

IoT活用

目次 (1)

1-3章はE-Learning

第4章 組込ボードの基礎

4-1. IoTでよく使用される組込ボード	7
4-2. Arduinoとは	10
4-3. Arduino IDEのダウンロードとインストール	15
4-4. Arduinoのメニュー画面	16
4-5. Arduinoのスケッチ例と動作検証	17
4-6. Arduinoとブレッドボードによる配線	18
4-7. ブレッドボードの通電箇所	19
4-8. Arduinoにおける回路設計	21
4-9. Arduinoにおけるオームの法則	27
演習 1 Arduinoを使った電気回路の設計	28

目次 (2)

第5章 組込ボードとセンサ

5-1. センサ	30
5-2. 環境センサ	31
5-3. 入力モジュール	34
5-4. 出力モジュール	37
演習2 Arduinoとセンサを使った回路設計	39

第6章 IoTのセキュリティ

6-1. IoTデバイスを標的としたマルウェア	41
6-2. Miraiウィルス	42
6-3. IoTセキュリティガイドライン	45
6-4. IoTセキュリティガイドラインの目的	46
6-5. サービス提供者のための指針	47
6-6. 一般利用者のための指針	48

目次 (3)

第7章 IoTプラットフォームを使ったデータ通信

7-1. IoTプラットフォームの例	50
7-2. IoTプラットフォーム sakura.io	52
7-3. sakura.ioの特徴	53
7-4. さくらのLTE通信モジュール	54
7-5. さくらの通信モジュールの位置付け	57
7-6. sakura.ioの物理的構成	59
7-7. IoTシステムの物理的構成	60
7-8. sakura.io 料金と通信ポイント	62
7-9. ポイント管理例	63
7-10. ライブラリとマニュアル	64
7-11. ログインとプロジェクト	65
7-12. 基本的な考え方	67
7-13. コード例	68

目次 (4)

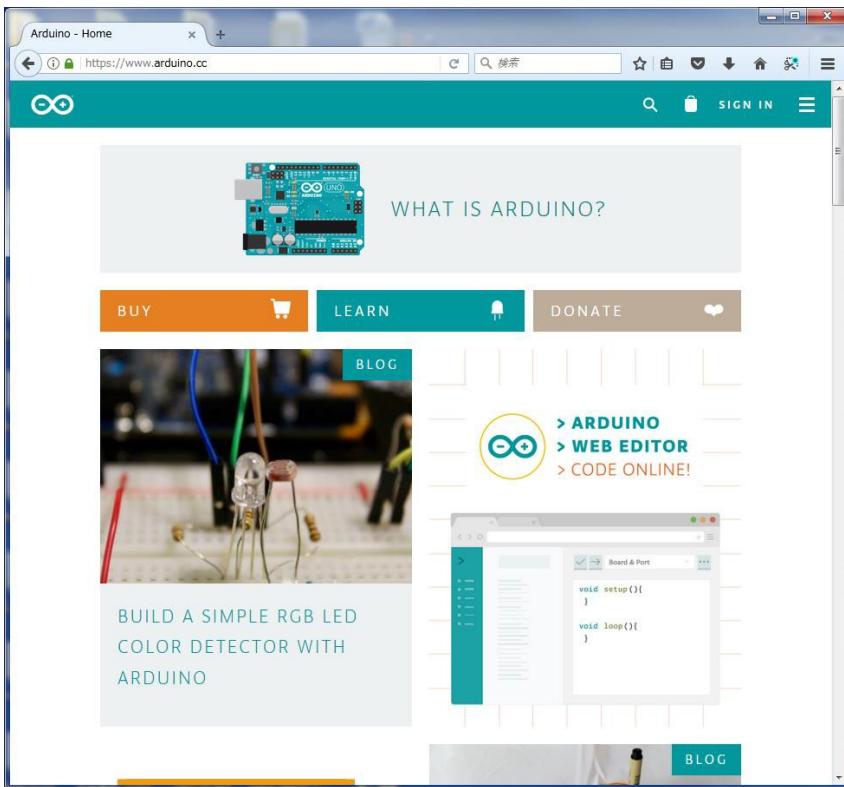
第7章 IoTプラットフォームを使ったデータ通信

7-14. 連携サービス	69
7-15. WebSocket	70
7-16. データ形式	71
7-17. JSON例（データが単数）	72
7-18. JSON例（データが複数）	73
7-19. 連携サービスの作成	74
7-20. WebSocketのURLとToken	75
7-21. JSON例（データが単数）	76
7-22. JSON例（データが複数）	77
7-23. 開発ツール Node-RED	78
演習3 さくらLTEモジュールの回路設計と利活用	80
演習4 総合演習	81

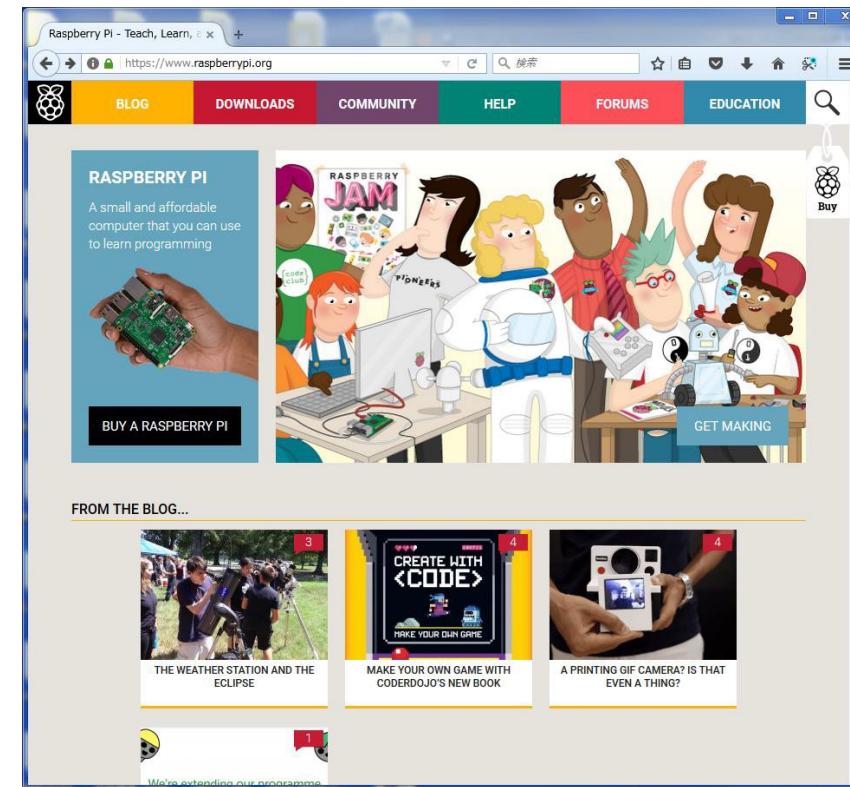
第4章 組込ボードの基礎

4-1. IoTで使用される組込ボードの例

Arduino

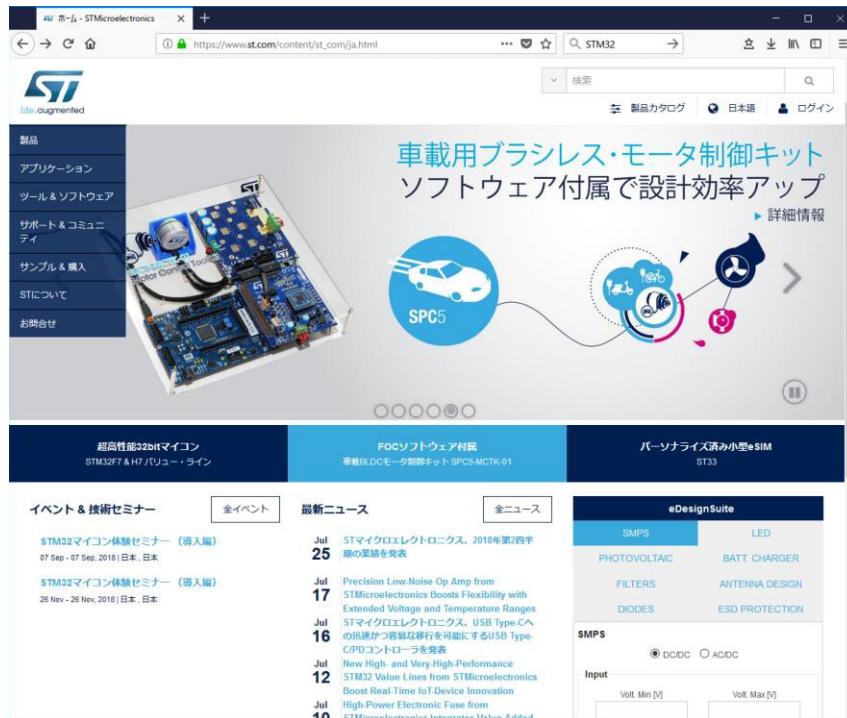


Raspberry Pi

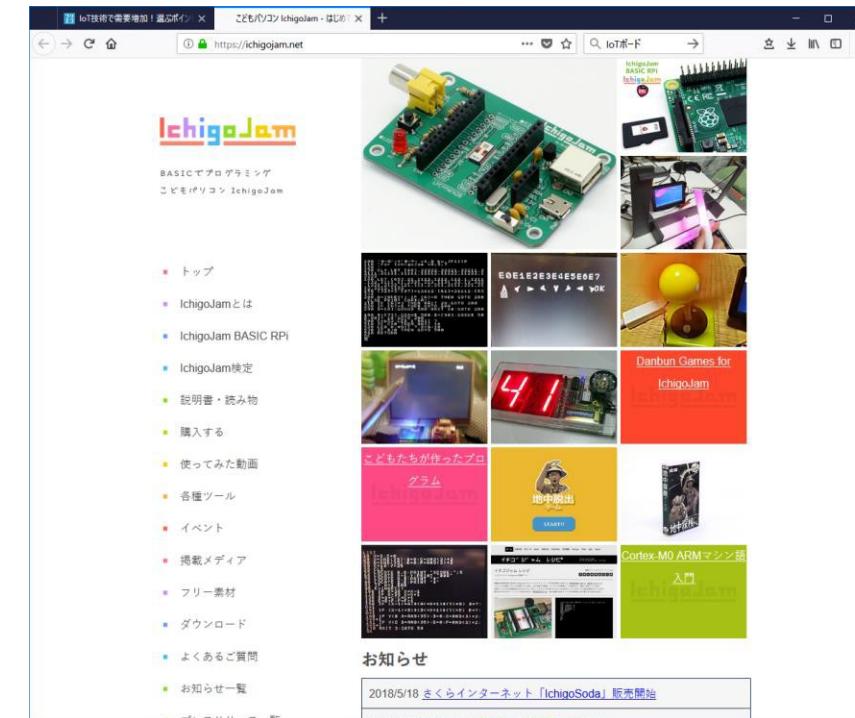


IoTで使用される組込ボードの例（続き）

STM32



IchigoJam



IoTで使用される組込ボードの例（続き）

単独で開発が可能な**Raspberry Pi**（※初期設定時にPCが必要）

シングルボードコンピュータ≠PC

OS : Linux (DebianベースのRasbianなど)

ディスプレイやキーボードをつないでPCと同じように開発

様々なプログラミング言語が利用可 (C、C++、Python、Node.jsなど)

