

# CHIRIMEN for Raspberry Pi 3

## ハンズオン講習会

@Web×IoT メイカーズチャレンジ 2019-20 in 群馬  
講師：宮崎 典行

## 本日のゴール

各種センサーやアクチュエーターの制御方法を理解すること

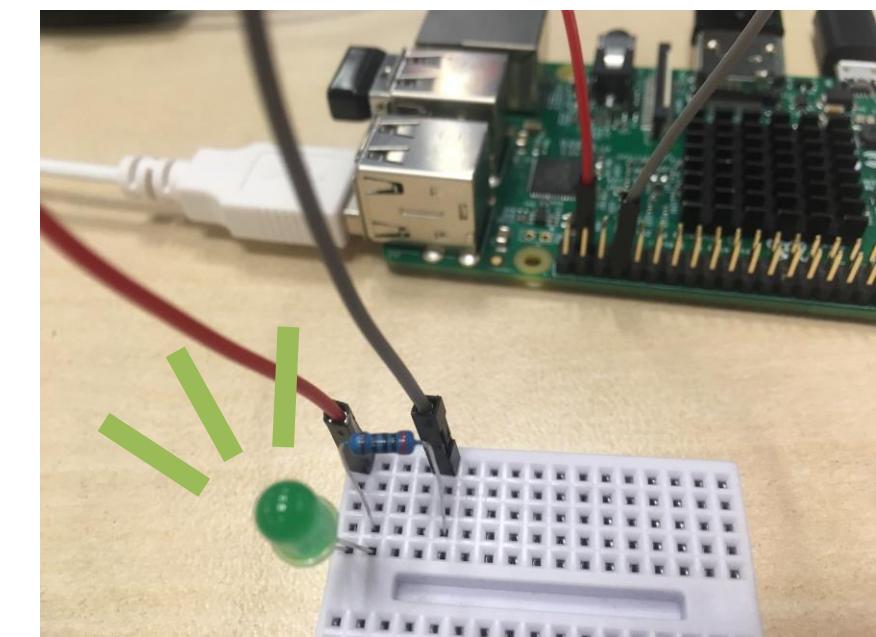
### 基礎編

0. Lチカしてしてみよう(初めてのGPIO) ※全体講習
1. GPIOの使い方
2. センサーを使ってみよう(初めてのI2C)

### 応用編

3. I2Cの使い方
4. GPIO/I2Cのまとめ

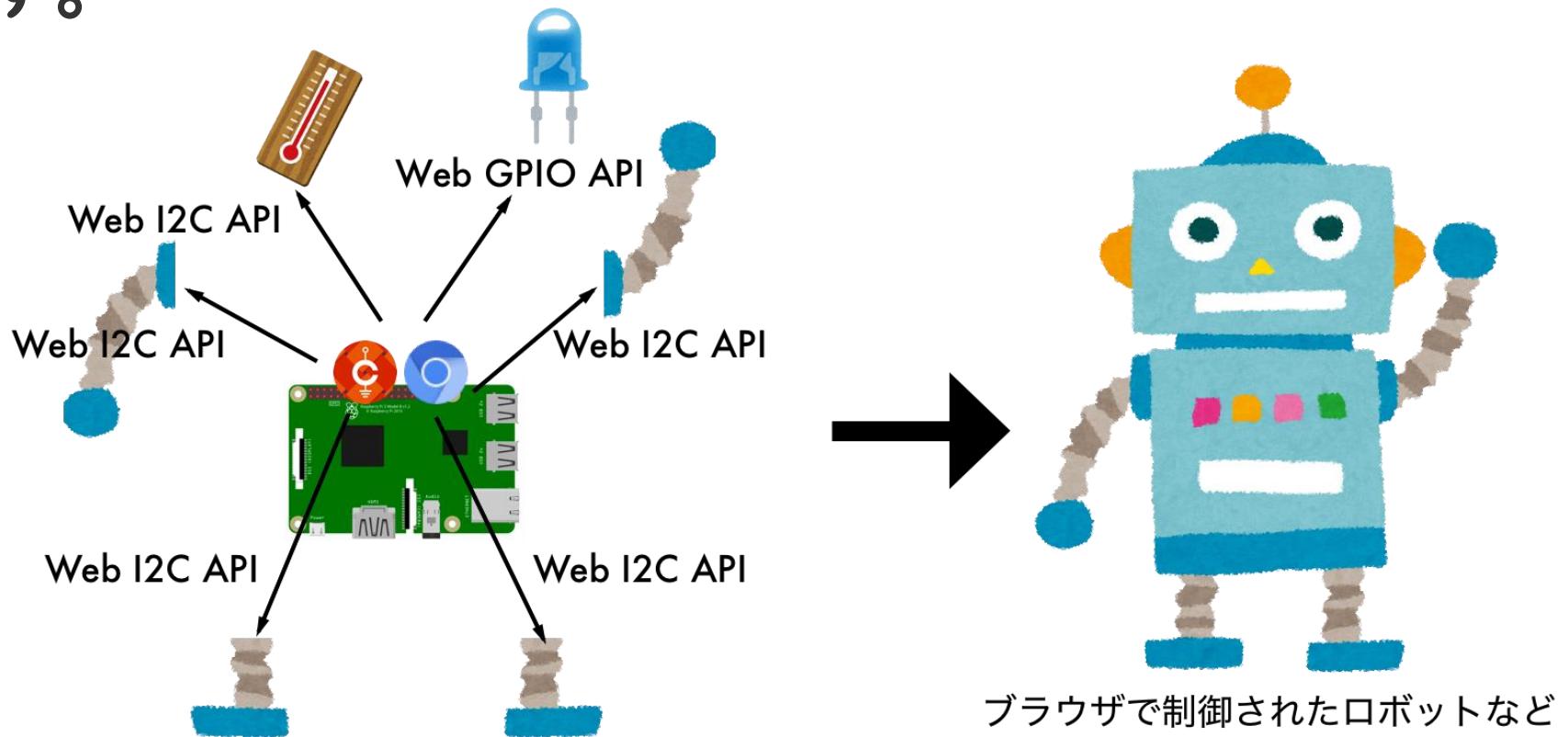
Examples (時間があれば)





CHIRIMEN for Raspberry Pi 3は、Web ブラウザからハードウェアを制御するプロトタイピング環境です。

センサーやモーターなどを組み合わせて様々なデバイスを作ることができます。





# Web ブラウザからハードウェアを制御できること

## ■ Lチカプログラムの例

```
window.onload = async function() {
    var gpioAccess = await navigator.requestGPIOAccess(); // GPIO を操作する
    var port = gpioAccess.ports.get(26); // 26 番ポートを操作する
    var v = 0;

    await port.export("out"); // ポートを出力モードに設定
    for (;;) { // 無限に繰り返す
        v = v === 0 ? 1 : 0; // ポートの出力値を 0/I 交互に変更
        port.write(v); // LED を ON/OFF する
        await sleep(1000); // 繰り返し毎に 1000ms 待機
    }
};
```



