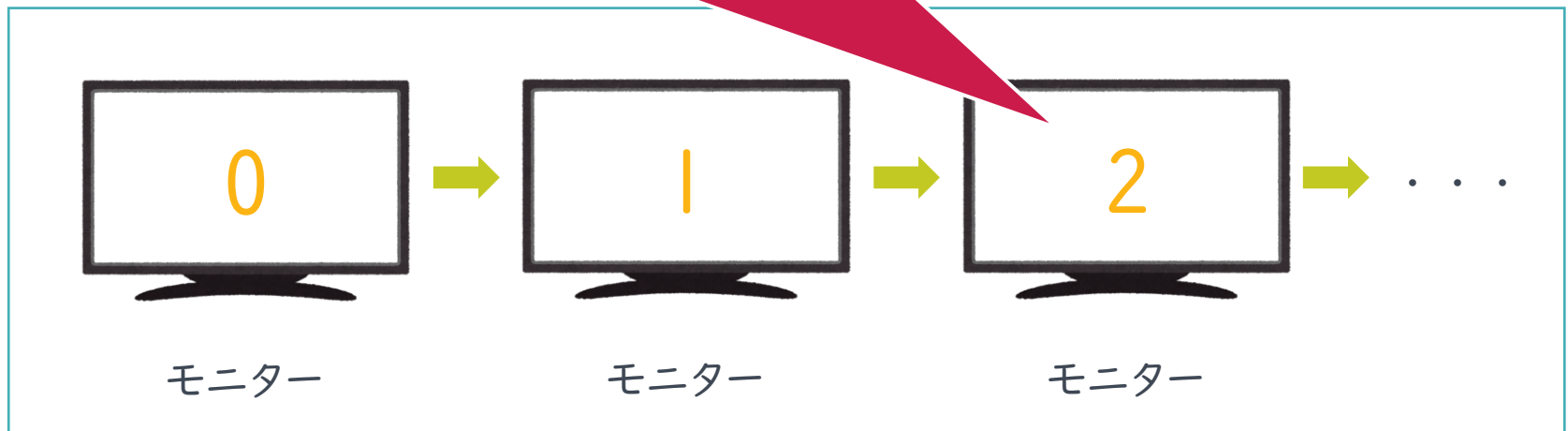
# 1. スイッチの状態を見える化



## 2. スイッチが押された回数を見える化



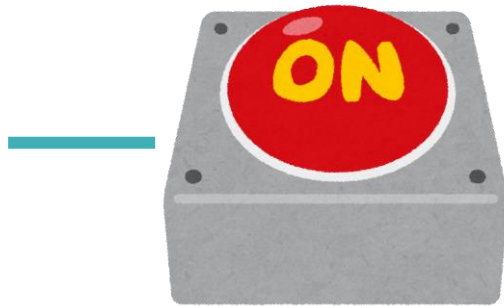
スイッチが押された回数を表示



# 3. スイッチが押された回数をグラフ化

スイッチが押された回数の時間変化を表示

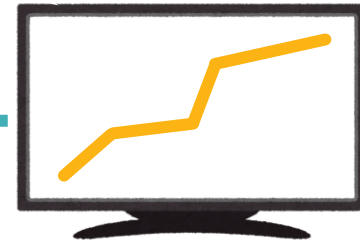
指



スイッチON



ラズベリーパイ



モニター

# Node-RED ダッシュボード

Webブラウザにより見える化を行うためのライブラリを使用します。

ライブラリ名: **node-red-dashboard**



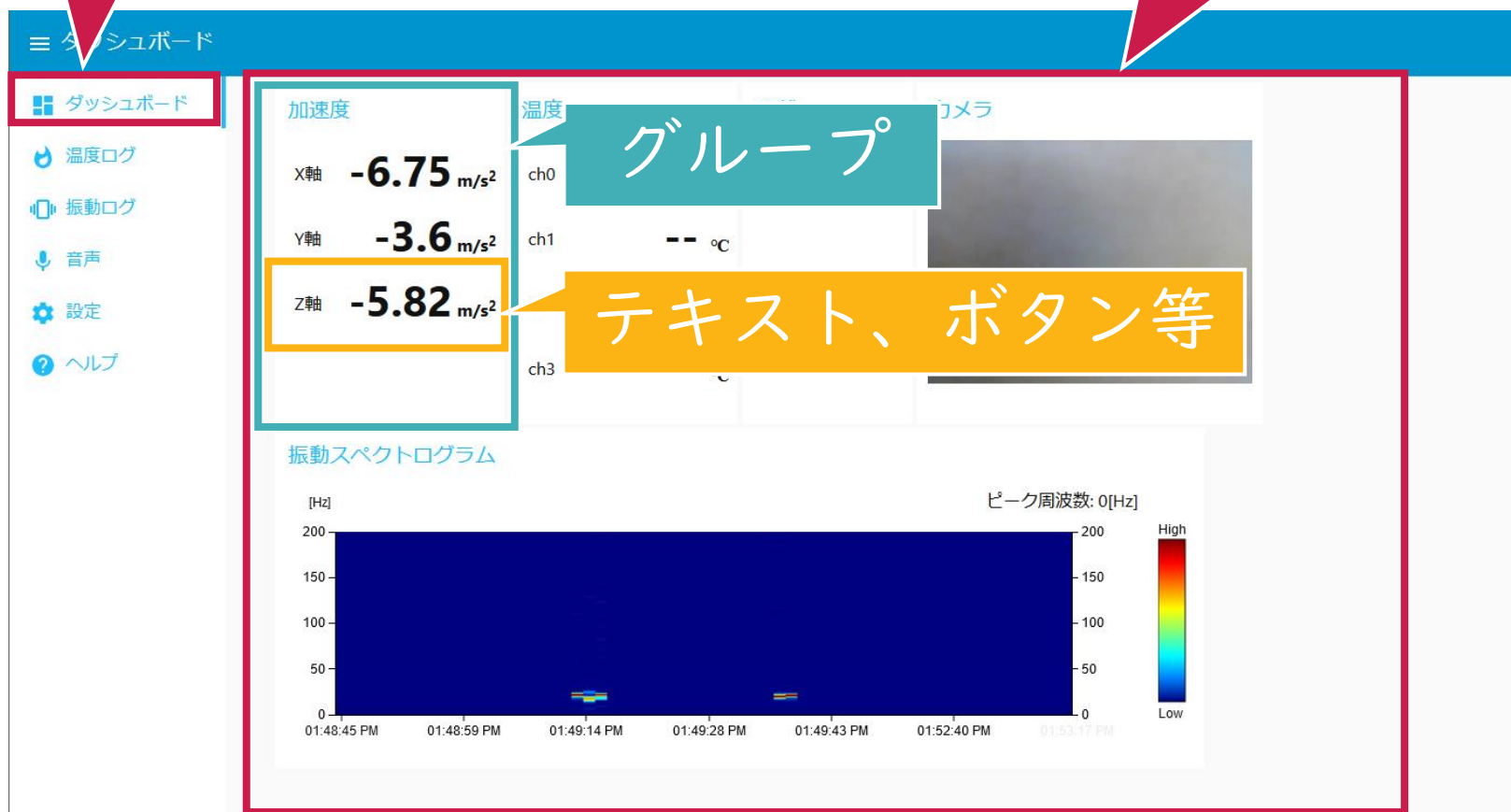


# Node-RED 画面構成について

ダッシュボードの画面構成についてです。

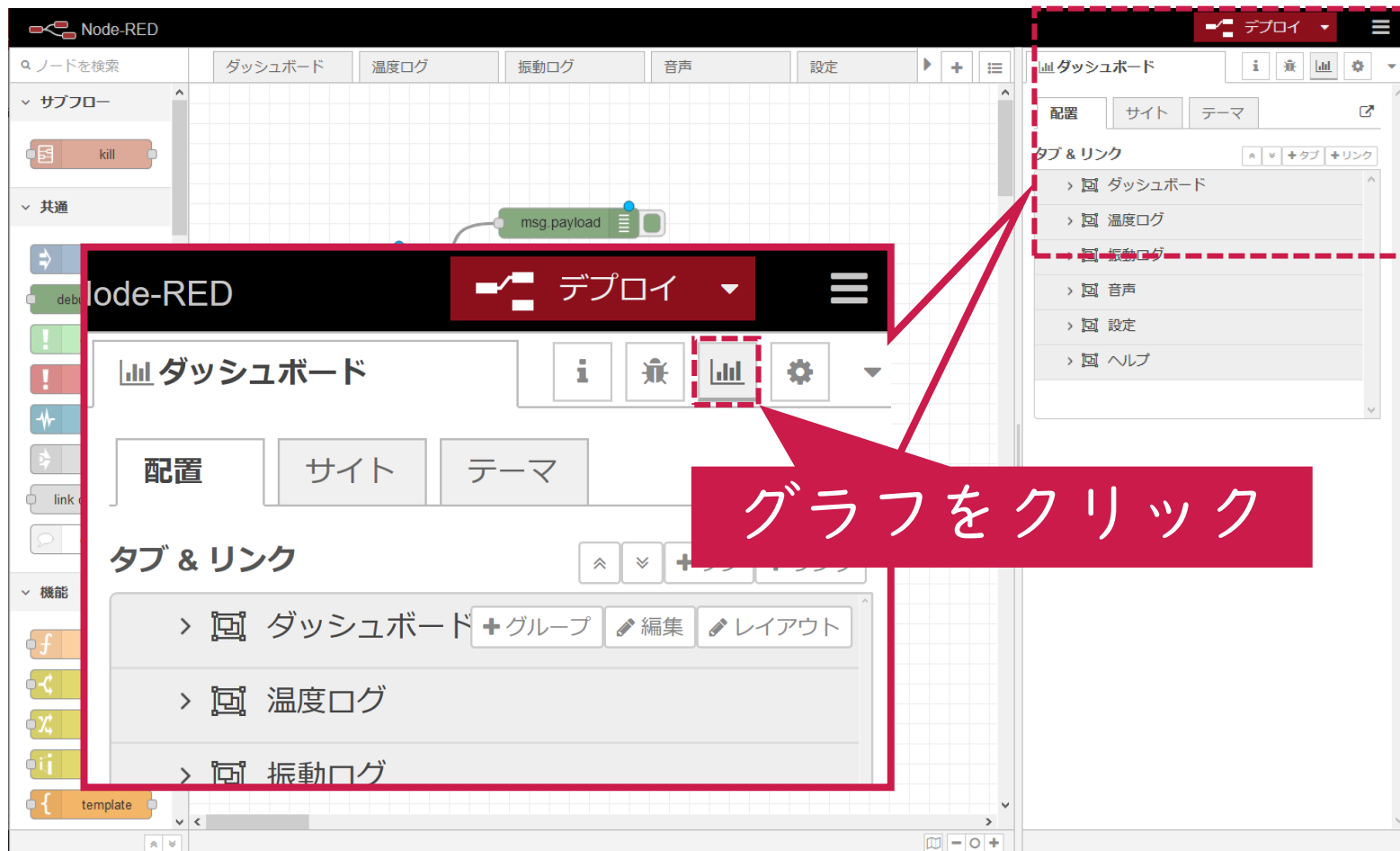
タブ

タブ画面



# Node-RED タブの作成①

セミナー用のタブを作成します。



# Node-RED タブの作成②

セミナー用のタブを作成します。

The screenshot shows the Node-RED interface with a focus on the 'タブ & リンク' (Tabs & Links) section. A red dashed box highlights the top right corner of the interface, including the 'デプロイ' (Deploy) button and the 'タブ & リンク' section. A red solid box highlights the '+タブ' button in the 'タブ & リンク' section. A red callout bubble points to the '+タブ' button with the text '「+タブ」をクリック' (Click '+Tab').

Node-RED

デプロイ

ダッシュボード

温度ログ

振動ログ

音声

設定

サブフロー

kill

共通

msg payload

node-RED

デプロイ

ダッシュボード

配置

サイト

テーマ

タブ & リンク

+タブ +リンク

ダッシュボード +グループ 編集 レイアウト

温度ログ

振動ログ

機能

template

「+タブ」をクリック