良くも悪くもPCと同じ開発環境

母艦（PC）からプログラムを書き込む**Arduino**

ワンボードマイコン

OS非搭載（その分、省電力）

母艦のPCにインストールした「Arduino IDE」からプログラムを書き込む

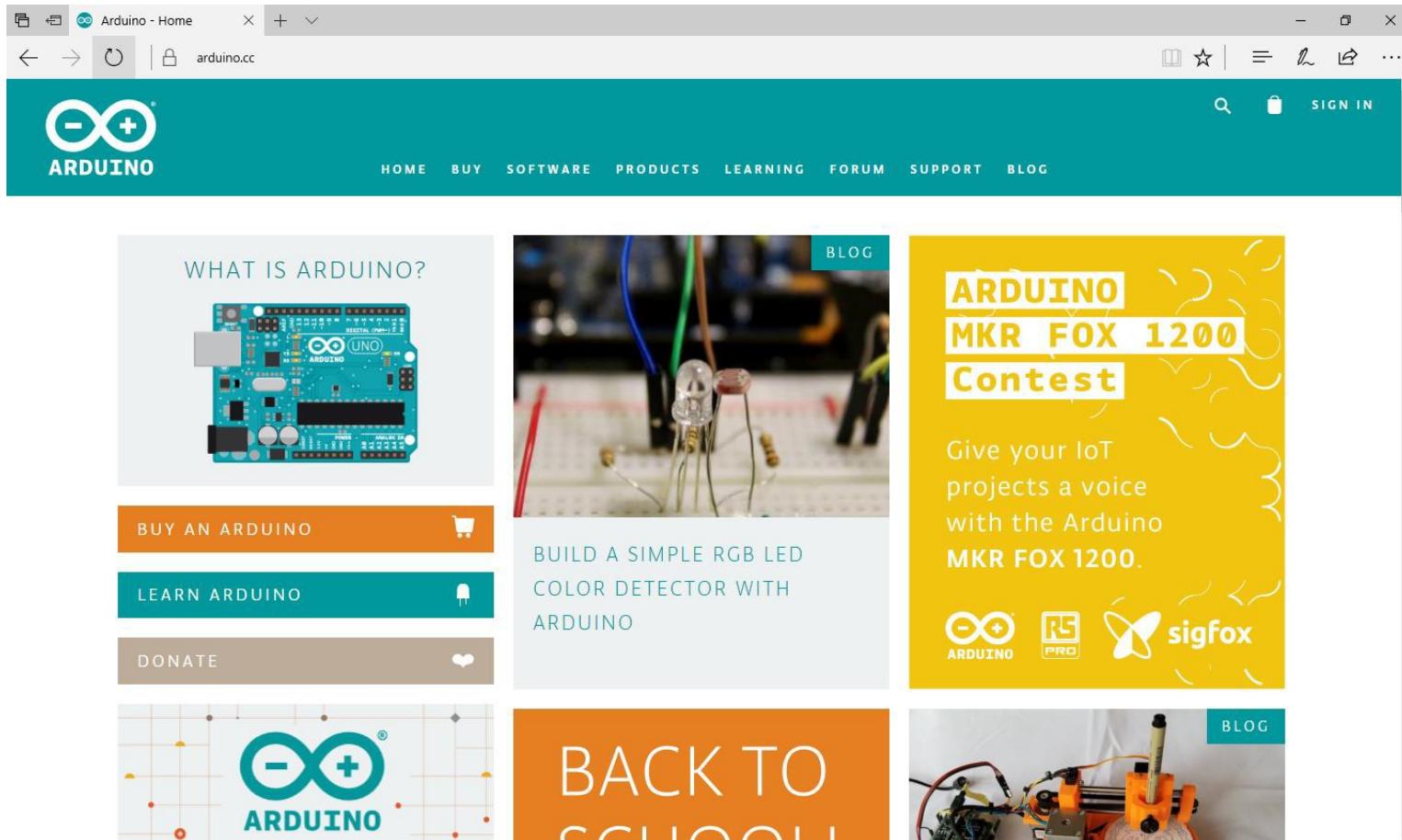
C言語風のArduino言語を利用

ライブラリを読み込めば簡単に実現ができる開発環境



ヘッダーを意識せずにTCP/IP通信が可能

4-2. Arduino



(<https://www.arduino.cc/>)

Arduino (続き)

アルドウイーノ

2005年に、Massimo Banzi・David Mellis（当時はイタリアのIDIIの学生）、David Cuartiellesによってプロジェクトスタート、後にTom Igoeが加わる

- 派生元

- 2003年 IDII修士論文プロジェクトWiring (Hernando Barragán)
- 「Arduinoの語られざる歴史」より
<https://arduinohistory.github.io/ja.html>

一枚のプリント基盤の上に、電子部品と入出力がついたマイクロコンピュータ

- Processingベースの開発環境 (Javaアプリケーション)
- プログラミング言語 : C++風言語 (Arudino言語とも呼ばれる、元はWiring)

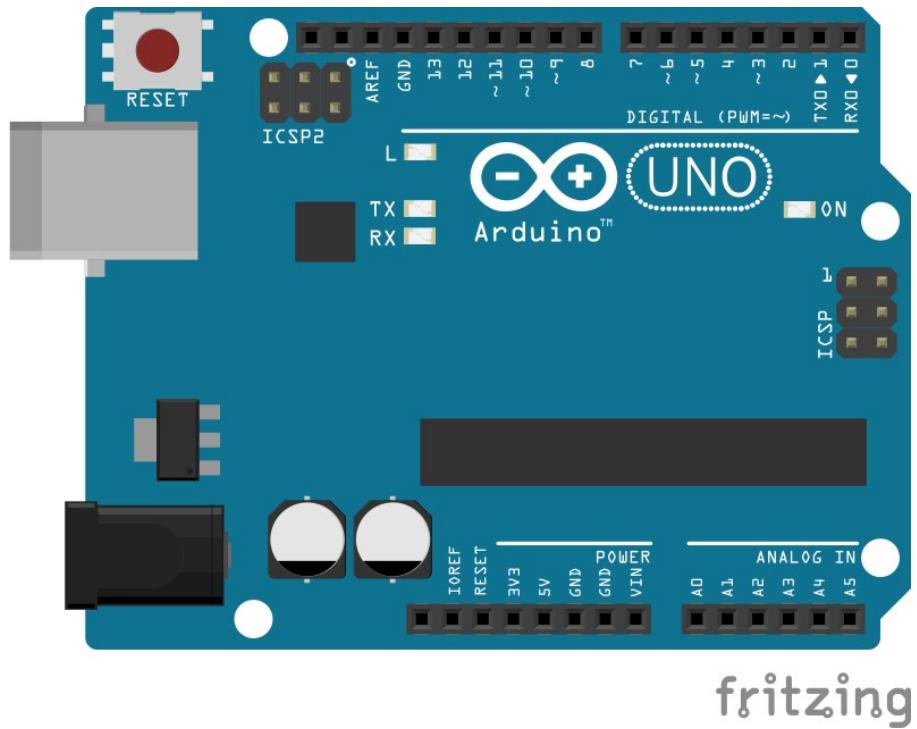
Arduino (続き)

The screenshot shows the Arduino Store website with a teal header. The header includes the Arduino logo, a search icon, a shopping bag icon, a location pin icon, and a 'SIGN IN' button. Below the header, the main navigation menu has links for HOME, STORE, SOFTWARE, EDUCATION, RESOURCES, COMMUNITY, and HELP. A prominent orange button labeled 'BOARDS & MODULES' is centered above the product grid. The left sidebar contains a breadcrumb trail: 'Store Home > Arduino > Boards & Modules'. It also features a 'NEW PRODUCTS' section with three categories: 'Most Popular' (indicated by a yellow dot), 'Special Offers' (indicated by a green dot), and 'Retired' (indicated by a grey dot). The 'ARDUINO' section is expanded, showing 'Boards & Modules' (indicated by a blue dot, which is highlighted with a yellow border), 'Shields', 'Kits', 'Accessories', 'Spare Parts', and 'Retired'. Other sections like 'Arduino Education', 'Goodies', 'Components', 'Other Brands', and 'Books & Manuals' are also listed. On the right, a 'Sort By' dropdown is set to 'FEATURED'. The main content area displays six Arduino boards in a 2x3 grid. The top row includes the Arduino Uno Rev3 (\$22.00), Arduino Nano Every (\$9.90), and Arduino Nano 33 BLE Sense with headers (\$31.50). The bottom row includes an Arduino Uno in a clear acrylic case, an Arduino Nano 33 BLE Sense (indicated by a 'PRE-ORDER' badge), and another Arduino Nano 33 BLE Sense board.

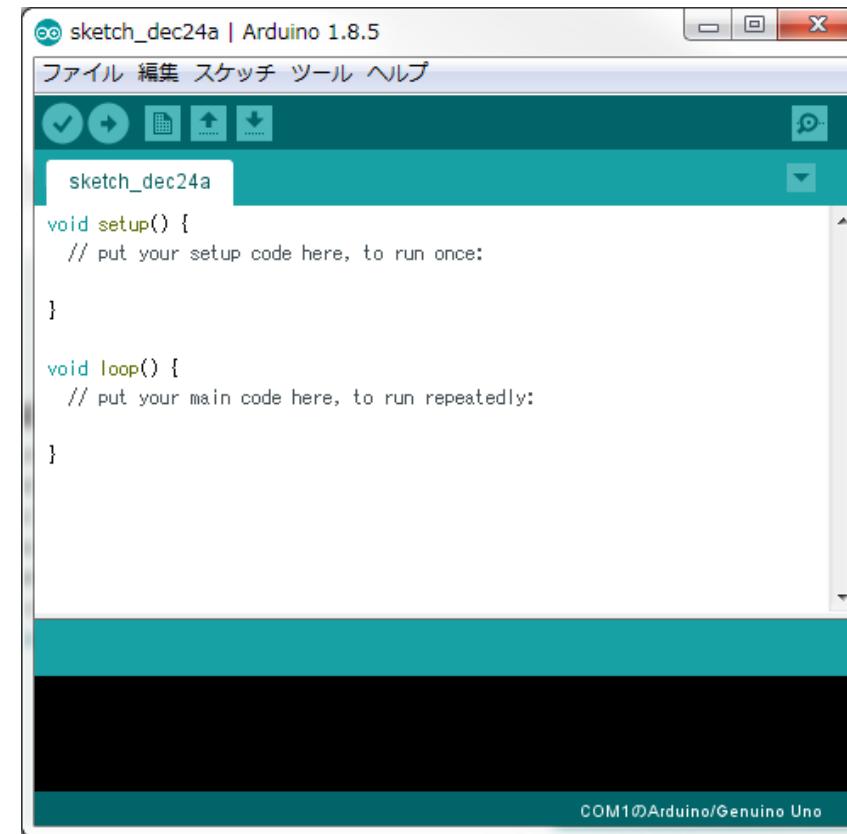
<https://store.arduino.cc/usa/arduino/boards-modules>

Arduino (続き)

Arduinoのプリント基板



Arduinoの開発環境(IDE)



Arduino (続き)

IO

デジタルIO 0 ~ 13 (最大負荷 40 mA)

アナログIO 0 ~ 5

※～のあるデジタルピンはPWM (Plus Width Modulation : パルス幅変調) が使えるピンを表す。通常、3、5、6、9、10、11でPWM出力ができる。

電源・・・外部電源またはUSB経由で供給

3.3V出力 (最大負荷 50 mA, 一部 150 mA)

5V出力 (最大負荷 50 mA)

GND

電圧の基準 (0 V)

※電気が流れて帰ってくる場所のイメージ (下水)

