

CHIRIMEN

Getting started

2016.9.5

JS Board勉強会 主宰 村岡 正和

むらおか まさかず

村岡正和



@bathtimefish

Webアプリケーション開発 IT業務システム設計/開発

Webサービス導入/事業戦略コンサルティング

神戸デジタル・ラボ 社外取締役

HTML5-WEST.jp代表 / html5j マークアップ部 部長 / HTML5 Experts.jp メンバー

NPO法人日本ウェアラブルデバイスユーザー会理事

神戸市ウェアラブルデバイス推進会議メンバーなど

HTML5-WEST.jp



HTML5 Experts.jp

日本に、もっとエキスパートを。

白石俊平

CSSNITE
in OSAKA

GTUG
京都

Kobe IT
festival

Kobe
Digital
Labo

Firefox OS 開発ガイドブック

http://amzn.to/292MeXI

amazon.co.jp
プライム

Kindleストア



BUFFALO P

Amazonポイント: 279

マイストア ギフト券 タイムセール Amazonで売る ヘルプ In English

村岡正和さん
アカウントサービス

カテゴリー

Kindleストア Kindle を購入 アクセサリー Kindle本 コミック 女性向けコミック 雑誌 日替わりセール まとめ買い Kindle Singles Amazonランキング Kindle無料アプリ

本 > コンピュータ・IT > OS



Firefox OS 【決定版】 開発ガイドブック Kindle版

村岡正和 (著), 藤下正美 (著), 山本祐輔 (著)

カスタマーレビューを書きませんか?

すべての2フォーマットおよびエディションを表示する

Kindle版
¥ 3,200

単行本 (ソフトカバー)
¥ 3,456 **プライム**

今すぐお読みいただけます: **無料アプリ**

¥ 2,700 より 3 中古品の出品

¥ 3,456 より 1 新品

(概要)

本書は、Mozilla Foundationが開発する「Firefox OS」の、日本初となる本格的な開発ガイドブックです。iOSやAndroidとの比較を交えながらFirefox OSの魅力を解説する第1部、Firefox OSの内部構造をはじめアプリ開発に必要な諸知識を解説する第2部、オープンソースであることを最大限利用したOSのカスタマイズ手法について解説する第3部に分けて、Firefox OSの魅力や構成技術を体系的に解説します。

続きを読む

紙の本の長さ: 384ページ

利用可能な端末

この本はファイルサイズが大きいため、ダウンロードに時間がかかる場合があります。Kindle端末では、この本を3G接続でダウンロードすることができませんので、Wi-Fiネットワークをご利用ください。

【無料配信】



東京の多様な地域特性、都市構造、都民のライフスタイルなどを考慮してつくられた、完全東京仕様の防災ブック。



どこでも
読書を始
めよう。

> Kindle無料アプリをダウンロード

HTML



WEB of Things



これからのHTML5はモノにアプローチすべき
だと突然思った。(2013年ごろ)

勉強会をつくって興味ある人たちと情報交換

ATND

PRODUCED BY RECRUIT

イベント検索



ログイン or 無料登録

日本語

English

注目のキーワード [unreal engine](#) [アプリ](#) [swift](#) [AE](#) [データ](#) [HTML5](#) [デザイン](#) [UX](#) [python](#)

JS board 勉強会 東京 #1

組込み系JavaScriptの勉強会

Tweet

Like 264

G+ 0

B! 2



日時: 2014/11/16 (日) 14:00 ~ 17:30

[Googleカレンダーに追加](#)

定員: 40人

会場: [ニフティ株式会社 \(東京都新宿区北新宿2-21-1 新宿フロントタワー \(18階受付\)\)](#)

URL: <http://www.nifty.co.jp/company/outline/access.htm>

主催者: [bathtimefish](#) [+フォローする](#)

ハッシュタグ: [#html5j](#)



Google Mapsで表示

このイベントは終了しました

主催者をフォロー
新規イベントのお知らせが受け
取れます



組込み系JavaScriptの勉強会を開催します。

たぶんTesselとかEspruinoとかみれると思います。

開催概要は以下のとおりです。

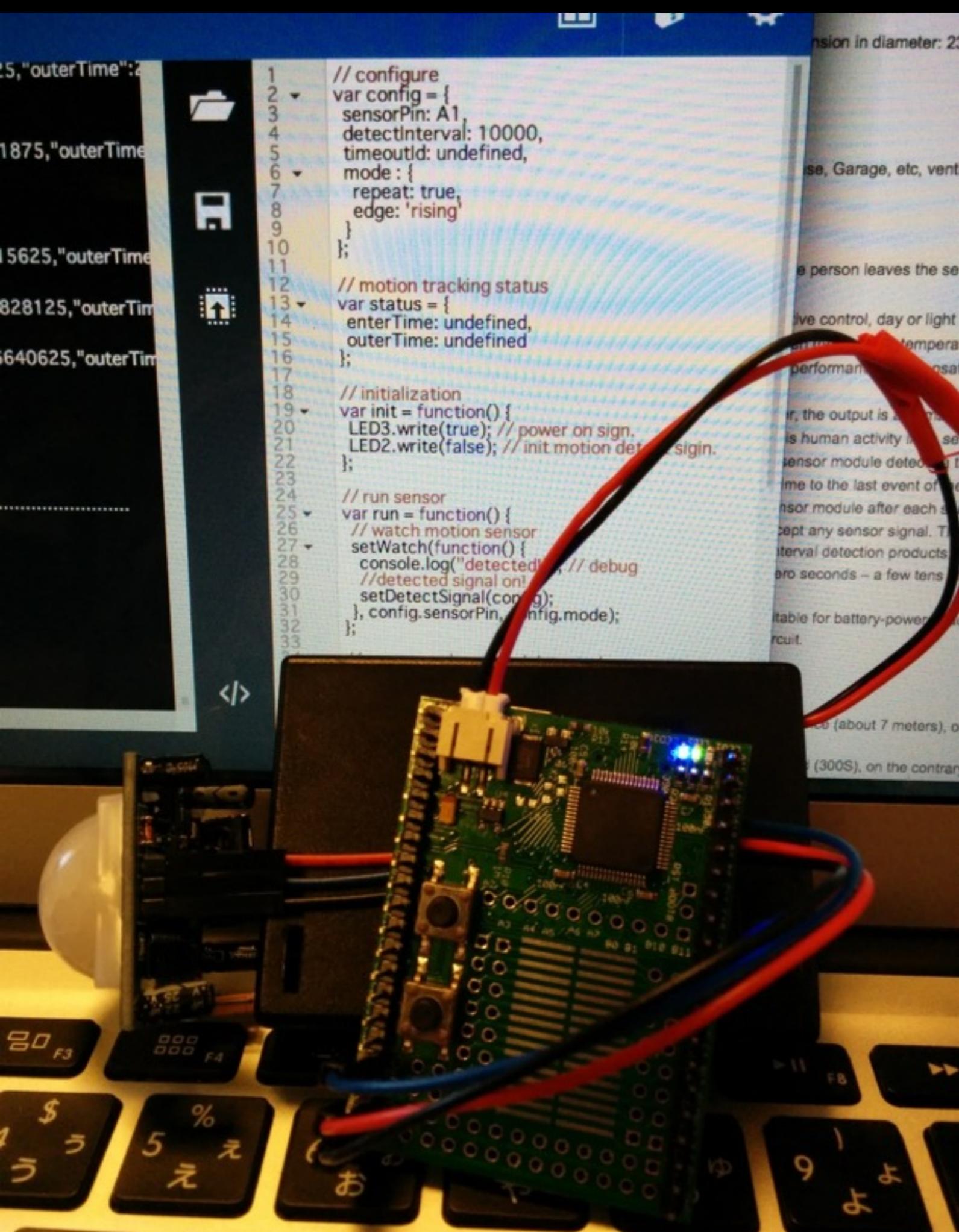
2014年11月16日 (日) 14:00~17:30 (13:40 開場)

場所: [東京都新宿区北新宿2-21-1 新宿フロントタワー \(18階受付\)](#)

[アクセス](#)

参加希望者 **30** / 40人

参加: **30** / キャンセル: **18**



```
1 // configure
2 var config = {
3   sensorPin: A1
4   detectInterval: 10000,
5   timeoutId: undefined,
6   mode: {
7     repeat: true,
8     edge: 'rising'
9   }
10 };
11
12 // motion tracking status
13 var status = {
14   enterTime: undefined,
15   outerTime: undefined
16 };
17
18 // initialization
19 var init = function() {
20   LED3.write(true); // power on sign.
21   LED2.write(false); // init motion detect sign.
22 };
23
24 // run sensor
25 var run = function() {
26   // watch motion sensor
27   setWatch(function() {
28     console.log("detected"); // debug
29     //detected signal on!
30     setDetectSignal(config);
31   }, config.sensorPin, config.mode);
32 };
33
```

JavaScriptでマイコンを制御しはじめたら、

日経エレクトロニクス さんに載ったり、

日経エレクトロニクス
2014.6.23 号

Web開発者はなぜハードウェア開発にひかれるのか

“画面の外”に作用できる

村岡 正和氏

バスタイムフィッシュ チーフアーキテクト

10 数年にわたってWeb関連のソフトウェア開発を手掛けてきた。ハードウェア開発にはまったくなじみがなかった。そんな私が、Kickstarterのプロジェクトに出ていた「Espruino」を見つけて、迷うことなく購入した。プログラムが“画面の外”に影響を及ぼすことに強い興味を持っていたからだ。

JavaScriptであれば、手足のように使いこなせるという自信がある。JavaScriptでハードウェアを制御できるEspruinoは、自分の知識と興味が合致する開発基盤だった。

Web開発者は、画面の中での人の行動履歴を蓄積し、それをお金に変えることに知恵を絞ってきた。画面の外の情報を集めたり、画面の外に働きかけたりできるようになれば、Webサービスの可能性は大きく広がる。



我々はマネタイズの方法を考える部分には長けている一方で、ユーザーにとって魅力あるハードウェアを開発したり、ハードウェアの信頼性を量産レベルまで高めたりする部分にはほとんど貢献できない。

だから、ハードウェアの専門家たちと組みたい。一緒に「Web of Things (モノのWeb)」の世界を作っていきたいと思っている。うまく協力できれば、機器とサービスを組み合わせた、これまでにないビジネスモデルも生み出せるだろう。(談)

Web時代

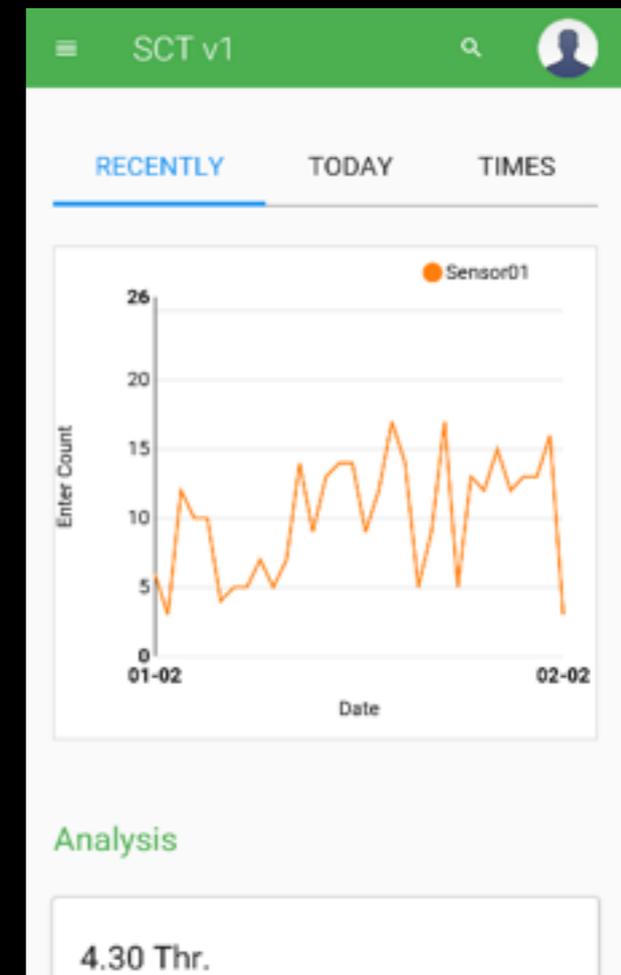
ユーザーの
画面内での行動の履歴





いろいろ応用して猫の
トイレを監視するセ
ンサーを開発したら、

↓専用スマホアプリ





村岡正和氏
バスタイムフィッシュ
チーフアーキテクト

では実際にハードウェアにあまり触れたことのないエンジニアが、このVirtualBenchの直感性をどのように評価するのだろうか。これを検証するために、1975年生まれのソフトウェア・プログラマで、バスタイム フィッシュチーフアーキテクトの村岡正和氏に、VirtualBenchに触れてもらった。

同氏は、Webシステムの設計開発やIT関連の技術コンサルティングなどを手掛ける一方で、関西のHTML5コミュニティ HTML-5WEST.jpの代表も務め、HTML5を中心としたWeb技術の普及にも取り組んでいる。さらにハッカソンなどのIT関連のイベントを主催したり、様々なIT関連のコミュニティ活動にかかわったりするなど、関西を中心に多

彩な活動を展開している。プログラマである同氏は、ソフトウェアの技術に精通しており、パソコンやタブレット端末などのデジタル機器を自由に使いこなすことができる。ただし、電気回路を扱った経験は、ほとんどなかった。

ところが、最近になって小型コンピュータを使ったパーソナルアプリケーションに取り組むようになった。

「Arduinoなど安価な小型コンピュータや、それに組み合わせて使える様々なセンサ・モジュールや通信モジュール、さらに使いやすい開発ツールが次々と登場したことで、電気回路の知識がないソフトウェア・エンジニアでも様々なハードウェアを開発できるようになったからです」（村岡氏）。す

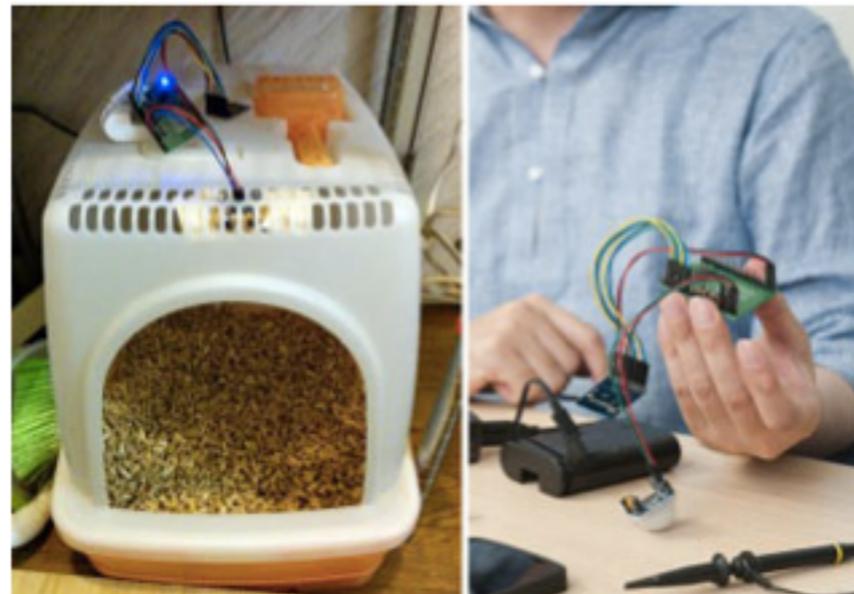


図2 監視システムを取り付けた猫用トイレと村岡氏が自作したハードウェア

日経ソフトウェアさんに取材されたり、

日経テクノロジーオンラインSPECIAL

http://special.nikkeibp.co.jp/atcl/TS/T/TEC_t_TMI03873/01/

Maker Faire 2015に出展したら、



CodeZineで記事書かせてもらったりした。

特集記事

JS BoardではじめてみようJavaScript組み込みプログラミング

JavaScript IoT

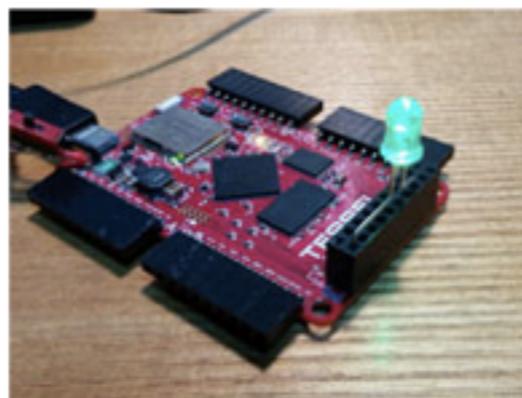
🖨️ 印刷用を表示

🐦 ツイート 📄 シェア 106 🍷 G+ 0 📄 B! 54

村岡 正和[著]