# 【参考】作品例 (Web×IoT メイカーズチャレンジ 2018-19 in 鳥取) 5

## IoTの基礎知識とアイディアで多彩な作品が生まれました！

**デジタルサイネージ**  
照度センサー×人感センサー



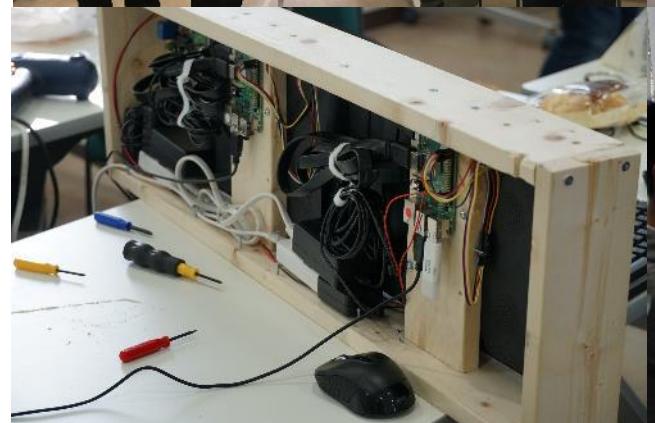
**雪かき頑張りの可視化**  
加速度センサー×温度センサー



**スマートごみ箱**  
画像認識×メール通知



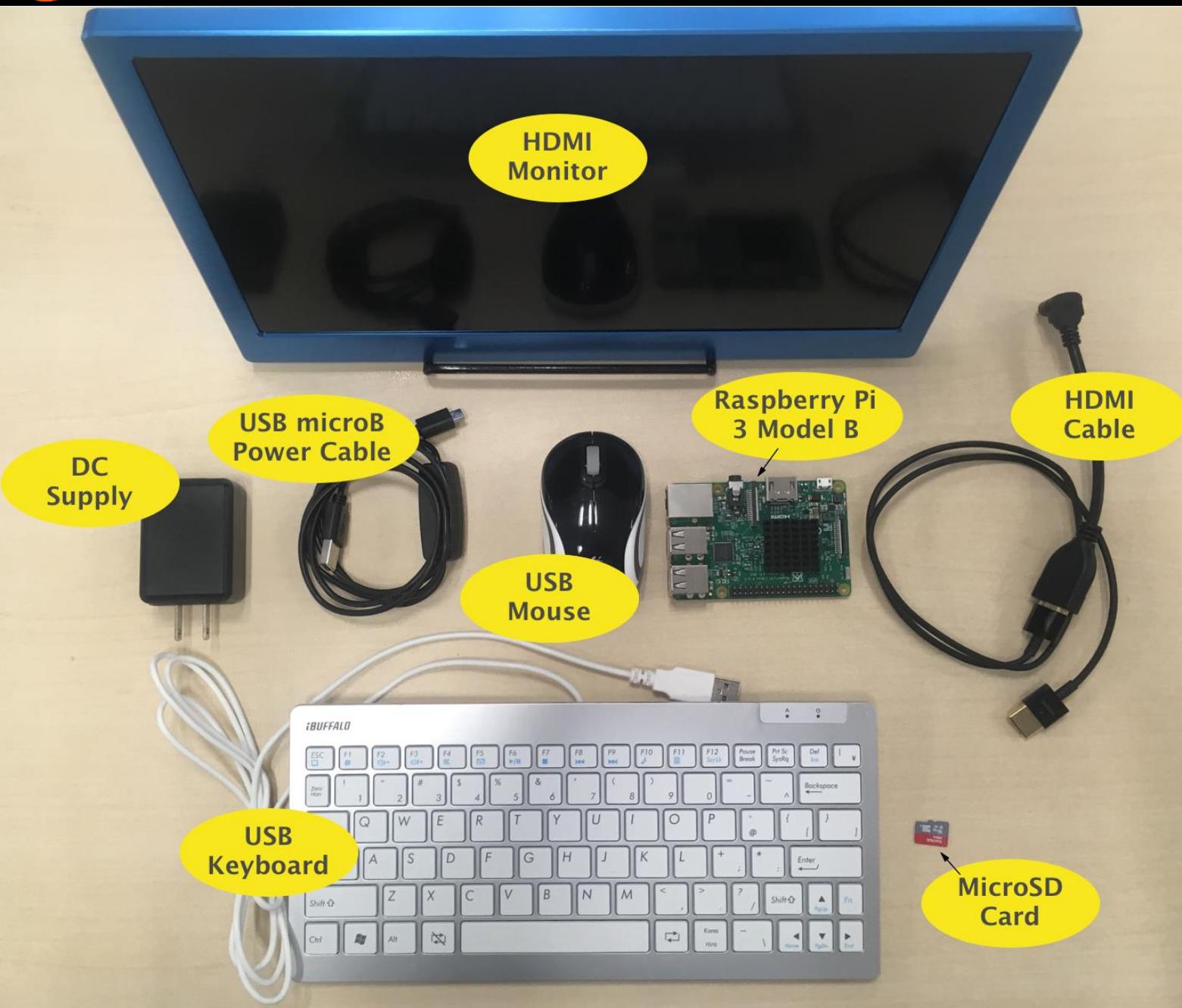
**忘れ物防止かばん**  
距離センサー×ディスプレイ通知





# 基本ハードウェア

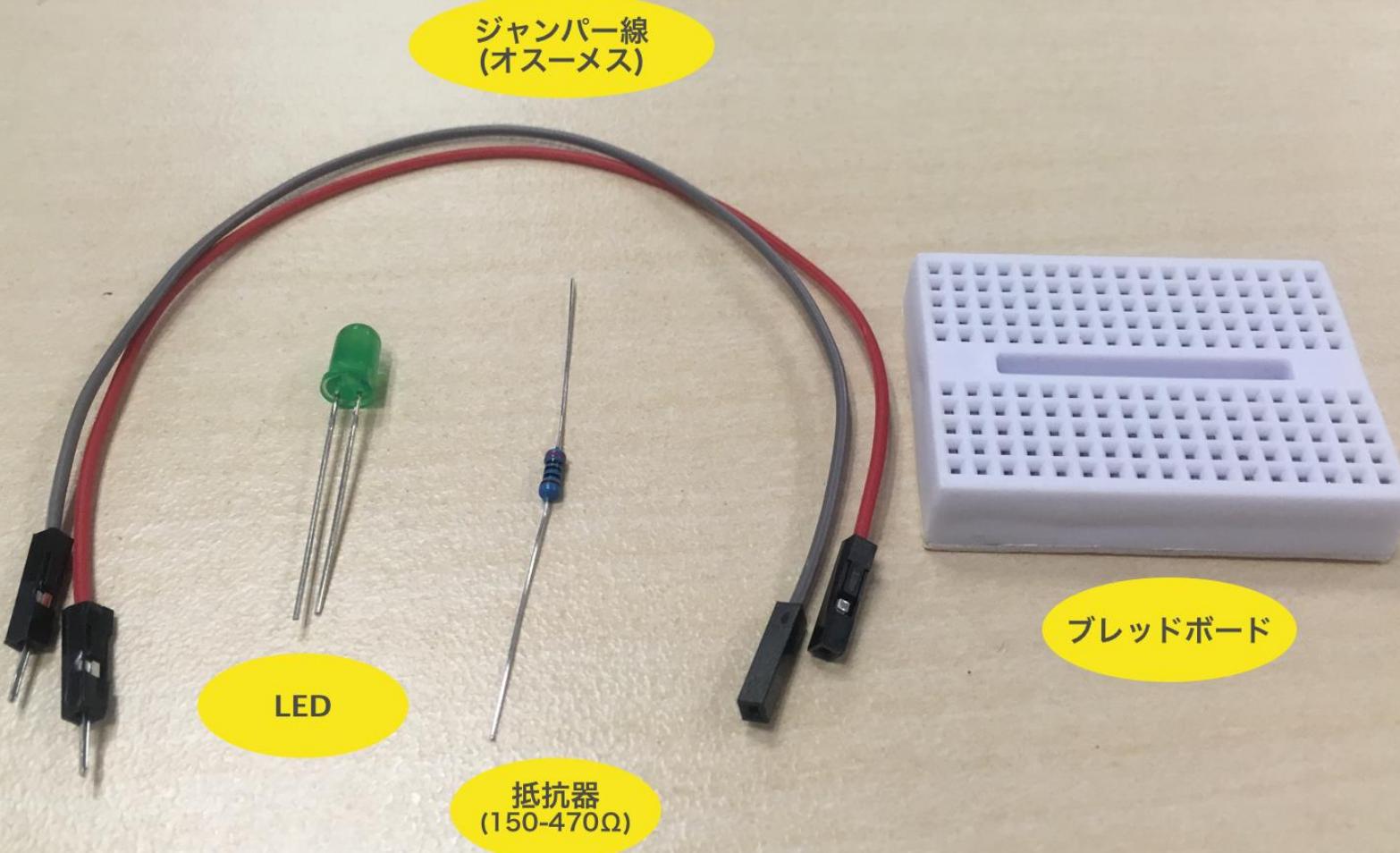
6



- Raspberry Pi 3 Model B×1
- ACアダプタ×1
- micro B USB電源ケーブル×1
- HDMIケーブル×1
- HDMI入力つきのモニタ×1
- USBマウス×1
- USBキーボード×1
- micro SDカード(8GB以上)×1

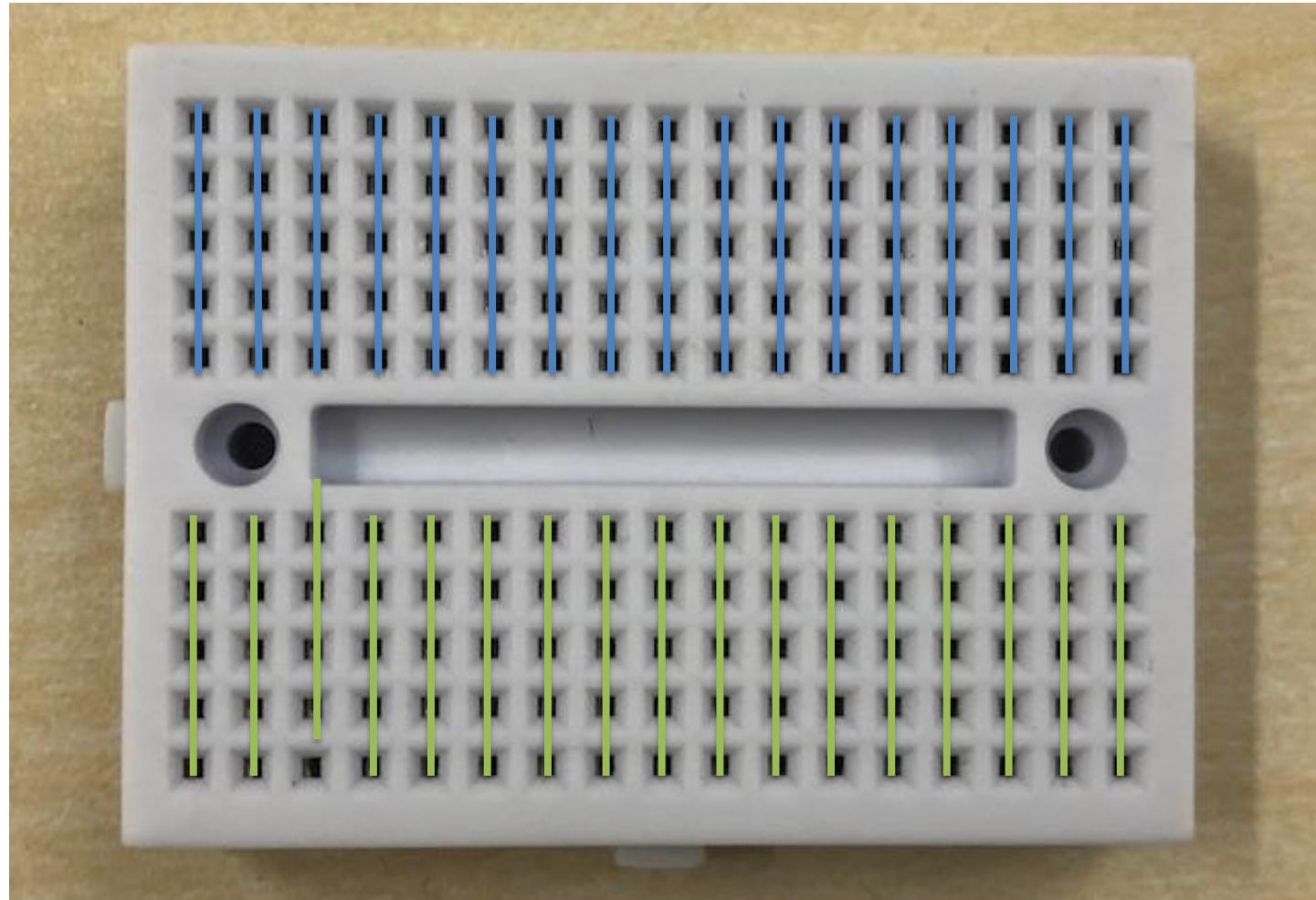
# C Lチカに必要となるパーツ

7



- ブレッドボード×1
- リード付きLED×1
- リード付き抵抗器  
(150-470Ω)×1
- ジャンパーウイヤー<sup>（オス-メス）</sup>×2