4-3. Arduino IDEのダウンロードとインストール

Download the Arduino IDE

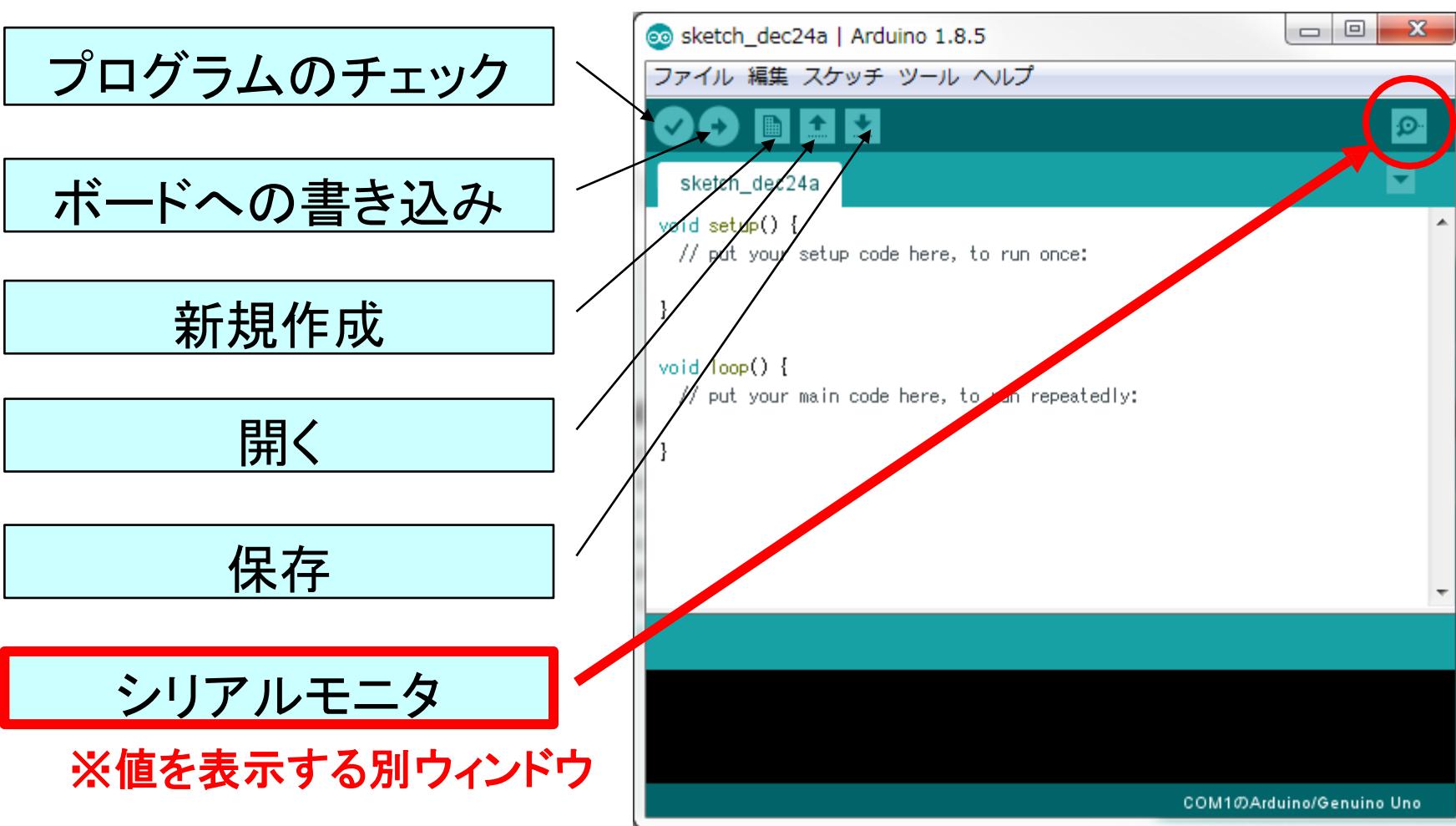
The screenshot shows the official Arduino website's download page. On the left, there is a large teal circular logo with a white infinity symbol containing a minus sign (-) on the left and a plus sign (+) on the right. To the right of the logo, the text "ARDUINO 1.8.5" is displayed in bold capital letters. Below this, a paragraph of text describes the Arduino Software (IDE) as open-source, Java-based, and compatible with Windows, Mac OS X, and Linux. It also mentions its use with Processing and other open-source software, and provides a link to the "Getting Started" page for installation instructions. On the right side of the page, there are download links for different operating systems. For Windows, there are links for "Windows Installer" (a .msi file) and "Windows ZIP file for non admin install". For Mac OS X, there is a link for "Mac OS X 10.7 Lion or newer". For Linux, there are links for "Linux 32 bits", "Linux 64 bits", and "Linux ARM". At the bottom right, there are links for "Release Notes", "Source Code", and "Checksums (sha512)".

インストール版・ZIP版

※通常、インストールや解凍をすればすぐに利用できる

※もし必要がある場合は、PCのデバイスマネージャーから
Arduinoのデバイスを更新する

4-4. Arduinoのメニュー画面

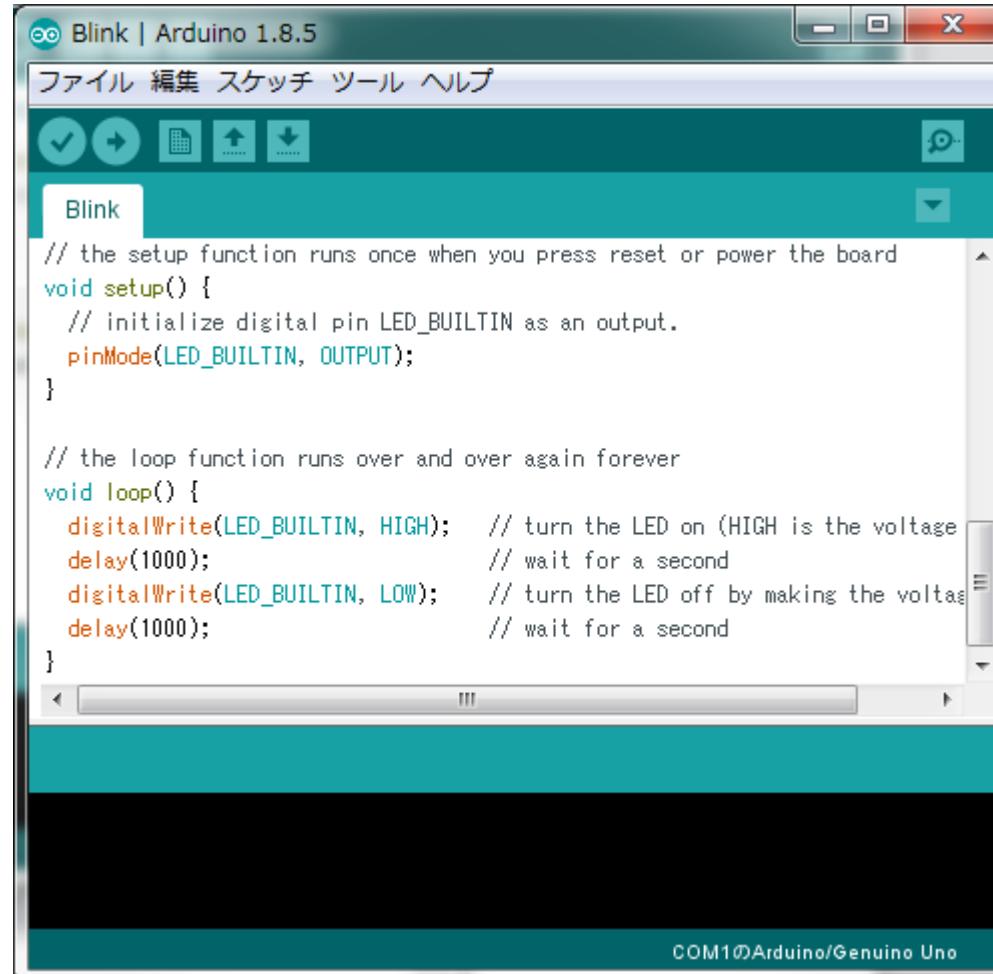


4-5. Arduinoのスケッチ例と動作検証

メニュー [ファイル] → [スケッチ例] → [Basics] → [Blink]