2015/11/25 14:00

JS Boardとは、HTML5、JavaScript等のWeb技術を中心に据えているマイコン、デバイスのことを指します。もともと筆者がそういう趣の勉強会を開きたくてつくった造語です。この記事では主にJavaScriptを利用した組み込み開発に興味がある方、電子工作をやってみたいWebクリエイターなどを対象に、JavaScriptで動作する組み込みデバイスを紹介したいと思います



[会員登録\(無料\)](#) [ログインはこちら](#)

GrapeCity.
フィールドをどこに配置しても
1レコードはしっかりキープ。
すごいデータグリッド

フィールド自由配置型データグリッド
MultiRow [マルチロウ]

Special Contents

PR

今年も出ました。Maker Faire 2016

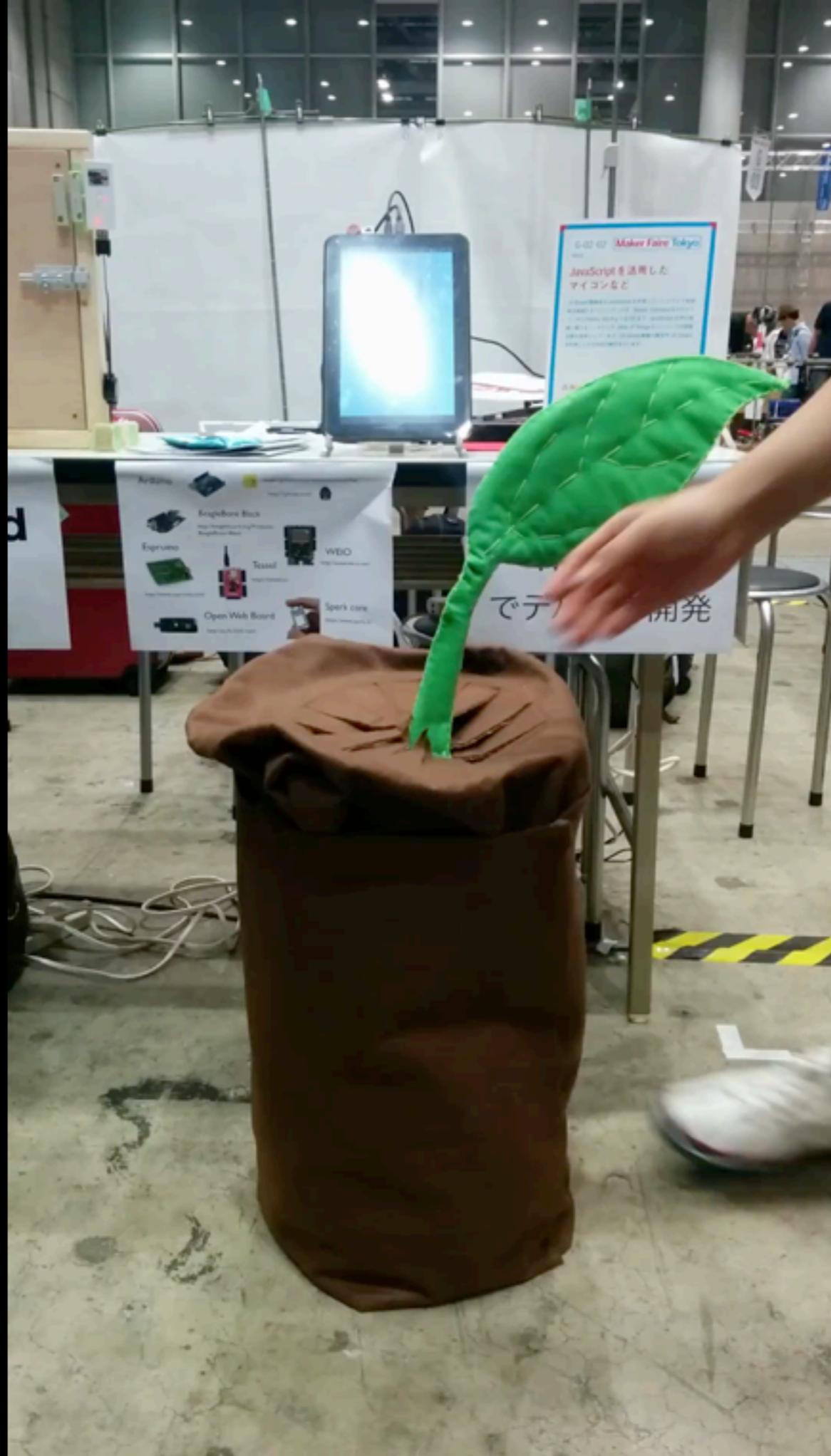


IoTはJS一択

すべてのデバイスは
JSで動いたら
いいと思う。

**JS Board
勉強会**

JavaScript
Web技術
でデバイス開発



6-02-02 Make Fair Tokyo
JavaScriptを使用した
アイコンなど

Arduino
Raspberry Pi
Eggs
Tessel
WEDO
Open Web Board
Spark core

でアプリ開発

思いつきで

トイレ監視のためのIoTドアセンサーをつくったら、



日経新聞に載って問い合わせに追われる毎日なう

日本経済新聞

2016年6月16日 (木)

Web刊 **速報** ビジネスリーダー マーケット テクノロジー アジア スポーツ マネー・ライフ 朝刊・夕刊 Myニュース New 人事ウオッチ

全て 経済 企業 国際 政治 株・金融 スポーツ 社会 ニュース18時 地域ニュース▼ 記事 株価

速報 > 地域ニュース > 近畿 > 記事

空きトイレ、スマホで確認 神戸デジタル・ラボ

2016/6/16 6:02 [有料会員限定]

小 中 大 保存 印刷 リプリント 共有

システム開発を手掛ける神戸デジタル・ラボ(神戸市)は、スマートフォン(スマホ)やパソコンから個室トイレの空き状況を確認できるシステムを開発した。あらゆるモノがインターネットにつながる「インターネット・オブ・シングス(IoT)」の技術を活用。オフィスビルだけでなく公共施設のトイレへの導入を見込む。

6月から同社が入居するオフィスビルの1フロアで実証実験を始めた。今秋のサービス開始を目指す。

個室トイレのドアに開閉センサーを設置し、そのデータをクラウド上に構築したシステムへ送信。アプリケーションを介してスマホやパソコンから、各トイレの利用状況をリアルタイムで確認する仕組みだ。システムを利用するための技術仕様「API(アプリケーション・プログラミング・インターフェース)」を提供し、API利用料を核とする事業モデルを想定する。

オフィスビルでは混雑緩和につながり、公共施設では管理者側がデータを監視することで防犯や急病人の早期発見に役立てられる。東京五輪が開催される2020年に向け、トイレの場所と空室状況が分かる仕組みを整えてインバウンド(訪日外国人)を含む観光客にも使いやすいサービスとして打ち出す。

小 中 大 保存 印刷 リプリント 共有

主な市場指標		世界の市況一覧 >	
日経平均(円) >> 6/16 10:04	15,785.72	-133.86	-0.84%
NYダウ(ドル) >> 6/15 終値	17,640.17	-34.65	-0.19%
ドル(円) >> 6/16 9:44	105.58-59	-0.66円高	-0.62%
ユーロ(円) >> 6/16 9:44	119.02-06	-0.17円高	-0.14%
長期金利(%) >> 6/16 9:20	-0.210	-0.015	
NY原油(ドル) >> 6/15 終値	48.01	-0.48	-0.98%

日経平均について (銘柄一覧) [Quick](#)

新たな経済潮流と
事業用地

千葉ニュータウン立地の期待と
可能性を検証する

UR都市機構

Wearable WEB



これからのHTML5はウェアラブルになるべき
だと突然思った。(2014年ごろ)

NPO法人日本ウェアラブルデバイスユーザー会設立

日本ウェアラブルデバイスユーザー会

📧 [メールリスト](#)



ウェアラブルデバイスデバイスの最新情報を共有しましょう。

w-ug.jp

注目のキーワード [unity](#) [アプリ](#) [swift](#) [データ](#) [HTML5](#) [デザイン](#) [UX](#)

日本ウェアラブルデバイスユーザー会 (WUG) NPO法人設立キックオフカンファレンス

Tweet Like 185

G+ 0

B! 0



日時: 2016/04/29 (金) 13:00 - 17:30

[Googleカレンダーに追加](#)

定員: 30人

会場: 株式会社 神戸デジタル・ラボ セミナールーム (兵庫県神戸市中央区京町72番地 新クレセントビル10階)

URL: <http://w-ug.jp/>

主催者: [ころわん](#) [+フォローする](#)

ハッシュタグ: [#wearableug](#)



[Google Mapsで表示](#)

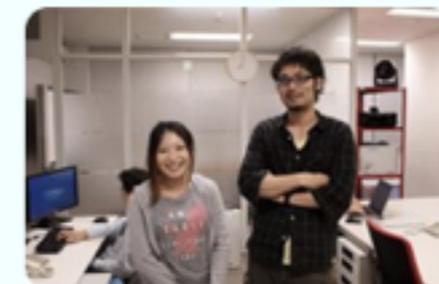
イベントに補欠登録する

現在、定員以上の参加がありますが、補欠として登録することができます。

参加登録期限: 2016/04/29 (金) 17:30

技術志向のITエンジニアのための求人サイト

CodeIQ JOBS



株式会社GINGA soft.

趣味のゲームやパソコン好きが内定のきっかけに！独学エンジニアがゲーム開発プロジェクトで活躍している会社です。

[C初級](#) [C#初級](#) [C++初級](#)

[Java初級](#) [PHP初級](#) [Ruby初級](#)

日本ウェアラブルデバイスユーザー会 (WUG) NPO法人設立キックオフカンファレンス

日本ウェアラブルデバイスユーザー会 (WUG) は、日本国内のウェアラブルデバイス振興に貢献することを目的として2014年から活動して参りましたが、このたび活動をさらに加速させるため特定非営利法人 (NPO法人) として申請、認証を受けました。

神戸市ウェアラブルデバイス推進会議メンバーになったり

現在位置 [トップページ](#) > [総合メニュー](#) > [市政情報](#) > [計画・事業](#) > [企画調整局の計画・事業等](#) > 神戸市ウェアラブルデバイス推進会議

神戸市ウェアラブルデバイス推進会議

ツイート

シェア 56

最終更新日2016年5月18日

[5月18日]第6回神戸市ウェアラブルデバイス推進会議の議事内容を公開しました。

神戸市では、ウェアラブルデバイスを使用した市民参加型の実証事業に取り組み、新たな市民サービスの創出を目指しています。実証事業を行う分野・テーマ、さらにはその方法を検討するため、ウェアラブルデバイスに関して専門的な見地から幅広く意見をいただく有識者会議を開催しています。

委員

- ・神戸大学大学院工学研究科教授 塚本 昌彦 (つかもと まさひこ) 【座長】
 - ・東京大学大学院情報理工学系研究科教授 稲見 昌彦 (いなみ まさひこ)
 - ・大阪電気通信大学総合情報学部デジタルゲーム学科教授 上善 恒雄 (じょうぜん つねお)
 - ・株式会社ブリリアントサービス代表取締役 杉本 礼彦 (すぎもと よしひこ)
 - ・神戸大学大学院工学研究科准教授 寺田 努 (てらだ つとむ)
 - ・神戸大学学術研究戦略企画室特命教授 富田 克彦 (とみた かつひこ)
 - ・株式会社ユニックス デザイン オフィス 代表取締役 中内 淑文 (なかうち よしふみ)
 - ・株式会社社会システム総合研究所代表取締役 西田 純二 (にしだ じゅんじ)
 - ・ウエストユニティス株式会社 代表取締役 福田 登仁 (ふくだ たかひと)
 - ・バスタイムフィッシュ 村岡 正和 (むらおか まさかず)
- (順不同・敬称略)

なぜか神戸市ポケモンGo対応チームのメンバーにされた

『「ポケモンGO」等スマートフォン及びその利用サービスの進化への対応プロジェクトチーム』の設置について

記者資料提供（平成28年8月2日）

企画調整局創造都市推進部（ICT創造担当） 松崎・長井

TEL：078-322-6462 内線：2365

E-MAIL: ict-sozo@office.city.kobe.lg.jp

『「ポケモンGO」等スマートフォン及びその利用サービスの進化への対応プロジェクトチーム』の設置について

1. 目的

「ポケモンGO」などスマートフォン及びその利用サービスの進化に対し、安全・安心、地域の活性化、子育て・教育など多様な観点から適切に対応していくため、民間事業者、職員等から構成されるプロジェクトチームを設置します。つきましては、以下のとおり、第1回プロジェクトチームミーティングを開催し、スマートフォンの進化が自治体行政、地域社会にどのような影響を与えるのかを把握・分析し、必要な対応の検討を行います。

2. メンバー

・外部人材 3名（50音順・敬称略）

田居 夏生 株式会社NTTドコモR&Dイノベーション本部イノベーション統括部企業連携担当部長

村岡 正和 バスタイムフィッシュ、株式会社神戸デジタル・ラボ 取締役

（オブザーバー参加）

河本 博幸 兵庫県警本部警務部監察官室長兼神戸市警察部庶務課長

じつはFirefox OSってスマートグラスのOSとして良かったかも



Firefox OS



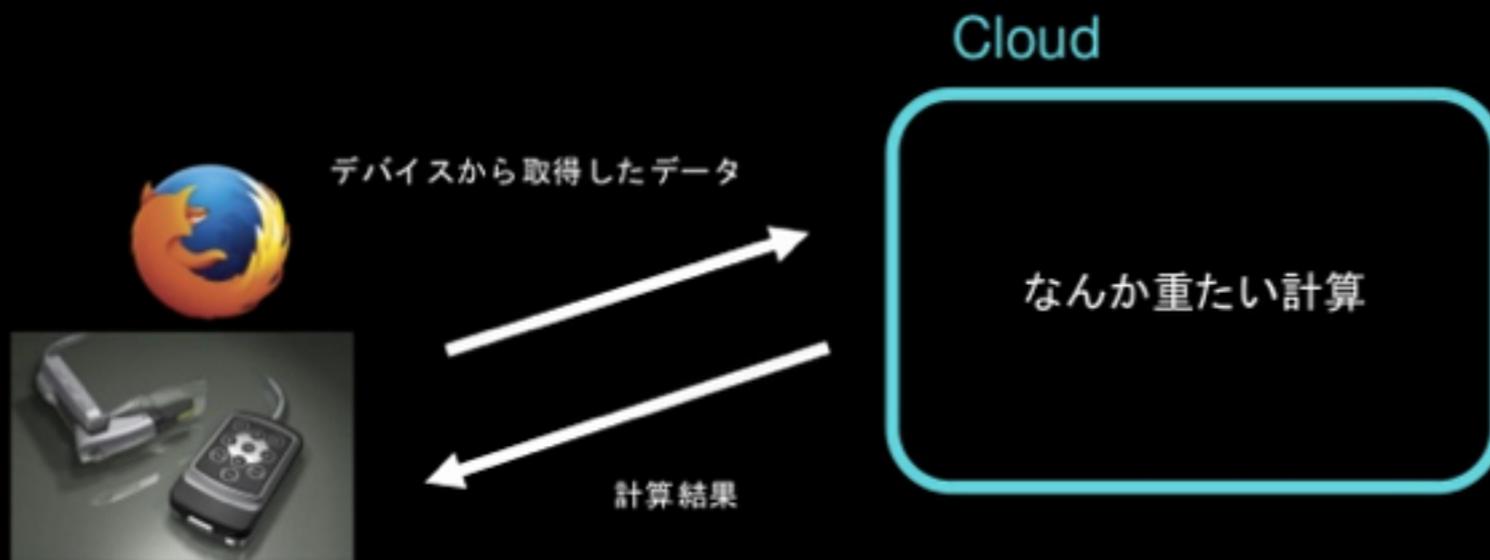
つい先日、勉強会で不平不満をグチりました。

<http://bit.ly/2ciOqjm>

Be the first to clip this slide

ウェアラブルデバイスは低リソースな反面重い計算処理が求められることが多い（映像配信とか音声認識とか）計算処理はできるだけクラウド側に逃してウェアラブル側は結果だけ表示するような設計が有効

mozappsを使うと最新のHTML5 API(DAPやPeerConnedtion etc)が使えるapkがカジュアルに開発できる。フルスクリーンで一発起動なので画面が狭く操作系が弱いスマートグラスでいいかんじ