# Node-RED タブの編集

セミナー用のタブの編集を行います。

The screenshot shows the Node-RED interface with the 'ダッシュボード' (Dashboard) tab selected. The 'タブ & リンク' (Tabs & Links) panel is open, displaying a list of tabs. The 'Tab 7' tab is expanded, showing options for '+グループ' (Add Group), '編集' (Edit), and 'レイアウト' (Layout). A red callout box points to the '編集' button with the text '「編集」をクリック'.

タブ & リンク

- > ダッシュボード
- > 温度ログ
- > 振動ログ
- > 音声
- > 設定
- > ヘルプ
- ▼ Tab 7
  - +グループ
  - 編集
  - レイアウト

# Node-RED タブの名称変更

セミナー用のタブの名称を変更します。

dashboard tab ノードを編集

削除 中止 更新

プロパティ

名前 セミナー

アイコン dashboard

状態  有効

メニュー  表示

名前を編集して更新

# Node-RED グループの作成

セミナー用のグループを作成します。

The image shows the Node-RED interface with the 'Dashboard' configuration panel open. The panel includes tabs for '配置' (Configuration), 'サイト' (Site), and 'テーマ' (Theme). Under the '配置' tab, there is a 'タブ & リンク' (Tabs & Links) section with a list of items: 'ダッシュボード', '温度ログ', '振動ログ', '音声', '設定', 'ヘルプ', and 'セミナー'. The 'セミナー' item is expanded, showing a '+グループ' button, a '編集' (Edit) button, and a 'レイアウト' (Layout) button. A red callout box points to the '+グループ' button with the text 「+グループ」をクリック (Click '+Group').