setup()
初期設定

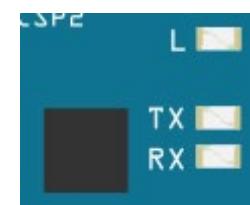
loop()
繰返し処理



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level)
  delay(1000);                         // wait for a second
  digitalWrite(LED_BUILTIN, LOW);       // turn the LED off by making the voltage level
  delay(1000);                         // wait for a second
}
```

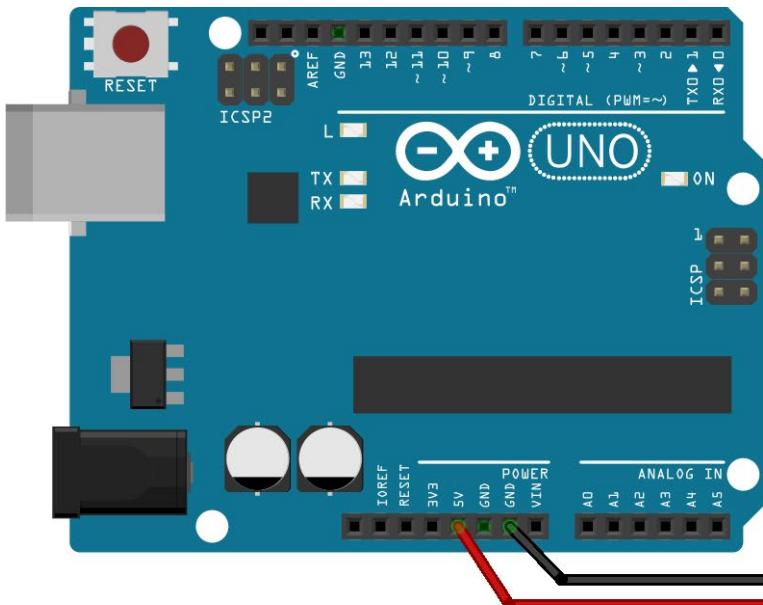
COM1のArduino/Genuino Uno



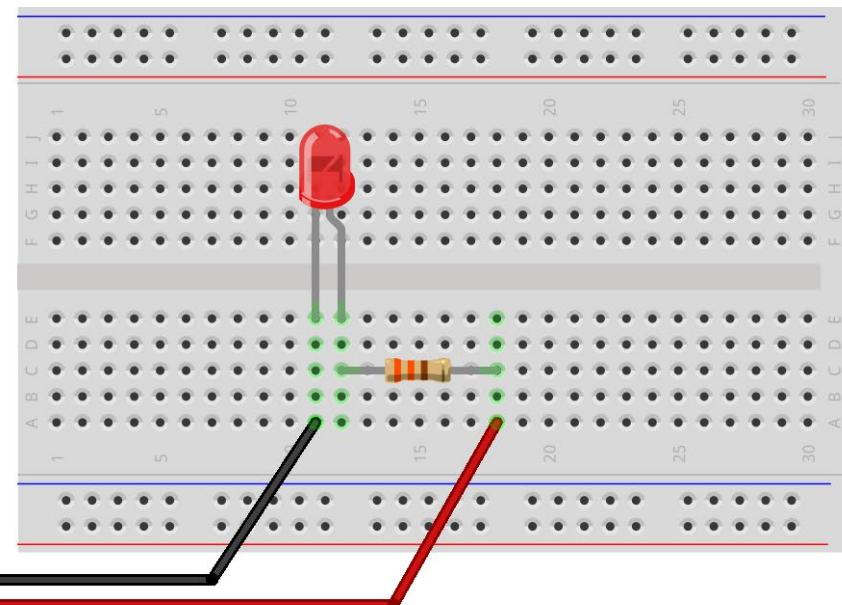
ボード上の
LEDが点滅
すればOK

4-6. Arduinoとブレッドボードによる配線

Arduino UNO

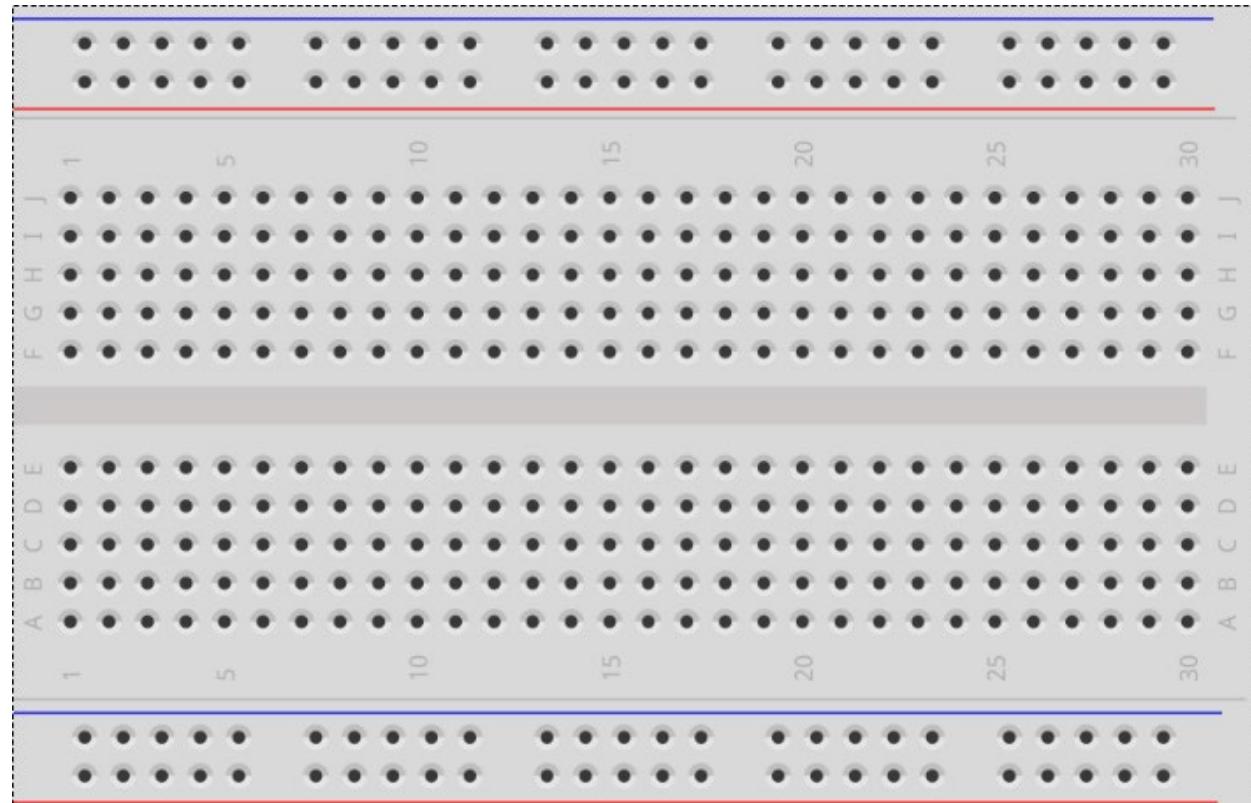


ブレッドボード



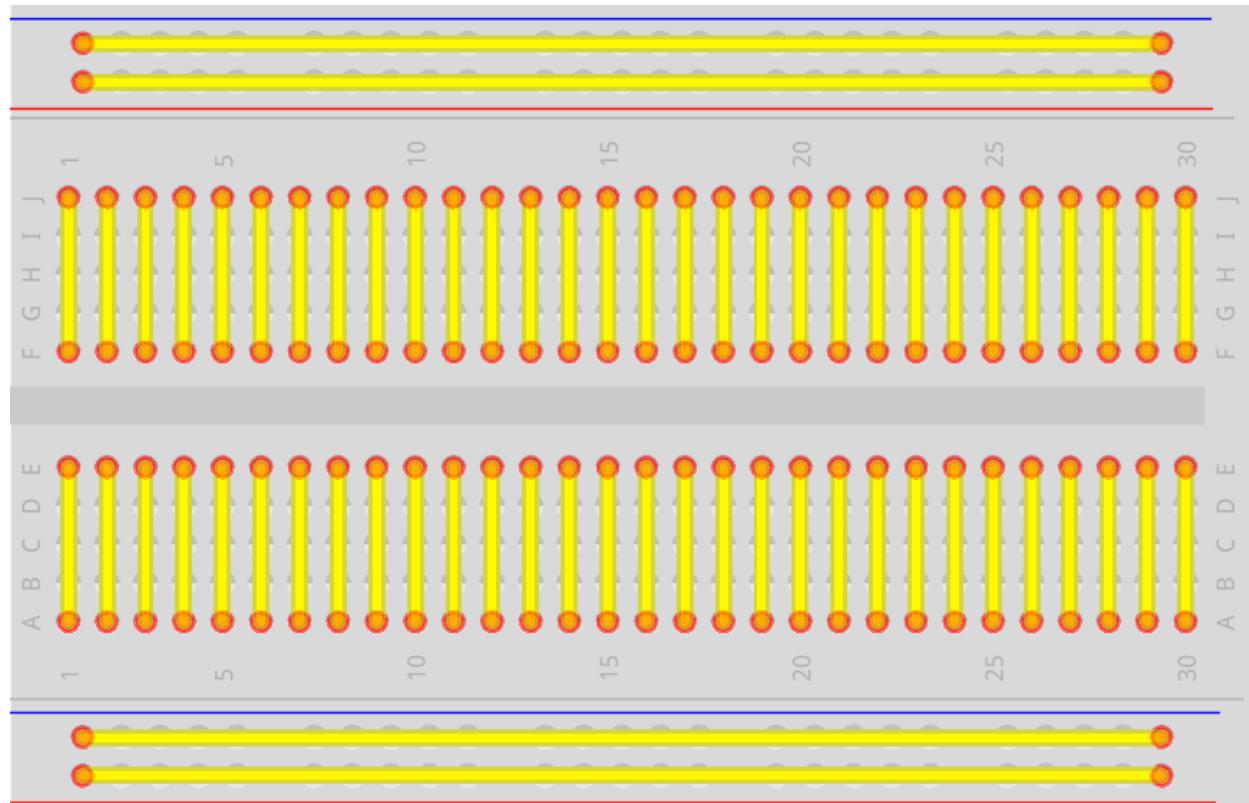
fritzing

4-7. ブレッドボードの通電箇所



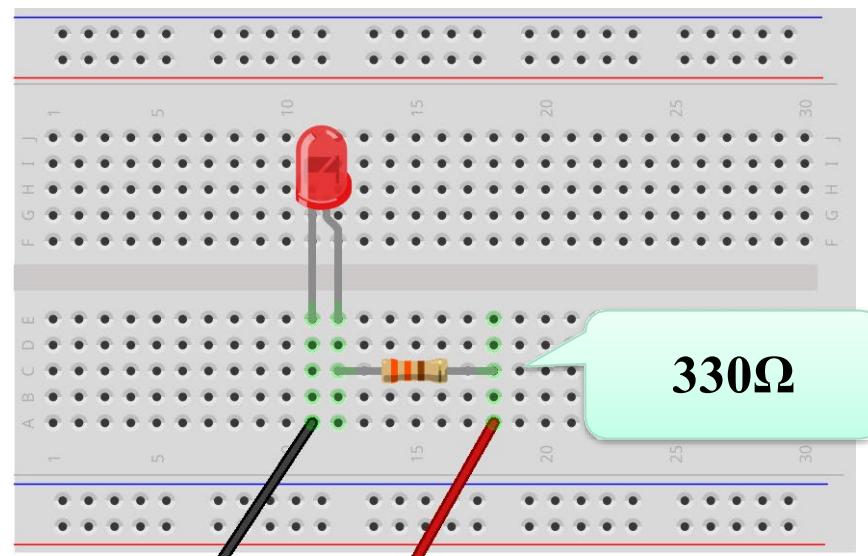
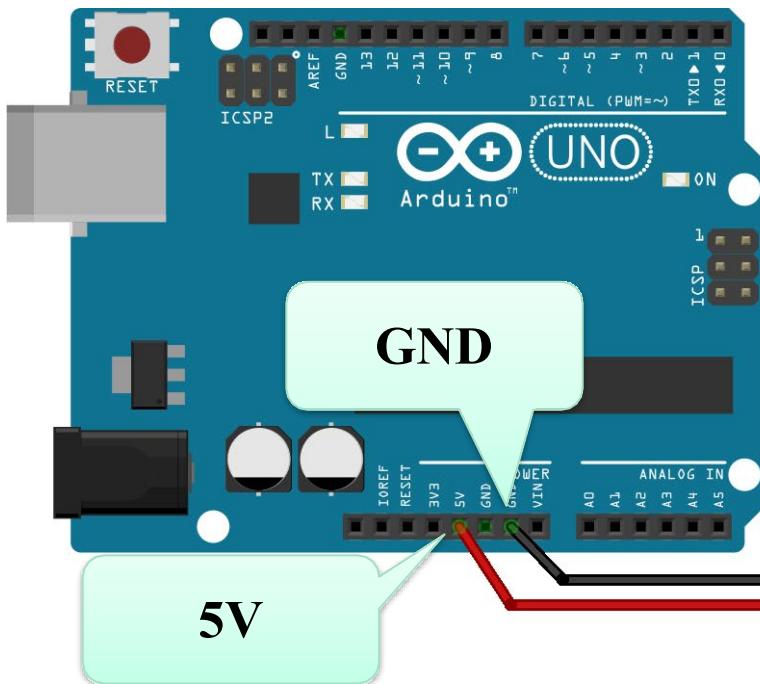
fritzing

ブレッドボードの通電箇所（続き）



fritzing

4-8. Arduinoにおける回路設計



fritzing

電圧～電流～抵抗

電圧 (E)

- 電気を押す力 (単位 : V)

電流 (I)

- 流れる電気の量 (単位 : A)

抵抗 (R)

- 電気の出力の穴の大きさ (単位 : Ω)

直列回路

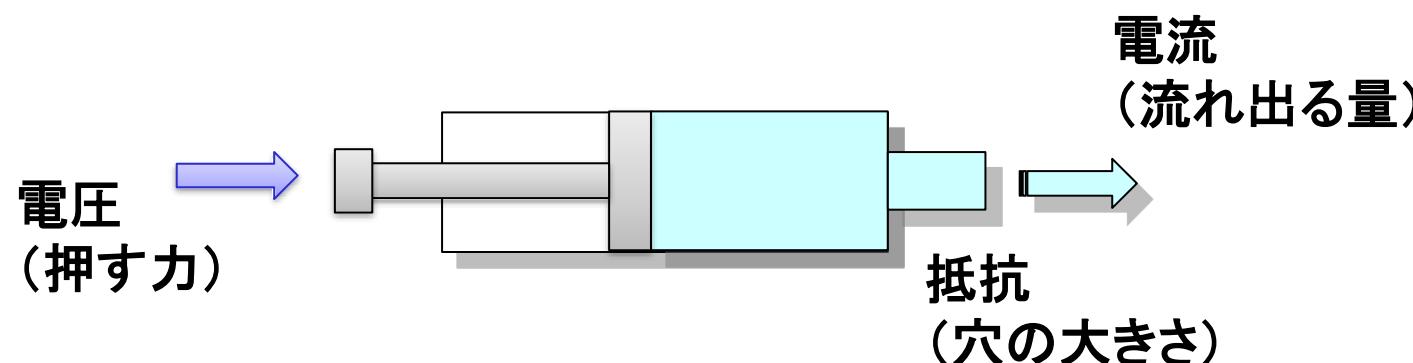
電流はどこも同じ値

電圧の和=全体の電圧

並列回路

電流の和=全体の電流

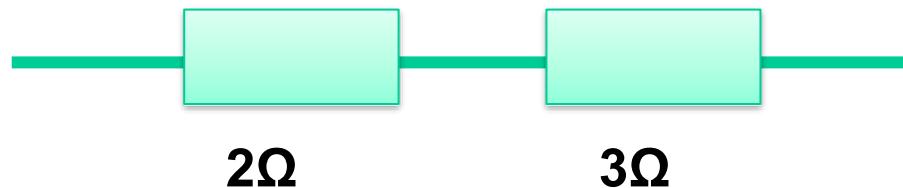
電圧はどこも同じ値



オームの法則

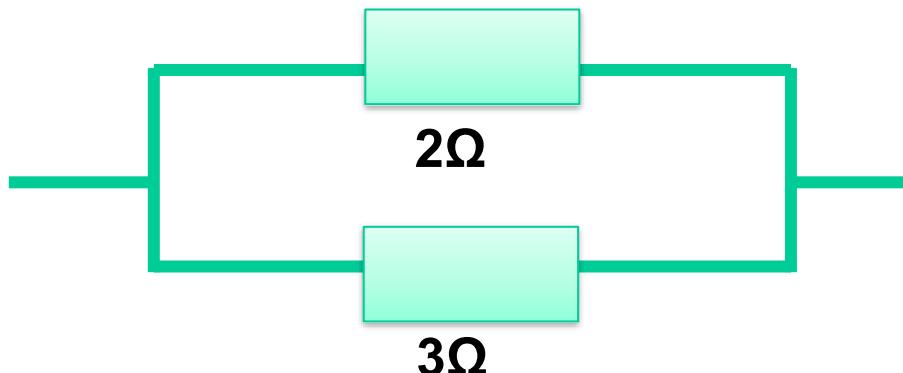
電圧 (V) = 電流 (A) × 抵抗 (Ω)

並列接続・・・和



$$2\Omega + 3\Omega = 5\Omega$$

並列接続・・・和分の積 (or 公式)



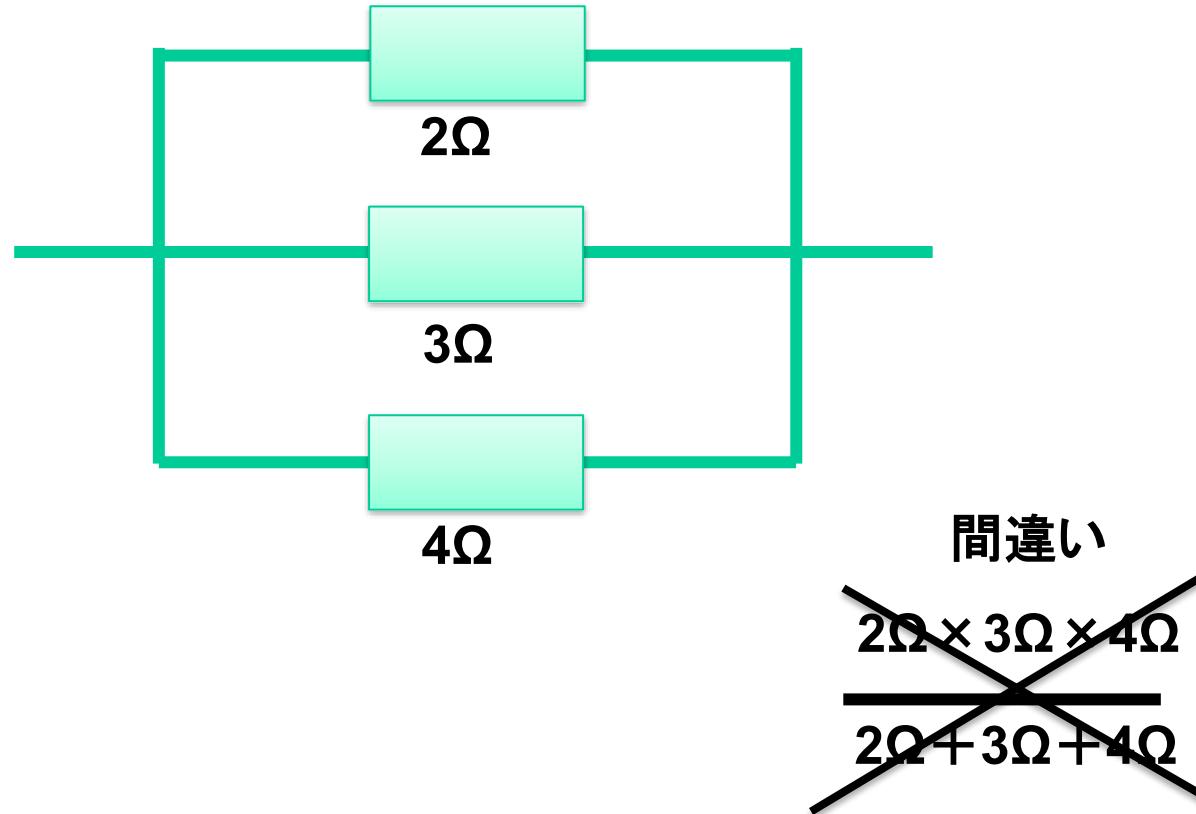
一本道が二本道に
増えるイメージ

$$\frac{2\Omega \times 3\Omega}{2\Omega + 3\Omega} = \frac{6\Omega}{5\Omega} = 1.2\Omega$$

※3つ以上の並列がある場合は2つずつ順番に計算

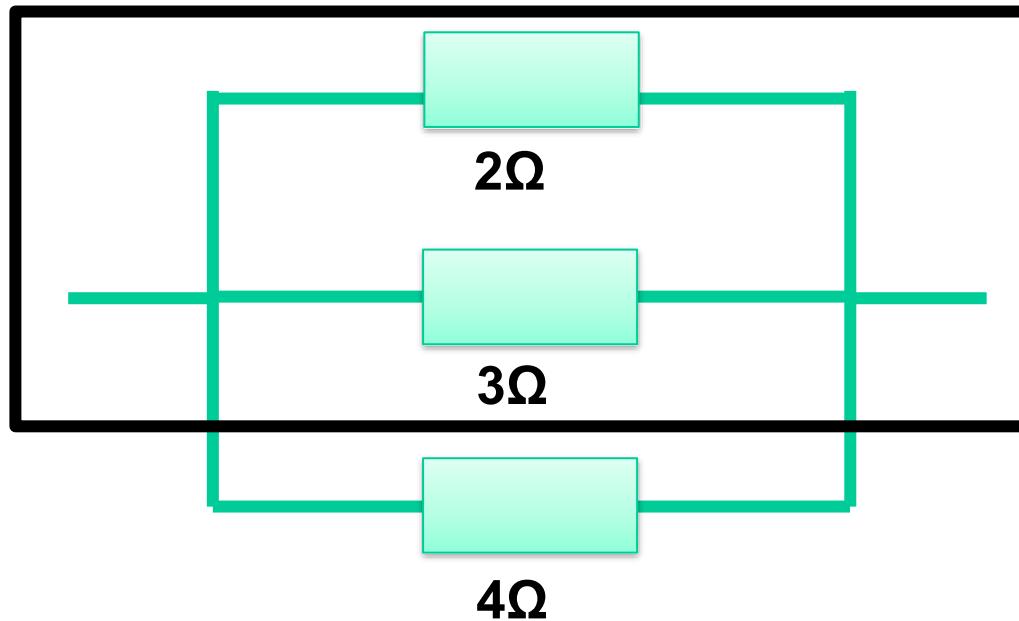
並列接続における和分の積

並列接続・・・3つ以上を和分の積で計算するのは間違い



並列接続における和分の積（続き）

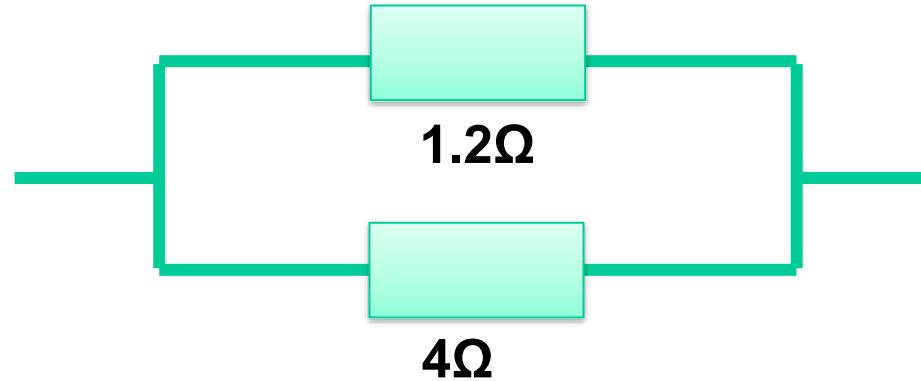
並列接続・・・まず一部分を和分の積で計算する



$$\frac{2\Omega \times 3\Omega}{2\Omega + 3\Omega} = \frac{6\Omega}{5\Omega} = 1.2\Omega$$

並列接続における和分の積（続き）

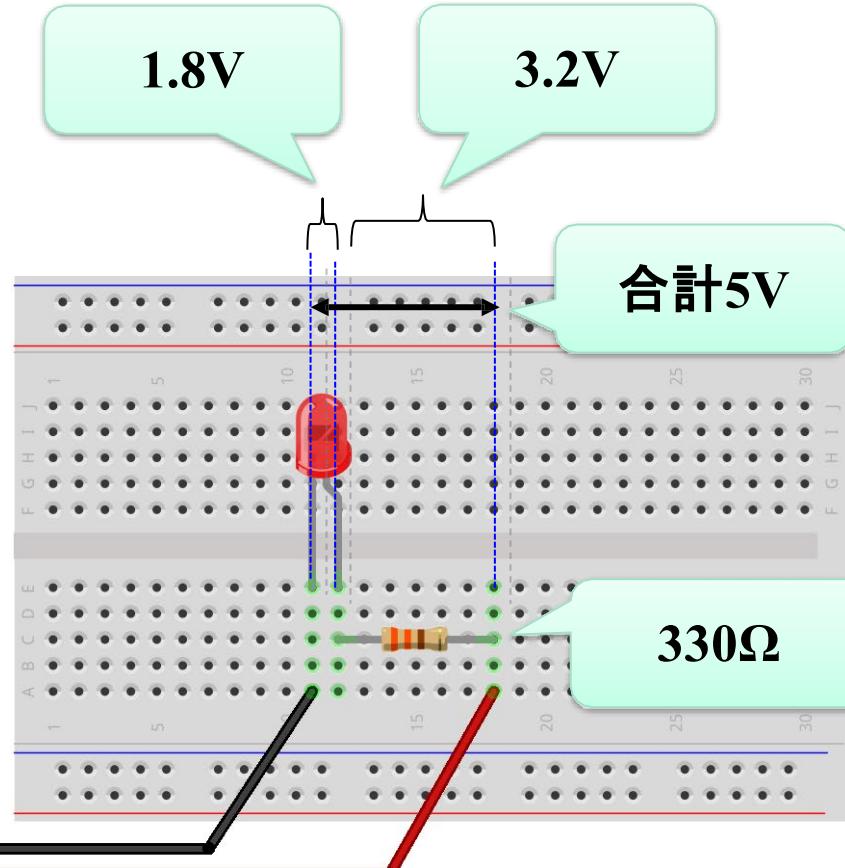
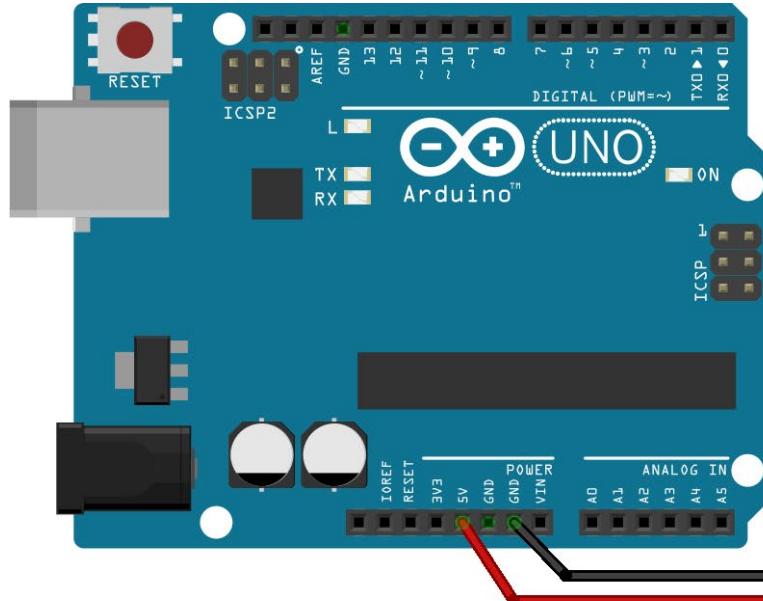
並列接続・・・残りの部分を2回目の和分の積で計算する



$$\frac{1.2\Omega \times 4\Omega}{1.2\Omega + 4\Omega} = \frac{4.8\Omega}{5.2\Omega} = 0.92\Omega$$

4-9. Arduinoにおけるオームの法則

電圧 (V) = 電流 (I) × 抵抗 (R)



赤色LEDの最大順電流は10~30mA前後
(ここでは10mAを流すものとする)

fritzing

演習 1 Arduinoを使った電気回路の設計

- ① LEDが点灯する回路
- ② スイッチでLEDをON・OFFする回路
- ③ スイッチを押したときにLEDをONする回路とプログラム
- ④ アナログ出力によるLED点灯
- ⑤ 応用編：LEDの種類や個数を変更

※作成した回路の回路図を描き抵抗値を書き込む