◀ 21 of 27 ▶

優良企業の人事から直接スカウト
Linkedinユーザーなら簡単登録・簡単レジュメアップでスカウト倍増！【登録無料】

IT分野専門の翻訳・ローカリゼーション
製品のドキュメント、マーケティング資料、教育用コンテンツ、Webサイトまで翻訳・制作。IT分野専門に20年の実績がある株式会社シーブレイン。

New offer
We offer financial services worldwide. Apply now

Ads by LinkedIn

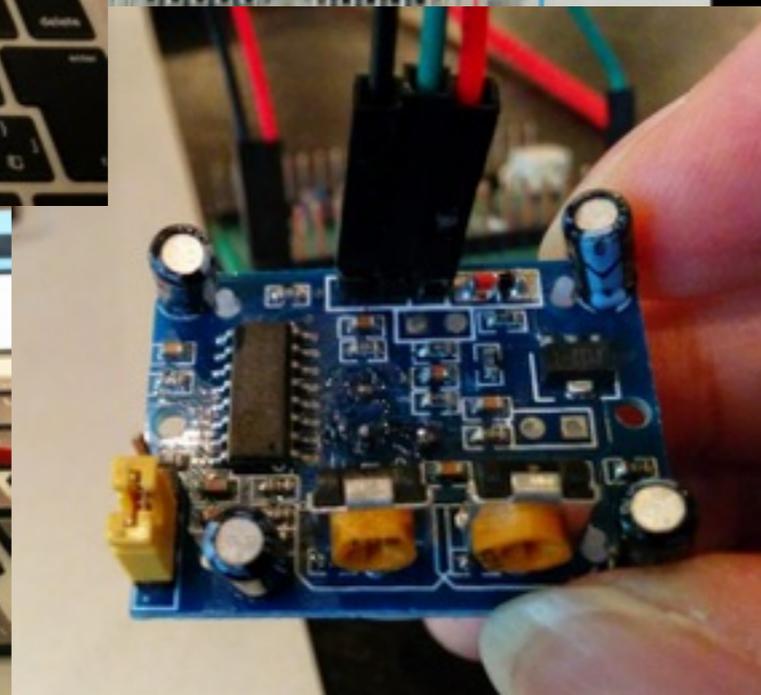
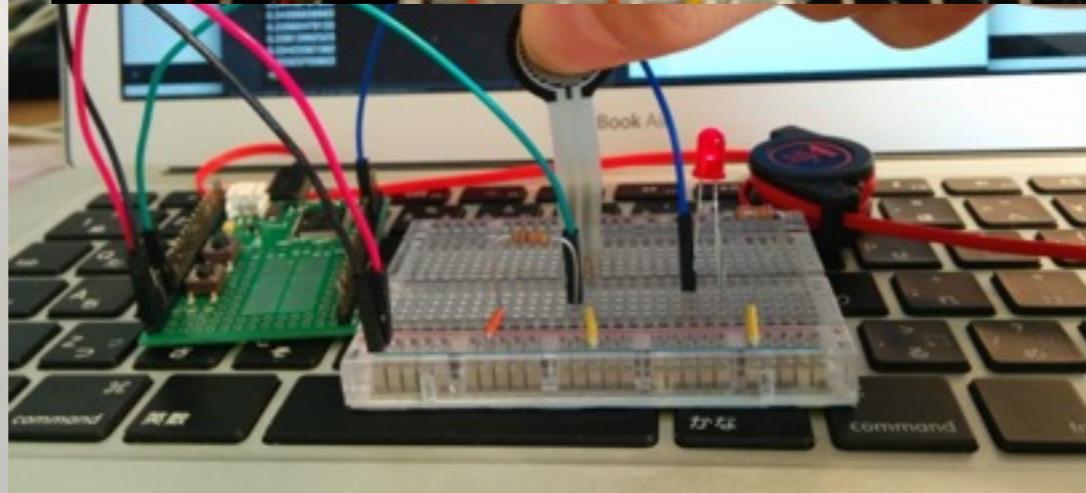
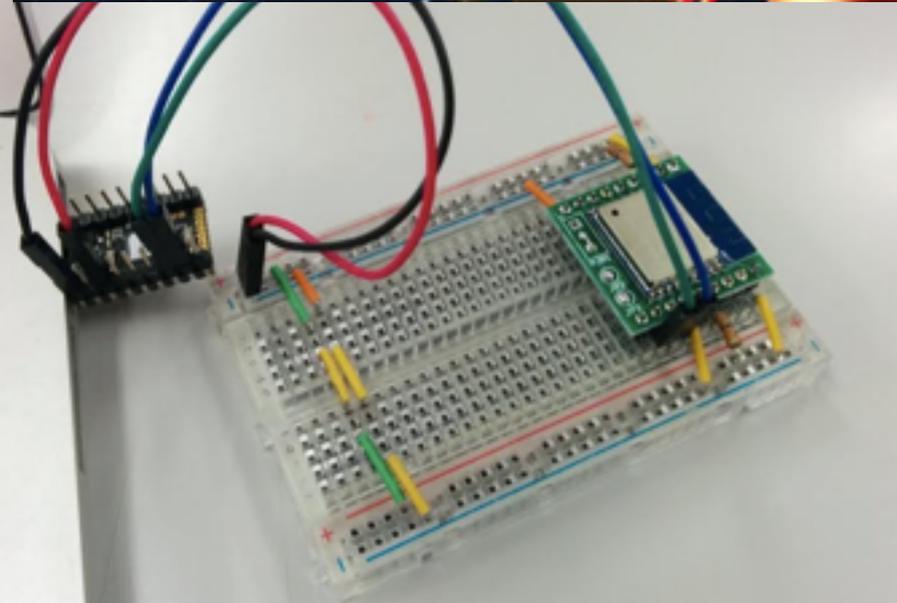
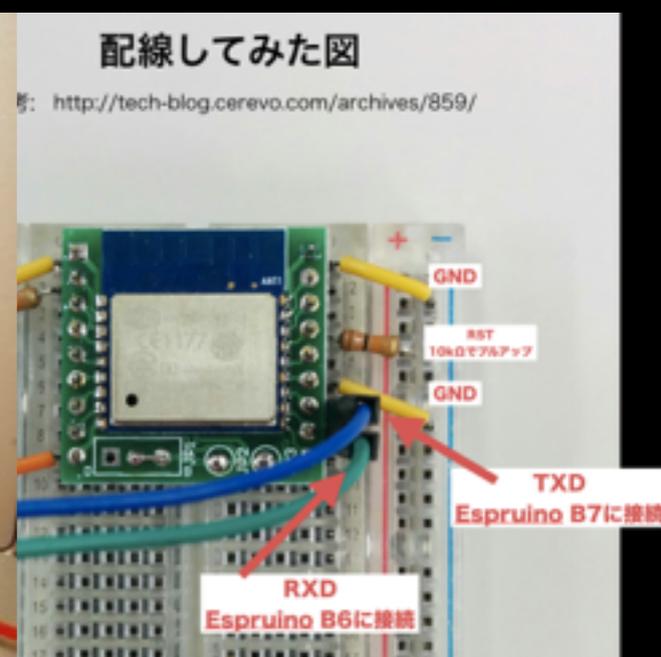
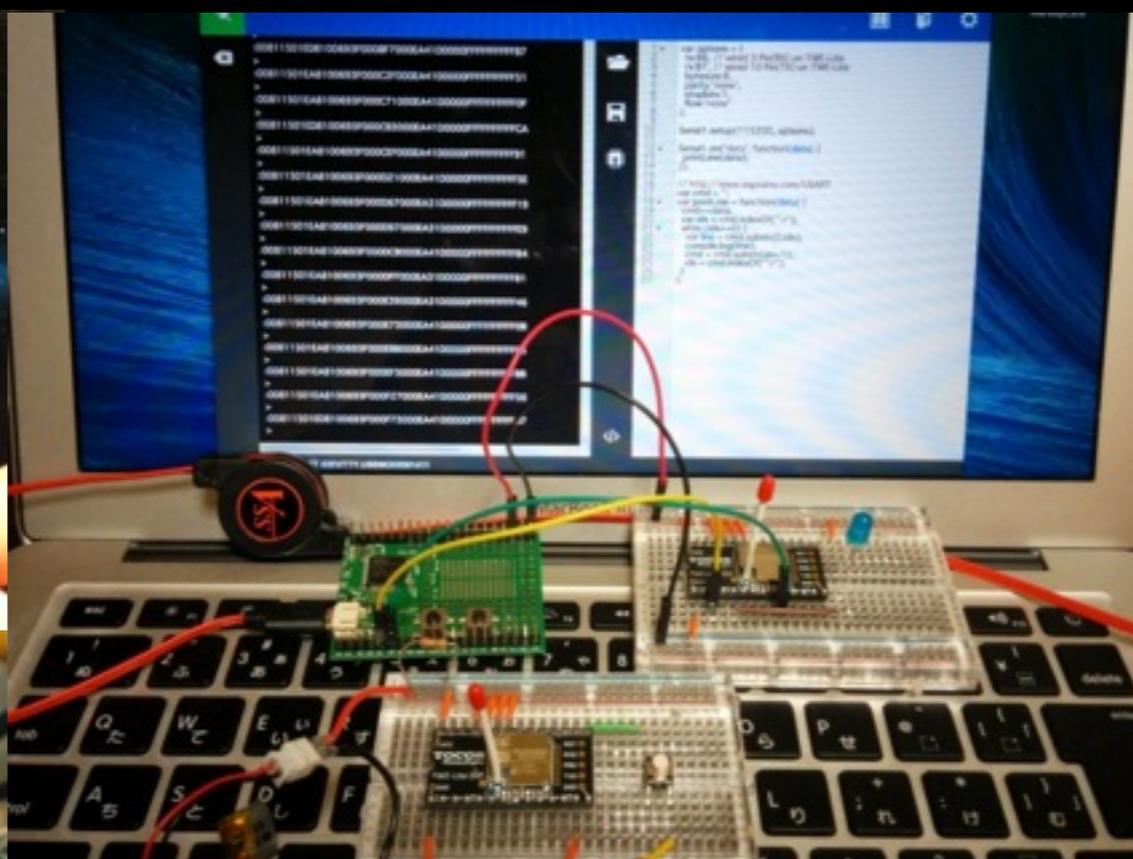
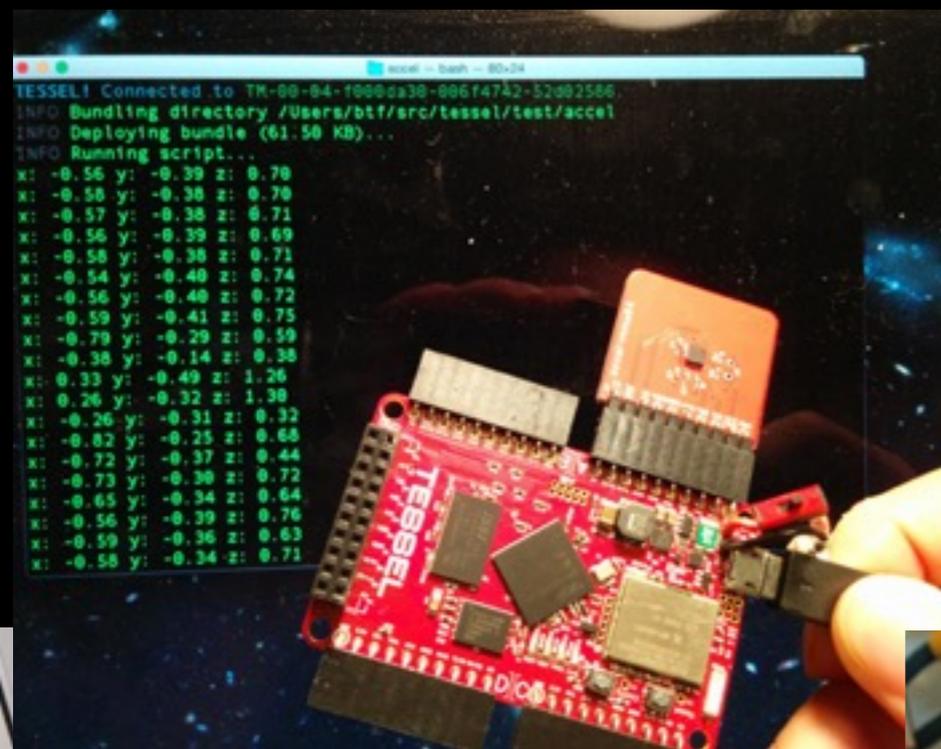
Recommended

Success Awaits
We're giving you \$50 in free LinkedIn advertising credits to get started with Sponsored Content. Click the following LinkedIn Marketing Solutions **Sponsored Content**