手軽に電気回路を作ることができます。



内部電気回路接続図

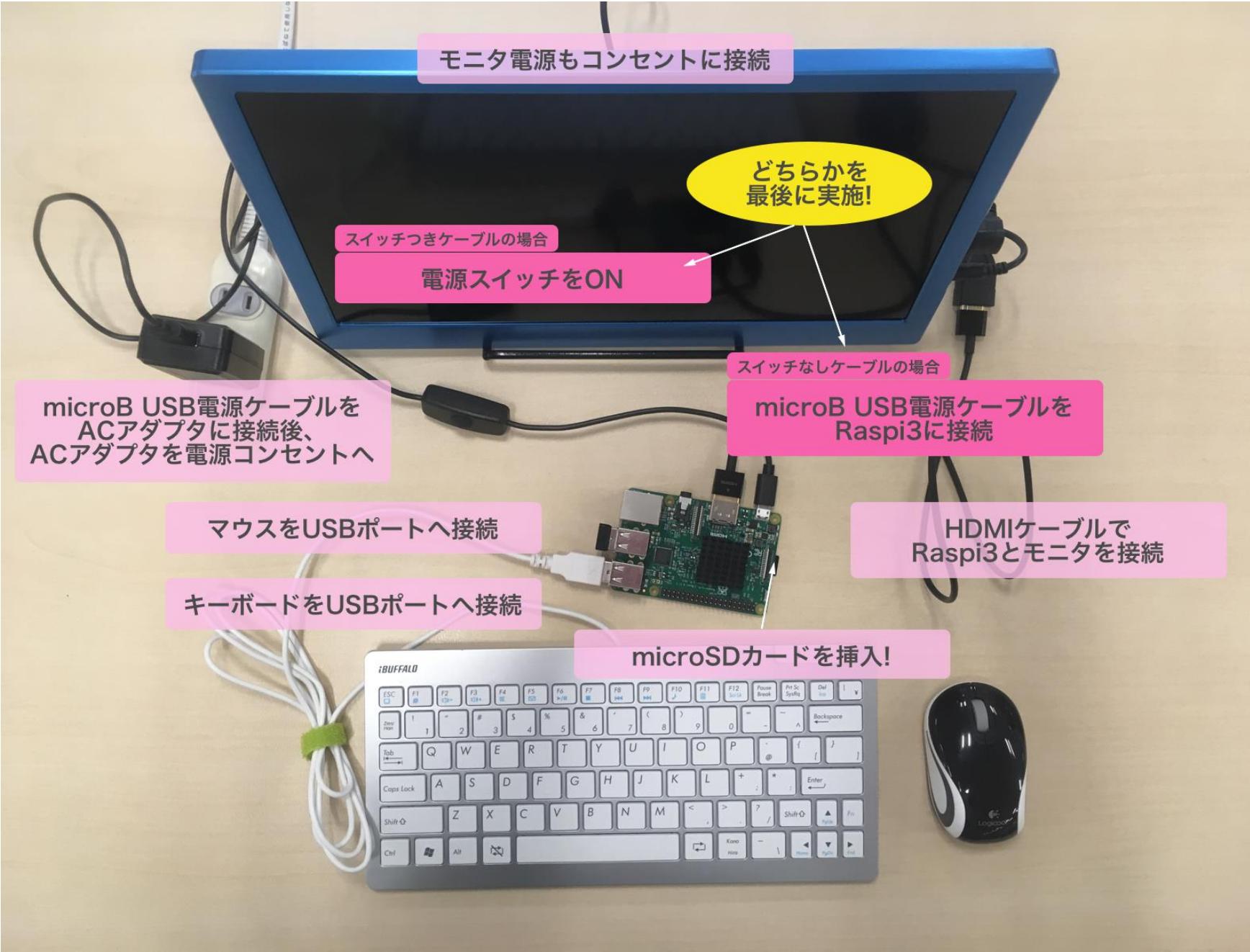


ブレッドボード裏面



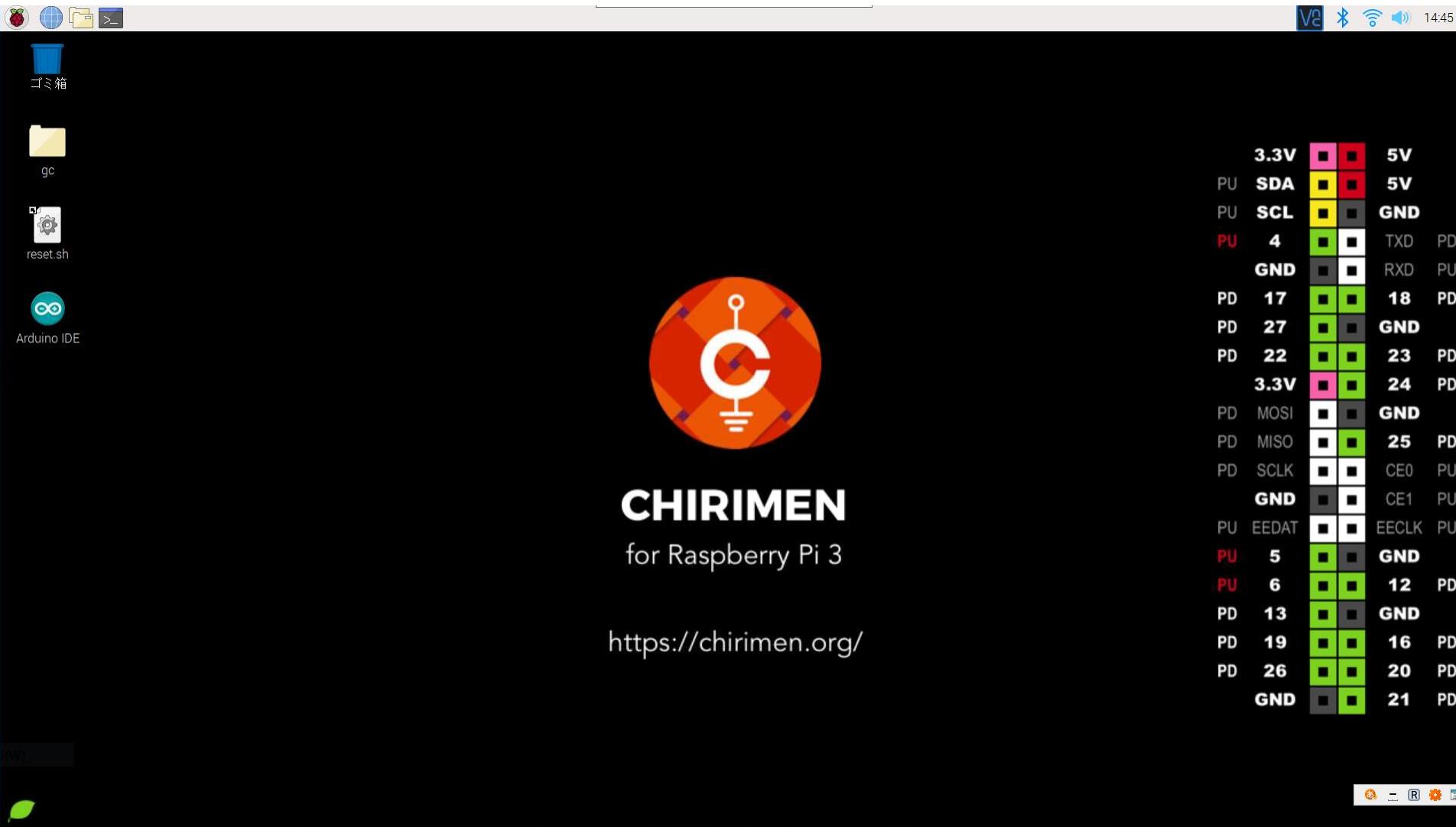
# CHIRIMEN for Raspberry Pi 3 の起動

10



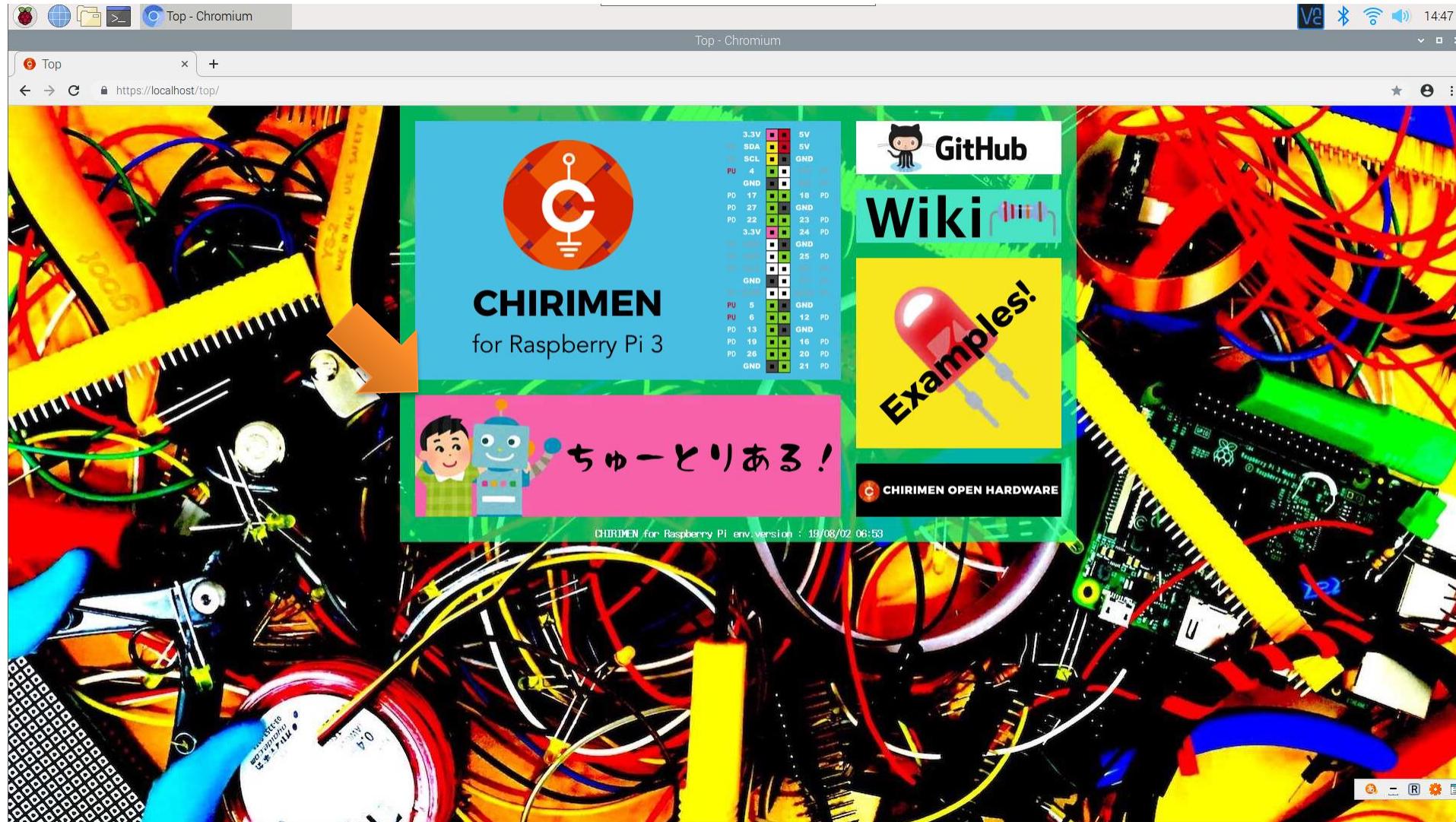


## デスクトップ画面





ブラウザ自動起動後、「ちゅーとりある！」を選択





# 「CHIRIMEN for Raspberry Pi 3」用を選択

```
await port.export('out'); // ポートを出力に設定
for (;;) { // 無限に繰り返す
    v = v === 0 ? 1 : 0; // ポートの出力値を 0/1 交互に変更
    port.write(v); // LED を ON/OFF する
    await sleep(1000); // 繰り返し毎に 1000ms 待機
}
};
```

少しスクロール



ブラウザだけで簡単に[サンプルコードの確認・編集](#)から公開までできます。Web開発の知識と環境が全てそのまま活かせる  
CHIRIMENは、素早くハードとソフトを融合させたプロトタイピングを行ったり、最も人気で実践的なプログラミング言語JavaScript  
とIoTをプログラミング初心者でも楽しく簡単に学ぶのに最適な環境です。詳しくは[CHIRIMENについて](#)ページをご覧ください。

## チュートリアル

上記のようにWebGPIO, WebI2Cが使えるCHIRIMEN環境は現在Raspberry Pi, TY51822r3, micro:bit向けに実装されています。お  
ののボードに合わせたものをご覧ください:

- 
- [CHIRIMEN for Raspberry Pi 3](#)
  - [CHIRIMEN with ty51822r3](#)
  - [CHIRIMEN with micro:bit](#)

注: Raspberry Pi 3ではボード上のブラウザからハードを制御しますが、TY51822r3やmicro:bitではPCやスマホなど他の端末のブ  
ラウザからWeb Bluetoothを使いリモート制御します。





# 基礎編 > 「0. Lチカしてみよう(初めてのGPIO)」を選択

The screenshot shows a VNC session connected to 192.168.11.4 (raspberrypi). The browser window title is 'CHIRIMEN for Raspberry Pi3 チュートリアル | CHIRIMEN チュートリアル - Chromium'. The URL is https://tutorial.chirimen.org/raspi3/. The page content is as follows:

LED と温度センサーを使ってみる最小限のお試しページです。より詳しくは以下のチュートリアルをご覧ください。

**基礎編**

まずはシンプルな GPIO 入出力や I2C センサーの操作方法を学びましょう。

- 0. Lチカしてみよう (初めての GPIO)
  - WebGPIO API を使って定期的に LED を点滅するサンプルを動かしてみます。
- 1. GPIO の使い方
  - マウスクリックで操作するブラウザ画面のボタンと物理スイッチ(タクトスイッチ)の両方で LED やモーターを制御するサンプルを通じて GPIO の基本を学びます。
- 2. センサーを使ってみよう (初めての I2C)
  - 温度センサーの値をドライバーを使う場合と WebGPIO API を直接操作する場合の 2 パターンで読み取ることで、I2C デバイス操作の基本を学びます。

**応用編**

いろいろな I2C デバイスを繋いだり組み合わせたりすることで IoT プロトタイピングを体験しましょう。

- 3. I2C の使い方
  - いろいろな I2C デバイスを接続したり、複数の I2C デバイスの同時操作について学びます。

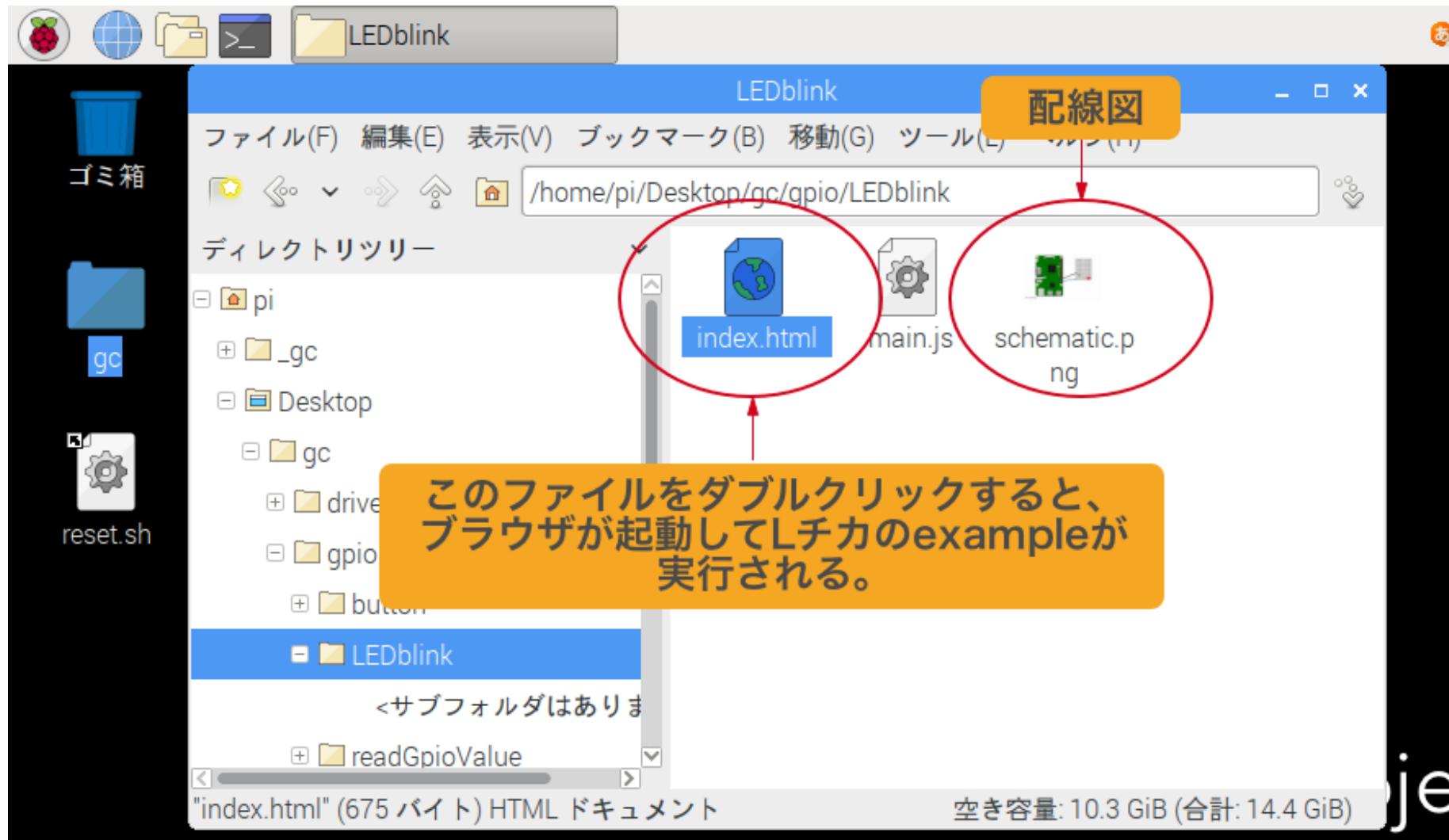
On the right side of the screenshot, there are two orange arrows: one pointing left labeled '少しスクロール' and one pointing down.