# Node-RED グループの編集

セミナー用のグループの編集をします。

The screenshot shows the Node-RED interface with a dashboard group named 'ダッシュボード' (Dashboard) selected. The group list on the right side of the dashboard configuration panel includes items like 'ダッシュボード', '温度ログ', '振動ログ', '音声', '設定', 'ヘルプ', 'セミナー', and 'Group 1'. At the bottom of this list, there is a '+ スペース' button and an '編集' (Edit) button, which is highlighted with a red dashed box. A red callout bubble points to this '編集' button with the text '「編集」をクリック'. Another red dashed box highlights the top part of the dashboard configuration panel, including the 'ダッシュボード' title, 'i', '🔍', '📊', and '⚙️' icons, and the '配置', 'サイト', and 'テーマ' tabs. The 'タブ & リンク' section is also visible, showing a list of tabs and links.

# Node-RED グループの名称変更

セミナー用のグループの名称を変更します。

dashboard group ノードを編集

削除 中止 更新

⚙️ プロパティ

📌 名前 見える化

📄 タブ セミナー

↔️ 幅 6

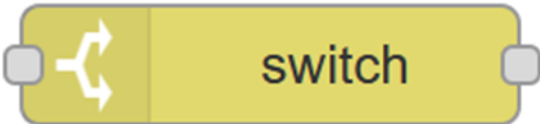
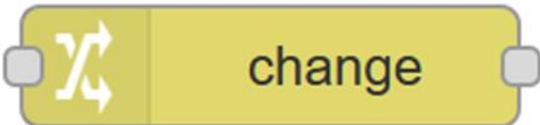
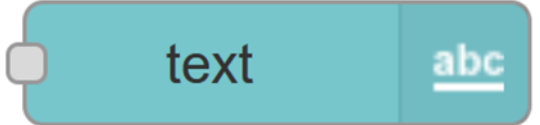
- グループ名を表示する
- グループの折りたたみを有効にする

名前を編集して更新



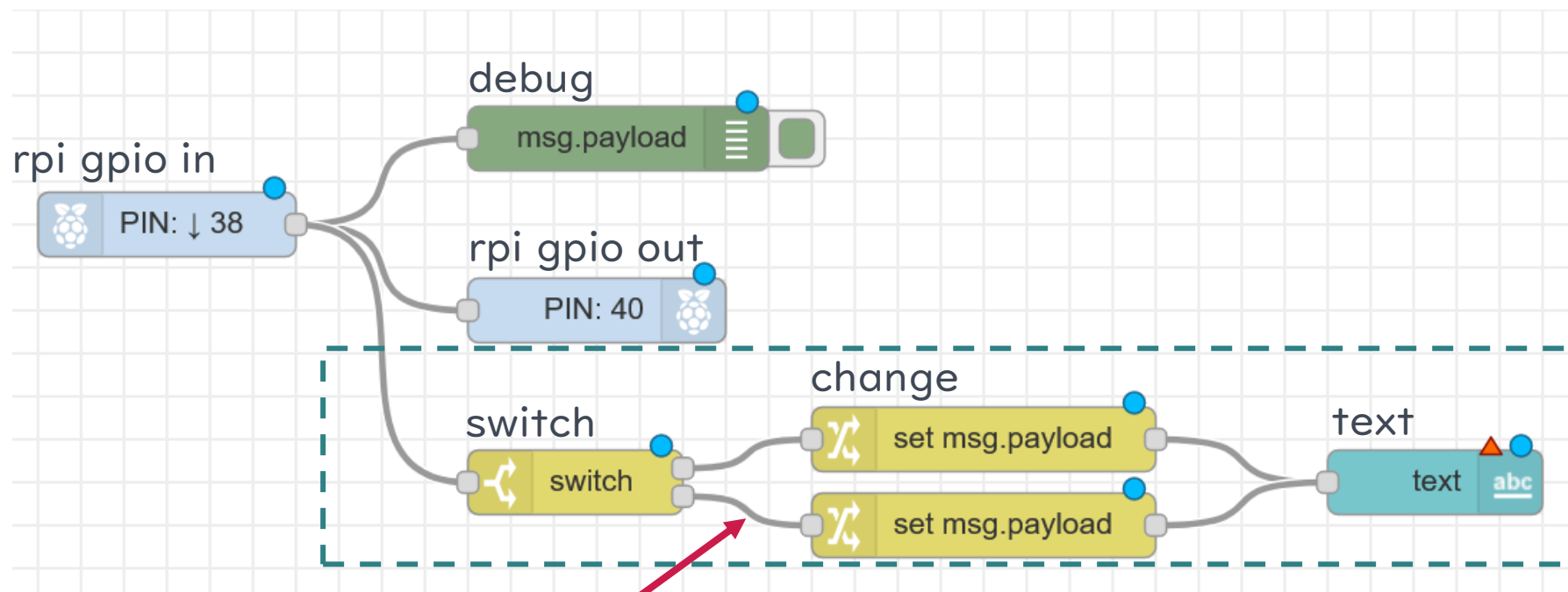
# Node-RED 使用するノード

スイッチの状態を見える化するためのプログラミングを行います。  
各ノードをダブルクリックすることで詳細設定ができます。

項目	ノード	機能	説明
機能		スイッチ	フローの分岐処理用ノード。 スイッチの状態による分岐に使用。
機能		チェンジ	フローのデータ変更用ノード。 ON/OFFの文字列変更に使用。
ダッシュボード		テキスト出力	テキスト表示用ノード。 画面に表示する文字列表示に使用。

