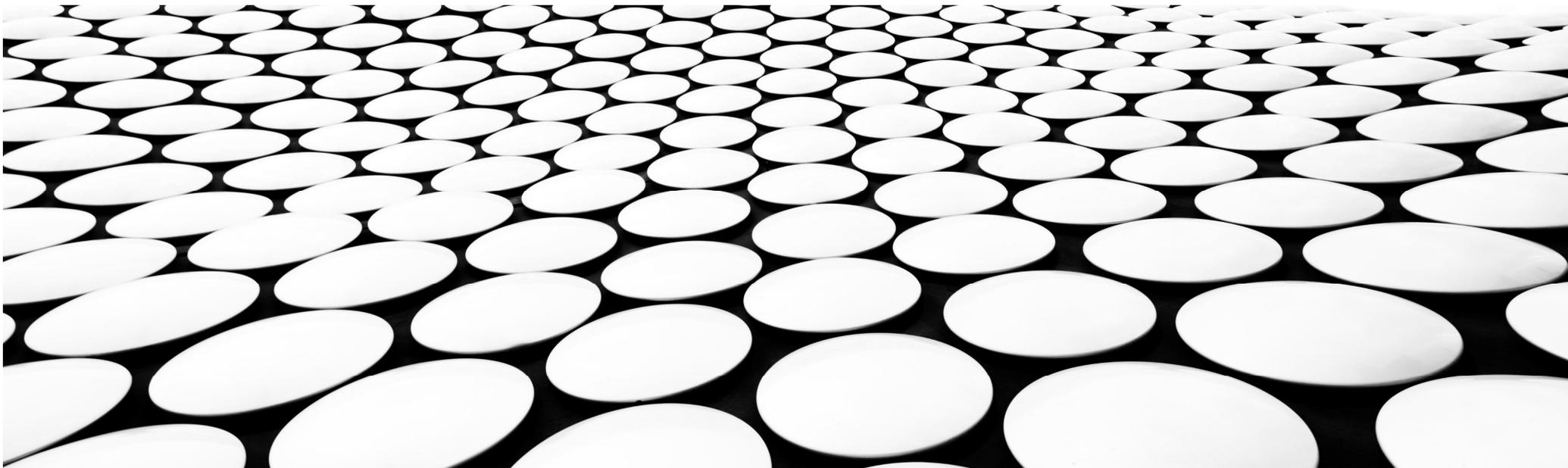
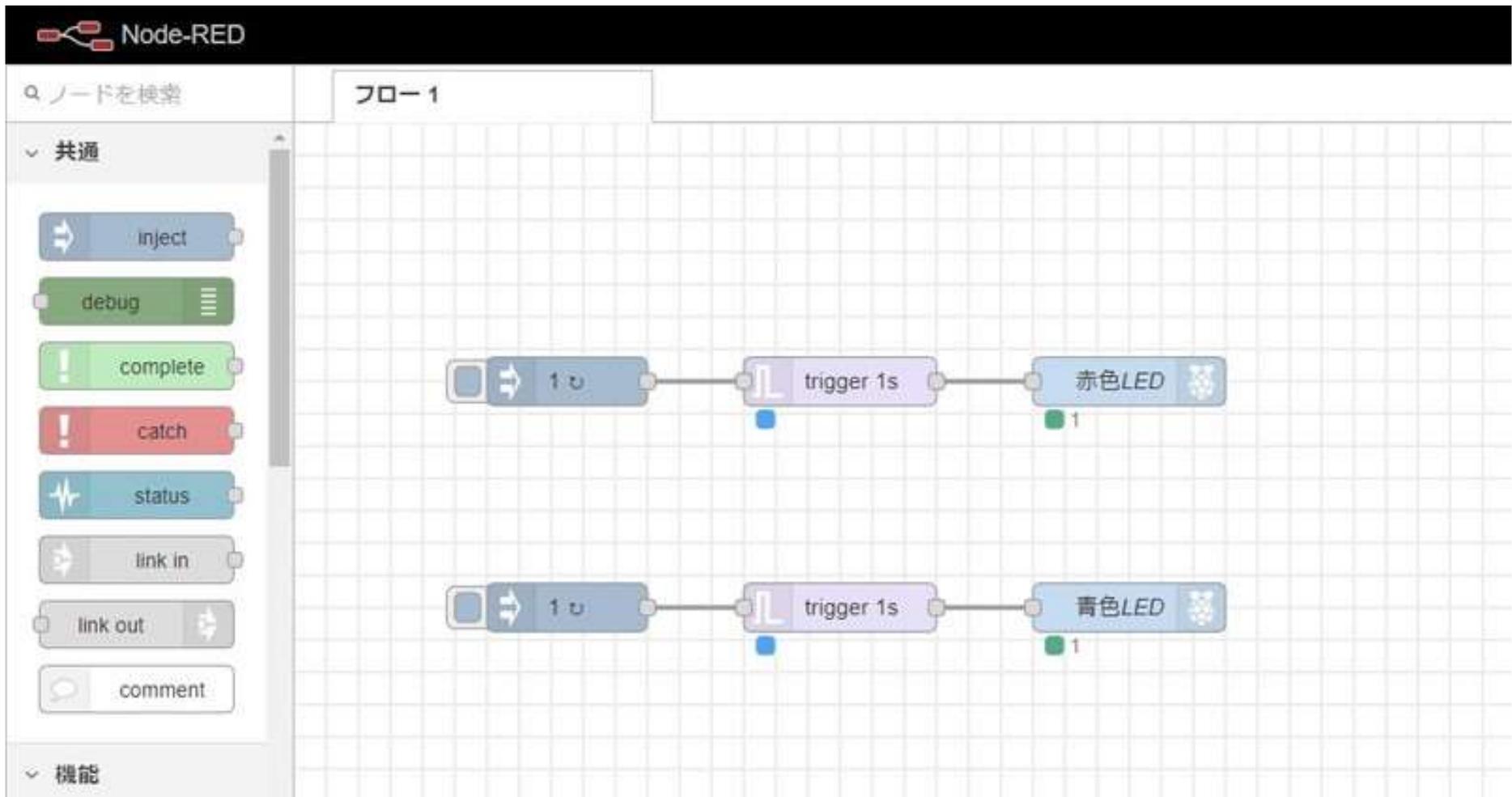