# Lチカしてみよう!!

15

1. デスクトップから gc/gpio/LEDblink/ フォルダを開きます。
2. 配線図 (schematic.png) を開きます。





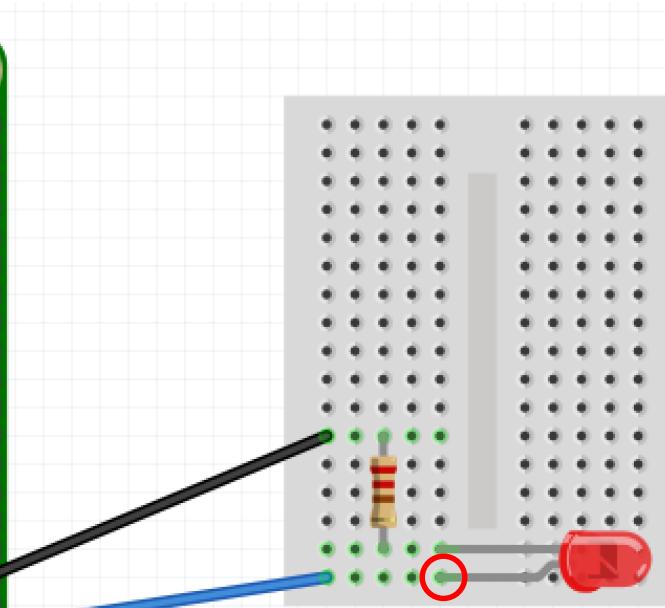
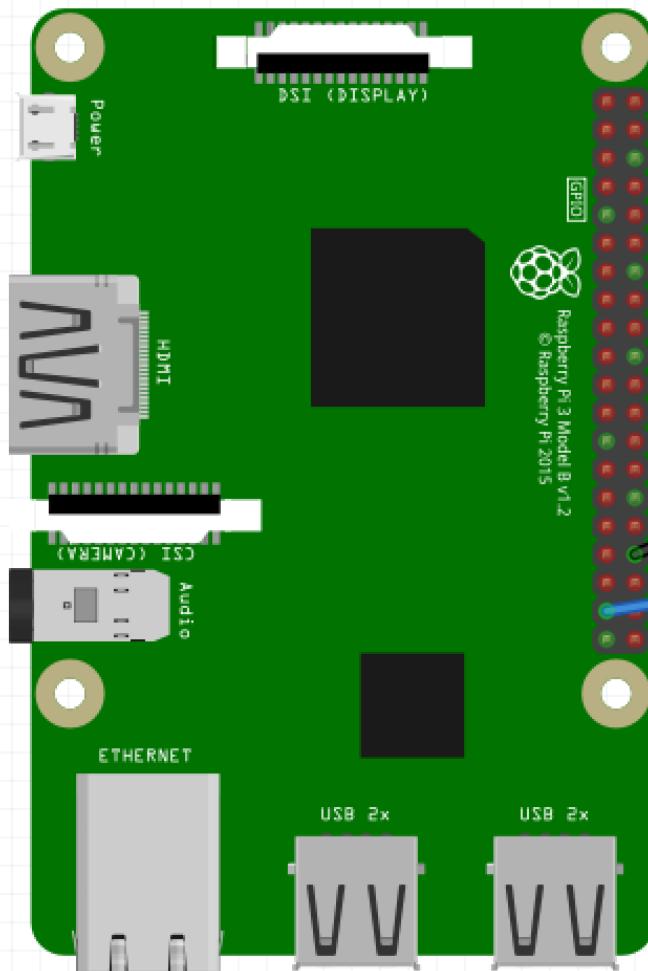
Lチカしてみよう!!

16

LED、抵抗器、ジャンパー線をRaspi3に接続してみよう!!

|       |          |         |       |  |
|-------|----------|---------|-------|--|
| 3.3V  | [Pink]   | [Red]   | 5V    |  |
| SDA   | [Yellow] | [Black] | 5V    |  |
| SCL   | [Yellow] | [Black] | GND   |  |
| 4     | [Green]  | [Black] | TxD   |  |
| GND   | [Black]  | [Black] | RxD   |  |
| 17    | [Black]  | [Black] | 18    |  |
| 27    | [Black]  | [Black] | GND   |  |
| 22    | [Black]  | [Black] | 23    |  |
| 3.3V  | [Pink]   | [Green] | 24    |  |
| MOSI  | [Black]  | [Black] | GND   |  |
| MISO  | [Black]  | [Green] | 25    |  |
| SCLK  | [Black]  | [Black] | CE0   |  |
| GND   | [Black]  | [Black] | CE1   |  |
| EEDAT | [Black]  | [Black] | EECLK |  |
| 5     | [Green]  | [Black] | GND   |  |
| 6     | [Green]  | [Black] | 12    |  |
| 13    | [Black]  | [Black] | GND   |  |
| 19    | [Black]  | [Black] | 16    |  |
| 26    | [Green]  | [Black] | 20    |  |
| GND   | [Black]  | [Black] | 21    |  |