第5章 組込ボードとセンサ

5-1. センサ

環境センサ

入力モジュール

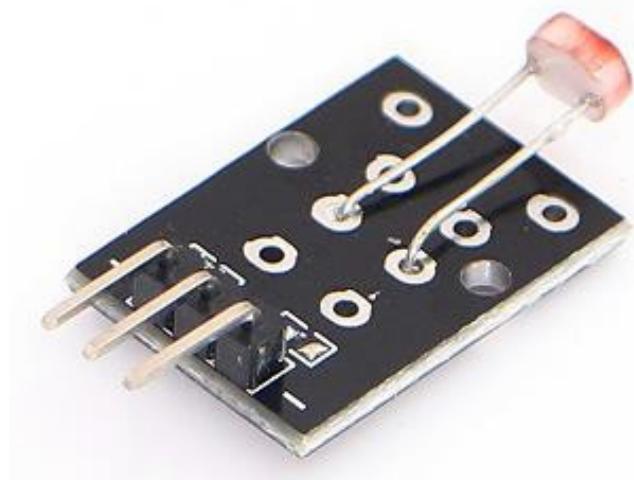
出力モジュール

※次スライド以降はKumanのデータファイルよりの抜粋

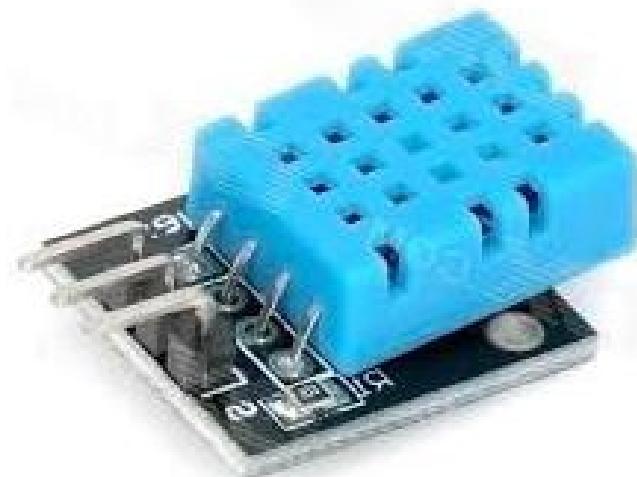
※Kumanのマニュアルは付属のCD-ROM内にある

5-2. 環境センサ

光センサ

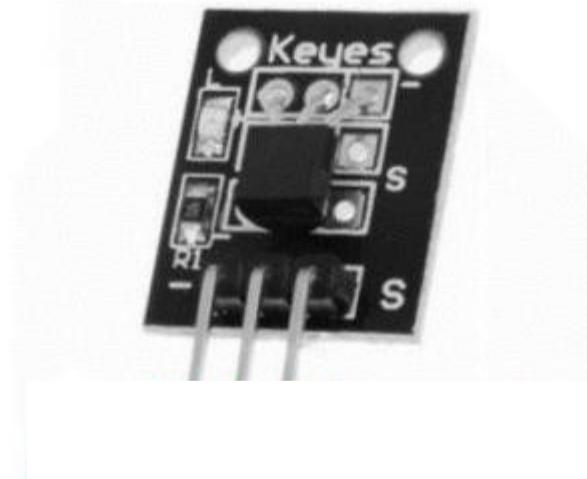


温湿度センサ

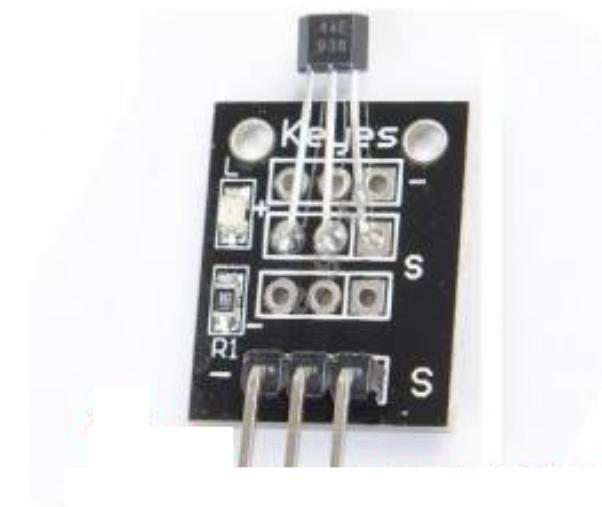


環境センサ（続き）

温度センサ

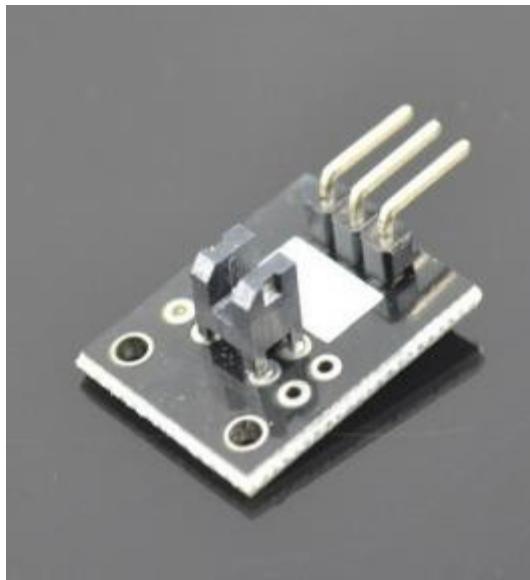


磁場センサ



環境センサ（続き）

光遮断センサ



その他

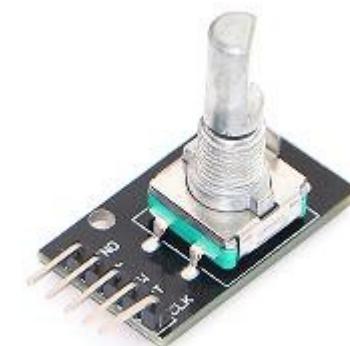
- ・ アナログ磁場センサ
 - ・ アナログ温度センサ
 - ・ 地磁気センサ
 - ・ 超音波センサ
 - ・ 赤外線センサ
- など

5-3. 入力モジュール

ジョイスティック

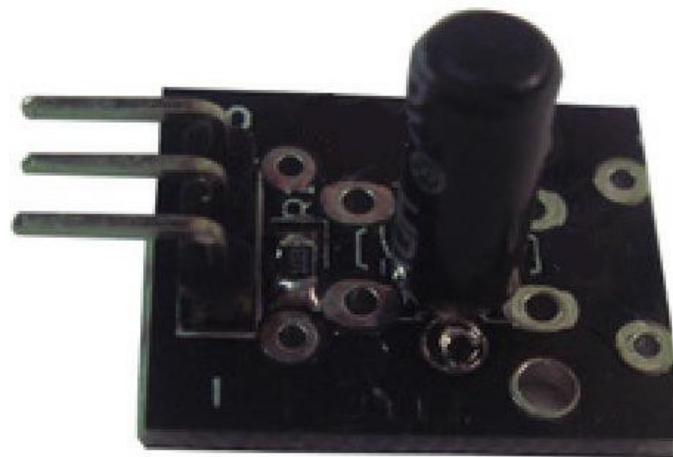


ロータリーエンコーダ

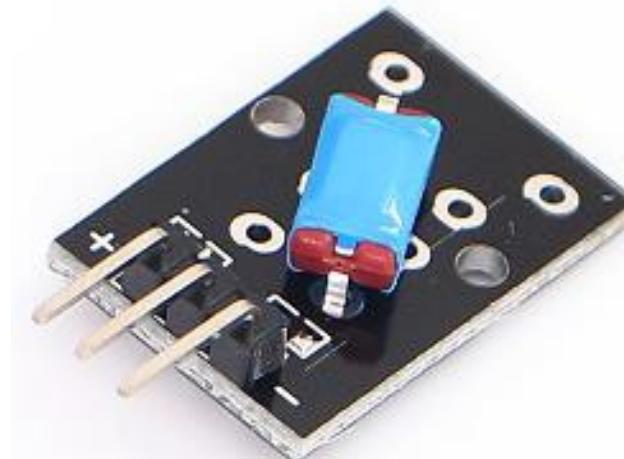


入力モジュール（続き）

衝撃センサ

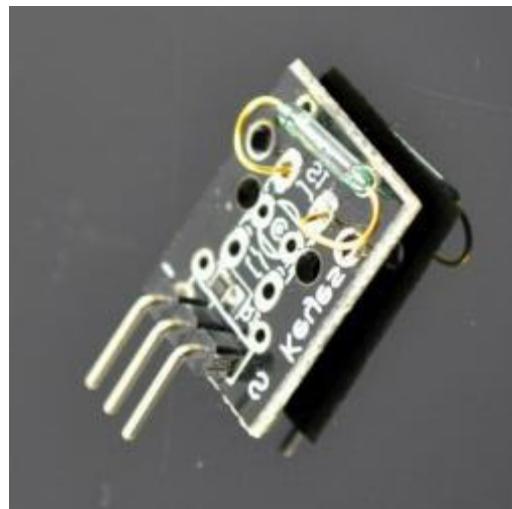


傾斜スイッチ



入力モジュール（続き）

リードスイッチ



その他

- ・ボタン
- ・タッチセンサ
- ・水センサ

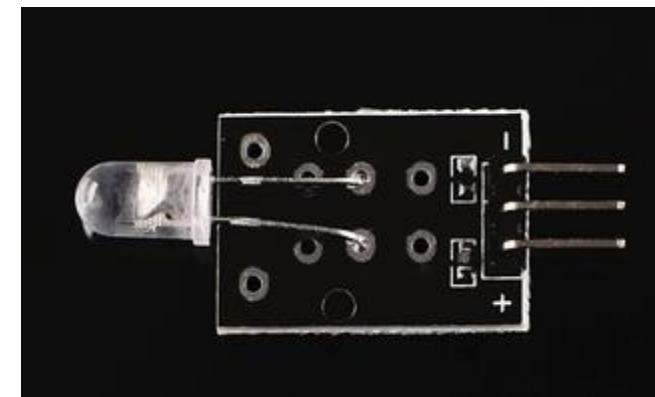
など

5-4. 出力モジュール

レーザ

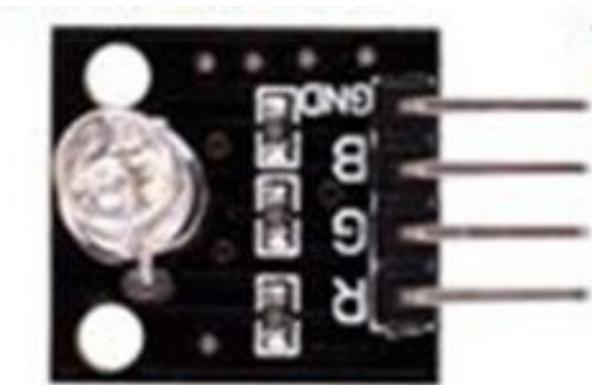


7色LED



出力モジュール（続き）

RGB LED



その他

- LCD
 - サーボ
 - モータ
- など

演習2 Arduinoとセンサを使った回路設計

環境センサ

- ① 光センサの利用
- ② 光センサによるLED点灯

入力モジュール

- ③ 傾斜スイッチの利用
- ④ ロータリーエンコーダによるLED点灯

出力モジュール

- ⑤ LCD出力
- ⑥ 応用編：各自で色々なセンサやモジュールを組合せて利用

第6章 IoTのセキュリティ

6-1. IoTデバイスを標的としたマルウェア

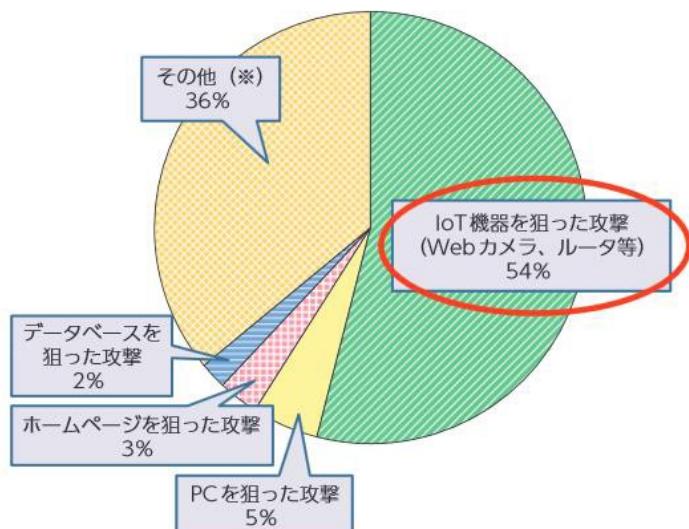
IoTデバイスの普及に伴って、MiraiウィルスのようなIoTデバイスを標的としたマルウェアが流行

図表6-5-2-2

NICTERによる観測結果

観測された全サイバー攻撃1,504億パケットのうち、

半数以上がIoTを狙っている！



※IoT機器特有のポートを狙った攻撃から、特定のIoT機器の脆弱性を狙ったより高度な攻撃も観測されるようになっており、単純にポート番号だけから分類することが難しいIoT機器を狙った攻撃が「その他」に含まれている。

NICT(国立研究開発法人情報通信研究機構)が運用するサイバー攻撃観測網(NICTER)が平成29年に観測したサイバー攻撃パケット、1,504億パケットのうち、半数以上がIoT機器を狙ったものであるという結果が示されている

情報通信白書平成30年版(総務省)

<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/pdf/30honpen.pdf>

6-2. Miraiウィルス

Miraiウィルス

IoTデバイスに感染しボットネットを作るマルウェア
ボットネットから攻撃目標に対してDDoS攻撃を行う
2016年にMiraiウィルスのボットネットが発見される
プロバイダやIT企業、ジャーナリストなどへの大規模かつ破壊的な攻撃が
観測された

Miraiウィルスの挙動

対象外を除いてIPアドレスをランダムに走査
脆弱性のある機器を調査（工場出荷時・デフォルト状態、辞書攻撃など）
感染したデバイスはC&Cサーバ（指令＆制御）から遠隔操作
DDoS攻撃を実行（UDPフラット攻撃、HTTPフラット攻撃、DNSフラット攻撃）
増殖を繰り返し感染を拡大

Miraiウィルス（続き）

マルウェア

コンピュータウィルスやワーム、トロイの木馬、スパイウェア、ボット、ランサムウェアなどの悪意のあるソフトウェアのこと
総合的な名称としてマルウェア（Malware）と呼ぶ

ボット

感染したコンピュータを外部から遠隔操作し不正アクセスの手足として利用し、迷惑メールの送信や特定サイトへの攻撃などを行うプログラム