NODE-REDのインストール

SIT DOLOR AMET



NODE-REDとはオープンソースソフトウェアです。

- 2013年、IBM社の「英国ハースリー研究所」で開発されました。
- IoTの一部としてハードウェアデバイス、API、オンラインサービスを相互に接続するための開発ツールを目的として作られました。
- その後2016年、IBMはオープンソースとして「JS Foundation」に移管されました。
- 日本の企業ではIoTのプラットフォーム「Lumada」の開発に取り組んでいる日立製作所が積極的に開発に取り組んでいます。



特徴

- Node-REDの特徴として以下の事が挙げられます。
- **コーディング未経験者**でも、開発が容易
- **PC・タブレット・スマホのブラウザ上**で開発可能
- **IoT、Webサービスが簡単に**開発可能
- **ハードウェアとソフトウェアを簡単に繋げるための架け橋**としてNode-REDはとても簡単で強力なツールとなっています。
- 現在では、本格的なプロセス制御や産業機器制御などの**工場内でのロボットの制御**や、**プログラミング教育現場**での使用用途として拡大し、**教育から産業用途まで幅広く使用できるツール**です。

ラズパイにNODE-REDをインストールする

ラズパイの環境は、Raspberry Pi OSです。
2つのインストール方法をご紹介します。
1. コマンドでインストールする方法
2. マウス操作してインストールする方法