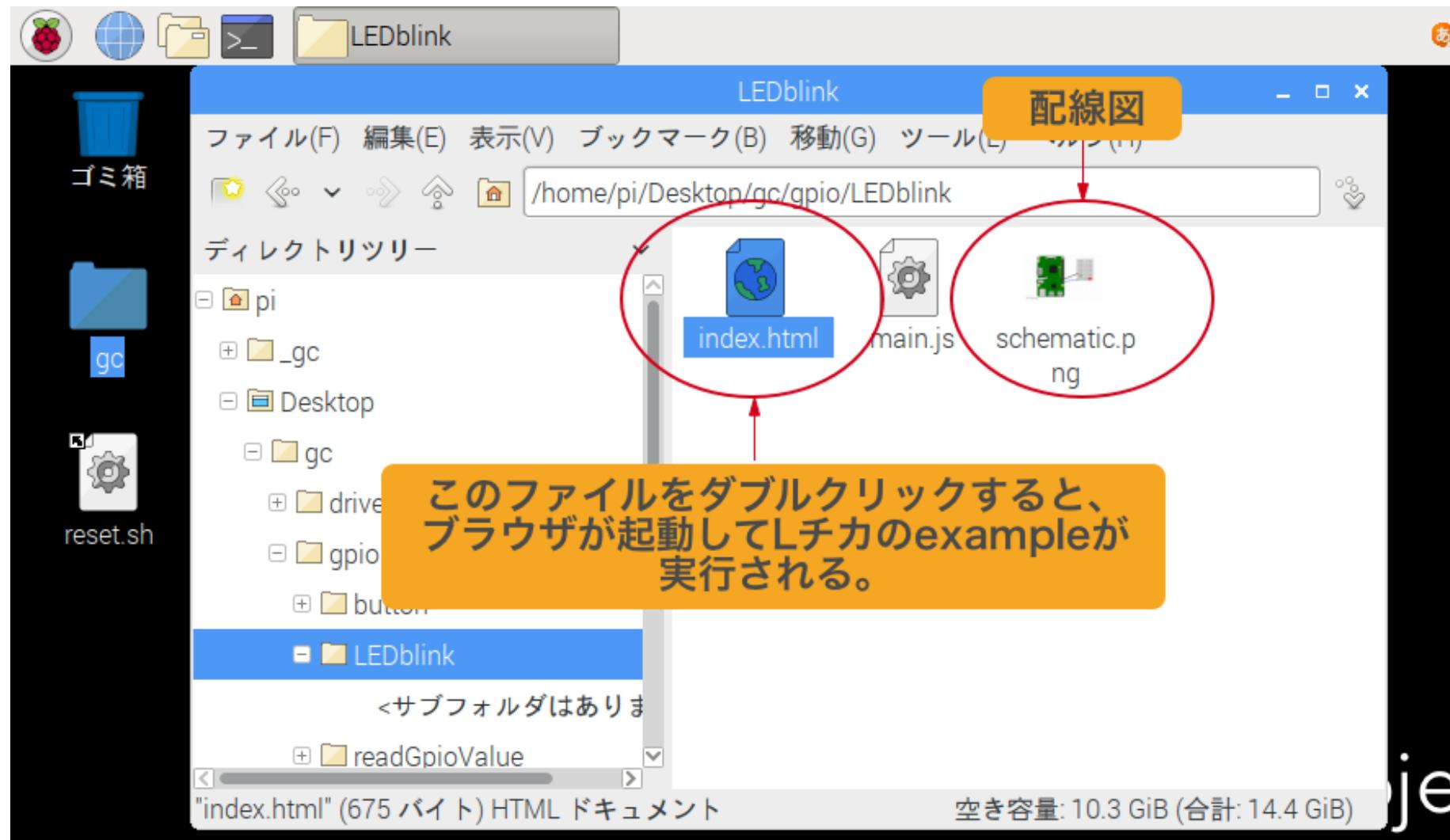
fritzing



LEDの接続方向に注意!!  
長い方がアノード(+)側  
※この絵では下側



3. リチカWebプログラム(index.html)を開きます。

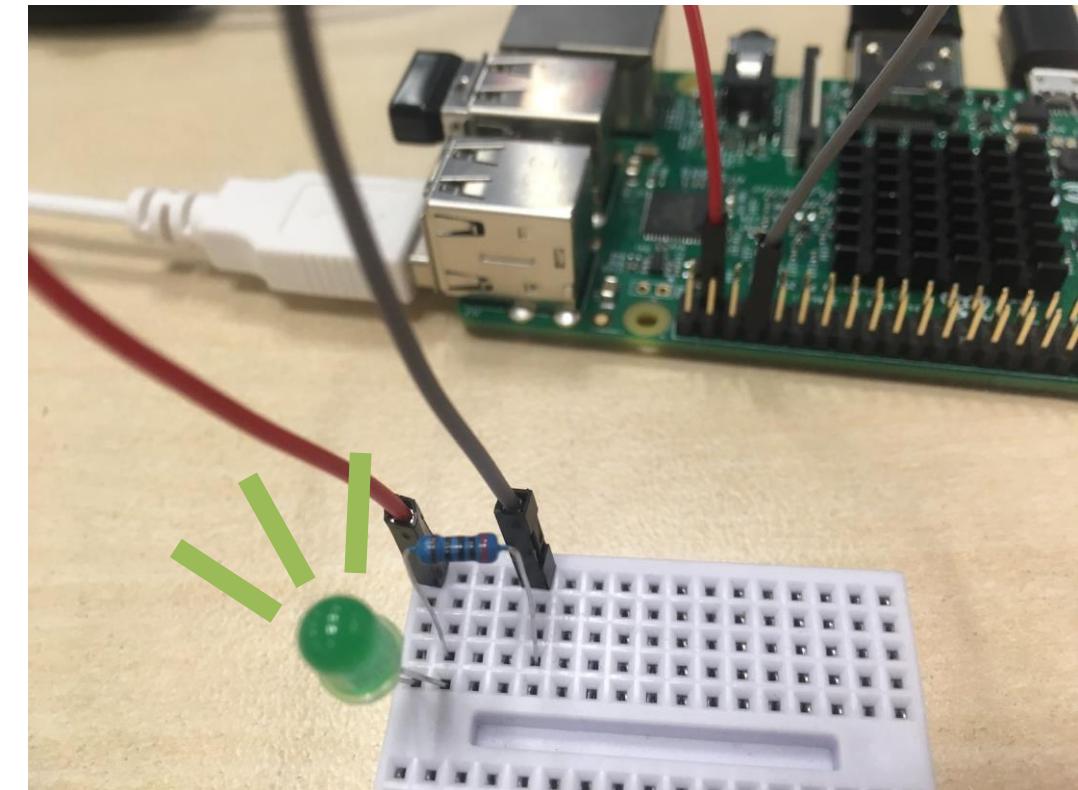
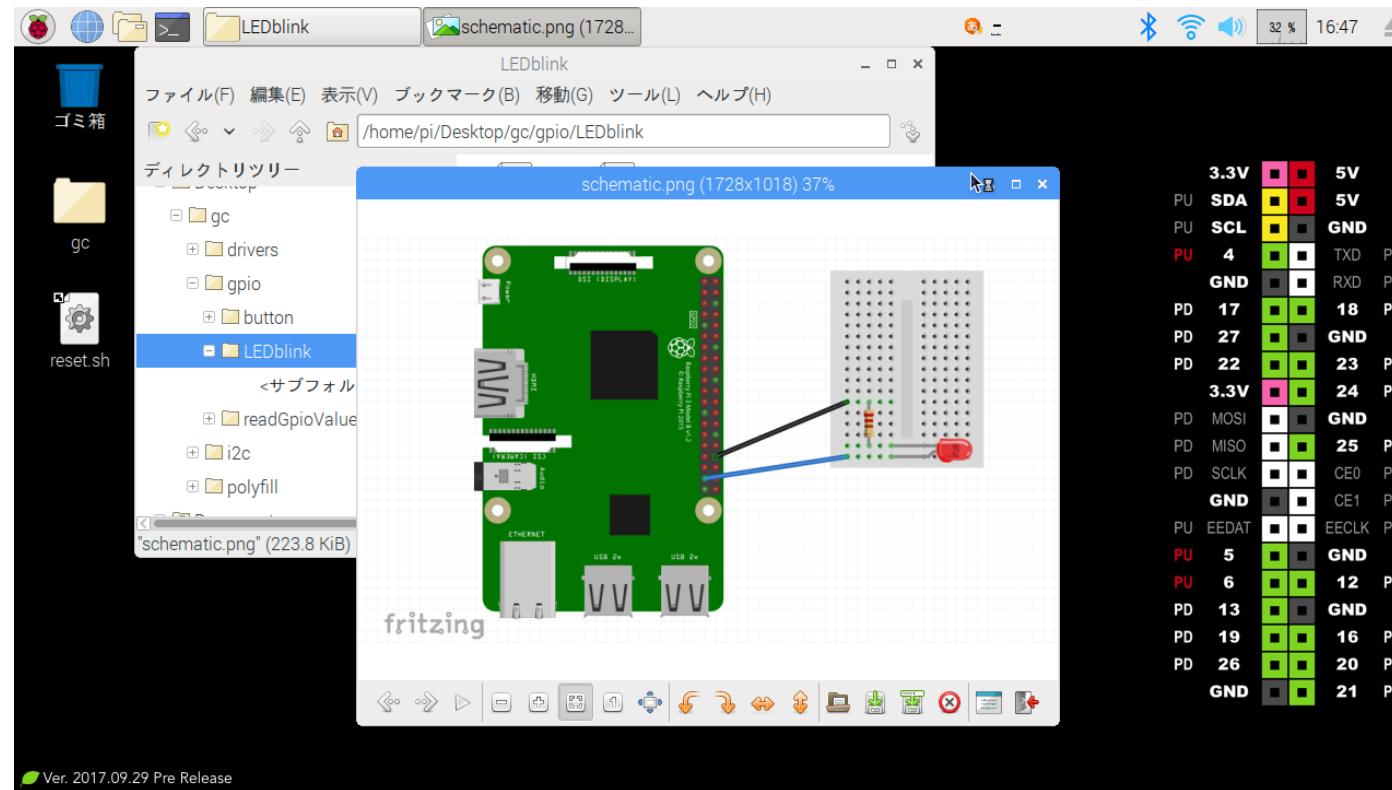




# Lチカしてみよう!!

| 18

## LEDチカチカしていますか？



|       |  |       |  |
|-------|--|-------|--|
| 3.3V  |  | 5V    |  |
| SDA   |  | 5V    |  |
| SCL   |  | GND   |  |
| 4     |  | TXD   |  |
| GND   |  | RXD   |  |
| 17    |  | 18    |  |
| 27    |  | GND   |  |
| 22    |  | 23    |  |
| 3.3V  |  | 24    |  |
| MOSI  |  | GND   |  |
| MISO  |  | 25    |  |
| SCLK  |  | CE0   |  |
| GND   |  | CE1   |  |
| EEDAT |  | EECLK |  |
| 5     |  | GND   |  |
| 6     |  | 12    |  |
| 13    |  | GND   |  |
| 19    |  | 16    |  |
| 26    |  | 20    |  |
| GND   |  | 21    |  |

GPIOは、「General-purpose input/output」の略で**汎用的な入出力インターフェース**のことです。

CHIRIMEN for Raspberry Pi 3ではRaspi3が提供する40本のピンヘッダのうち、左記緑色の17本のピンをWebアプリから利用可能なGPIOとして設定しています。

※今回は26番を使用していますが、他のポートも同じように使用することができます。

1. LチカWebプログラム(index.html)のタブを閉じます
2. ブラウザのブックマークから「Examples」を選択



3. 「GPIO-Blink」から「JSBin」を選択

A screenshot of a web page titled 'GPIO Examples (API Spec)'. The page features a blue header with the title and a tip: 'Tips: デスクトップのピン配置図を見ながら配線するとミスしにくくなります。'. Below the header, there are three examples: 'GPIO-Blink', 'GPIO-Button', and 'GPIO-readGpioValue'. Each example has a thumbnail image of a breadboard setup on the left and a detailed description on the right. The 'GPIO-Blink' section is highlighted with an orange arrow pointing to its 'JSBin' button. Each example also has '回路図' (Circuit Diagram), 'デモ' (Demo), 'JSBin' (highlighted in orange), and 'CSB' buttons. The 'GPIO-Blink' section includes a '概要' (Summary) section and a '使用パーツ' (Used Parts) list: 'LED x 1', '抵抗 (100Ω~470Ω) x 1' (with a note '(220Ωのカラーコード:赤赤茶金)'), and 'ジャンパー (オス・メス) ケーブル x 2'. The 'GPIO-Button' section includes a '概要' section and a '使用パーツ' list: 'タクトスイッチ x 1' and 'LED x 1'. The 'GPIO-readGpioValue' section includes a '概要' section and a '使用パーツ' list: 'タクトスイッチの状態 (押されているか、押されていないか) を読み込みコンソールへLog出力します。' and 'LED x 1'. The URL at the bottom is 'https://r.chirimen.org/jsbin-gpio-blink'.



## 3.JS Binに登録されたJavaScriptプログラムが表示されます

The screenshot shows a browser window for JS Bin. On the left, the HTML tab displays the following code:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta content="width=device-width, user-scalable=no, initial-scale=1.0, name='viewport'" name="viewport" />
    <title>LED Blink</title>
    <script src="https://r.chirimen.org/polyfill.js"></script>
    <style>
      p {
        color: blue;
        text-align: center;
        font-size: 24px;
      }
      img {
        display: block;
        margin-left: auto;
        margin-right: auto;
      }
    </style>
  </head>
  <body>
    <p>LED-GPIO-26</p>
  </body>
</html>
```

On the right, the JavaScript tab displays the following code:

```
// noprotect (JSBin の無限ループ検知・停止機能を無効化)
main();
async function main() {
  var gpioAccess = await navigator.requestGPIOAccess();
  var port = gpioAccess.ports.get(26); // 26 番ポートを操作
  var v = 0;
  await port.export("out"); // ポートを出力モードに設定
  for (;;) {
    v = v === 0 ? 1 : 0; // ポートの出力値を 0/1 交互に変更
    port.write(v); // LED を ON/OFF する
    await sleep(1000); // 繰り返し毎に 1000ms 待機
  }
}
```

Two orange callout boxes are present. The top one points to the "Console" and "Output" tabs at the top of the JS Bin interface, with the text: "ConsoleとOutputを非表示にすると画面が広く使用できます。". The bottom right one points to a translation dialog box with the text: "翻訳は不要なので閉じる".

At the bottom of the screenshot, the words "HTML" and "JavaScript" are displayed in large orange text.



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta
      content="width=device-width, user-scalable=no, initial-scale=1"
      name="viewport"
    />
    <title>LED Blink</title>
    <script src="https://r.chirimen.org/polyfill.js"></script>
    <style>
      ~省略~
    </style>
  </head>
  <body>
    <p>LED→GPIO-26</p>
  </body>
</html>
```

ハードウェアにアクセスするための機能  
Web GPIO, Web I2C  
機能を提供するライブラリ



タイマーが満了するたびにLEDがチカチカします。

```
main();  
  
async function main() {  
  var gpioAccess = await navigator.requestGPIOAccess(); // GPIO を操作する  
  var port = gpioAccess.ports.get(26); // 26 番ポートを操作する  
  var v = 0;  
  
  await port.export("out"); // ポートを出力モードに設定  
  for (;;) {  
    v = v === 0 ? 1 : 0; // ポートの出力値を 0/I 交互に変更  
    port.write(v); // LED を ON/OFF する  
    await sleep(1000); // 繰り返し毎に 1000ms 待機  
  }  
}
```

LEDがチカチカする周期  
を変更してみよう!!



## 本日のゴール

各種センサーやアクチュエーターの制御方法を理解すること

### 基礎編

0. Lチカしてしてみよう(初めてのGPIO) ※全体講習→完了
1. GPIOの使い方
2. センサーを使ってみよう(初めてのI2C)

### 応用編

3. I2Cの使い方
4. GPIO/I2Cのまとめ

Examples (時間があれば)



# 続きを読むは自分のペースで理解を深めながら進めてください。

V 192.168.11.4 (raspberrypi) - VNC Viewer  
CHIRIMEN for Raspb...  
CHIRIMEN for Raspberry Pi3 チュートリアル | CHIRIMEN チュートリアル - Chromium  
CHIRIMEN for Raspberry x +  
← → C https://tutorial.chirimen.org/raspi3/  
LED と温度センサーを使ってみる最小限のお試しページです。より詳しくは以下のチュートリアルをご覧ください。

## 基礎編

まずはシンプルな GPIO 入出力や I2C センサーの操作方法を学びましょう。

- 0. Lチカしてみよう (初めての GPIO)
  - WebGPIO API を使って定期的に LED を点滅するサンプルを動かしてみます。
- 1. GPIO の使い方
  - マウスクリックで操作するブラウザ画面のボタンと物理スイッチ (タクトスイッチ) の両方で LED やモーターを制御するサンプルを通じて GPIO の基本を学びます。
- 2. センサーを使ってみよう (初めての I2C)
  - 温度センサーの値をドライバーを使う場合と WebGPIO API を直接操作する場合の 2 パターンで読み取ることで、I2C デバイス操作の基本を学びます。

## 応用編

いろいろな I2C デバイスを繋いだり組み合わせたりすることで IoT プロトタイピングを体験しましょう。

- 3. I2C の使い方
  - いろいろな I2C デバイスを接続したり、複数の I2C デバイスの同時操作について学びます。



わからないときは迷わず  
周りのチユーター＆メンター陣に  
お声がけください！



2日目

## 1日目のゴール

各種センサーやアクチュエーターの制御方法を理解すること

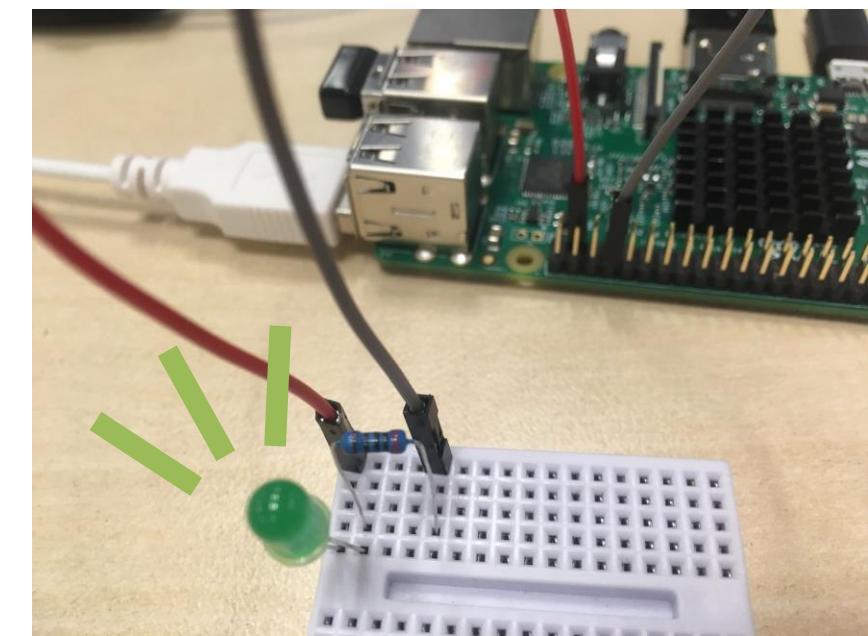
### 基礎編

0. Lチカしてしてみよう(初めてのGPIO) ※全体講習
1. GPIOの使い方
2. センサーを使ってみよう(初めてのI2C)

### 応用編

3. I2Cの使い方
4. GPIO/I2Cのまとめ

Examples (時間があれば)