# Node-RED ノードの配線を追加

ノードをワークスペースに配置し、ノードの間を線(フロー)で接続します。

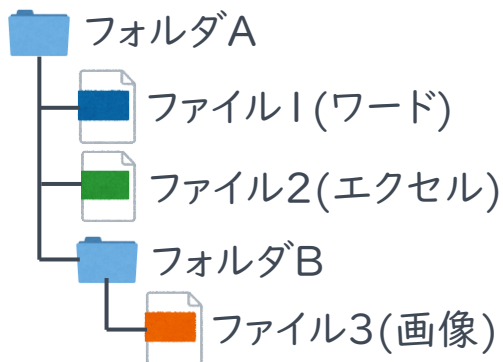


フローを流れるデータは **msg** と名前が付けられています。

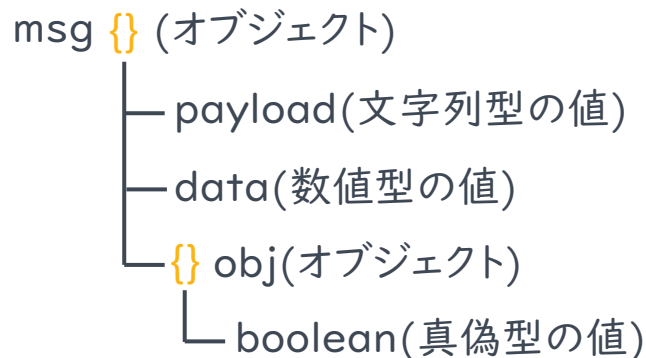
# Node-RED フローを流れるデータ

フローを流れるデータについて説明します。

## ファイルシステム(Windowsなど)



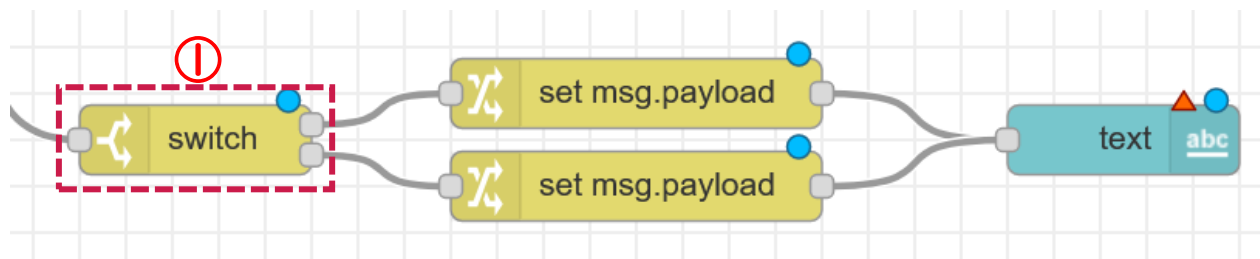
## フローを流れるデータ(JSONフォーマット)



- **オブジェクト** (フォルダのようなイメージ)  
オブジェクト内には複数の値とオブジェクトを持ちます。
- **値** (ファイルのようなイメージ)  
数値、文字列などの値を持ちます。

# Node-RED ノードの設定①-1

スイッチの状態で分岐する設定を行います。



- ① ノードをダブルクリックする。
- ② **+追加**をクリックする。
- ③ **数値**を選択する。(2箇所)
- ④ 数字の**0**を設定する。
- ⑤ 数字の**1**を設定する。
- ⑥ **最初に合致した~**を選択する。
- ⑦ **完了**をクリックする。

switch ノードを編集

削除 中止 完了

プロパティ

名前 名前

プロパティ msg. payload

== 0 → 1

== 1 → 2

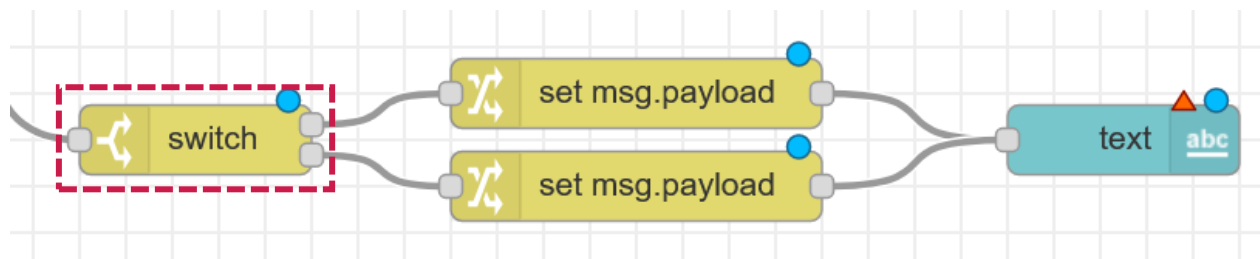
+追加

最初に合致した条件で終了

メッセージ列の補正

# Node-RED ノードの設定①-2

スイッチの状態で分岐する設定について説明します。

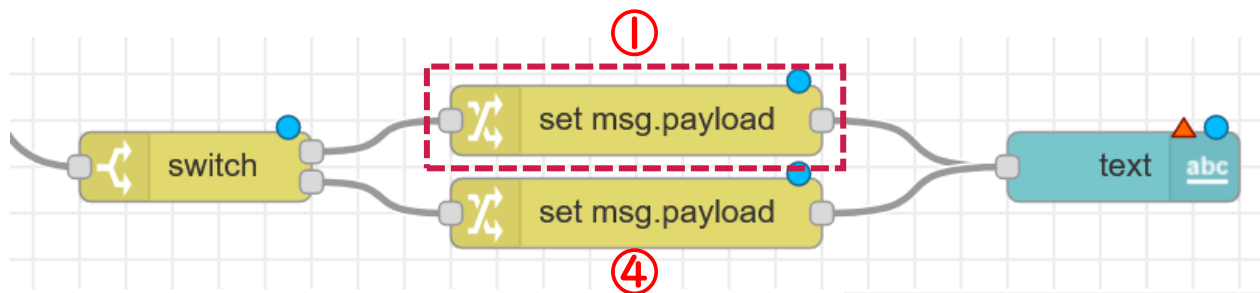


- ① `msg.payload`のデータが
- ② 数字の0と
- ③ 等しい(==)場合は
- ④ 1番のフロー(上)へ分岐し、
- ⑤ 数字の1と
- ⑥ 等しい(==)場合は
- ⑦ 2番のフロー(下)へ分岐する。

The screenshot shows the 'switch' node configuration panel titled 'switch ノードを編集'. At the top, there are buttons for '削除', '中止', and '完了'. Below is the 'プロパティ' section. The '名前' field contains '名前'. The 'プロパティ' dropdown is set to 'msg. payload'. There are two rows of configuration for the switch conditions. The first row has a menu icon (③), the operator '==' (⑥), a dropdown set to '0' (②), and a '→ 1' button (④). The second row has a menu icon (⑥), the operator '==' (⑤), a dropdown set to '1' (⑤), and a '→ 2' button (⑦). At the bottom, there is a dropdown menu set to '最初に合致した条件で終了' and a checkbox for 'メッセージ列の補正'.

# Node-RED ノードの設定②-1

データ変更する設定を行います。

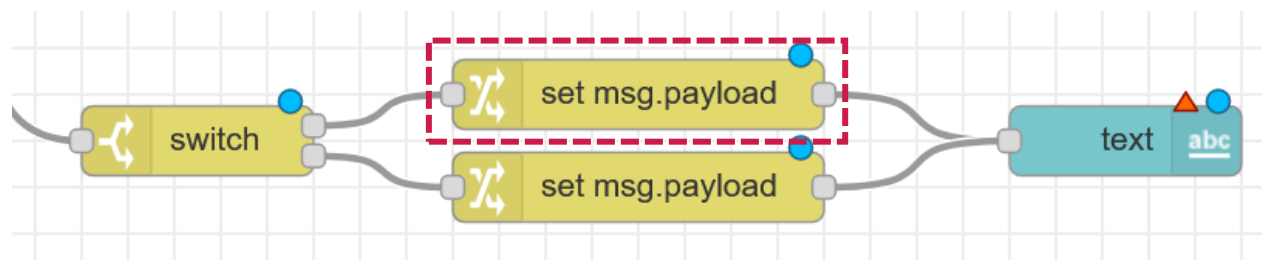


- ① ノードをダブルクリックする。
- ② 対象の値にOFFを設定する。
- ③ 完了をクリックする。
- ④ 同様にONを設定する。

The screenshot shows the configuration interface for a 'change' node in Node-RED. The window title is 'change ノードを編集'. At the top right, there are three buttons: '削除' (Delete), '中止' (Cancel), and '完了' (Done), with the '完了' button circled in red and labeled with a circled 3. Below the buttons is a 'プロパティ' (Properties) section with a '名前' (Name) input field. The 'ルール' (Rules) section is expanded, showing a rule for 'msg. payload'. The rule has a dropdown menu set to '値の代入' (Value substitution) and a dropdown menu set to 'a\_z'. Below this, there is a radio button for 'OFF', which is circled in red and labeled with a circled 2. At the bottom left, there is a '+ 追加' (Add) button.