react-jsonschema-formについて
Masakazu Muraoka

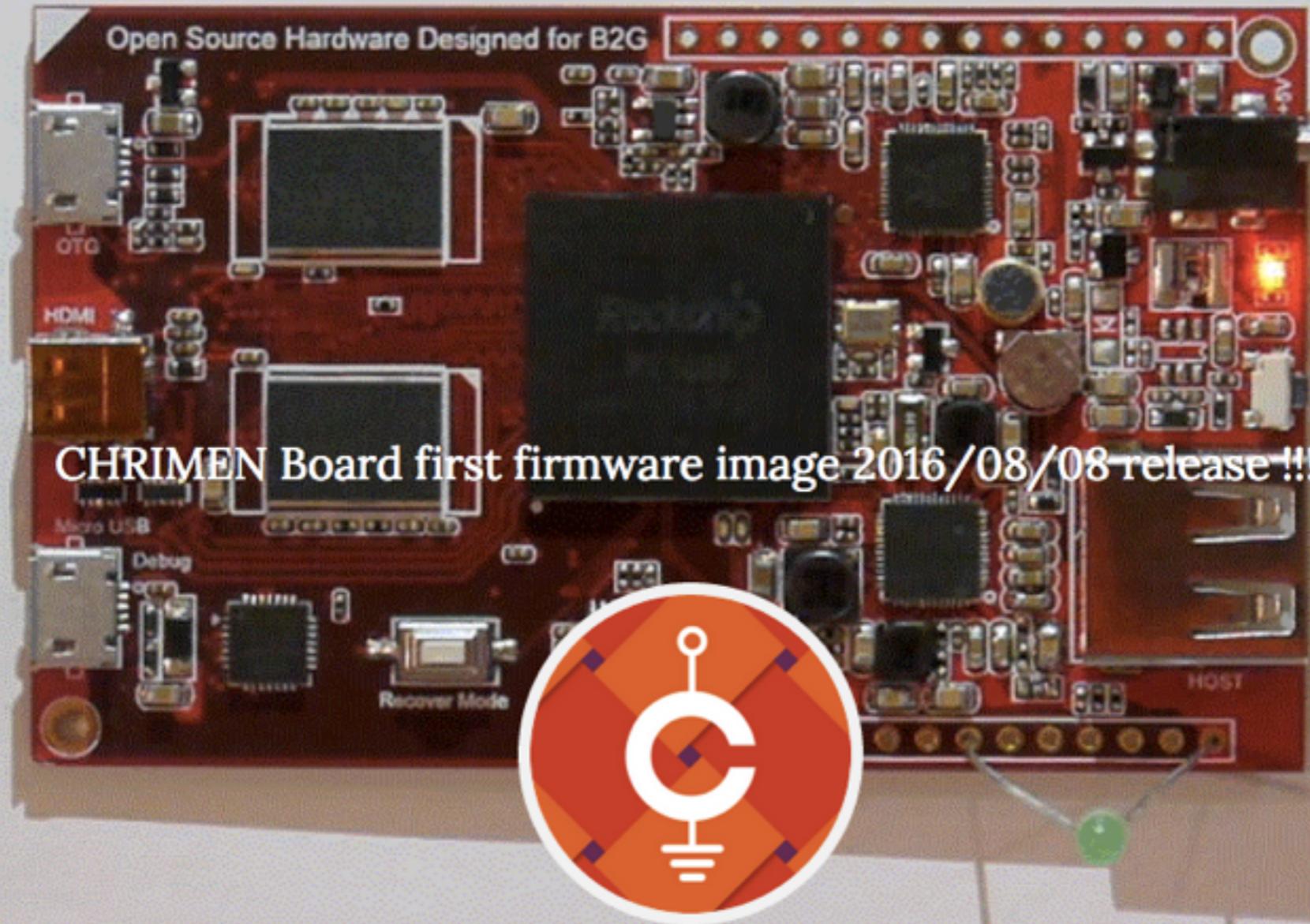
Fetch apiについて
Masakazu Muraoka

まあウェアラブルはおいといて、
ハードウェア関係に知り合いたくさんできたり、
IoT関係の案件や講演の相談がきたり、
趣味だったことが仕事になってる最近です。



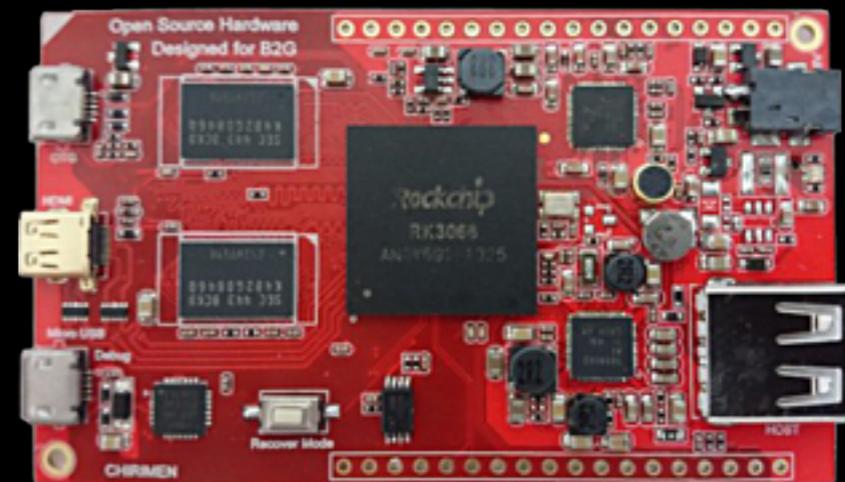
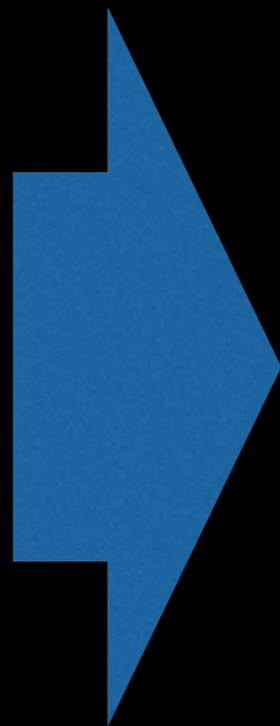
A dark blue background featuring a smartphone on the right side. The phone's screen displays the Firefox OS logo and a stylized fox mascot. The fox is orange and white, with its tail curled around the phone. The text 'Firefox OS' is visible on the phone's screen and in the background.

**Firefox OSの開発資源は
モノづくりに活かされようとしています。**



CHIRIMEN Board first firmware image 2016/08/08 release !!!

CHAIRMEN for WoT



フルスタックなHTML5でセンサーなどのハードウェア制御
が可能なシングルボードコンピュータ

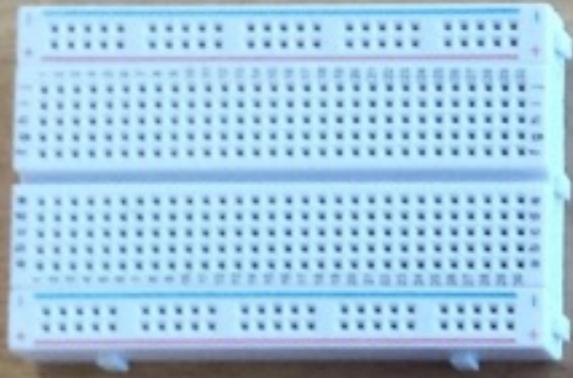
A red printed circuit board (PCB) for the CHIRIMEN project. The board is populated with various electronic components including microchips, capacitors, and connectors. At the top, it says "Open Source Hardware Designed for B2G". On the left side, there are connectors for "Micro-USB" and "Debug". At the bottom, there is a "Recover Mode" button and a "HOST" port. The word "CHIRIMEN" is printed at the bottom left. The board is shown against a dark background.

**今日はCHIRIMENを使って
Web技術でハードウェア制御という
めったにない (誰得? な)
体験をしてみましよう!**

Getting Started

LEDを点滅させる

用意するもの



ブレッドボード



電源ケーブル



CHRIMEN



ジャンプワイヤ(2本)

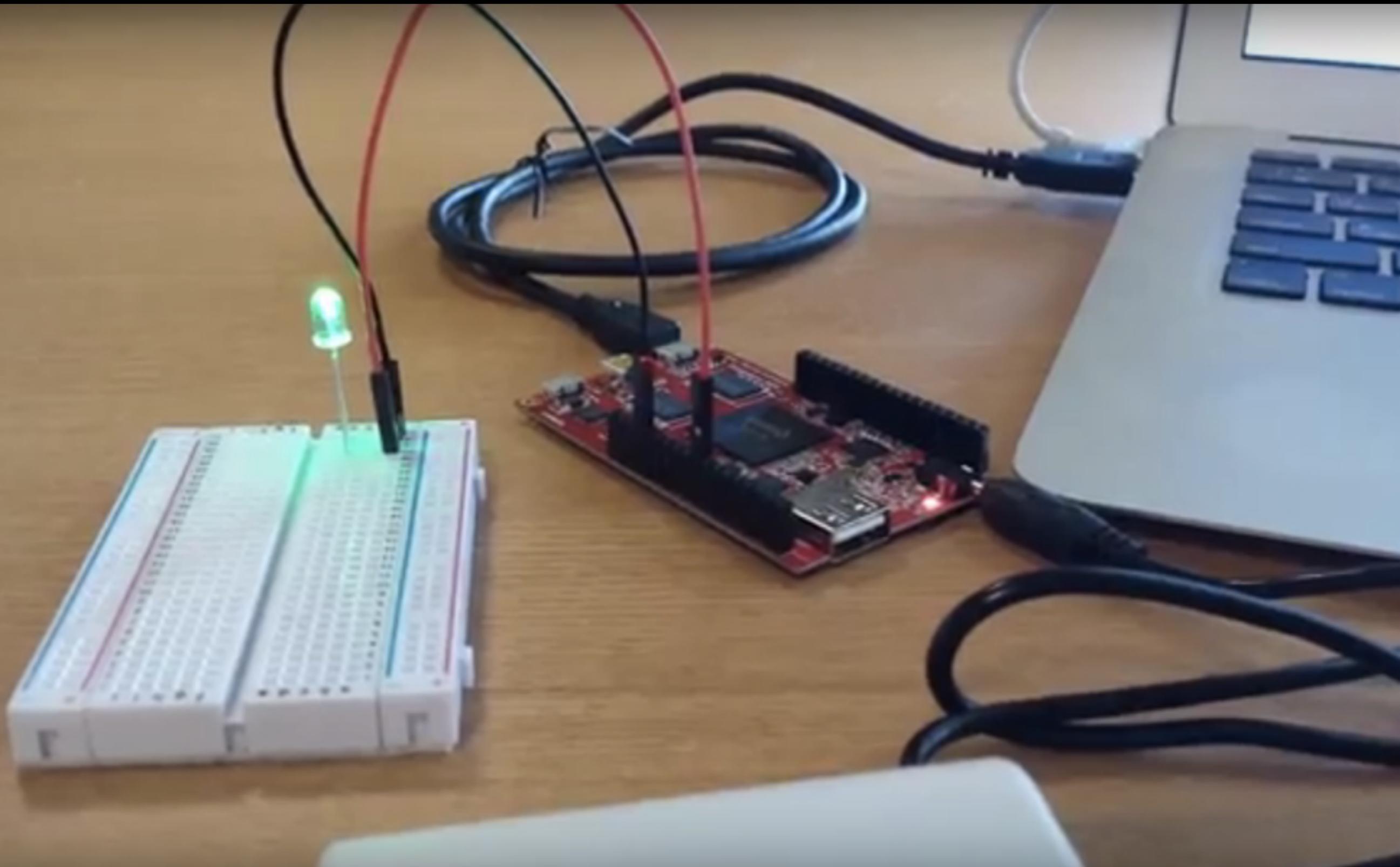


LED



USBケーブル

完成图



予備知識

LEDの足の長さをチェックする



**長いほうがアノード
(+極)**

**短いほうがカソード
(-極)**

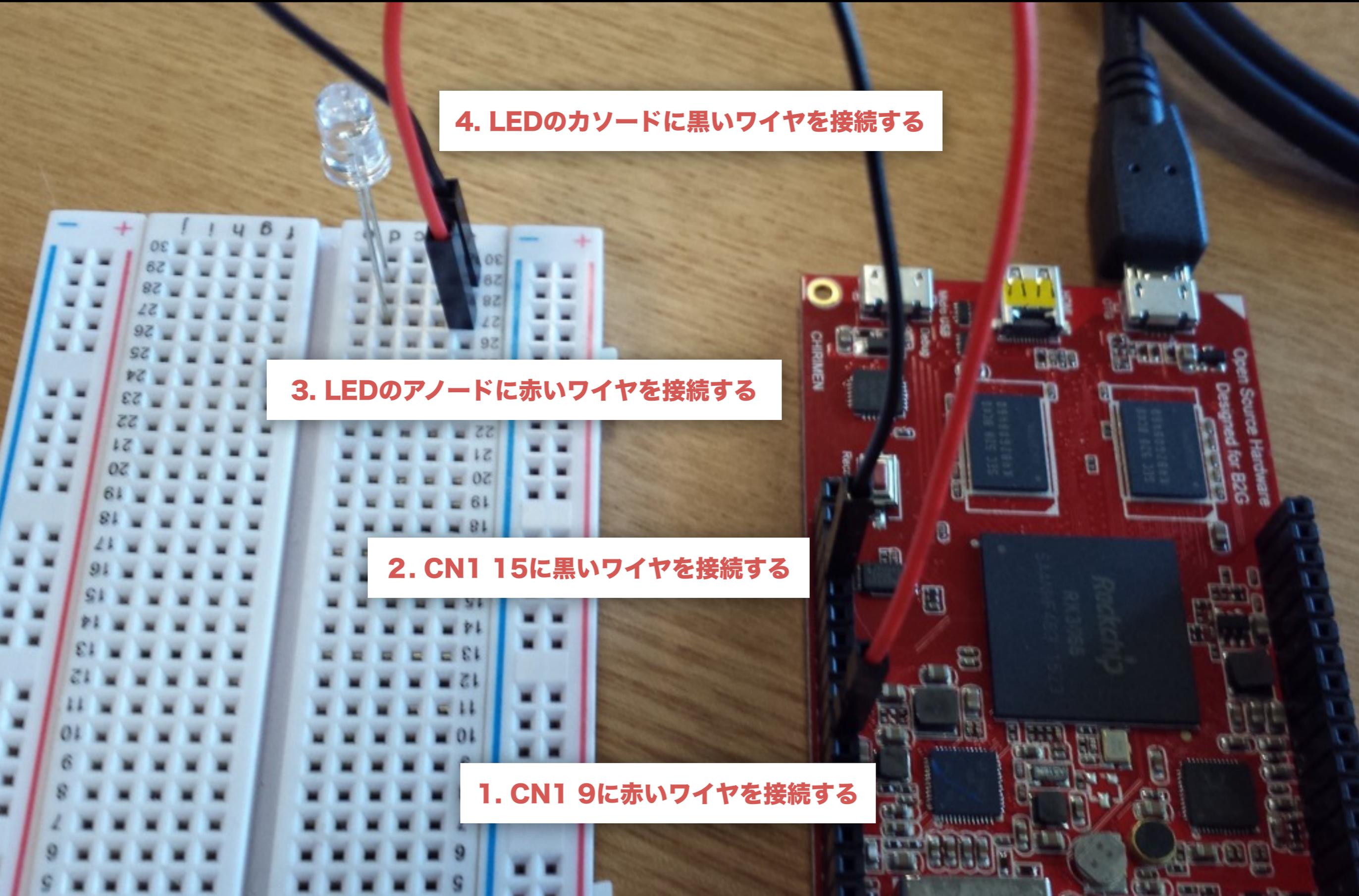
まずは配線 (CHIRIMEの電源は入れない)