本日のゴール

センサーの組み合わせを試してみること

応用編

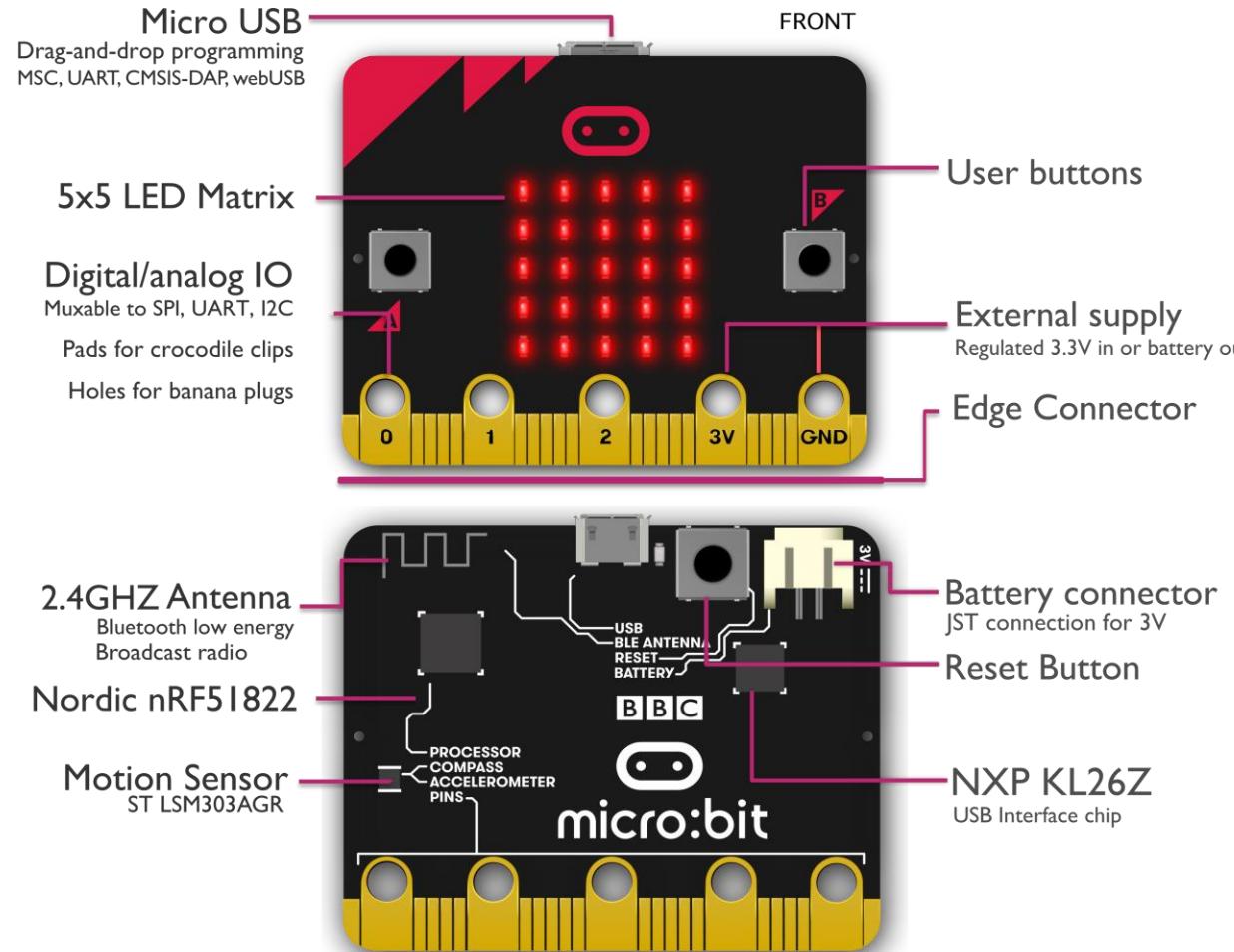
3. I2Cの使い方
4. GPIO/I2Cのまとめ

発展編

5. CHIRIMEN with micro:bit

Examples (時間と興味の限り)

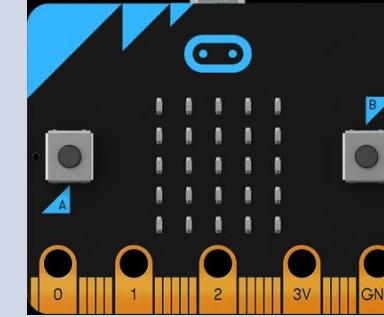
# イギリス生まれの小型コンピューター ボード



- 5x5 LED
- 温度センサー
- 明るさセンサー
- 地磁気センサー
- 加速度センサー
- Bluetooth LE
- 各種入出力端子 (GPIO/I2C)

✗ 無線LAN

# 無線通信の国際規格と技術適合証明（技適）

|  | Raspberry Pi 3B+  | micro:bit   |
|--|---|---|
| 外観   |   |  |
|   | IEEE802.11ac (Wi-Fi 5)<br>周波数帯: 5GHz, 2.4GHz  | ✗<br>-  |
|  | Bluetooth 4.2 LE<br>周波数帯: 2.4GHz  | Bluetooth 4.1 LE<br>周波数帯: 2.4GHz  |
| 技術適合<br>証明   |  | ○   |



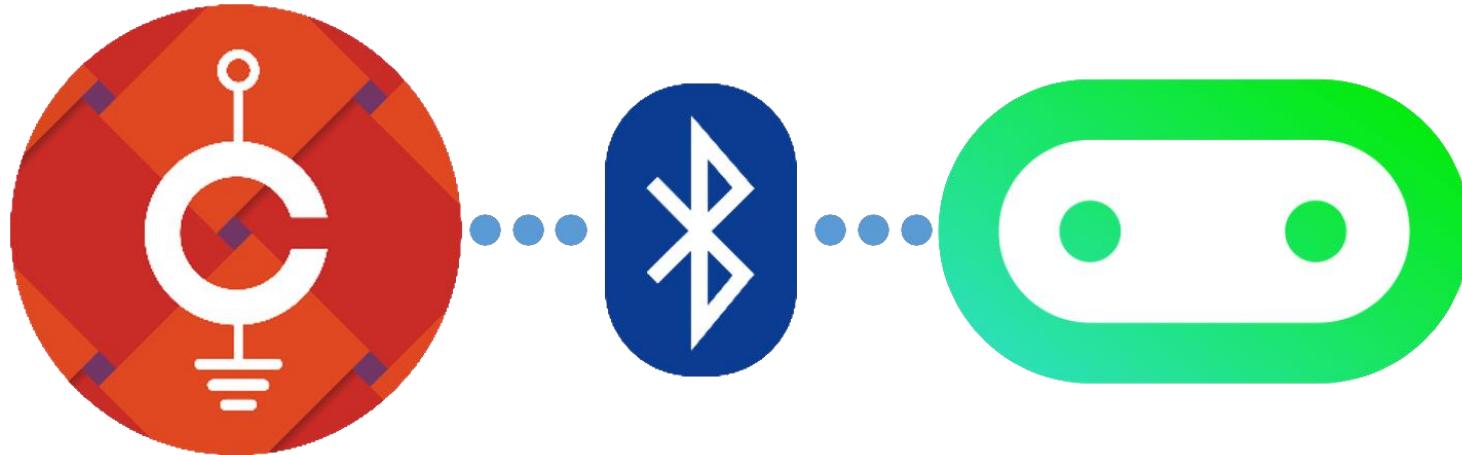
## ブロックとJavaScriptでプログラミング可能

This screenshot shows the Microsoft Scratch-like interface for micro:bit. It features a script editor with two scripts. The top script uses a 'ボタン A が押されたとき' (Button A pressed) event. Inside, it has a 'もし...なら' (If...Then) control block. The condition part is 'じゃんけん = 0 なら' (Rock paper scissors = 0). The main body contains two 'LED画面に表示' (Display on LED screen) blocks, each with a different pattern. The bottom script is a 'ループ' (Loop) with a single 'LED画面に表示' (Display on LED screen) block showing a pattern. The interface includes a sidebar with categories like 基本 (Basic), 入力 (Input), 音楽 (Music), LED, 無線 (Wireless), ループ (Loop), 論理 (Logic), 変数 (Variables), and 計算 (Math). A preview window at the bottom shows the micro:bit board with the current script logic.

This screenshot shows the Microsoft Scratch-like interface for micro:bit with the generated JavaScript code. The code implements a rock-paper-scissors game where the computer randomly chooses between 0 (Rock) and 1 (Paper). It then displays the result on the LED screen based on the button press. The code is as follows:

```
1 letじゃんけん = 0
2 input.onButtonPressed(Button.A, function () {
 じゃんけん = Math.randomRange(0, 2)
if (じゃんけん == 0) {
  basic.showLeds(`
    . . .
    . . #
    # # #
  `)
} else if (じゃんけん == 1) {
  basic.showLeds(`
    . # .
    . # .
    # # #
  `)
} else {
  basic.showLeds(`
    # . #
    # . #
    # . #
    # # #
  `)
}
30 basic.forever(function () {
```