ボットネット

ボットに感染したコンピュータからなるネットワークはボットネットと呼ばれる

ボットネットのコンピュータは特定サイトの一斉攻撃（DDos攻撃）などに利用される

Miraiウイルス（続き）

DDoS攻撃

Distributed Denial of Service攻撃の略

ウィルスに感染して遠隔操作可能な複数の端末から一斉にDoS攻撃（サービス拒否攻撃）を行う

UDPフラット攻撃、HTTPフラット攻撃、DNSフラット攻撃など、通信プロトコルの手続きのパケットを一斉に大量に送りつけることで、相手が処理しきれなくなりサービスが停止してしまう

6-3. IoTセキュリティガイドライン

経済産業省及び総務省が「IoT推進コンソーシアム IoTセキュリティワーキングルループ」を開催

IoTを活用した革新的なビジネスモデルを創出

国民が安全で安心して暮らせる社会を実現

必要な取組等について検討

「IoTセキュリティガイドライン ver1.0」が策定（平成28年7月5日）

<https://www.meti.go.jp/press/2016/07/20160705002/20160705002.html>

6-4. IoTセキュリティガイドラインの目的

本ガイドラインの目的は、IoT特有の性質とセキュリティ対策の必要性を踏まえて、IoT機器やシステム、サービスについて、その関係者がセキュリティ確保観点から求められる基本的な取組を、セキュリティ・バイ・デザインを基本原則としつつ、明確化することによって、産業界による積極的な開発等の取組を促すとともに、利用者が安心してIoT機器やシステム、サービスを利用できる環境を生み出すことにつなげるもの。

なお、本ガイドラインの目的は、サイバー攻撃などによる被害発生時における関係者間の法的責任の所在を一律に明らかにすることではなく、むしろ関係者が取り組むべきIoTのセキュリティ対策の認識を促すとともに、その認識のもと、関係者間の相互の情報共有を促すための材料を提供することである。

本ガイドラインは、その対象者に対し、一律に具体的なセキュリティ対策の実施を求めるものではなく、守るべきものやリスクの大きさ等を踏まえ、役割・立場に応じて適切なセキュリティ対策の検討が行われることを期待する

6-5. サービス提供者のための指針

	指針	主な要点
方針	IoTの性質を考慮した基本方針を定める	<ul style="list-style-type: none">・経営者がIoTセキュリティにコミットする・内部不正やミスに備える
分析	IoTのリスクを認識する	<ul style="list-style-type: none">・守るべきものを特定する・つながることによるリスクを想定する
設計	守るべきものを守る設計を考える	<ul style="list-style-type: none">・つながる相手に迷惑をかけない設計をする・不特定の相手とつなげられても安全安心を確保できる設計をする・安全安心を実現する設計の評価・検証を行う
構築・接続	ネットワーク上での対策を考える	<ul style="list-style-type: none">・機能及び用途に応じて適切にネットワーク接続する・初期設定に留意する・認証機能を導入する
運用・保守	安全安心な状態を維持し、情報発信・共有を行う	<ul style="list-style-type: none">・出荷・リリース後も安全安心な状態を維持する・出荷・リリース後もIoTリスクを把握し、関係者に守ってもらいたいことを伝える・IoTシステム・サービスにおける関係者の役割を認識する・脆弱な機器を把握し、適切に注意喚起を行う

6-6. 一般利用者のための指針

- 問合せ窓口やサポートがない機器やサービスの購入・利用を控える
- 初期設定に気をつける
- 使用しなくなった機器については電源を切る
- 機器を手放す時はデータを消す

第7章 IoTプラットフォームを使ったデータ通信

7-1. IoTプラットフォームの例

sakura.io

The screenshot shows the homepage of sakura.io. At the top, there's a navigation bar with links for "sakura.ioとは?", "パートナー", "商品の購入", "料金", "お問い合わせ", "開発者向け", and "ログイン". Below the navigation is a large banner with the text "だれもが、データを活かせる世の中へ。" (Everyone can live in a world where data is utilized). A subtext below it reads: "sakura.ioは、これまで気付けなかった「モノ・コト」の相関性や関係性を見出し、それを世界中でシェアできるプラットフォームを目指します。". There's a "会員登録はこちら" button. On the right side, there's a news card for "sakura.io評価ボードの販売を開始しました." dated 2018/5/31. Below the banner, there's a section for "イベント・セミナー・ワークショップ情報" with three entries: "TU-22プログラミング・コンテスト2018" (2018/10/21), "「まわりなカフェ～サポートスタッフと直接お話しして、問題解決しませんか?～」(大阪)" (2018/08/20), and "「決済サービスおよび市場ニーズへのアプローチ」(沖縄)に、エバンジェリスト横田が登壇いたします" (2018/08/04). At the bottom, there's a callout box stating: "sakura.ioは、通信モジュールからデータの保存/連携まで IoTに関わるネットワークとデータのやり取りを統合的に実現します。". The footer features a diagram titled "sakura.io の提供範囲" showing "モノづくりの領域 (組み込み系エンジニア)" and "コトづくりの領域 (Web系エンジニア)".

IIJ IoT

The screenshot shows the IIJ IoT services page. The header includes links for "法人のお客様", "個人のお客様", "IIJについて", "IIJの技術", and "株主・投資家向け情報". The main content area has a breadcrumb trail: "ホーム > 法人のお客様 > WAN・ネットワーク > IIJ IoTサービス". Below that is a navigation bar with "WAN・ネットワーク", "モバイル", and "クラウド". A section titled "IIJ IoTサービス" features a sub-section "IoTビジネスに必要な機能をワンストップで". Below this, there's a "お知らせ" (Notice) box with a "【セミナー情報】" entry: "8月9日(木) 東京 IIJ IoTサービスを使い倒す会". At the bottom, there's a "IIJ IoTサービスの概要" section with a diagram illustrating the connectivity between "もの (センサーデバイス)", "つなぐ (IIJ IoTサービス 必要な機能を選択・組み合わせ)", and "クラウド (データ活用)".

IoTプラットフォームの例（続き）

AWS IoT



IoT とは

IoT（モノのインターネット）は生産性と効率性を高めるために、物理的な世界をインターネットと接続してデバイスからのデータを活用します。広範な接続オプションがあり、接続コストが下がっているうえに、より多くのデバイスがデータを取得できるため、様々なモノをインターネットに接続することができます。IoT アプリケーションが使用されているのは、冷蔵庫、監視カメラ、ケーブルテレビのセットトップボックスのような消費者製品、コンベアベルトや製造装置のような商業システム、信号機やスマートメーターのような商用デバイスなど多岐に渡ります。電源をオンにできるすべてのデバイスが IoT アプリケーションの一部なのです。

消費者は自宅にあるすべてのモノ、運転するすべてのモノ、着用するモノにまで IoT の機能を求めていきます。たとえば、スマートブレード田舎白動型ホームペーパーなどの

SORACOM IoT



IoT とは、センサーやデバイスといった「モノ」をインターネットにつなぎ、取得した

7-2. IoTプラットフォーム sakura.io

sakura.ioとは? パートナー 商品の購入 料金 お問い合わせ 開発者向け ログイン

だれもが、データを活かせる世の中へ。

sakura.ioは、これまで気付けなかった「モノ・コト」の相関性や関係性を見出し、それを世界中でシェアできるプラットフォームを目指します。

会員登録はこちら >

サービスサイト >さくらインターネットが提供するIoTプラットフォームサービス、sakura.io

news

2018/5/31
sakura.io評価ボードの販売を開始しました。

イベント・セミナー・ワークショップ情報