4. LEDのカソードに黒いワイヤを接続する

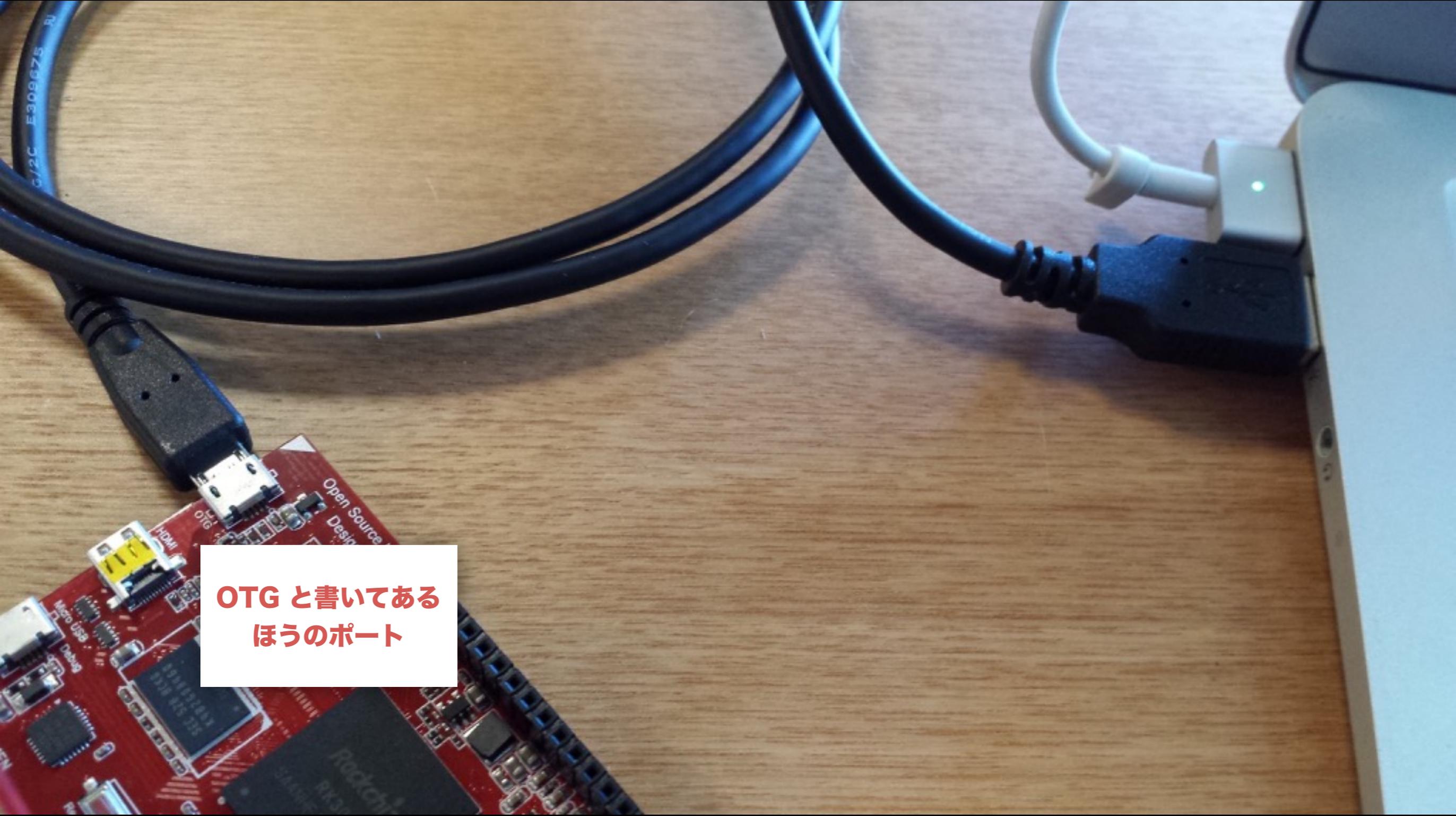
3. LEDのアノードに赤いワイヤを接続する

2. CN1 15に黒いワイヤを接続する

1. CN1 9に赤いワイヤを接続する

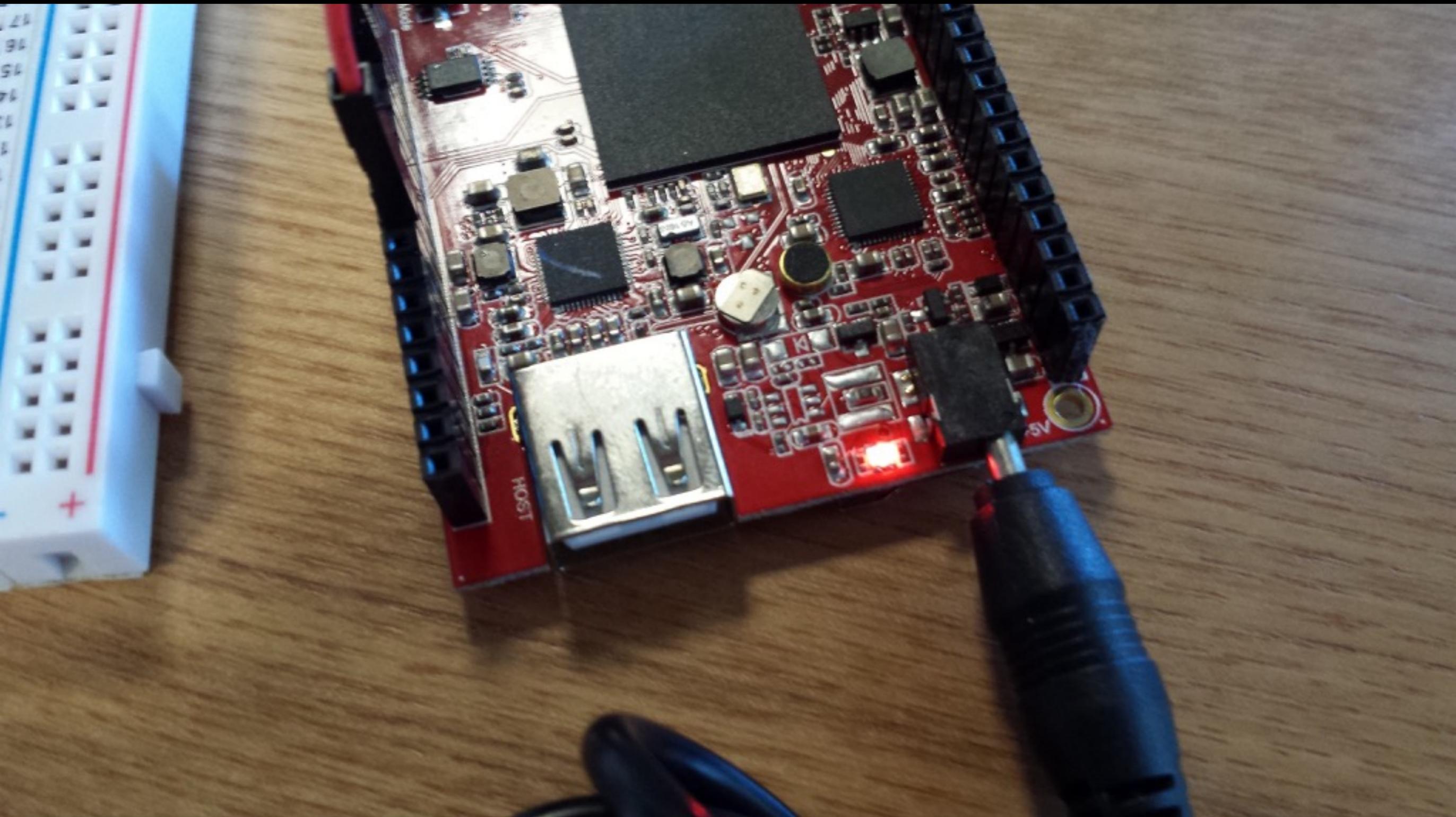


USBケーブルでPCとつなぐ

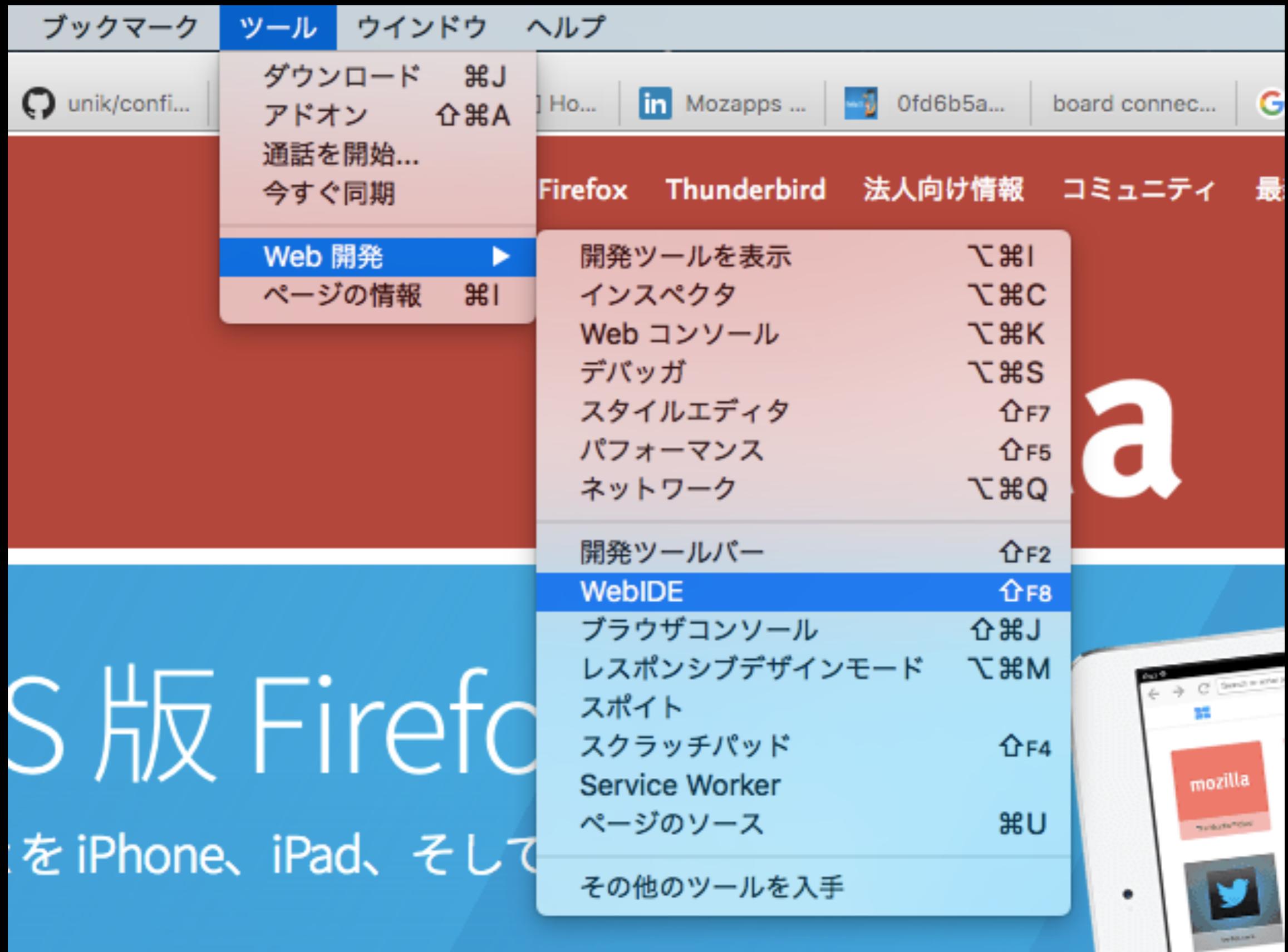


OTG と書いてある
ほうのポート

電源ケーブルをつないで電源をいれる



FirefoxのWebIDEを起動する



USB デバイスから CHRIMEN を選択する



WebIDE と CHRIMEN が接続される

The screenshot shows the Firefox WebIDE Settings interface. The browser title is "Firefox WebIDE: Settings". In the top right corner, there is a status indicator for "CHRIMEN" with a mobile phone icon. The main content area is titled "Settings" and includes a gear icon, the word "CERTIFIED", and the text "説明 Gaia Settings". Below this, the "App ID" is listed as "app://settings.gaiamobile.org/manifest.webapp".

The left sidebar, titled "ランタイムアプリ", lists various applications: メインプロセス, Bluetooth Manager, Bookmark, Browser, BuddyUp, Bugzilla Lite, Built-in Keyboard, Calculator, Calendar, Callscreen, Camera, Clock, Communications, Default Home Scr..., Default Theme, Downloads, and E-Mail.

The right sidebar, titled "USB デバイス", shows a list of connected devices. The first device is "CHRIMEN", which is highlighted. Below it, under "WI-FI デバイス", there is a "シミュレータ" section with a plus icon and the text "シミュレータをインストール". The "その他" (Other) section includes: リモートランタイム, Chrome Desktop, Safari, Firefox, and other WebViews on iOS, ランタイム情報, 許可設定一覧, 端末の設定, 端末オプション, スクリーンショット撮影, and 切断.

サンプルコードをダウンロードする

<https://github.com/bathtimefish/chirimen-getting-started-examples/releases/tag/0.1>

The screenshot shows the GitHub interface for the repository `bathtimefish / chirimen-getting-started-examples`. The page is viewed on the 'Releases' tab. The 'First release' section is highlighted, showing a release by `bathtimefish` made 4 minutes ago. The release details include the version `0.1` and the commit message `first commit`. Below the release information, there are two download links: `Source code (zip)` and `Source code (tar.gz)`. The left sidebar shows the 'Latest release' section with the version `0.1` and the commit hash `89a4c4c`.

This repository Search Pull requests Issues Gist

bathtimefish / chirimen-getting-started-examples Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Releases Tags Edit release Delete

Latest release

0.1
89a4c4c

First release

bathtimefish released this 4 minutes ago

0.1

first commit

Downloads

Source code (zip)

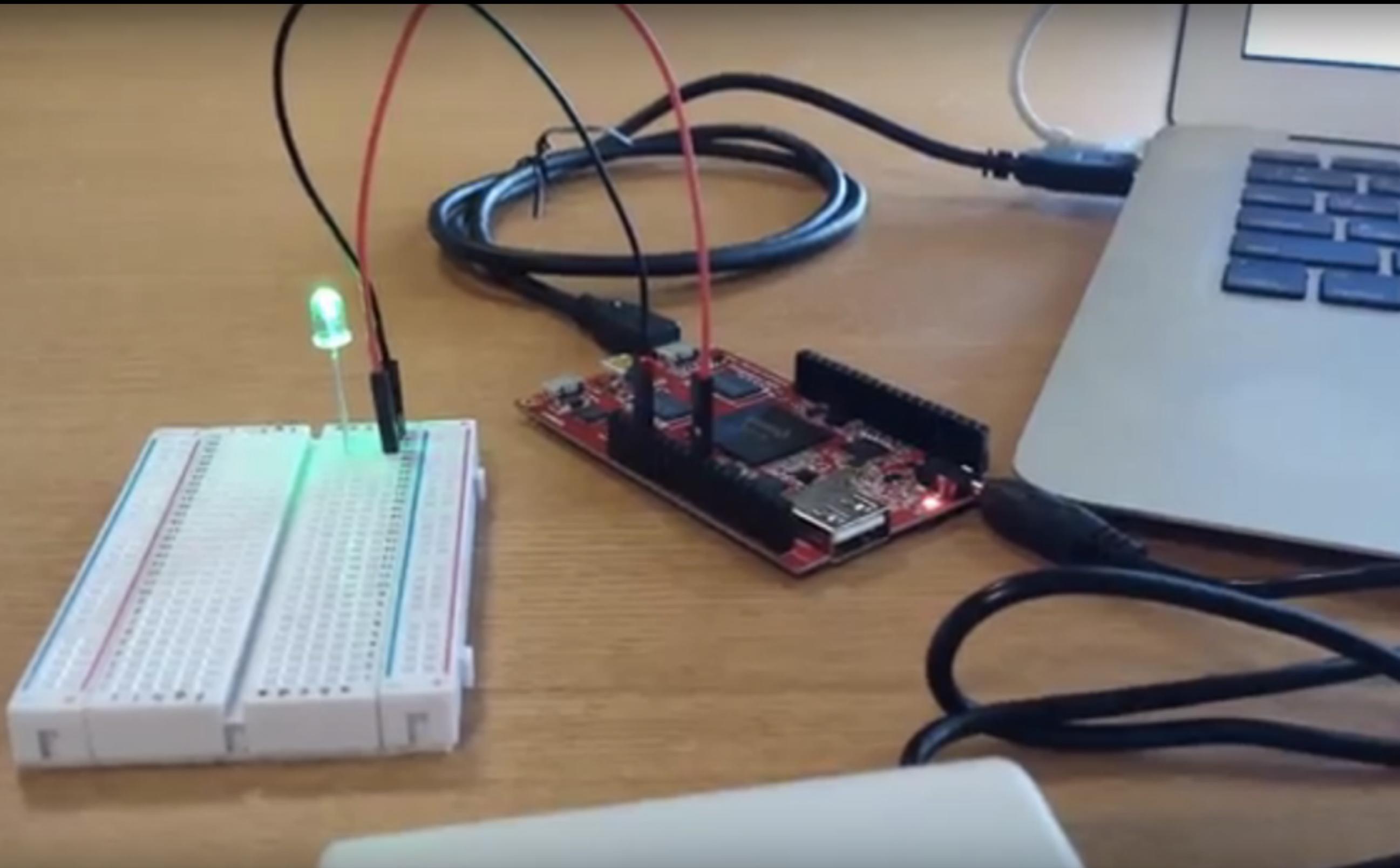
Source code (tar.gz)

blink-ledプロジェクトを読み込む

The screenshot displays the Firefox WebIDE interface for a project named "blink-led". The window title is "Firefox WebIDE: blink-led". At the top right, the user's name "CHIRIMEN" is visible. The interface is divided into several sections:

- Left Panel (Applications):** A list of system applications including "ランタイムアプリ", "メインプロセス", "Bluetooth Manager", "Bookmark", "Browser", "BuddyUp", "Bugzilla Lite", "Built-in Keyboard", "Calculator", "Calendar", "Callscreen", "Camera", "Clock", "Communications", "Default Home Scr...", "Default Theme", "Downloads", and "E-Mail".
- Project Explorer:** Shows the file structure for "blink-led":
 - ▶ bower_components/
 - ▶ icons/
 - app.js
 - bower.json
 - index.html
 - manifest.webapp
- Project Details:** Displays information for the "blink-led" project:
 - Project Name:** blink-led
 - Type:** PACKAGED WEB
 - 説明:** blink LED
 - 場所:** /Users/btf/src/FirefoxOS/chirimen/blink-led
 - App ID:** app://1f498415-0e77-9d40-b936-93b08e6a705e/manifest.webapp
- Right Panel (Settings):** A sidebar with various system settings:
 - USB デバイス (CHIRIMEN)
 - WI-FI デバイス
 - シミュレータ (シミュレータをインストール)
 - その他 (リモートランタイム, Chrome Desktop, Safari, Firefox, and other WebViews on iOS)
 - ランタイム情報
 - 許可設定一覧
 - 端末の設定
 - 端末オプション
 - スクリーンショット撮影
 - 切断

実行するとLEDが点滅する



プログラムの要点

```
1
2 window.addEventListener("load", function() {
3   let { spawn } = task;
4   spawn(function() {
5     const gpio = yield navigator.requestGPIOAccess();
6     // https://chirimen.org/docs/en/board\_
7     const led = gpio.ports.get(198);
8     yield led.export("out");
9     var on = 0;
10    setInterval(() => {
11      on = on ? 0 : 1;
12      console.log(on);
13      led.write(on);
14    }, 1000);
15  });
16 });
```

LEDを接続したデバイスの取得と設定

1秒間隔で点滅させる
ループ処理

CN1 9ポートはGPIO 198番

なので `gpio.ports.get(198);` となるんだけどわかりにくいよねこれなんとかしてください。

Note: Currently UART, SPI and PWM are re-assigned to GPIO.