# Node-RED ノードの設定②-2

データ変更する設定について説明します。



- ① `msg.payload`に
- ② `OFF`という文字列を
- ③ 代入する。

change ノードを編集

削除 中止 完了

プロパティ

名前 名前

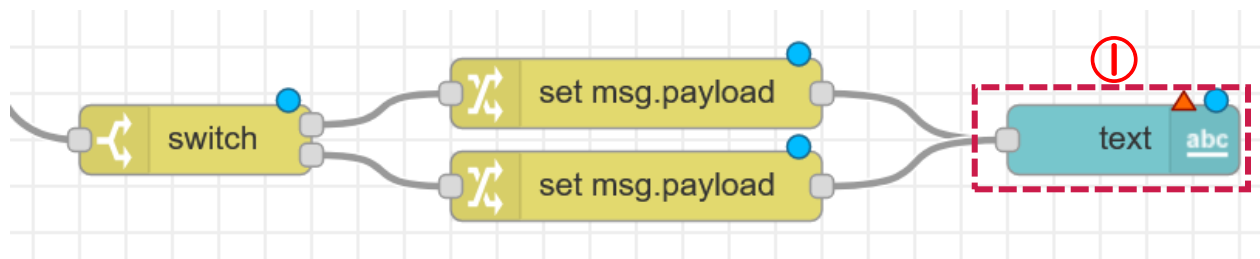
ルール

- ③ 値の代入
- ① msg. payload
- ② a-z OFF

+ 追加

# Node-RED ノードの設定③ーI

Webブラウザにスイッチの状態を表示する設定を行います。



- ① ノードをダブルクリックする。
- ② グループに**見える化**を設定する。
- ③ ラベルに**スイッチ**を設定する。
- ④ **完了**をクリックする。