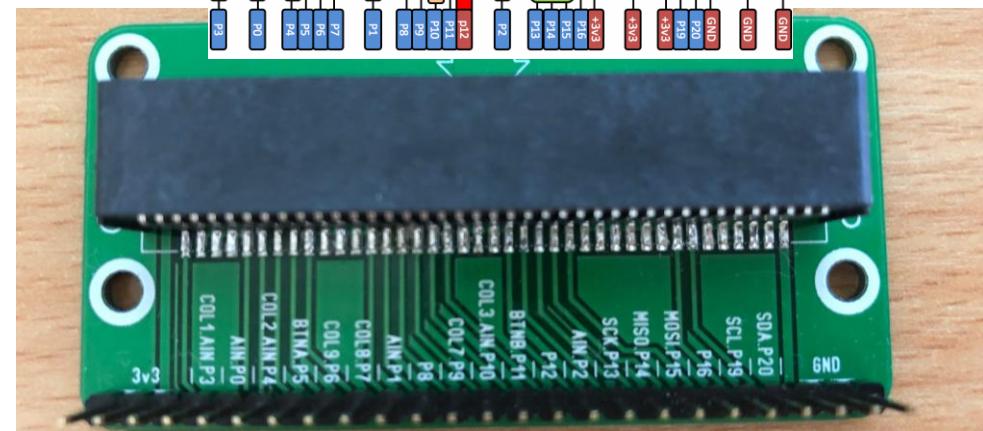
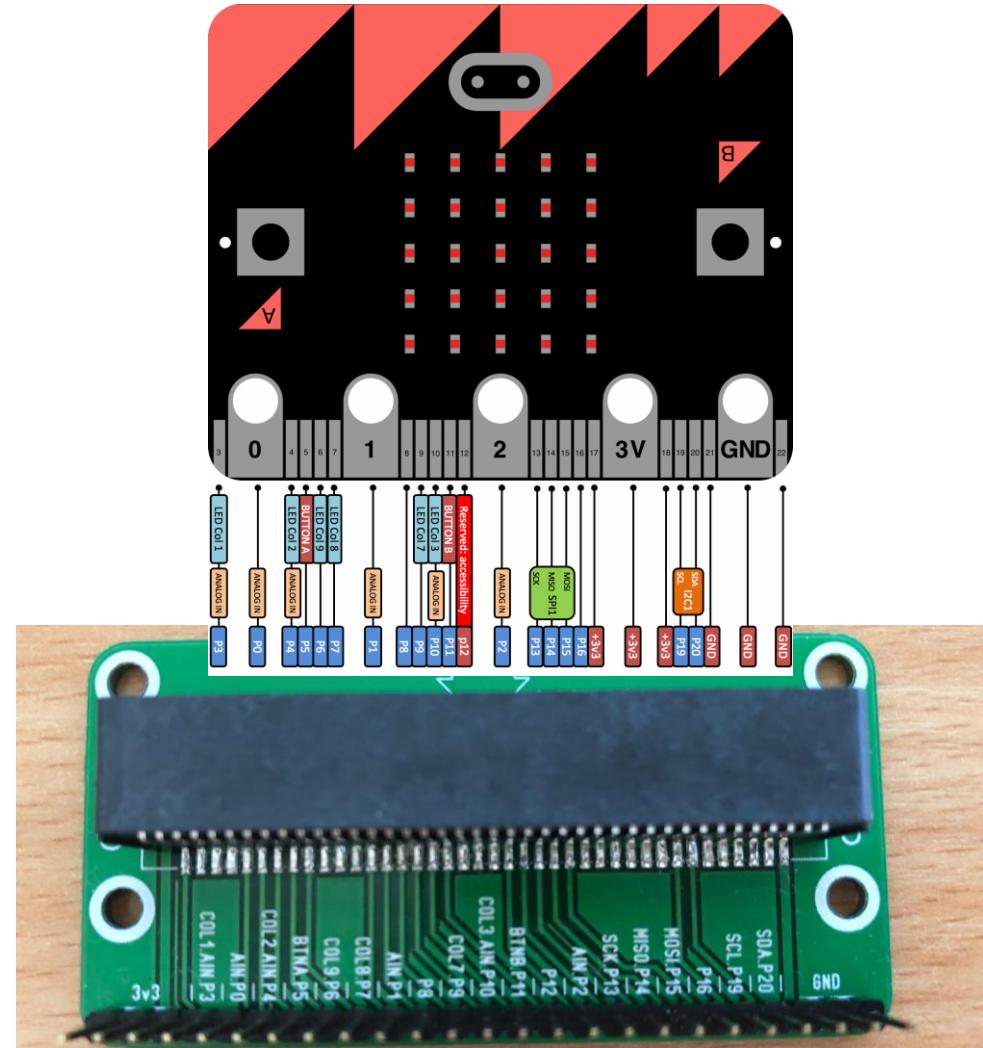
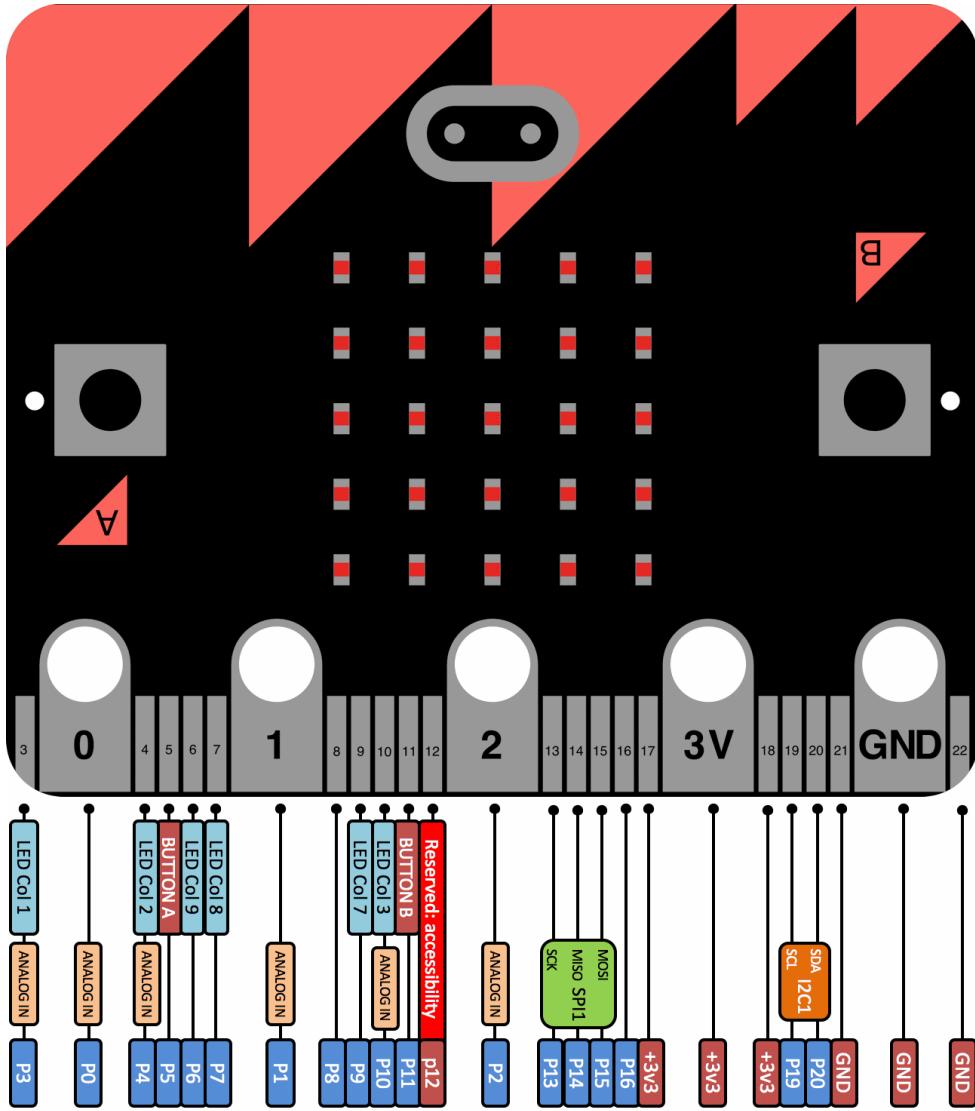
## CHIRIMEN の micro:bit 版



WebGPIO/WebI2C APIを介して  
(昨日学んだ) CHIRIMEN for Raspberry Pi 3  
とほぼ共通のコードでGPIO/I2Cデバイスを制御可能



## 入出力端子を使いやすくするためのボード





# ブラウザ自動起動後、「ちゅーとりある！」を選択





# 「micro:bit」を選択

CHIRIMEN チュートリアル

ブラウザからハードを制御する CHIRIMEN のチュートリアル集です  
対応ボード別チュートリアル: Raspberry Pi / TY51822r3 / micro:bit

Language: Japanese, English (Google Translation)

## CHIRIMEN とは

CHIRIMEN とは、Web ブラウザからハードウェア制御も可能にしたプロトタイピング環境です。ブラウザの JavaScript からセンサーや電子パーツを制御できるため、デジタルのソフトとフィジカルなハードを同じプログラムで容易に連携できます。例えば Lチカコードはこの通りです:

```
window.onload = async function() {
  var gpioAccess = await navigator.requestGPIOAccess(); // GPIO を操作する
  var port = gpioAccess.ports.get(26); // 26 番ポートを操作する
  var v = 0;

  await port.export("out"); // ポートを出力モードに設定
```

<https://chirimen.org/chirimen-micro-bit/>

# スタートアップガイド→Examplesの順に進めてください。

## CHIRIMEN with micro:bit

### 概要

micro:bitを利用したCHIRIMENの実装を公開しています。CHIRIMENとはボードコンピュータの端子にないセンサやアクチュエータを使用した(ブラウザ上で動作する)ウェブアプリを動作させる環境です。WebGPIOとWebI2C APIが使えるようになります。micro:bitではウェブブラウザは(多分:-)動きませんが、Bluetoothを介して、PC等で動くウェブブラウザ(ChromeなどwebBluetoothをサポートしたもの)とmicro:bitを接続することで、CHIRIMEN環境を構築します。また、CHIRIMEN with micro:bitでは、WebGPIO,WebI2Cに加え、micro:bitが内蔵しているセンサーやLEDをウェブアプリで使うことができる機能が追加されています。



### スタートアップガイド

- [こちらです](#)



### Examples

以下のページに使用例を掲載します。

- [Examples](#)

### その他情報

- [拡張機能](#)
- [CHIRIMEN for Raspberry Pi3との差分](#)
- [システム構成の説明](#)

### 注記

この実装は、webGPIOだけを実装した[webGPIO-etc-on-microbit-via-webBluetooth](#)を前身として持ちます。しかしmicro:bitのリソース上の制限から、この実装はそれとまったく異なるものになっています。



# ブラウザの設定を変更 (WebBluetoothを有効)

The screenshot shows the 'chrome://flags' page in a browser window. A large orange arrow points to the 'Experimental Web Platform features' section, which is currently set to 'Enabled'. Other sections visible include 'V8 Orinoco garbage collection features', 'Await optimization', 'GPU rasterization', 'Out of process rasterization', 'Developer Tools experiments', 'Silent Debugging', and 'Touch UI Layout'.

chrome://flags - Chromium

chrome://flags - Chro... Downloads MICROBIT

スタートアップガイド | chrome://flags

Search flags

Reset all to default

V8 Orinoco garbage collection features  
This enables the V8 Orinoco garbage collection features. – Mac, Windows, Linux, Chrome OS, Android  
[#enable-v8-orinoco](#)

Await optimization  
Enables await taking 1 tick on the microtask queue. – Mac, Windows, Linux, Chrome OS, Android  
[#harmony-await-optimization](#)

GPU rasterization  
Use GPU to rasterize web content. Requires impl-side painting. – Mac, Windows, Linux, Chrome OS, Android  
[#enable-gpu-rasterization](#)

Out of process rasterization  
Perform Ganesh raster in the GPU Process instead of the renderer. Must also enable GPU rasterization – Mac, Windows, Linux, Chrome OS, Android  
[#enable-oop-rasterization](#)

Experimental Web Platform features  
Enables experimental Web Platform features that are in development. – Mac, Windows, Linux, Chrome OS, Android  
[#enable-experimental-web-platform-features](#)

Developer Tools experiments  
Enables Developer Tools experiments. Use Settings panel in Developer Tools to toggle individual experiments. – Mac, Windows, Linux, Chrome OS  
[#enable-devtools-experiments](#)

Silent Debugging  
Do not show the infobar when an extension attaches to a page via chrome.debugger API. This is required to debug extension background pages. – Mac, Windows, Linux, Chrome OS  
[#silent-debugger-extension-api](#)

Touch UI Layout  
Enables touch UI layout in the browser's top chrome. – Mac, Windows, Linux, Chrome OS

microbit-We...hex



# CHIRIMEN with micro:bitの導入

The screenshot shows a Chromium browser window with the following details:

- Address Bar:** https://chirimen.org/chirimen-micro-bit/guidebooks/startup.html
- Tab Bar:** タブは「スタートアップガイド」(複数開いてる)と「Microsoft MakeCode for」。
- Content Area:**
  - 解説文: Examplesや以降の解説の実体配線図では、ブレークアウトボードは省略されています。GPIO番号やSCL,SDA,GND,3V(3.3V,VDD)などの端子の名称をもとにつないでください。
  - セクション: **mircro:bit に CHIRIMEN サポートプログラムを書き込む**
  - 説明文: CHIRIMEN with micro:bit では WebGPIO/WebI2C API を後述の Polyfill ライブラリと合わせて利用することでアプリケーションを開発します。各 API を呼び出すとライブラリ内では Web Bluetooth API を用いて micro:bit と通信しますが、micro:bit 側には PC からの制御命令を受け取るため専用の **CHIRIMEN サポートプログラム** を書き込んでおく必要があります。
  - 手順リスト:
    - 「こちらのページ」にアクセスして、micro:bitにサポートプログラムを書き込みます。  
書き込み方は通常のmicro:bit用プログラムと全く同じです。
      - [編集]ボタンを押す
      - [ダウンロード]ボタンを押し、サポートプログラムをダウンロード
      - micro:bitをUSBでPCに接続(USBドライブとしてPC上にマウント)
      - ダウンロードしたプログラムをマウントしたmicro:bit USB ドライブにコピー
      - 自動的に再起動しインストール完了。がmicro:bitに表示されます
      - これで準備完了です！USBケーブルを外して次に進みましょう。(micro:bitのプログラムは電源を落としても消えません)
  - 備考文: なお、このCHIRIMEN用のサポートプログラムは、micro:bit上で動かす他のプログラムと同じ動作環境(makecode)で構築してありますので、CHIRIMEN with micro:bitを使わないときは、特別な手順は必要なく自由に他のプログラムと置き換えることができます。
  - セクション: **ブラウザの設定を確認**
  - 説明文: コンピュータと micro:bit との無線通信に利用する Web Bluetooth API はまだ実験的な実装という扱いであり、ブラウザや環境によっては有効化されていません。Chromeなどのブラウザで chrome://flags ページを開き Experimental Web Platform features が有効になっていることを確認、無効の場合は有効化してからブラウザを再起動(環境によっては設定変更後に一度 Bluetooth のオンオフも)してください。開発者ツールのコンソールで navigator.bluetooth オブジェクトが見えるようになっていれば有効化されています。
- Bottom Status Bar:** https://chirimen.org/chirimen-micro-bit/guidebooks/link2original.html#https://makecode.microbit.org/\_0jhPcA3iX0gC
- Bottom Right Buttons:** メニュー、最大化、最小化、閉じる、すべて表示