①コマンドでインストールする方法

こちらの方法はssh接続でインストールできるので、**普段からssh接続で利用している方向けのインストール方法**です。

[Node-RED公式サイト](#)に、コマンドが用意されています。

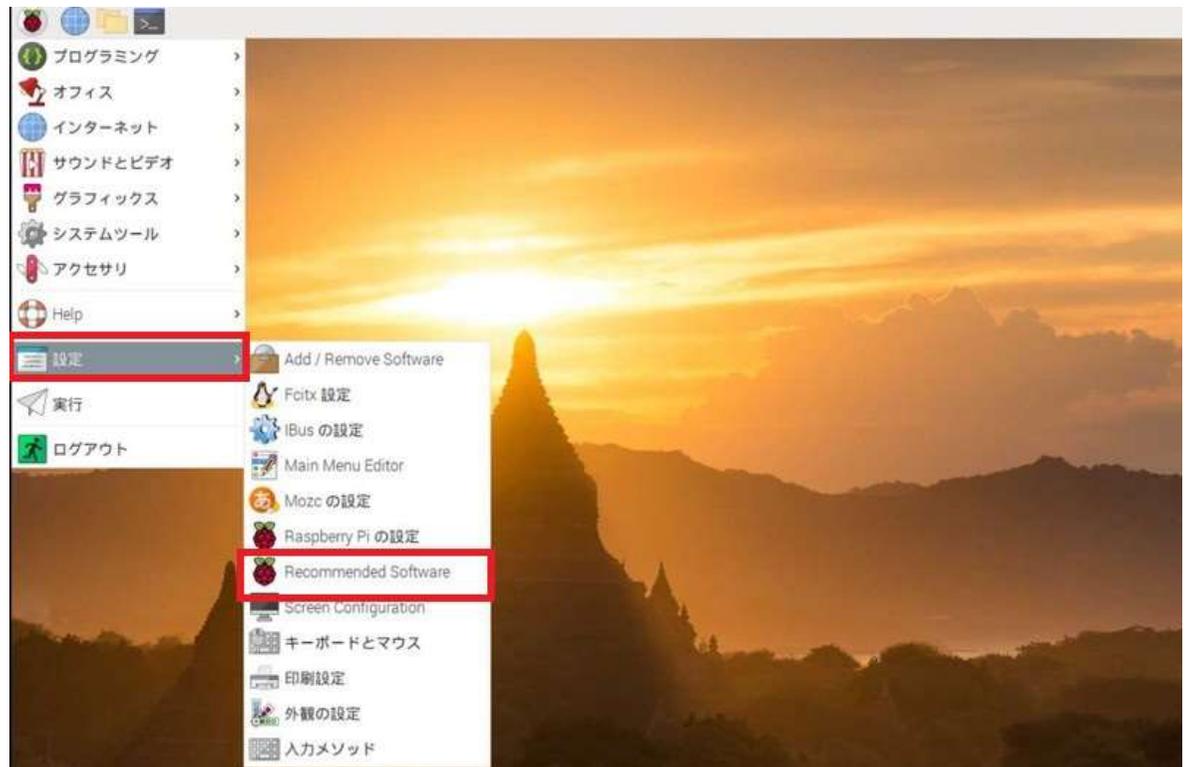
次のコマンドをターミナルで実行します。

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

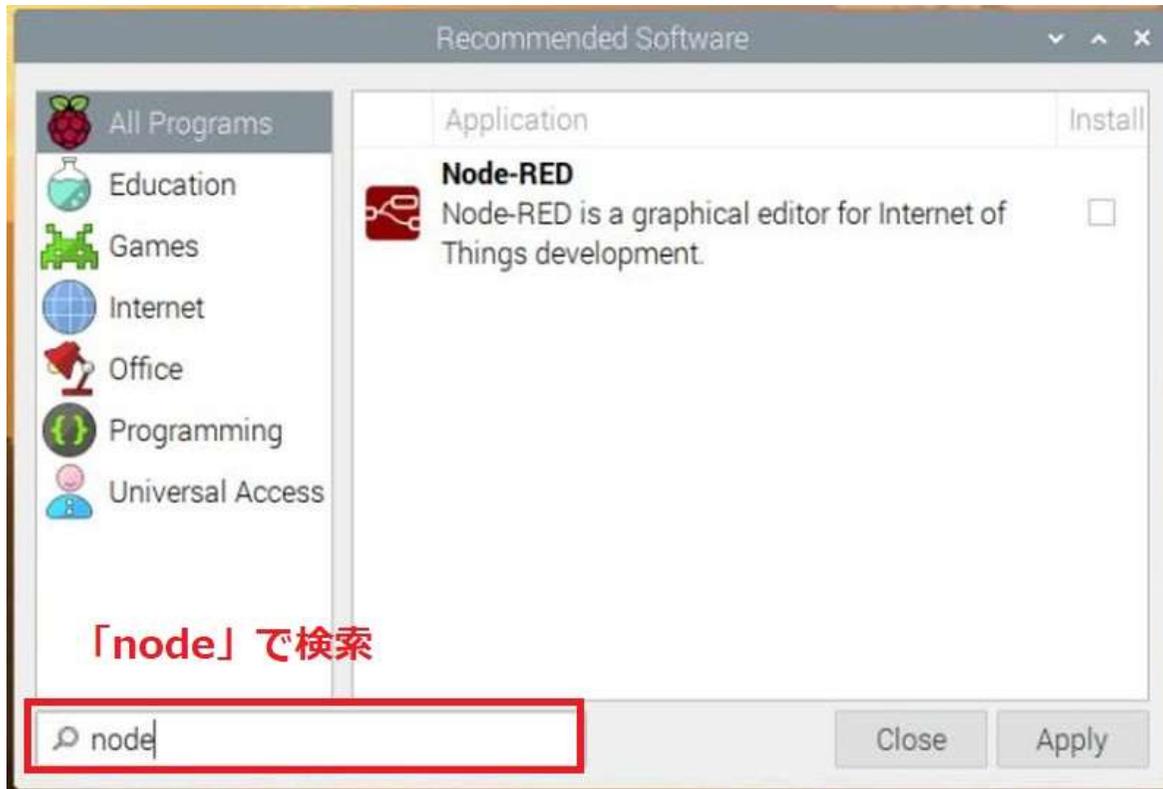
これでインストール完了です。

② マウス操作してインストールする方法

- こちらの 방법은、**普段からラズパイをマウスで操作している方向けのインストール方法**です。
- ラズパイの画面で「設定」→「Recommended Software」の順にクリック。



続いて、検索窓で「node」を入力し、チェックを入れ「Apply」をクリック。



これでインストールが完了です。

Lチカしてみる

Node-REDを起動する

ラズパイのターミナルで次のコマンドを実行して、Node-REDを立ち上げます。

```
node-red-start
```

すると下の画像のように沢山の情報が出力され、最後の行に「サーバは `http://xxx` で実行中です」という結果が表示されます。

NORD-REDの起動

- 起動時に表示される、一番初めの部分の「http://xxx:1880」の部分がNode-REDのサーバーのドメインになります。

xxxの部分は、ラズパイのIPアドレスです。

私の場合は、ラズパイのIPアドレスが「192.168.100.180」です。

NORD-REDの起動



- Node-REDはブラウザ上で開発をするので、PCやタブレットなどでChromeを開きます。
- Chromeで先ほどの赤枠の部分を入力すれば、ブラウザ上でNode-REDの画面になります。

NODE-REDのフローを作成する

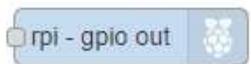
それでは、Node-REDでLチカのフローを組んでみましょう。
今回使うフローは以下の3つです。