text ノードを編集

削除 中止 完了

プロパティ

Group ② [セミナー] 見える化

Size 自動

Label ③ スイッチ

Value format {{msg.payload}}

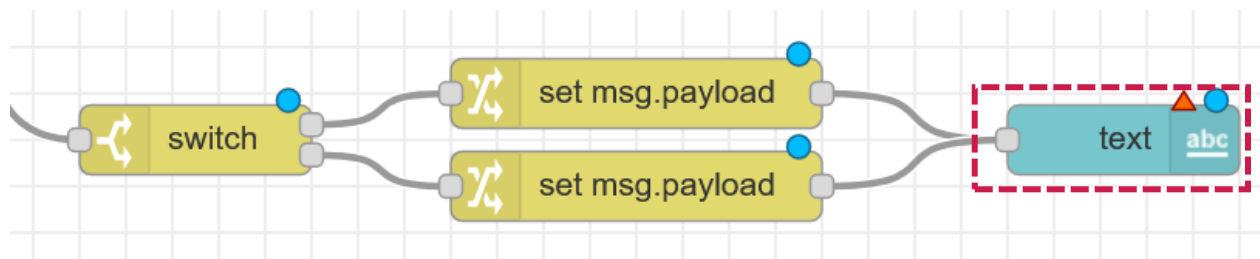
Layout

Name



# Node-RED ノードの設定③-2

Webブラウザにスイッチの状態を表示する設定について説明します。



- ① 選択したタブ、グループの
- ② **スイッチ**というラベルに
- ③ **msg.payload**の中身を表示する。

text ノードを編集

削除 中止 完了

プロパティ

Group ① [セミナー] 見える化

Size 自動

Label ② スイッチ

Value format ③ {{msg.payload}}

Layout

label value label value label value

label value label value

Name

# Node-RED 動作確認

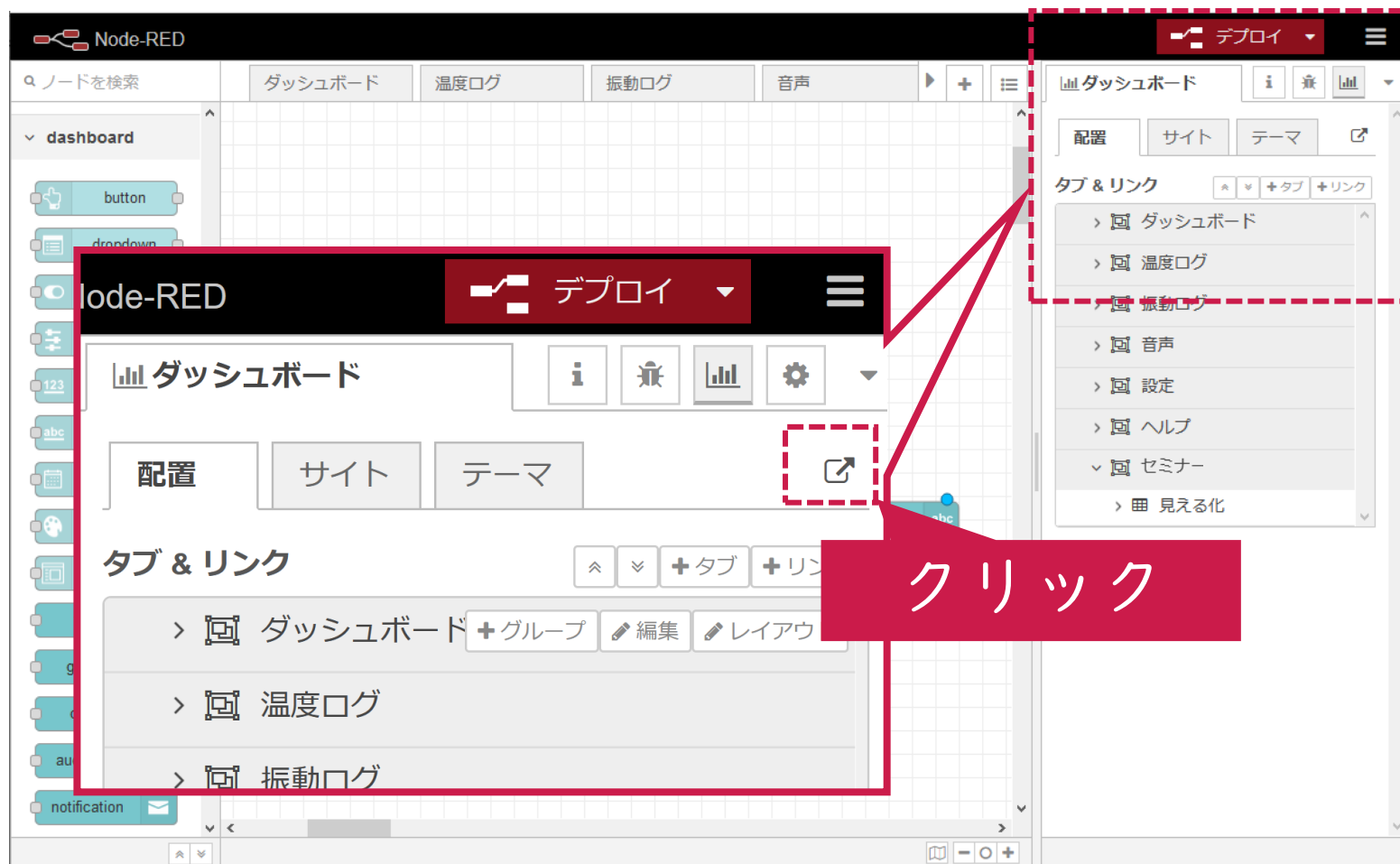
デプロイにより設定を確定します。

クリック

The screenshot displays the Node-RED web interface. On the left, a sidebar lists various widgets under the 'dashboard' category, including button, dropdown, switch, slider, numeric, text input, date picker, colour picker, form, text, gauge, chart, audio out, and notification. The main workspace shows a flow diagram starting with a '38' node connected to a 'msg.payload' node. This is followed by a 'PIN: 40' node, then a 'switch' node. The 'switch' node has two outgoing paths, each leading to a 'set msg.payload' node. Finally, these two paths converge into a 'スイッチ' (Switch) node. On the right side, a 'ダッシュボード' (Dashboard) sidebar is visible, showing a list of tabs: 'ダッシュボード', '温度ログ', '振動ログ', '音声', '設定', 'ヘルプ', and 'セミナー'. The 'セミナー' (Seminar) tab is currently selected and expanded, showing a sub-item '見える化' (Visualization).

# Node-RED 動作確認

動作確認を行います。



スイッチの押下による画面表示の変化を確認しましょう。

# Node-RED 動作確認結果①

作成したタブを選択します。

① クリック



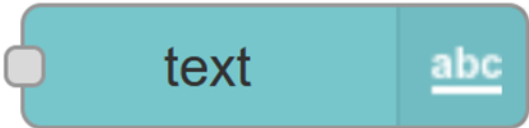
# Node-RED 動作確認結果②

スイッチの押下により画面にOFFとONが交互に表示されます。



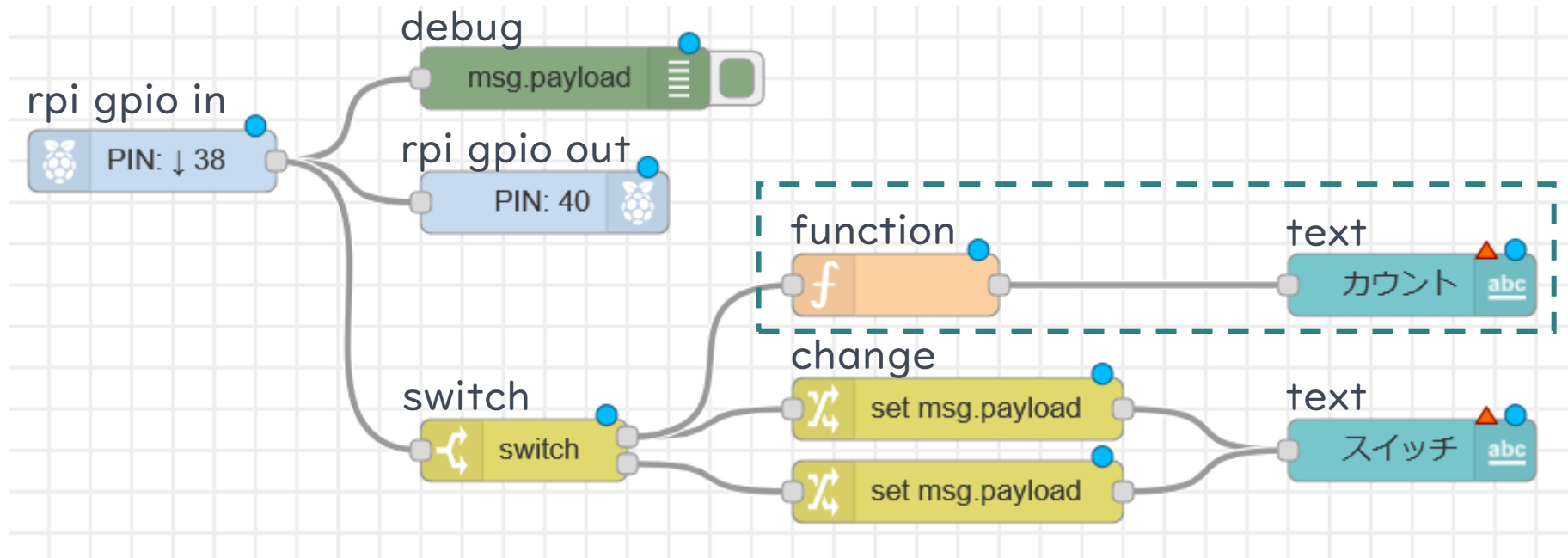
# Node-RED 使用するノード

スイッチの押下回数を見える化するためのプログラミングを行います。  
各ノードをダブルクリックすることで詳細設定ができます。

項目	ノード	機能	説明
機能		ファンクション	プログラミング処理用ノード。 JavaScript(Node.js)が記述可能。 スイッチのカウンタに使用。
ダッシュボード		テキスト出力	テキスト表示用ノード。 画面に表示するテキストに使用。

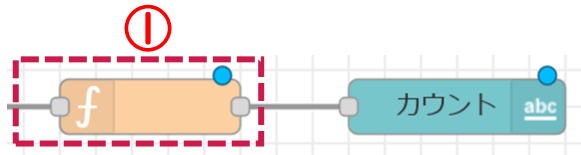
# Node-RED ノードの配線の追加

ノードをワークスペースに配置し、ノードの間を線(フロー)で接続します。



# Node-RED ノードの設定④-1

スイッチの押下回数を増加させる設定を行います。



① ノードをダブルクリックする。

② コードを入力する。

③ 完了をクリックする。

```
function ノードを編集
削除 中止 完了
プロパティ
名前 名前
コード
1 // countという名前の場所に保存している、カウンターの値を取得する。
2 // countという名前の場所が存在しない場合には0を取得する。
3 // 取得したデータをdataという名前の一時的な場所に代入する。
4 const data = flow.get('count') || 0;
5 // 取得したデータに1を加算して、msg.payloadという名前の場所に代入する。
6 msg.payload = data + 1;
7 // 取得したデータに1を加算して、countという名前の場所にカウンターを保存する。
8 flow.set('count', data + 1);
9 // 次のフローへmsgを流す。
10 return msg;
11
12
出力数 1
```