## プログラムをダウンロード

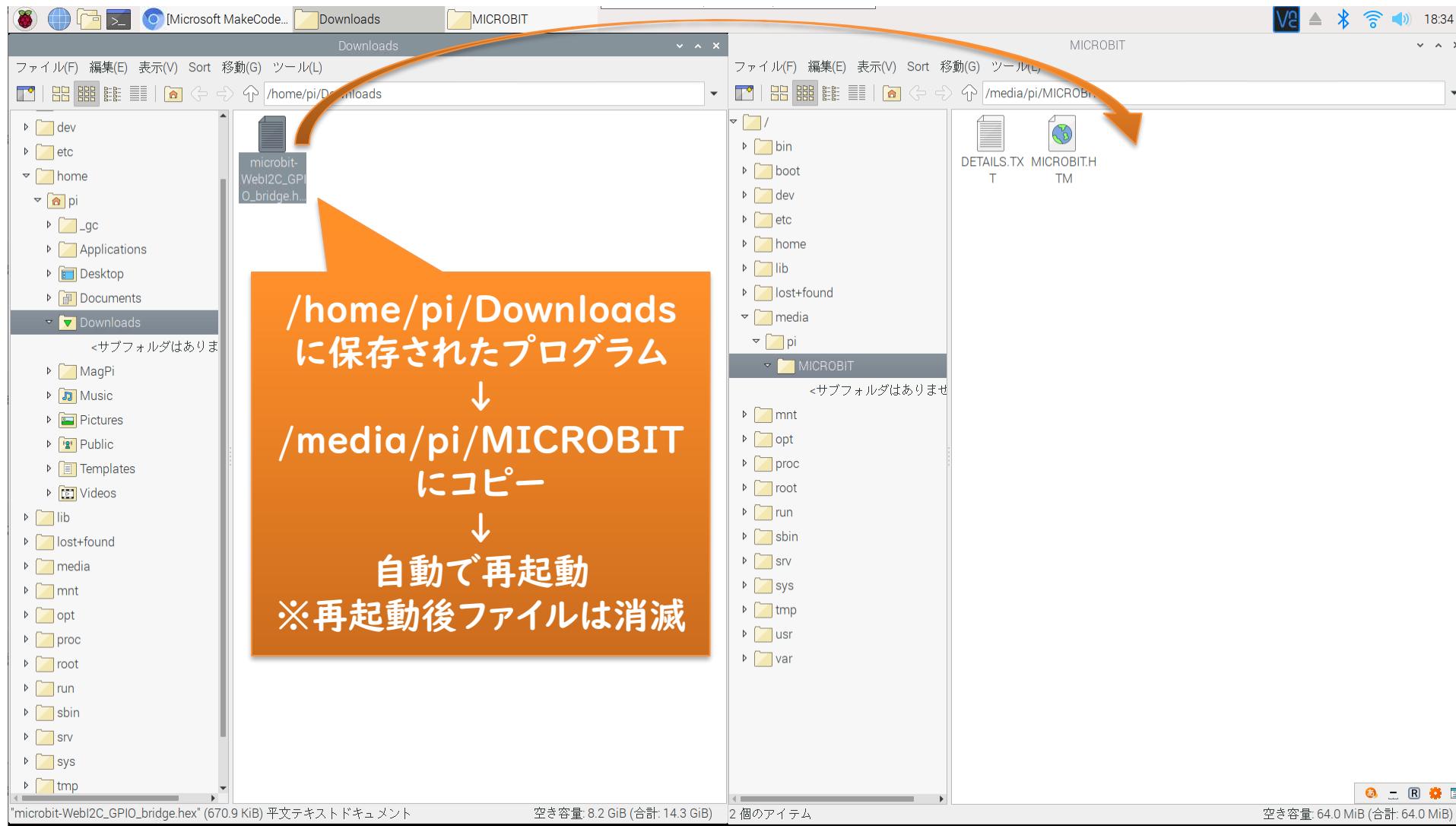
The screenshot shows the Microsoft MakeCode for micro:bit editor interface. On the left, there's a preview of the micro:bit board with various pins labeled (A, B, 0, 1, 2, 3V, GND). The main workspace displays the following JavaScript code:

```
1 input.onButtonPressed(Button.AB, function () {
2     input.calibrateCompass()
3 }
4 input.onButtonPressed(Button.A, function () {
5     button = 1
6 })
7 bluetooth.onBluetoothDisconnected(function () {
8     basic.showIcon(IconNames.No)
9     basic.pause(1000)
10 })
11 input.onButtonPressed(Button.B, function () {
12     button = 2
13 })
14 bluetooth.onBluetoothConnected(function () {
15     basic.showIcon(IconNames.Diamond)
16     basic.pause(2000)
17     basic.showLeds(`
18     . . .
19     . . .
20     . . .
21     . . .
22     . . .
23 `)
24 })
25 let ledMsg = ""
26 let bleMsg = ""
27 let button = 0
28 let bleRes = ""
29 let rp = 0
```

On the right side of the editor, there are several tabs: 基本 (Basic), 入力 (Input), 音楽 (Music), LED, Custom, Bluetooth, ループ (Loop), 論理 (Logic), 変数 (Variables), and 計算 (Math). Below the code, there's a green "ダウンロード" (Download) button with a downward arrow icon. At the bottom of the editor, there are tabs for "microbit-We...hex" and "WebI2C\_GPIO\_bridge". A status bar at the bottom right shows "すべて表示" (Show All).



# micro:bitにインストール(コピー)



## ゴール

各種センサーやアクチュエーターの制御方法を理解すること

## 基礎編

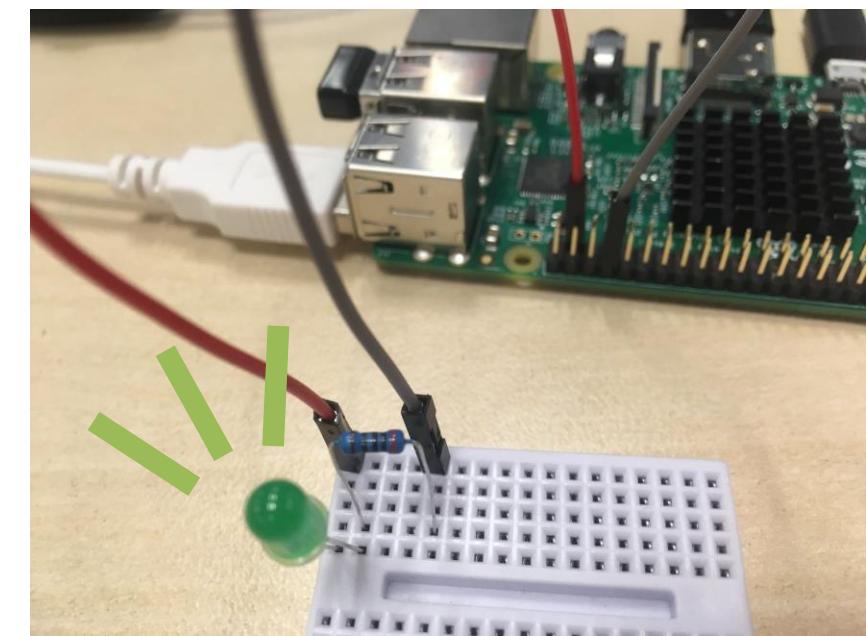
0. Lチカしてしてみよう(初めてのGPIO) ※全体講習
1. GPIOの使い方
2. センサーを使ってみよう(初めてのI2C)

## 応用編

3. I2Cの使い方
4. GPIO/I2Cのまとめ

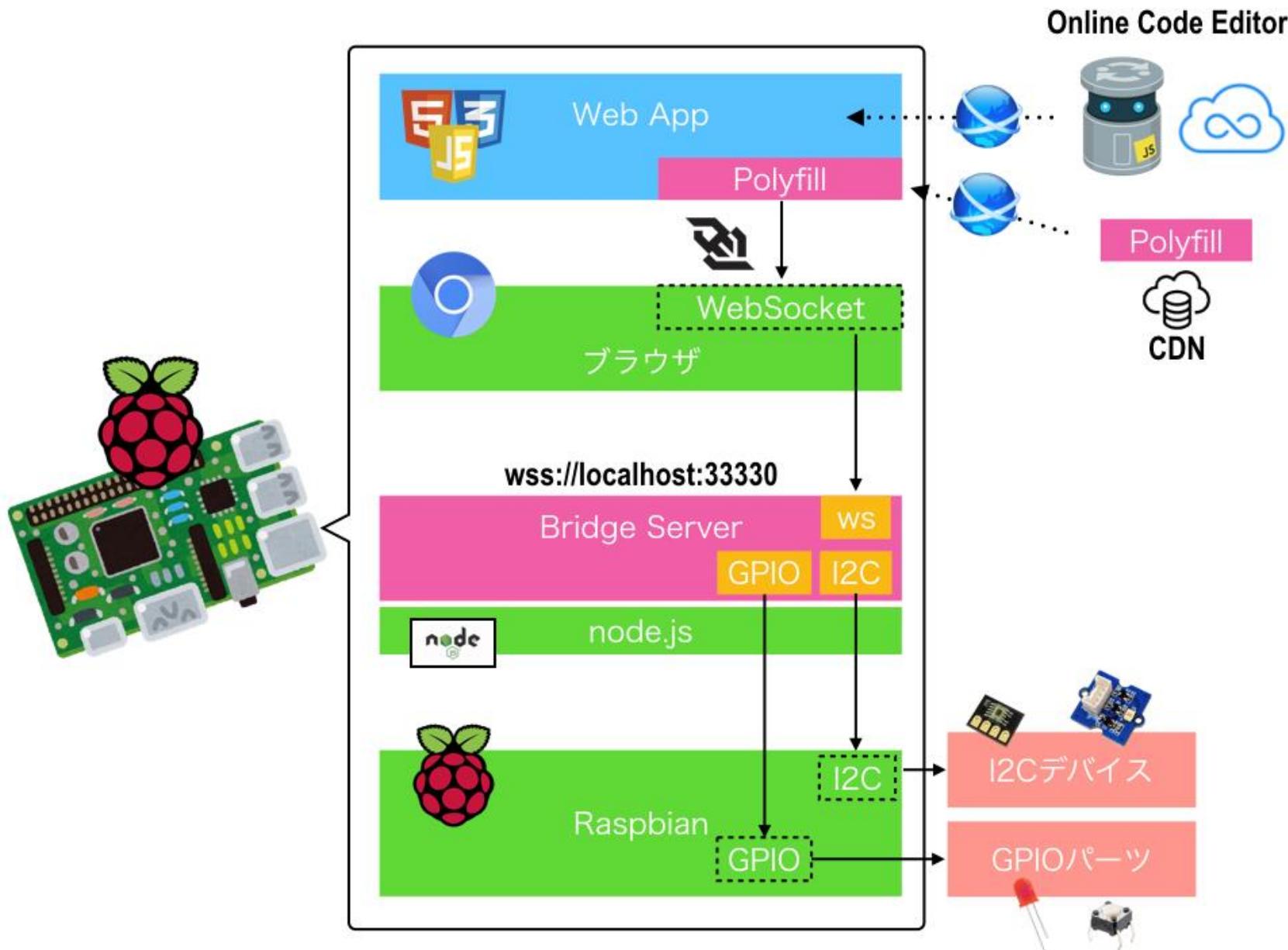
## 発展編

5. CHIRIMEN with micro:bit





# 參考資料



# この資料のライセンスについて

この資料は、総務省「IoT機器等の電波利用システムの適正利用のためのICT人材育成事業」の一環として実施された学生や若手エンジニアを対象としたIoTシステム開発のスキルアップイベント「Web×IoT メイカーズチャレンジ 2019-20 in 群馬」のハンズオン講習会用資料として講師の宮崎典行氏が作成されたスライドです。



クリエイティブ・コモンズ・ライセンス 表示 4.0 国際 (CC-BY) によって許諾されていますので、この資料を基に改変や再配布いただくことも可能です。

(CC-BY ライセンスの内容を知りたい方は <http://creativecommons.org/licenses/by/4.0/deed.ja> でご確認ください。)

なお、この資料のブラッシュアップやアップデート等をされた場合は、以下の Githubリポジトリに共有いただくことを推奨しています。

Web×IoT メイカーズチャレンジ教材共有リポジトリ  
<https://github.com/webiotmakers/material>