入力値を与えることができる



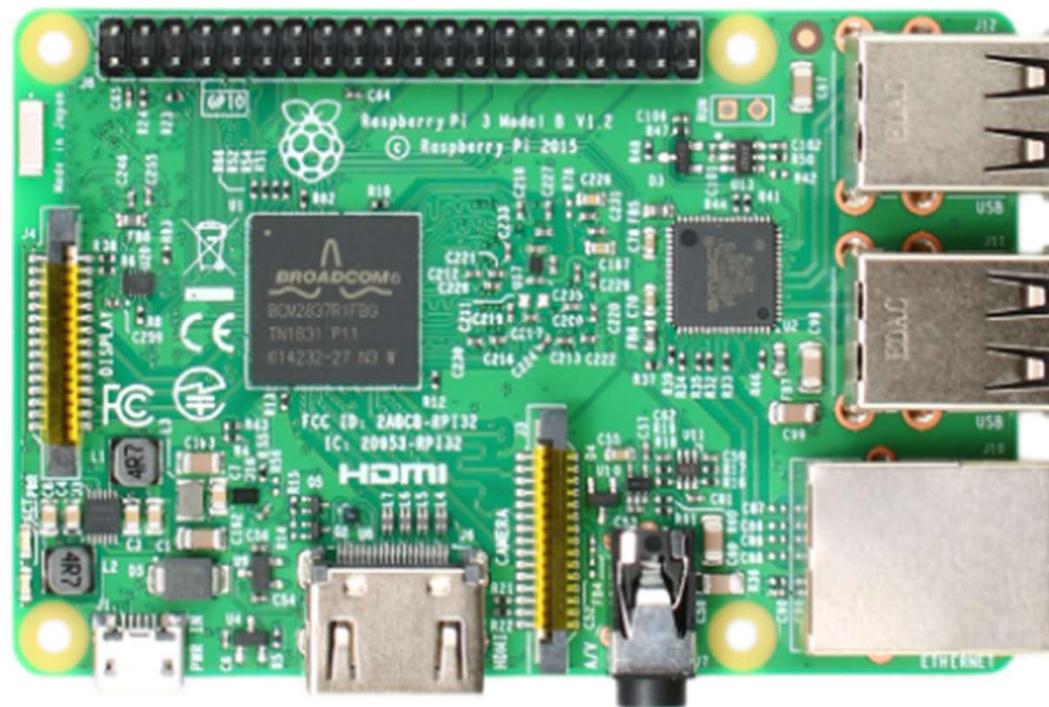
指定した時間でオンとオフを切り替えられる



ラズパイのgpioの出力値を与えることができる

Lチカの回路を組む

- 今回使用した物
- ラズパイ3を使用しましたが、ラズパイzeroのような安価なモデルでも十分に動作します。

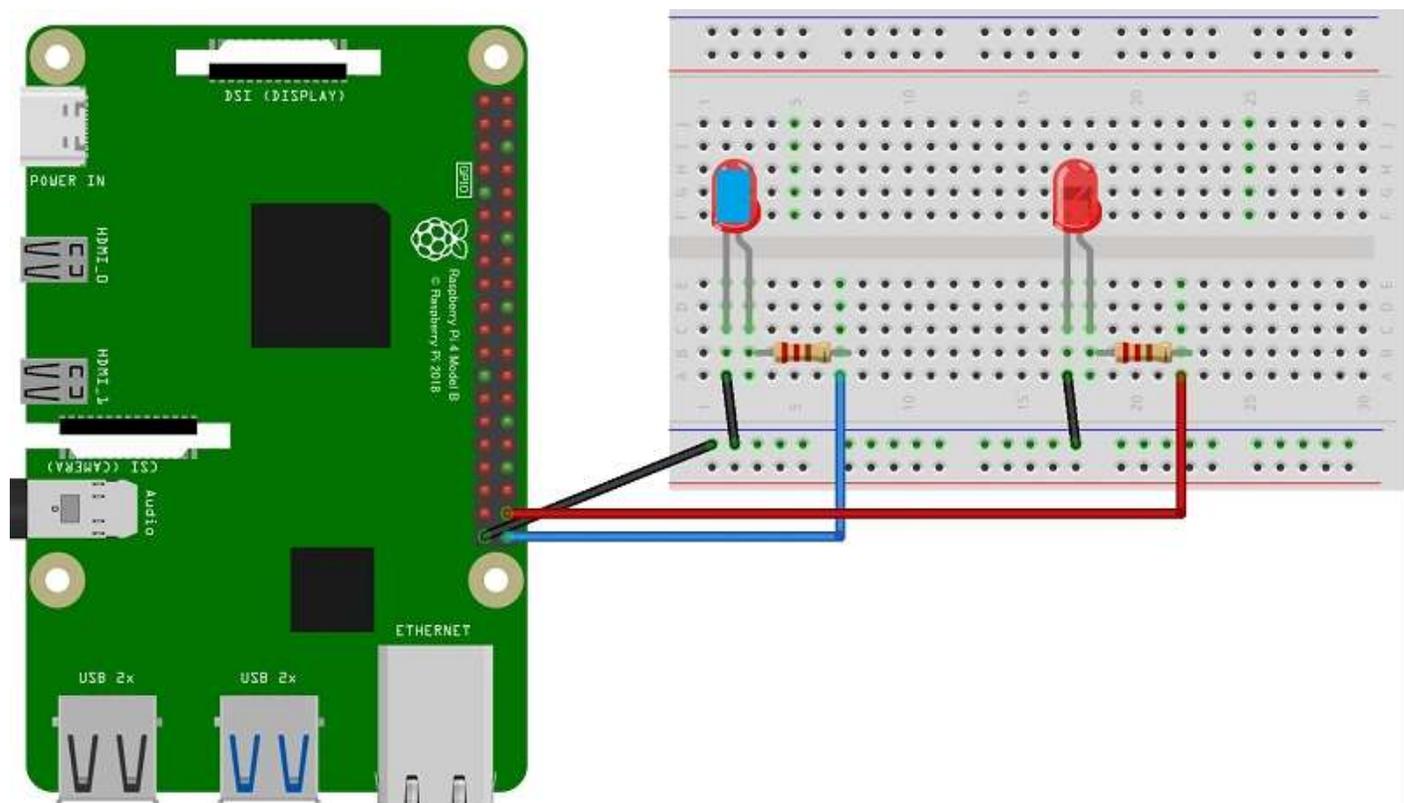


回路図

2種類のLEDをLチカしてみます。

抵抗は300Ω前後のものを
使用します。

赤色LEDはGPIO20、
青色LEDはGPIO21に
接続します。



まず、injectノードを①ドラッグアンドドロップし、②ダブルクリックし、③プロパティに入力します。

出力値は” 1” とし、2秒間の繰り返しをします。

The screenshot shows the Node-RED web interface. On the left, the 'Common' (共通) node palette is visible, with the 'inject' node highlighted. A green arrow points from the 'inject' node to the workspace, labeled '①ドラッグアンドドロップ'. In the workspace, a 'inject' node is placed, and a red arrow points to it, labeled '②ダブルクリック'. The right-hand panel, titled 'inject ノードを編集', shows the configuration for the selected node. The 'msg.payload' property is set to '1', and the 'Repeat' (繰り返し) section is configured with a duration of 2 seconds. The 'Repeat' section includes a dropdown menu for '指定した時間間隔' (Selected interval) and a numeric input field set to '2' with a unit of '秒' (seconds).

次に、同様にtriggerノードのプロパティに入力します。
1秒の待機とします。