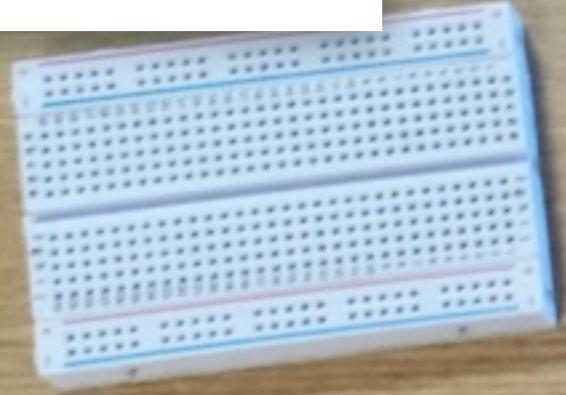
CN1 (Connector1)

Number	Description (sysfs name)	Drive	Pull up/down
1	GND	N/A	N/A
2	I2C-2 SDA	8mA	UP
3	I2C-2 SCL	8mA	UP
4	GPIO-3 D3 (gpio283)	8mA	UP
5	GPIO-3 D4 (gpio284)	8mA	DOWN
6	ADC-0 in	N/A	N/A
7	GPIO-1 A4 (gpio196)	8mA	UP
8	GPIO-1 A5 (gpio197)	8mA	DOWN
9	GPIO-1 A6 (gpio198)	8mA	UP
10	GPIO-1 A7 (gpio199)	8mA	UP
11	GPIO-1 C4 (gpio244)	8mA	DOWN
12	GPIO-1 C3 (gpio243)	4mA	DOWN
13	GPIO-1 C6 (gpio246)	8mA	DOWN
14	GPIO-1 C5 (gpio245)	8mA	DOWN
15	GND	N/A	N/A
16	VCC 3.3V	N/A	N/A
17	VCC 3.3V	N/A	N/A
18	VCC 5V	N/A	N/A

人が近づいたら光るやつにする

用意するもの

ブレッドボード



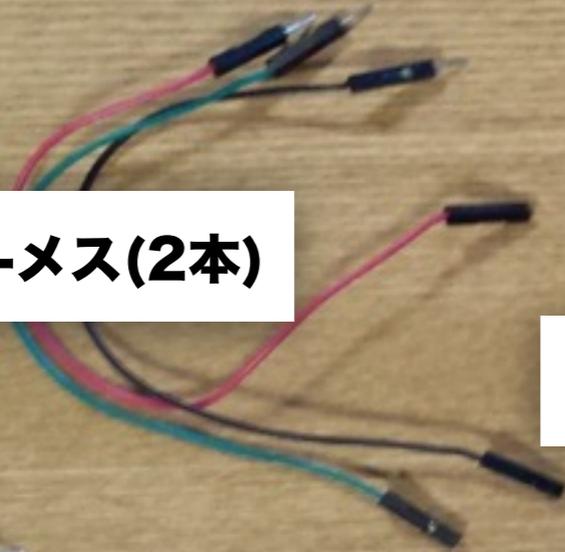
CHROMEN



電源ケーブル



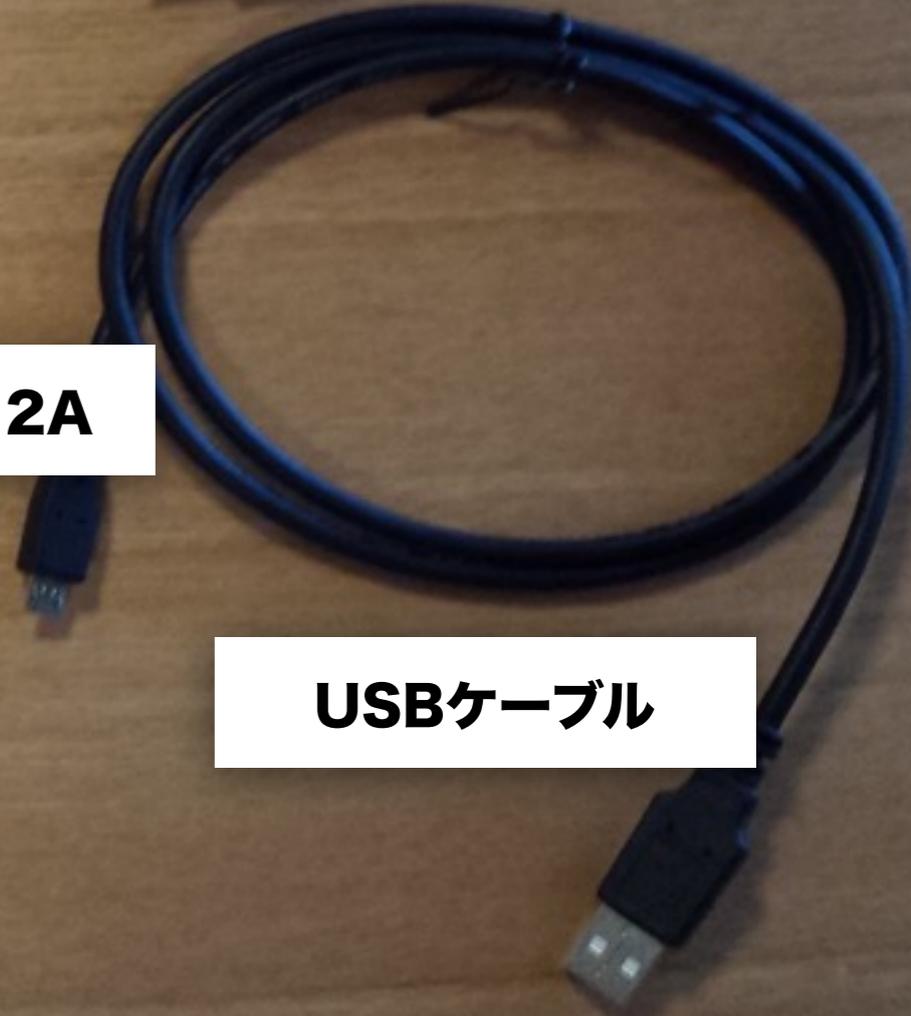
ジャンプワイヤ オス-メス(2本)



赤外線焦電センサ SB612A



USBケーブル



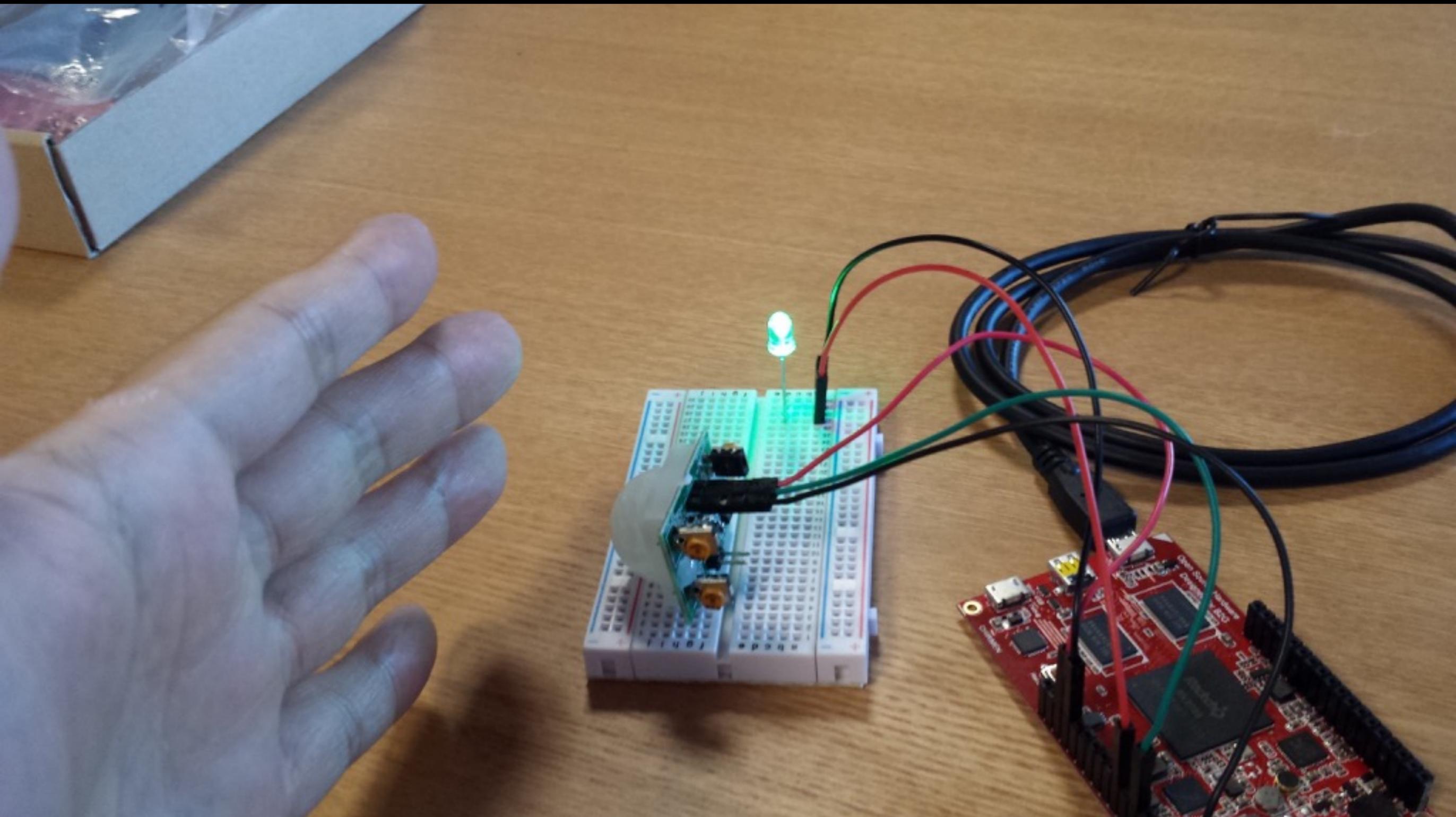
ジャンプワイヤ オス-オス(2本)



LED

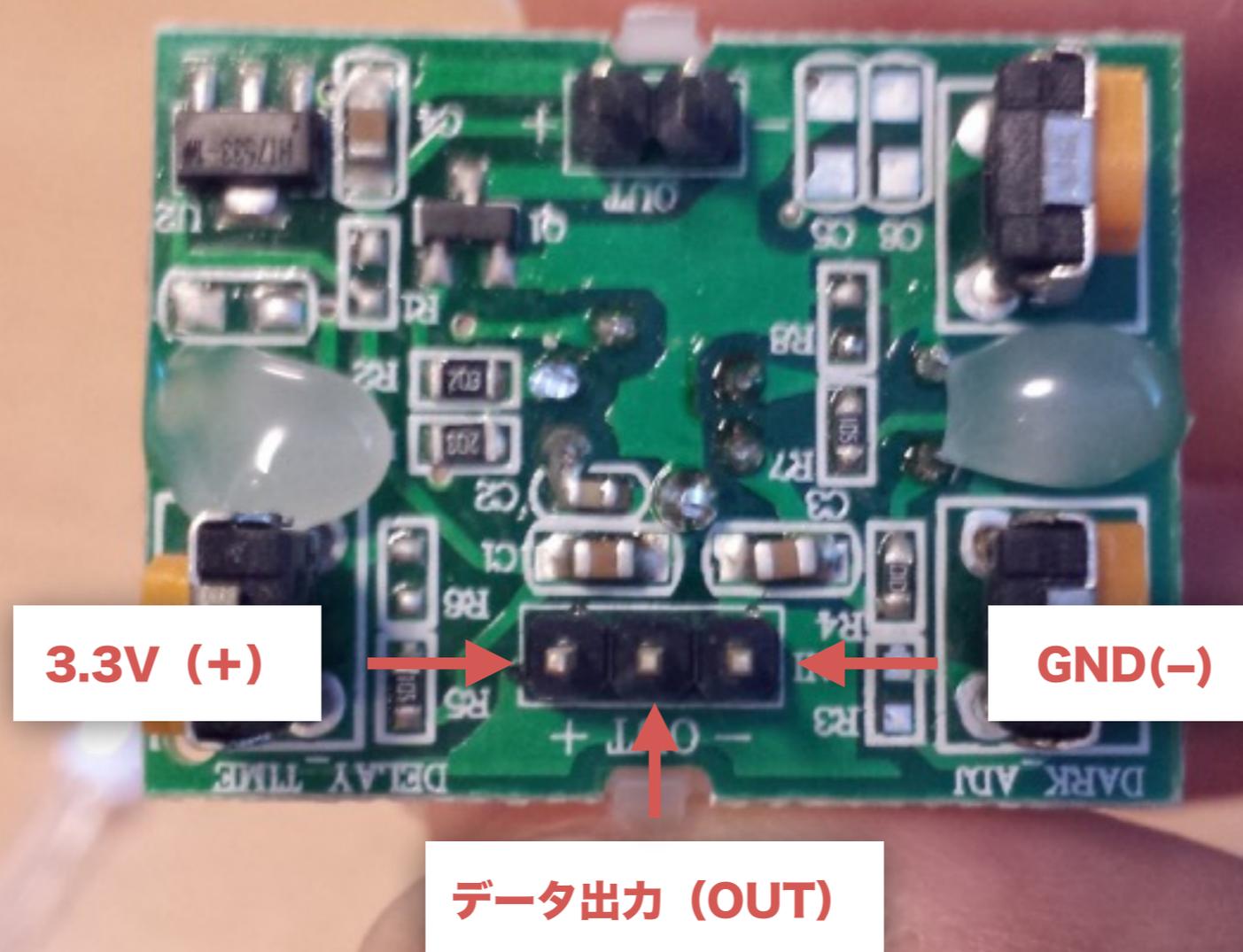


完成图



予備知識

SB612Aのピンレイアウト



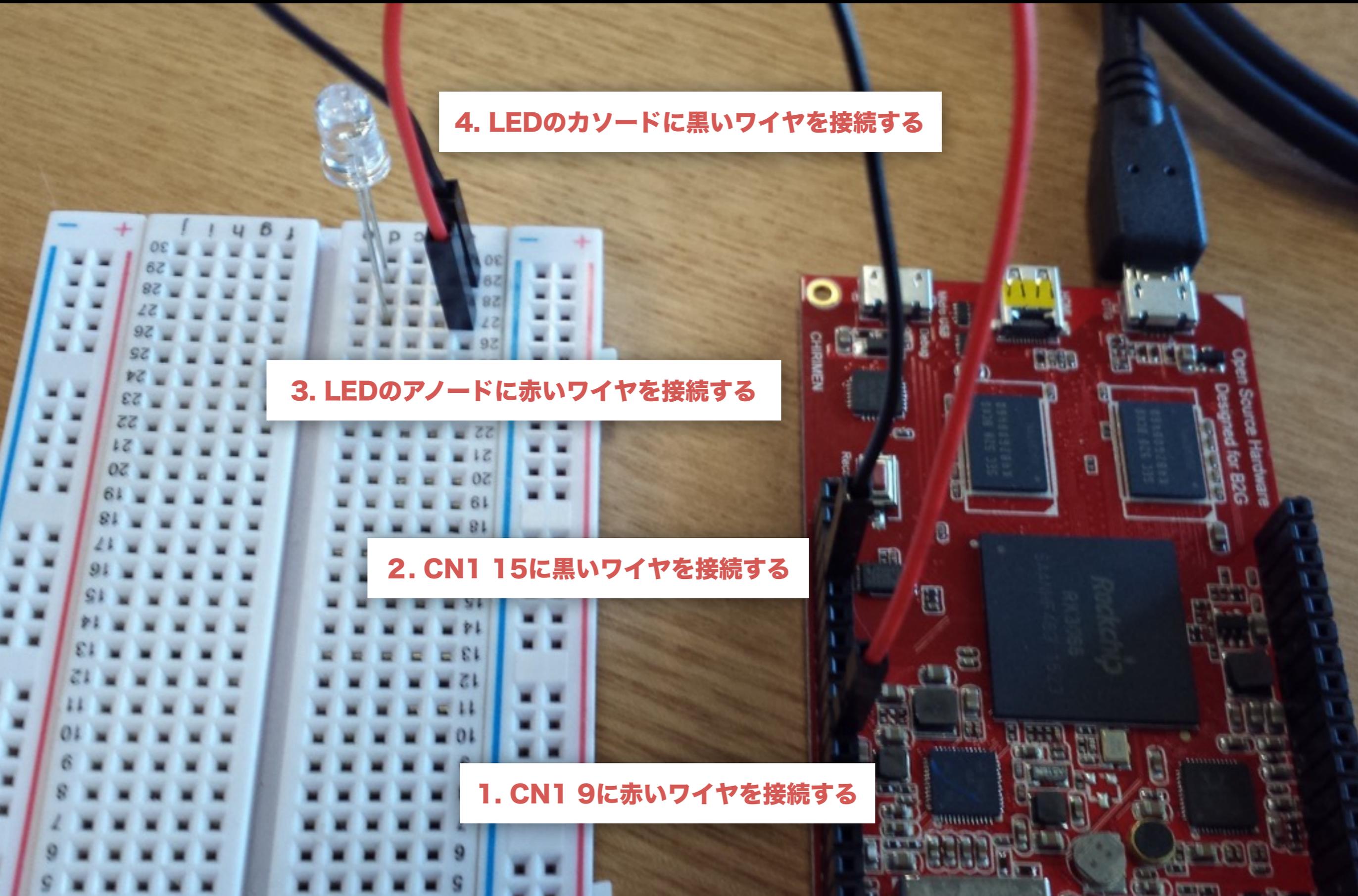
LEDの配線 (CHIRIMEの電源は入れない)