次のスライドにコードの詳細を載せています。



# Node-RED ノードの設定④-2

スイッチの押下回数を増加させるコードの詳細です。

🔧 コード

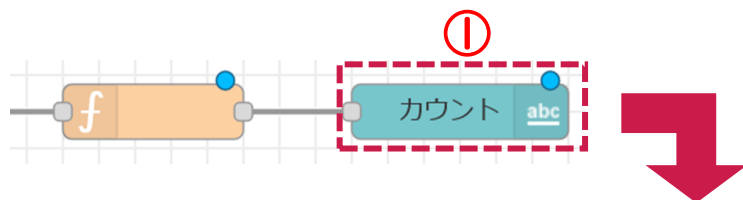
```
1 // countという名前の場所に保存している、カウンターの値を取得する。
2 // countという名前の場所が存在しない場合には0を取得する。
3 // 取得したデータをdataという名前の一時的な場所に代入する。
4 const data = flow.get('count') || 0;
5 // 取得したデータに1を加算して、msg.payloadという名前の場所に代入する。
6 msg.payload = data + 1;
7 // 取得したデータに1を加算して、countという名前の場所にカウンターを保存する。
8 flow.set('count', data + 1);
9 // 次のフローへmsgを流す。
10 return msg;
11
12
```

コメント

ソースコード(4, 6, 8, 10行目)

# Node-RED ノードの設定⑥

スイッチの押下回数を表示する設定を行います。



- ① ノードをダブルクリックする。
- ② グループに**見える化**を設定する。
- ③ ラベルに**カウント**を設定する。
- ④ **完了**をクリックする。

text ノードを編集

削除 中止 完了

プロパティ

Group ② [セミナー] 見える化

Size 自動

Label ③ カウント

Value format {{msg.payload}}

Layout

label value label value label value

label value label value

Name

# Node-RED 動作確認

デプロイにより設定を確定します。

クリック

The screenshot shows the Node-RED web interface. At the top right, there is a red button labeled "デプロイ" (Deploy). The main workspace contains a flow diagram with the following components:

- A blue "PIN: ↓ 38" node.
- A green "msg.payload" node.
- A blue "PIN: 40" node.
- A yellow "switch" node.
- An orange "function" node.
- Two yellow "set msg.payload" nodes.
- A cyan "カウント" (Count) node.
- A cyan "スイッチ" (Switch) node.

The flow starts with the "PIN: ↓ 38" node, which branches into three paths: one to "msg.payload", one to "PIN: 40", and one to "switch". The "switch" node has two outputs: one to the "function" node and one to the "set msg.payload" nodes. The "function" node outputs to the "カウント" node. The "set msg.payload" nodes output to the "スイッチ" node.

On the right side, there is an "i 情報" (Info) panel with the following details:

フロー	"48fcdcbc.1479a4"
名前	フロー 1
状態	有効

Below the info panel, there is a note: "設定により、ヒントの表示/非表示を変更できます。" (Depending on the settings, you can change the display/non-display of hints.)

スイッチの押下による回数の変化を確認しましょう。

# Node-RED 動作確認結果

スイッチの押下により画面のカウント数が増加していきます。

≡ セミナー		
見える化		
スイッチ	OFF	
カウント	0	



スイッチを押して離す

≡ セミナー		
見える化		
スイッチ	OFF	
カウント	1	





スイッチを押して離す

≡ セミナー		
見える化		
スイッチ	OFF	
カウント	2	

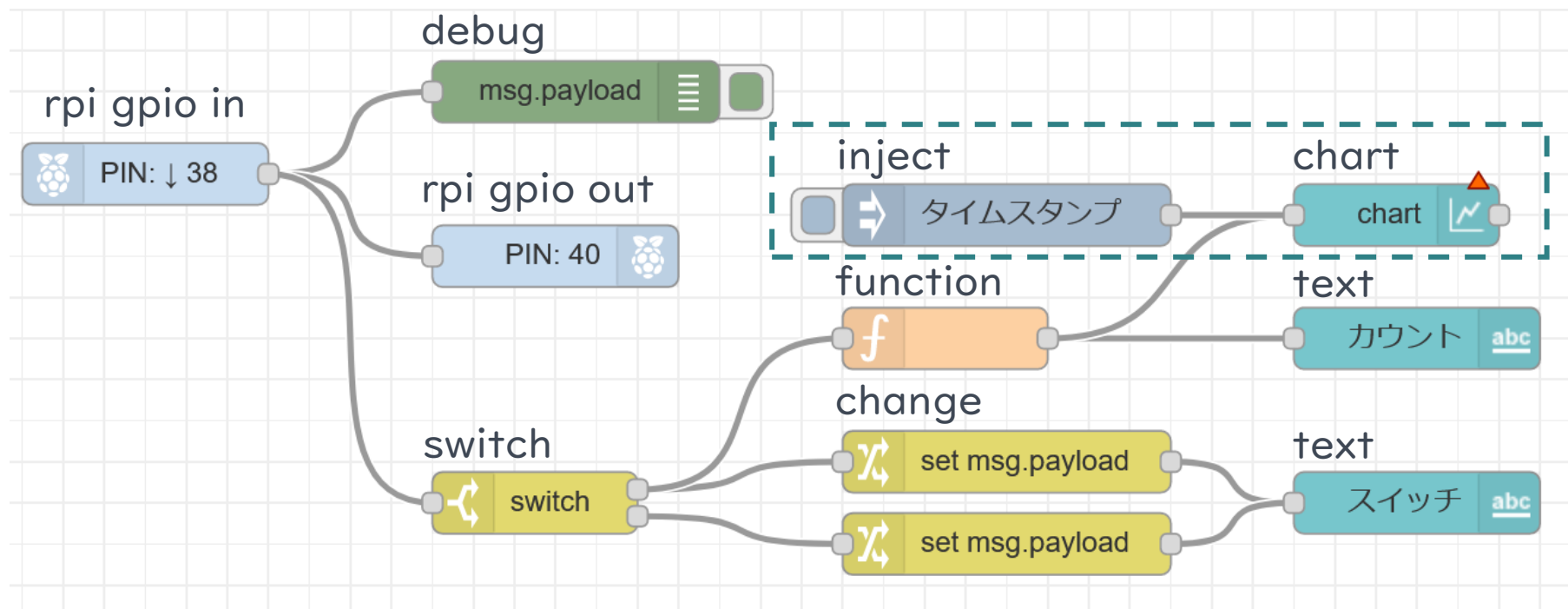
# Node-RED 使用するノード

スイッチの押下回数をグラフ化するためのプログラミングを行います。  
各ノードをダブルクリックすることで詳細設定ができます。

項目	ノード	機能	説明
共通		フロー 開始	手動でのフロー開始、定期実行、 起動時の自動実行等の フロー開始用のノード 起動時のグラフ初期化に使用。
ダッシュ ボード		グラフ 表示	グラフ表示用ノード。 押下回数の時系列表示に使用。