The screenshot displays the Node-RED web interface. On the left, a sidebar contains a search bar and a list of nodes including 'catch', 'status', 'link in', 'link out', 'comment', 'function', 'switch', 'change', 'range', 'template', 'delay', 'trigger', 'exec', and 'in'. The main workspace, titled 'フロー 1', shows a flow with two nodes: a '1 u' node and a 'trigger 250ms' node. On the right, the 'trigger ノードを編集' (Edit trigger node) panel is open. It features a 'プロパティ' (Properties) section with the following settings: '送信データ' (Send data) set to '1', '送信後の処理' (Action after sending) set to '指定した時間待機' (Wait for specified time), and a time field set to '1 秒' (1 second), which is highlighted with a red underline. Below this, there are two unchecked checkboxes: '新たなメッセージを受け取った時に遅延を延長' (Extend delay when receiving a new message) and 'msg.delayを用いて遅延時間を上書き' (Override delay time using msg.delay). The '再送信データ' (Resend data) field is set to '0'. The '初期化条件' (Initialization conditions) section includes 'msg.resetを設定' (Set msg.reset) and 'msg.payloadが次の値' (msg.payload is the following value) with a text input field containing '任意' (Arbitrary). The '処理対象' (Target for processing) is set to '全メッセージ' (All messages). The '名前' (Name) field is empty.