4. LEDのカソードに黒いワイヤを接続する

3. LEDのアノードに赤いワイヤを接続する

2. CN1 15に黒いワイヤを接続する

1. CN1 9に赤いワイヤを接続する



焦電センサの配線

(CHIRIMEの電源は入れない)

5. +に赤いワイヤを接続する

5. OUTに緑のワイヤを接続する

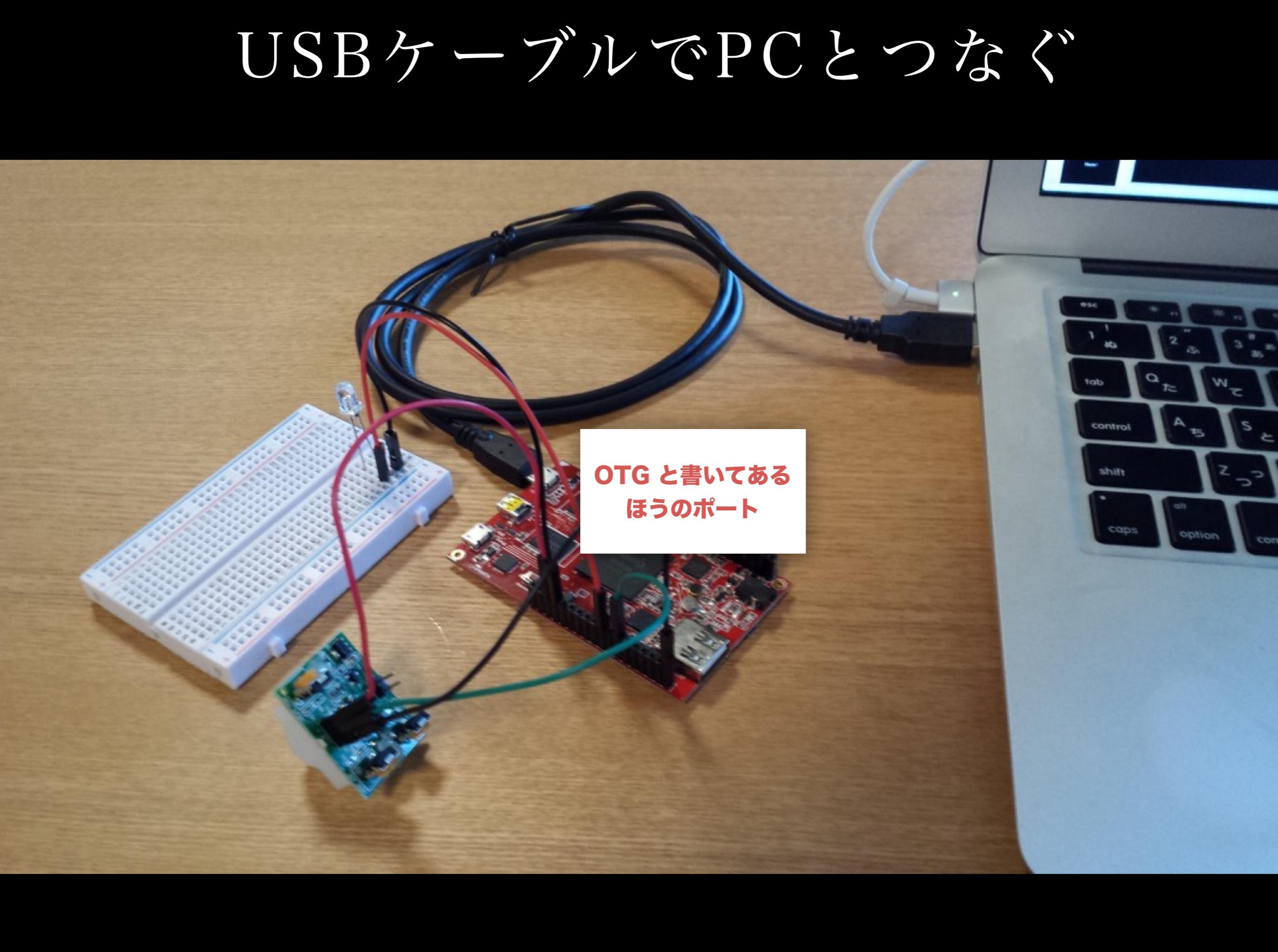
4. -に黒いワイヤを接続する

3. CN1 16に赤いワイヤを接続する

2. CN1 7に黒いワイヤを接続する

1. CN1 1に黒いワイヤを接続する

USBケーブルでPCとつなぐ



OTG と書いてある
ほうのポート

電源ケーブルをつないで電源をいれる



motion-sensor-ledプロジェクトを読み込む

The screenshot displays the Firefox WebIDE interface for the 'motion-sensor-led' project. The browser title is 'Firefox WebIDE: motion-sensor-led'. The top navigation bar includes a refresh button, a stop button, and a settings icon, along with the user name 'CHIRIMEN'.

The left sidebar contains a list of application components, with 'motion-sensor-led' selected and highlighted in blue. Below it, a file explorer shows the project structure:

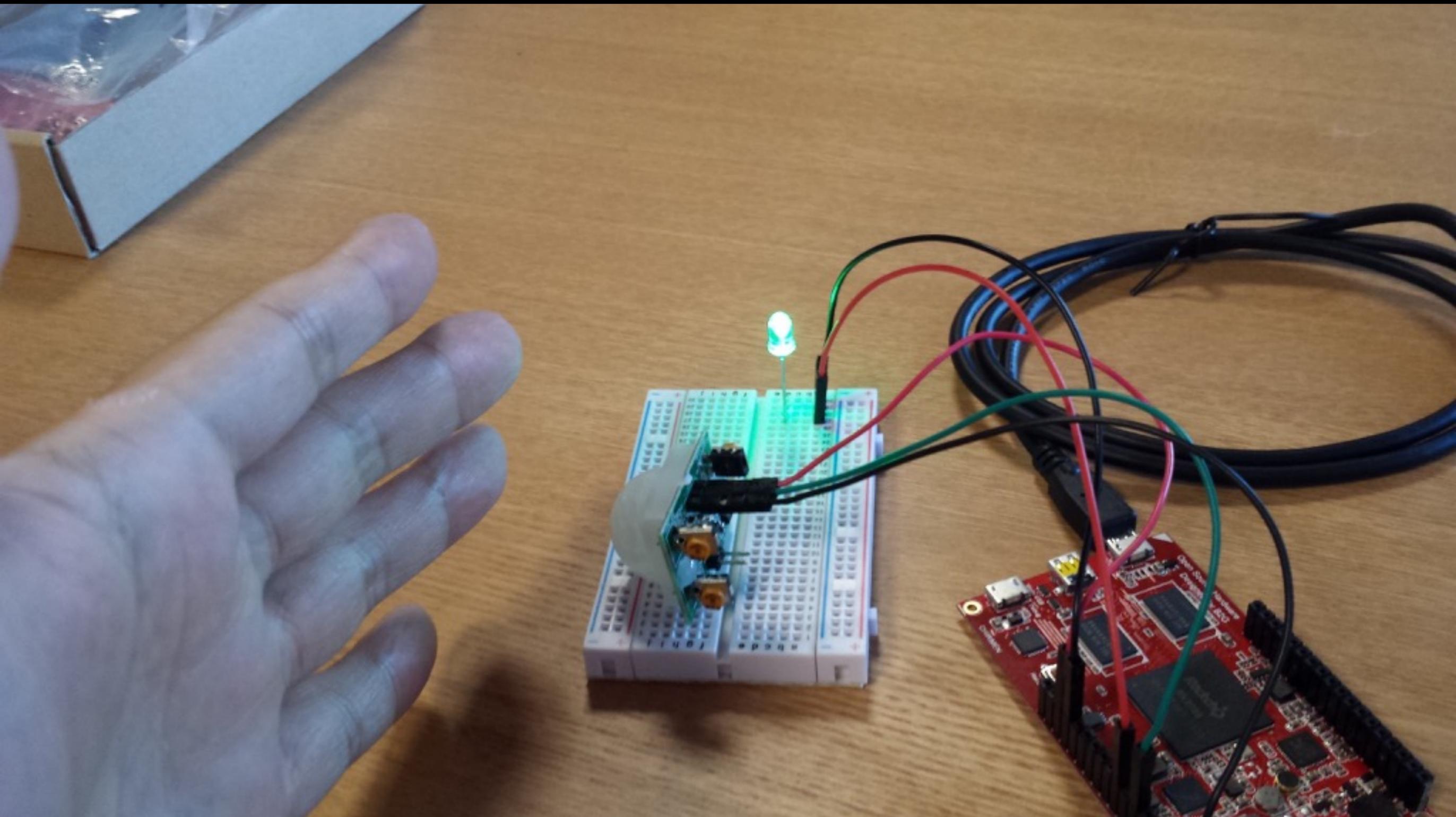
- ▶ bower_components/
- ▶ icons/
- app.js
- bower.json
- index.html
- manifest.webapp

The main workspace area shows the project details for 'motion-sensor-led', which is a 'PACKAGED WEB' application. The description is 'A Hello World app'. The location is '/Users/btf/src/FirefoxOS/chirimen/motion-sensor-led'. The App ID is 'app://26d77030-b3a4-7f4b-9214-c3ef3ba3c15d/manifest.webapp'. A '有効' (Active) status indicator is visible in the top right of the workspace area.

The right sidebar contains a list of system settings and options:

- USB デバイス
 - CHIRIMEN
- WI-FI デバイス
- シミュレータ
 - シミュレータをインストール
- その他
 - リモートランタイム
 - Chrome Desktop
 - Safari, Firefox, and other WebViews on iOS
- ランタイム情報
- 許可設定一覧
- 端末の設定
- 端末オプション
- スクリーンショット撮影
- 切断

実行すると人を感知して光る



プログラムの要点

```
1
2 window.addEventListener("load", function() {
3   let { spawn } = task;
4   spawn(function() {
5     const gpio = yield navigator.requestGPIOAccess();
6     console.info(gpio);
7     // https://chirimen.org/docs/en/board\_connectors.html
8     const sensor = gpio.ports.get(196); // Motion Sensor Data Pin: CN1 7
9     const led = gpio.ports.get(198); // LED Anode Pin: CN1 9
10    yield sensor.export("in");
11    yield led.export("out");
12    setInterval(() => {
13      sensor.onchange = v => {
14        console.info(v);
15        led.write(v);
16      };
17    }, 1000);
18  });
19 });
```

LEDとセンサーの
の取得と設定

← センサーは"in"

← LEDは"out"

1秒間隔でセンサーの
変化をみるループ処理

うまくいってる？

うまくいかない場合は配線がちゃんとできているか
プログラムにバグはないか調べてみよう。

HTTP通信をする

用意するもの

HDMIディスプレイ

CHROMION

USBハブ

マウス

電源ケーブル

Planex GW-USNano2

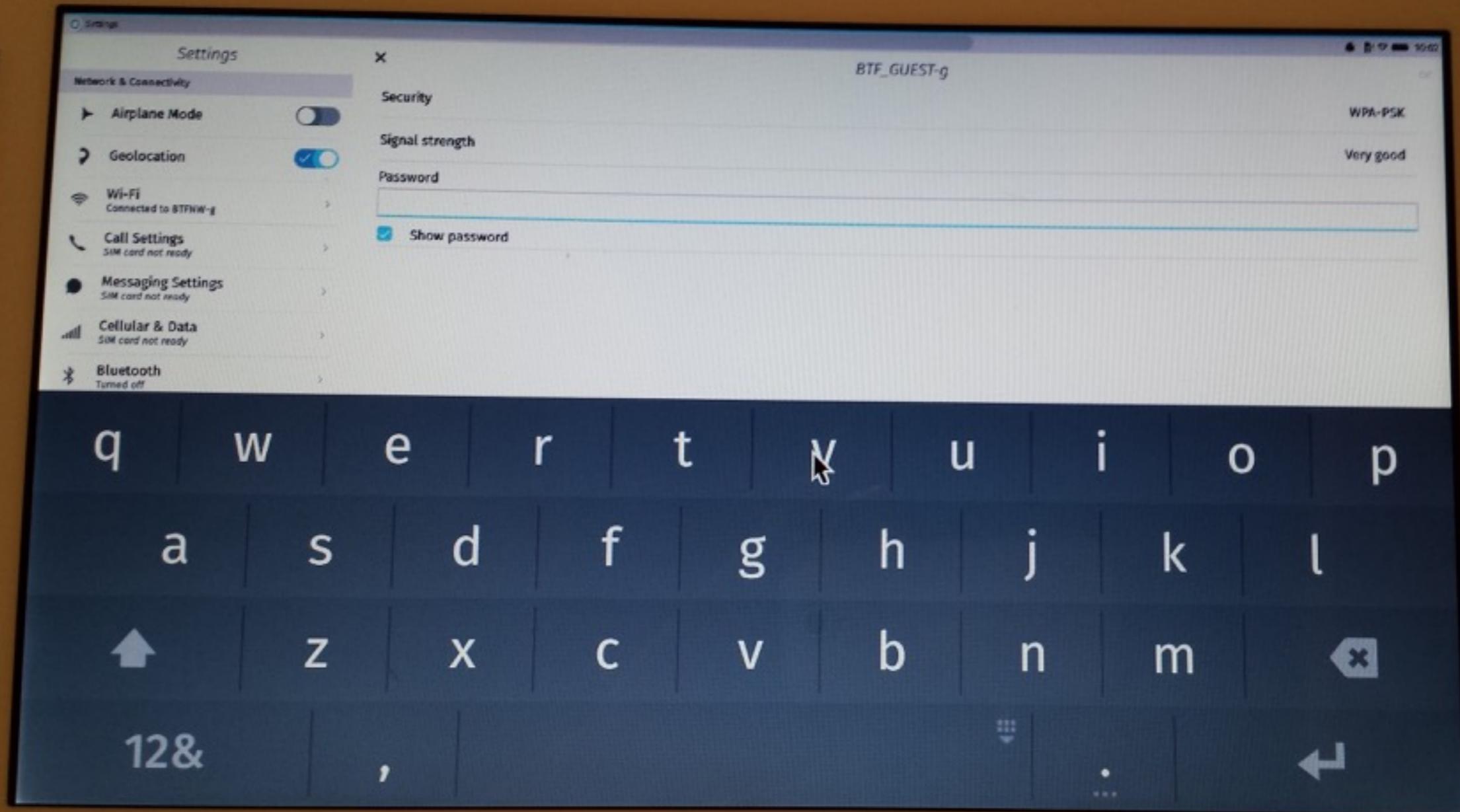
完成図

The screenshot displays the Firefox WebIDE interface for a project named "http-request-example". The browser title is "Firefox WebIDE: http-request-example". The interface is divided into several sections:

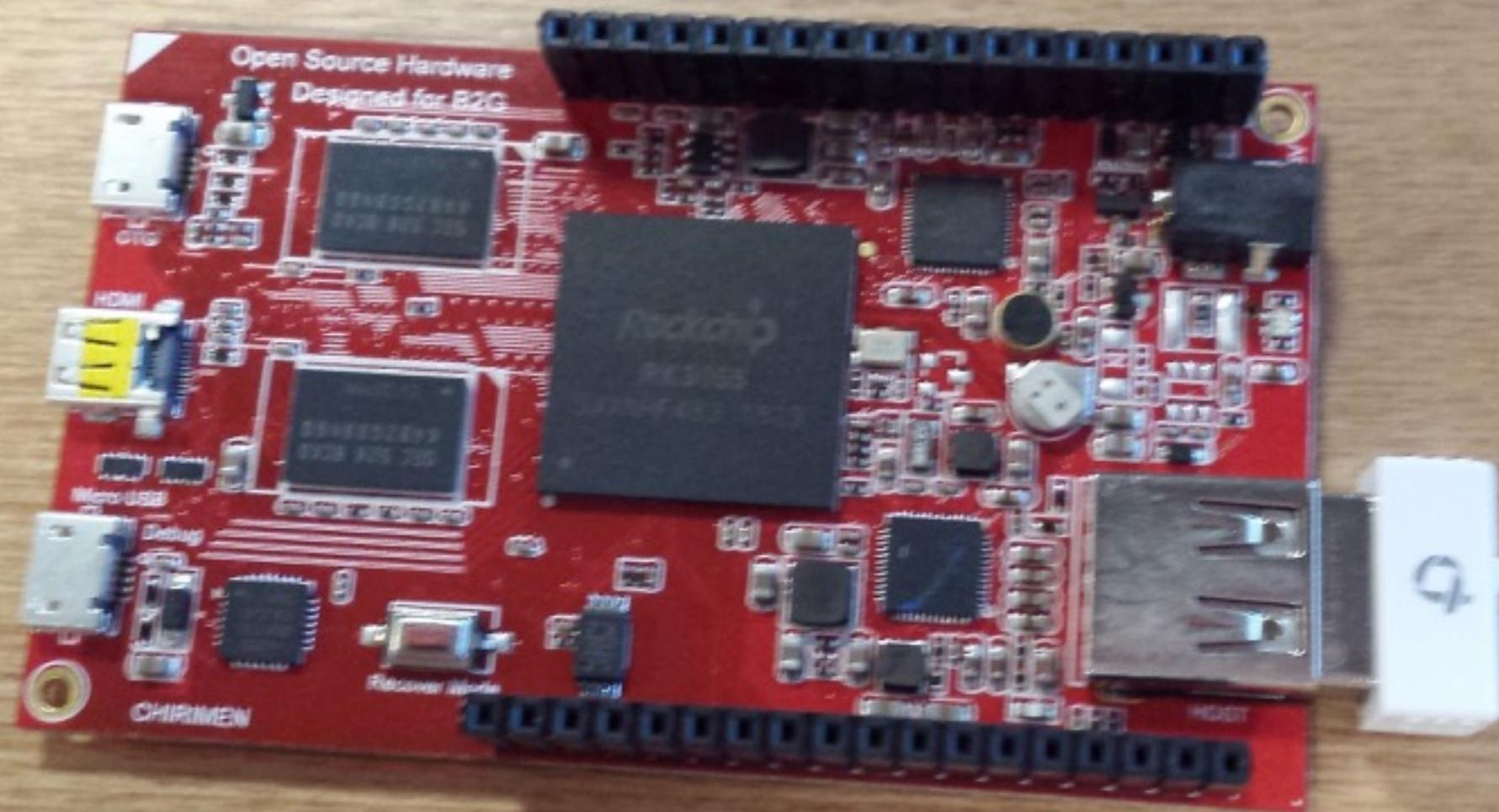
- Left Panel:** Contains navigation options like "新規アプリ...", "パッケージ型アプリを開く...", and "マイプロジェクト". The "http-request-example" project is selected and highlighted in blue.
- Project Explorer:** Shows the file structure of the project, including folders like "bower_components/" and "icons/", and files like "app.js", "axios-sysxhr.js", "bower.json", "index.html", and "manifest.webapp".
- Project Details:** Displays information for the selected project:
 - Icon:** A pink dashed circle.
 - Name:** http-request-example
 - Type:** PACKAGED PRIVILEGED
 - 説明 (Description):** A simple HTTP request
 - 場所 (Location):** /Users/btf/src/FirefoxOS/chirimen/http-request-example
 - App ID:** app://4f214941-9a52-224c-bbc5-7286ab33c11c/manifest.webapp
- Right Panel:** Shows system information and device options, including "USB デバイス" (with "CHIRIMEN" selected), "WI-FI デバイス", and "シミュレータ".
- Bottom Panel:** Contains development tools such as "インスペクタ", "コンソール", "デバッガ", "スタイルエ...", "パフォーマンス...", and "ネットワーク". The "コンソール" (Console) is active, showing the following output:

```
Hello World!
Object { success: true, status: 1, message: "Succeed HTTP request!" }
```


電源を入れてSettingsからWifiの設定を行う



USBハブ、ディスプレイを取り外し
GW-USNano2をCHIRIMENに取り付ける



USBケーブルでPCとつなぐ



電源ケーブルをつないで電源をいれる



http-request-exampleプロジェクトを読み込む

Firefox WebIDE: http-request-example

プロジェクトを削除 有効

http-request-example
PACKAGED PRIVILEGED

説明
A simple HTTP request

場所
/Users/btf/src/FirefoxOS/chirimen/http-request-example

App ID
app://4f214941-9a52-224c-bbc5-7286ab33c11c/manifest.webapp

USB デバイス
CHIRIMEN

Wi-Fi デバイス

シミュレータ
+ シミュレータをインストール

その他
リモートランタイム
Chrome Desktop
Safari, Firefox, and other WebViews on iOS
ランタイム情報
許可設定一覧
端末の設定
端末オプション
スクリーンショット撮影
切断

http-request-exa...
blink-led
motion-sensor-led
ランタイムアプリ
メインプロセス
Bluetooth Manager
Bookmark
Browser
BuddyUp
Bugzilla Lite
Built-in Keyboard
Calculator
Calendar
Callscreen
Camera
Clock
Communications
Default Home Scr...
Default Theme

bower_components/
icons/
app.js
axios-sysxhr.js
bower.json
index.html
manifest.webapp

実行するとHTTPレスポンスが返ってくる

The screenshot displays the Firefox WebIDE interface for a project named "http-request-example". The main panel shows the project details, including the description "A simple HTTP request" and the App ID "app://4f214941-9a52-224c-bbc5-7286ab33c11c/manifest.webapp". The console at the bottom shows the output of the application, which includes "Hello World!" and a JSON object: "Object { success: true, status: 1, message: 'Succeed HTTP request!' }". The console also indicates the source files and line numbers: "app.js:11:3" and "app.js:17:5".

Firefox WebIDE: http-request-example

プロジェクトを削除 有効

http-request-example
PACKAGED PRIVILEGED

説明
A simple HTTP request

場所
/Users/btf/src/FirefoxOS/chirimen/http-request-example

App ID
app://4f214941-9a52-224c-bbc5-7286ab33c11c/manifest.webapp

USB デバイス
CHIRIMEN

WI-FI デバイス

シミュレータ
+ シミュレータをインストール

その他
リモートランタイム
Chrome Desktop
Safari, Firefox, and other WebViews on iOS
ランタイム情報
許可設定一覧

インスペクタ コンソール デバッガ スタイルエ... パフォーマ... ネットワーク

● ネットワーク ● CSS ● JS ● セキュリティ ● ログギング ● サーバ

Hello World! app.js:11:3
Object { success: true, status: 1, message: "Succeed HTTP request!" } app.js:17:5

プログラムの要点

```
1
2 window.addEventListener("load", function() {
3   console.log("Hello World!");
4   var url = 'http://www.bathtimefish.com/get.php?status=1';
5   let { spawn } = task;
6   spawn(function() {
7     // new XMLHttpRequest({systemXHR: true}); にした axios-sysxhr.js
8     var res = yield axios.get(url);
9     console.log(res.data);
10  });
11 });
```

リクエストするURL

リクエストして
レスポンスを取得する

axios

https://github.com/mzabriskie/axios



This repository Search

Pull requests Issues Gist



mzabriskie / axios

Watch 124

Star 5,561

Fork 300

Code

Issues 22

Pull requests 7

Wiki

Pulse

Graphs

Promise based HTTP client for the browser and node.js

524 commits

6 branches

27 releases

58 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download



nickuraltsev Releasing 0.14.0

Latest commit c963486 15 hours ago

dist	Releasing 0.14.0	15 hours ago
examples	Splitting progress event handlers into upload and download handlers	5 days ago
lib	Adding support for http_proxy and https_proxy environment variables	4 days ago
sandbox	Adding custom URL support in sandbox client	2 years ago
test	Adding proxy to TypeScript definitions	3 days ago
.eslintrc	Updating dependencies	6 months ago
.gitignore	Adding TypeScript definitions for interceptors	12 days ago
.npmignore	Adding TypeScript definitions for interceptors	12 days ago

実はサンプルでは公式のコードとはちがうやり方をしています

Example

Performing a GET request

```
// Make a request for a user with a given ID
axios.get('/user?ID=12345')
  .then(function (response) {
    console.log(response);
  })
  .catch(function (error) {
    console.log(error);
  });
```

```
var url = 'http://www.bathtimefish.com/get.p
let { spawn } = task;
spawn(function() {

  var res = yield axios.get(url);
  console.log(res.data);

});
```

ずっと便利なライブラリを使っていました。

<http://taskjs.org/>

mozilla

Fork me on GitHub



generators + promises = tasks

task.js is an experimental library for ES6 that makes sequential, blocking I/O simple and beautiful, using the power of JavaScript's new `yield` operator.

Tasks are interleaved like threads, but they are **cooperative** rather than **pre-emptive**: they block on **promises** with `yield`. Here's an example using **jQuery**:

```
spawn(function*() {
  var data = yield $.ajax(url);
  $('#result').html(data);
  var status = $('#status').html('Download complete.');
```

```
  yield status.fadeIn().promise();
  yield sleep(2000);
  status.fadeOut();
});
```

task.js works with any framework or library that uses the **Promises/A** spec.

Quick Start

The power of **task.js** comes from ES6 generators, which are **currently only available in Firefox**. You can try it out in Firefox, although there are a few remaining incompatibilities with the current ES6 spec (which is still a work in progress).

Here's a "hello world" of tasks that will work in Firefox:

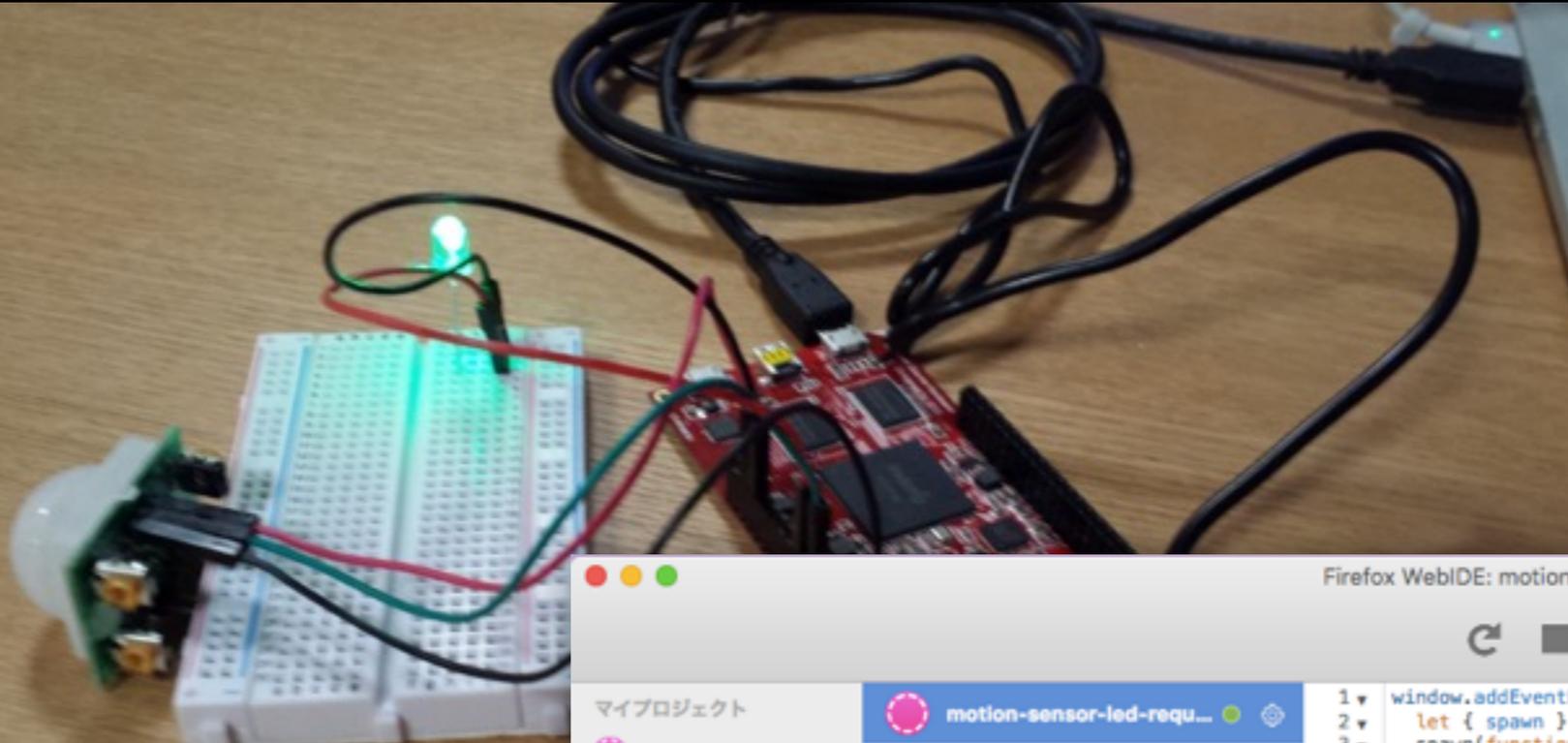
應用編

いままでのを組み合わせてみよう

問題.

人が近づくとLEDが光って、
焦電センサーから得られた値を
HTTPリクエストし、
成功レスポンスを得る
アプリケーションをつくりなさい。

完成図



Firefox WebIDE: motion-sensor-led-request

CHIRIMEN

マイプロジェクト

- blink-led
- LEDblink
- MotionSensor
- http-request-exa...
- motion-sensor-led
- motion-sensor-le...

ランタイムアプリ

- メインプロセス
- Bluetooth Manager
- Bookmark
- Browser

USB デバイス

- CHIRIMEN

Wi-Fi デバイス

- シミュレータ
- シミュレータをインストール

その他

- リモートランタイム
- Chrome Desktop
- Safari, Firefox, and other WebViews on iOS
- ランタイム情報
- 許可設定一覧
- 端末の設定

```
1 window.addEventListener("load", function() {
2   let { spawn } = task;
3   spawn(function() {
4     const gpio = yield navigator.requestGPIOAccess();
5     // https://chirimen.org/docs/en/board_connectors.html
6     const sensor = gpio.ports.get(196); // Motion Sensor Data Pin: CN1 9
7     const led = gpio.ports.get(198); // LED Anode Pin: CN1 9
8     yield sensor.export("in");
9     yield led.export("out");
10    setInterval(() => {
11      sensor.onChange = v => {
12        console.info(v);
13        led.write(v);
14        getRequest(v);
15      };
16    }, 1000);
17  });
18 });
19
20 var getRequest = function(v) {
21   let { spawn } = task;
22   var url = 'http://www.bathimefish.com/get.php?status=' + v;
23   spawn(function() {
24     var res = yield axios.get(url);
25     console.log(res.data);
26   });
27 }
```

インスペクタ コンソール デバッガ スタイルエ... パフォーマ... ネットワーク

ネットワーク CSS JS セキュリティ ログギング サーバ

出力を絞り込み

```
1 Object { success: true, status: 1, message: "Succeed HTTP request!" } app.js:12:9
0 Object { success: true, status: 0, message: "Succeed HTTP request!" } app.js:25:5
0 Object { success: true, status: 0, message: "Succeed HTTP request!" } app.js:12:9
0 Object { success: true, status: 0, message: "Succeed HTTP request!" } app.js:25:5
```

うまくできた？

ちょっと微妙にひねらないといけないところがありますw

最新のWeb技術を使って、
ハードウェアの制御な時代です。

Webの技術はまだまだ発達して
あらゆるジャンルにアプローチしていくでしょう。
その中から興味のある技術を積極的に学んでいって
将来のために役立てていってください。

Thanks !