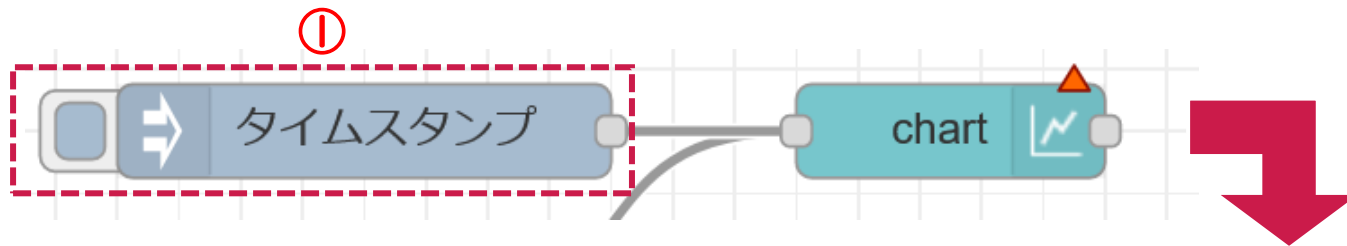
# Node-RED ノードの配線の追加

ノードをワークスペースに配置し、ノードの間を線(フロー)で接続します。



# Node-RED ノードの設定④ーI

Node-REDの起動時にグラフを初期化する設定を行います。

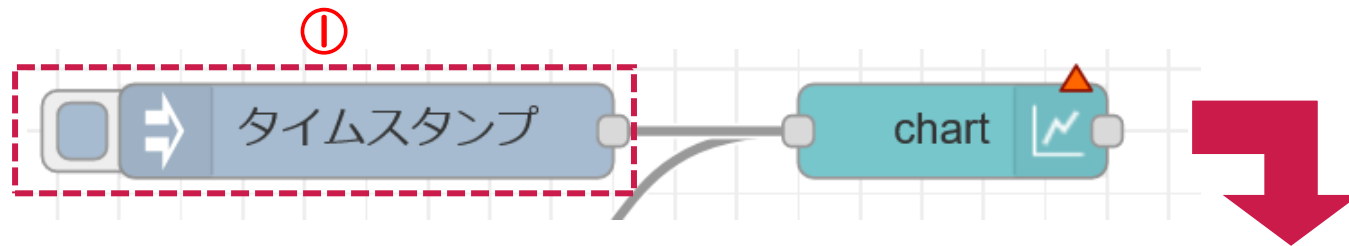


- ① ノードをダブルクリックする。
- ② **json**を設定する。
- ③ **[ ]**を設定する。
- ④ **チェック**をする。
- ⑤ **完了**をクリックする。



# Node-RED ノードの設定④-2

Node-REDの起動時にグラフを初期化する設定について説明します。



- ① Node-RED起動0.1秒後に
- ② jsonデータ[]を送信する。

inject ノードを編集

削除 中止 完了

プロパティ

ペイロード `{}`

トピック

Node-RED起動の 0.1 秒後、以下を行う

繰り返し なし

名前 名前

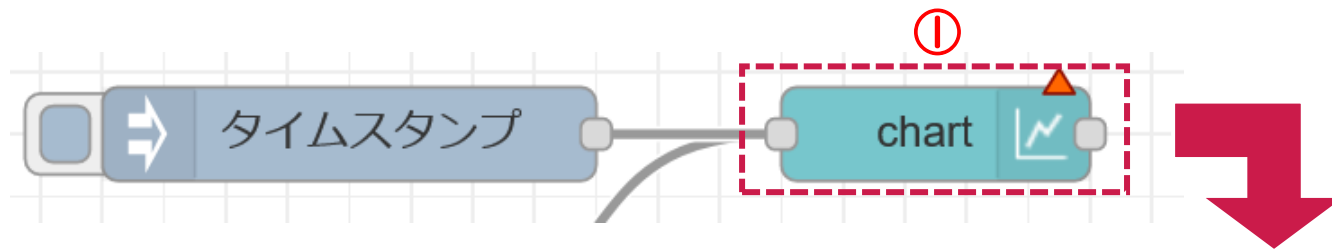
## \*注

- ・ chartノードは初期化が必要。
- ・ 初期化の方法はjsonデータ[]を入力。



# Node-RED ノードの設定⑤ - 1

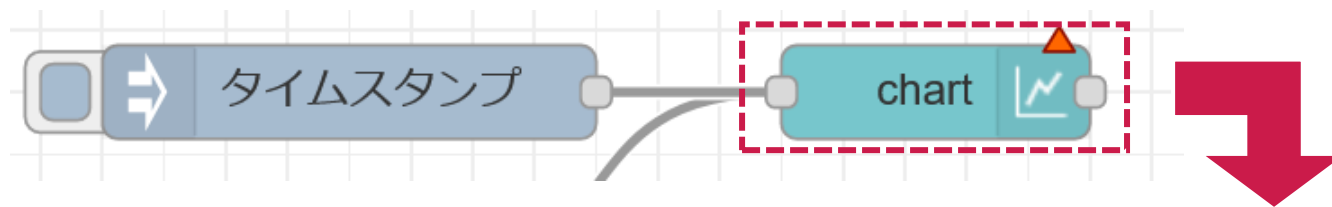
Webブラウザに累積回数をグラフ表示する設定を行います。



- ① ノードをダブルクリックする。
- ② グループに**見える化**を設定する。
- ③ ラベルに**累積回数**を設定する。
- ④ **minutes**に変更する。
- ⑤ **完了**をクリックする。

# Node-RED ノードの設定⑤-2

Webブラウザに累積回数をグラフ表示する設定について説明します。



- ① 選択したタブ、グループの
- ② 累積回数というラベルに
- ③ 折れ線グラフで
- ④ 過去1分間のデータを表示する。

chart ノードを編集

削除 中止 完了

プロパティ

Group ① [セミナー] 見える化

Size 自動

Label ② 累積回数

Type ③ Line chart  enlarge points

X-axis ④ last 1 minutes OR 1000 points

X-axis Label ▼ HH:mm:ss

# Node-RED 動作確認

デプロイにより設定を確定します。

クリック

The screenshot shows the Node-RED web interface. On the left, there is a sidebar with a search bar and a list of nodes categorized into 'サブフロー' (Subflows), '共通' (Common), and '機能' (Functions). The main workspace contains a flow diagram starting with a 'PIN: ↓ 38' node. This node branches into three paths: one to a 'msg.payload' node, one to a 'PIN: 40' node, and one to a 'switch' node. The 'switch' node has two outputs, each leading to a 'set msg.payload' node. The 'PIN: 40' node connects to a '1' node, which then connects to a '累積回数' (Accumulation Count) node. The 'set msg.payload' nodes connect to 'カウント' (Count) and 'スイッチ' (Switch) nodes. On the right side of the interface, there is a 'デプロイ' (Deploy) button in the top right corner, and a sidebar with a 'ダッシュボード' (Dashboard) section containing a list of tabs and links.

スイッチの押下によるグラフの変化を確認しましょう。

# Node-RED 動作確認結果

過去1分間の押下回数のグラフ表示と、押下ごとに表示が変化します。