次に、rpi-gpio-outノードも同様にプロパティに入力します。
gpio20を選択し、名前を「赤色LED」とします。

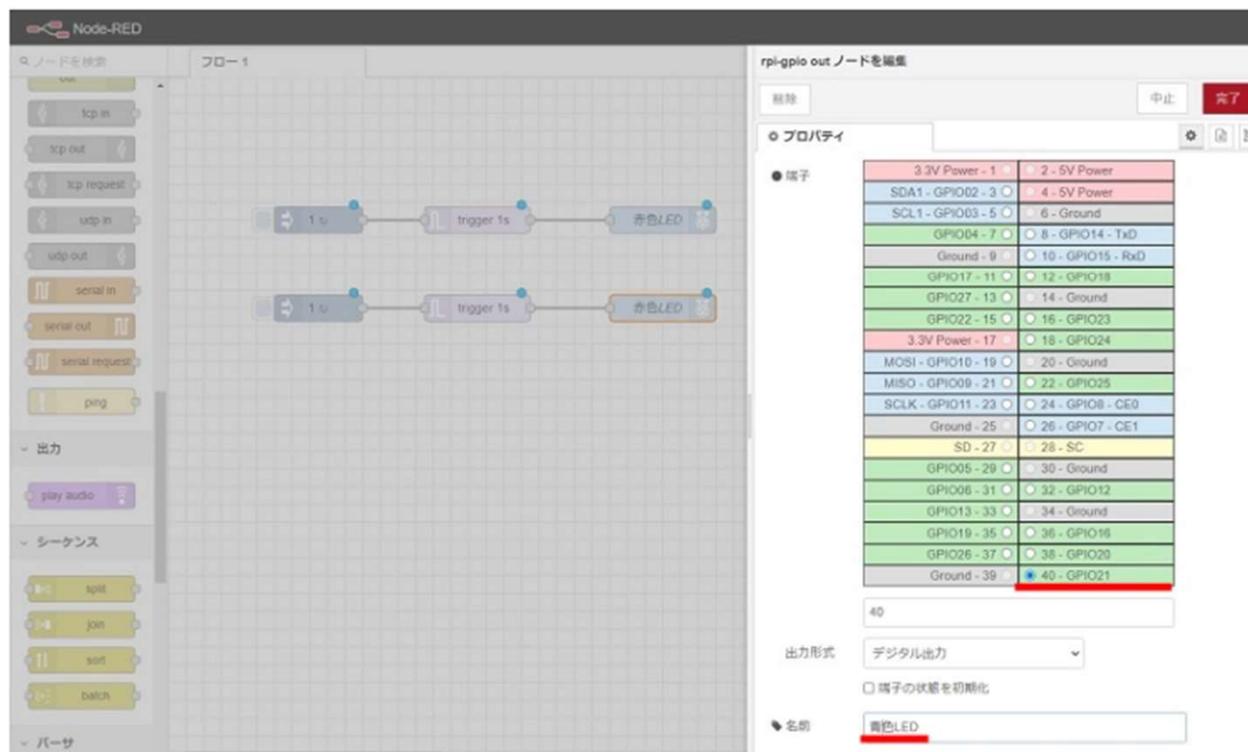
The screenshot shows the Node-RED web interface. On the left, a sidebar contains various node categories: 'batch', 'パーサー' (Parser) with nodes like 'csv', 'tsv', 'json', 'xml', 'yaml'; 'ストレージ' (Storage) with nodes like 'file', 'file in', 'watch', 'tail'; and 'Raspberry Pi' with nodes like 'rpi-gpio in', 'rpi-gpio out', 'rpi-mouse', and 'rpi-keyboard'. The main workspace shows a flow with three nodes: '1 s', 'trigger 1s', and 'PIN'. The 'PIN' node is selected, and its configuration panel is open on the right.

The configuration panel for the 'rpi-gpio out' node is titled 'rpi-gpio out ノードを編集'. It has buttons for '削除' (Delete), '中止' (Cancel), and '完了' (Done). Under the 'プロパティ' (Properties) section, there is a '端子' (Pin) list. The list contains 40 items, each with a radio button. The item '38 - GPIO20' is selected and highlighted with a red background. Below the list, the value '38' is entered in a text field. The '出力形式' (Output Format) is set to 'デジタル出力' (Digital Output). There is a checkbox for '端子の状態を初期化' (Initialize pin state) which is currently unchecked. At the bottom, the '名前' (Name) field contains the text '赤色LED' (Red LED), which is also highlighted with a red background.

端子	
<input type="radio"/> 3.3V Power - 1	<input type="radio"/> 2 - 5V Power
<input type="radio"/> 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 - 5C
<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 checked="" type="radio"/> 38 - GPIO20
<input type="radio"/> Ground - 39	<input type="radio"/> 40 - GPIO21

これで赤色LEDのノードの準備は整ったので、次はこれらをコピーして青色LEDのノードを作ります。

rpi-gpio-outのノードをダブルクリックして、gpio21を選択し名前を「青色LED」とします。



あとはこれらを線でつなげば、フローの完成です。

あとはこれらを線でつなげば、フローの完成です。

右上の「デプロイ」をクリックすれば動作が開始します。