2018/10/21 「U-22プログラミング・コンテスト2018」に、代表田中と技術本部副本部長 江草が委員として参加いたします

2018/08/20 「まりなカフェ～サポートスタッフと直接お話しして、問題解決しませんか？～」（大阪）を開催いたします

2018/08/04 「決済サービスおよび市場ニーズへのアプローチ」（沖縄）に、エバンジェリスト横田が登壇いたします

イベント・セミナー情報の一覧

sakura.ioは、通信モジュールからデータの保存/連携まで
IoTに関わるネットワークとデータのやり取りを統合的に実現します。



モノづくりの領域
(組み込み系エンジニア)

https://sakura.io/

7-3. sakura.ioの特徴

低価格&セキュア（閉域網を使用）

クラウド連携可能

最低月額料金 64円（税込み）



「電気信号」と「JSONデータ」の相互変換装置として動作し、これまでのIoTデバイス/サービス開発の中間層を補完することで、モノ/サービスづくり・連携に注力が可能です。

7-4. さくらのLTE通信モジュール



sakura.io Webサイト
<https://sakura.io/product/>

さくらのLTE通信モジュール（続き）



sakura.io Webサイト
<https://sakura.io/product/>

さくらのLTE通信モジュール（続き）

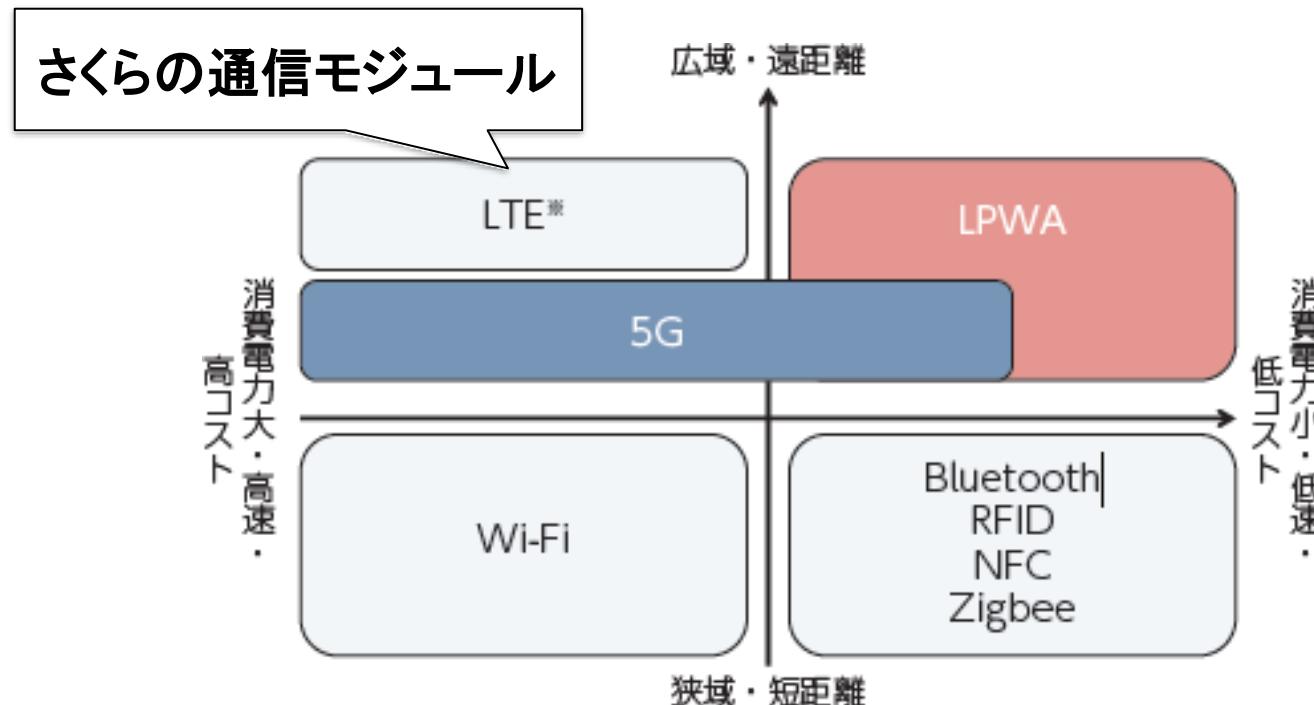
sakura.ioモジュール

LTE通信モジュール、LTEカテゴリー1（低速、小消費電力、IoT向き）

特徴（製品データシートより抜粋）

- ・sakura.ioにLTE網を通じてダイレクトに接続するため、ゲートウェイ装置がいらない
- ・コマンドのみでデータの送受信ができ、ホストMCU側で通信プロトコルを実装する必要がない
- ・ホストMCUインターフェースはI2C, SPI, およびUARTから選択可能
- ・小型モジュール（46W×34D×3H）内にLTEモデムやSIMなど必要な機能をすべて内蔵
- ・待ち受け時の消費電力が低い
- ・日本国内工事設計認証および電気通信端末機器認証済み

7-5. さくらの通信モジュールの位置付け

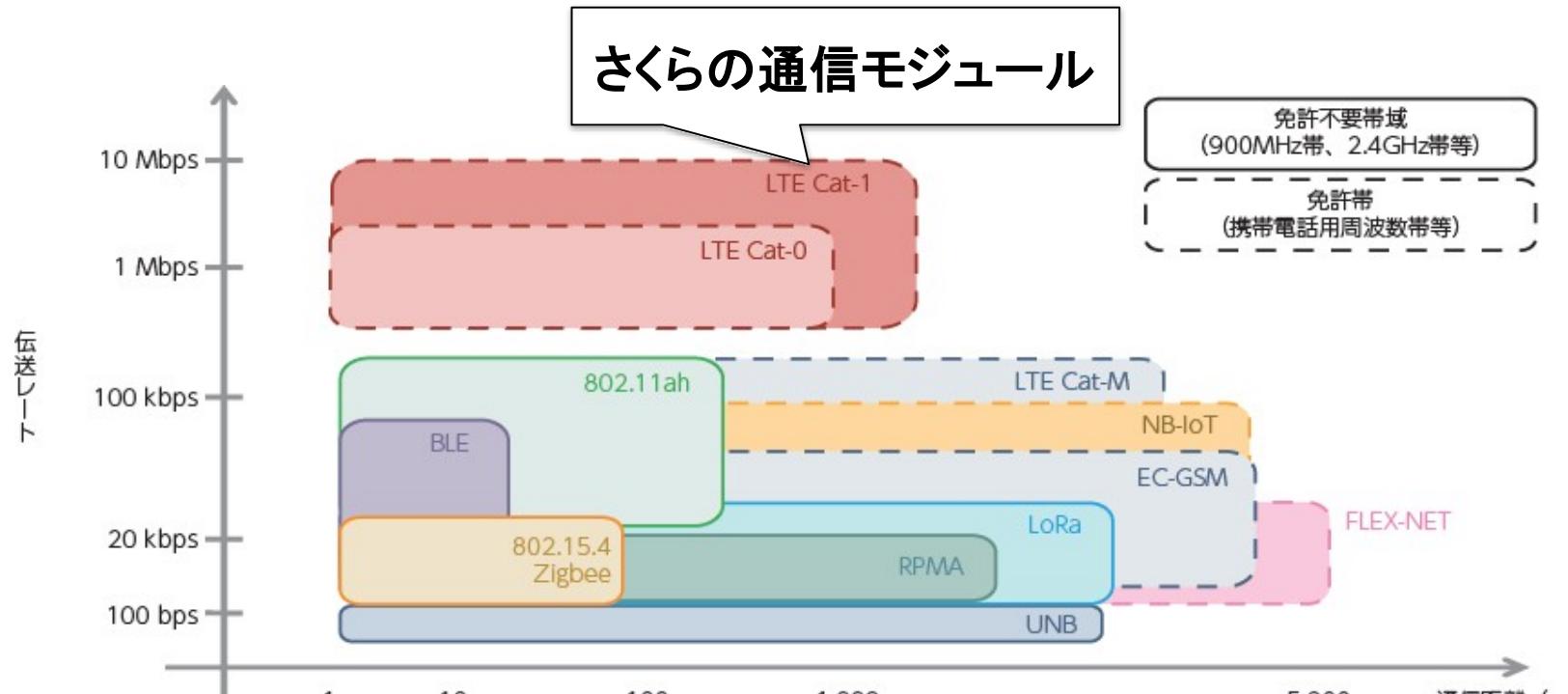


※既存のM2M接続は2G、3G、4Gが主流

(出典) 総務省「第4次産業革命における産業構造分析とIoT・AI等の進展に
係る現状及び課題に関する調査研究」(平成29年)

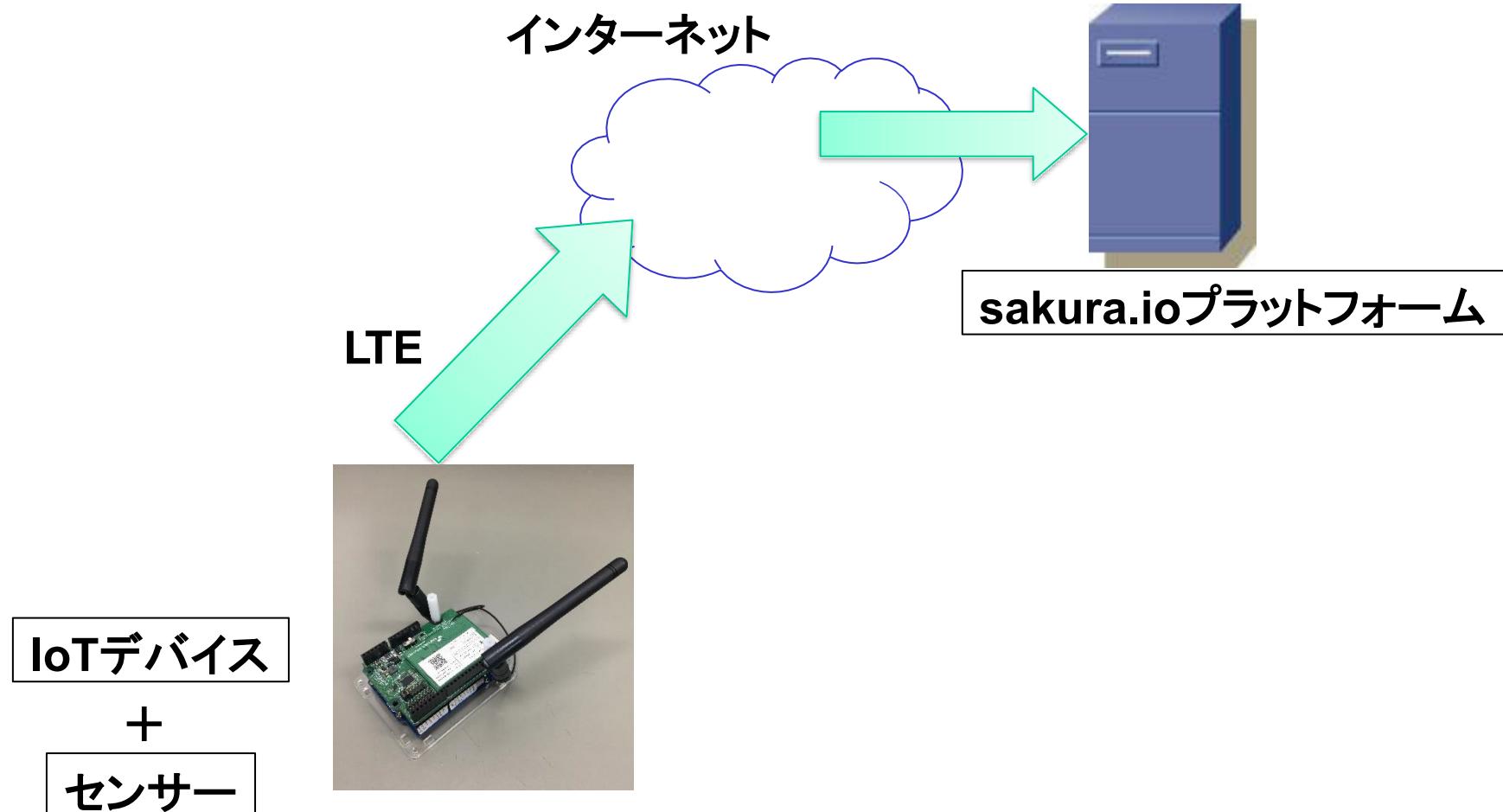
情報通信白書平成29年版(総務省)
<http://www.soumu.go.jp/johotsusintokei/whitepaper/h29.html>

さくらの通信モジュールの位置付け（続き）

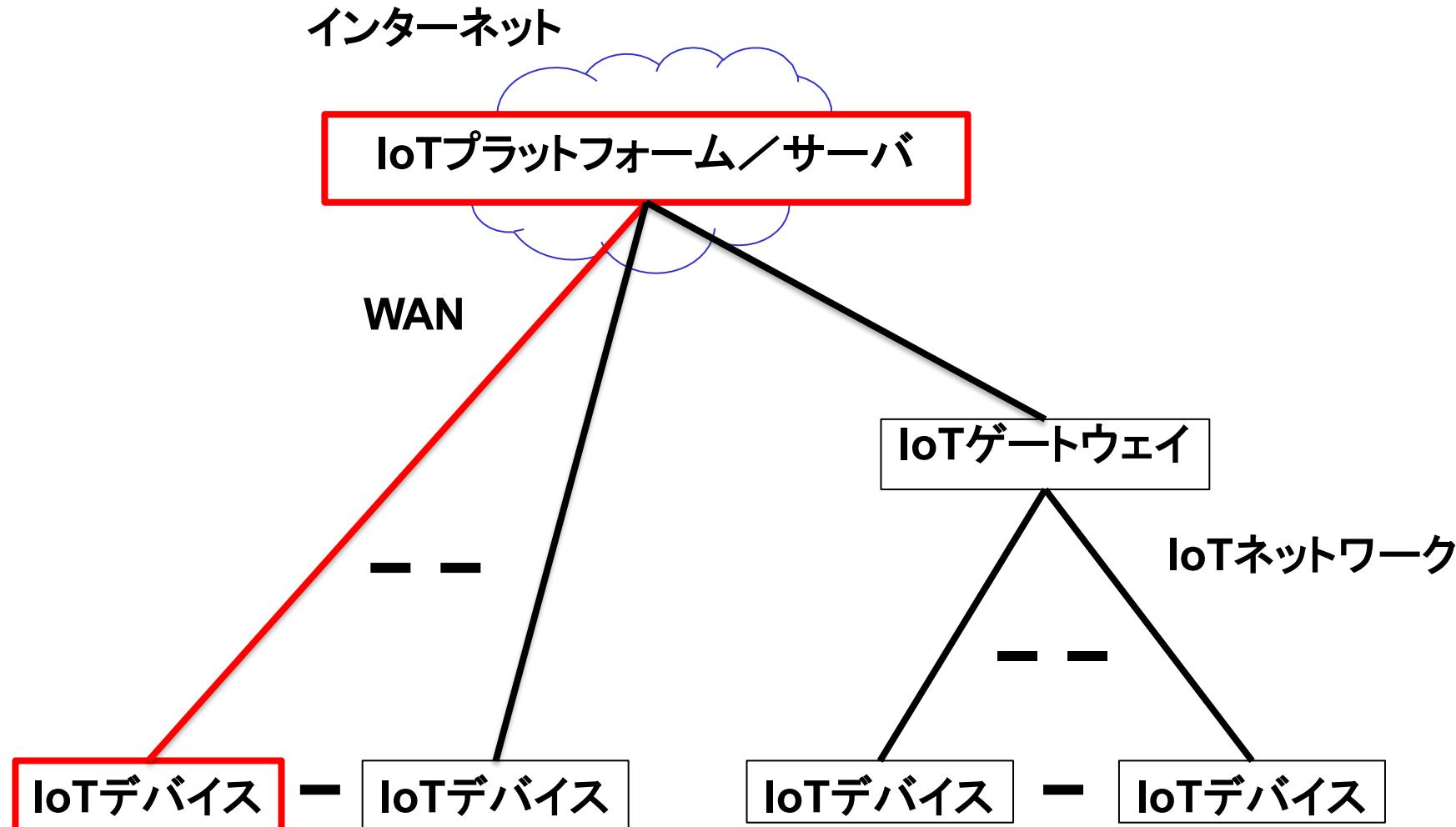


(出典) 総務省「第4次産業革命における産業構造分析とIoT・AI等の進展に係る現状及び課題に関する調査研究」(平成29年)

7-6. sakura.ioの物理的構成



7-7. IoTシステムの物理的構成



7-3. さくらのIoT Platformの特徴

低価格&セキュア（閉域網を使用）

クラウド連携可能

最低月額料金 64円（税込み）



「電気信号」と「JSONデータ」の相互変換装置として動作し、これまでのIoTデバイス/サービス開発の中間層を補完することで、モノ/サービスづくり・連携に注力が可能です。

7-8. sakura.io 料金と通信ポイント

1ヶ月につき通信ポイントが10,000pt付与

100回の通信ごとに100pt消費（100回未満は切り上げ）

別途購入する場合は20,000pt/100円

都度消費ではなく、月末に通信回数によってポイント引き落とし。不足すればその分を精算

10,000pt = 10,000回の通信

5分に1度の通信 → 1時間で12回 → 1日 288回

* 30日で8,640回

5分に1度の通信でも充分。データを貯めて定期的に送信することも可能

7-9. ポイント管理例

ポイント管理

現在のポイント 39,800pt 2018年7月末期限ポイント

10,000pt

有効期限	有償ポイント	無償ポイント
2018-07-31	0pt	10,000pt
2018-08-31	0pt	10,000pt
2018-09-30	0pt	10,000pt
2019-07-31	0pt	9,800pt

履歴

日付	区分	ポイント増減
2017-07-31	2017年7月 モジュール通信(200ポイント)	-200pt
2017-07-31	2017年8月 付与分	+10,000pt

7-10. ライブラリとマニュアル

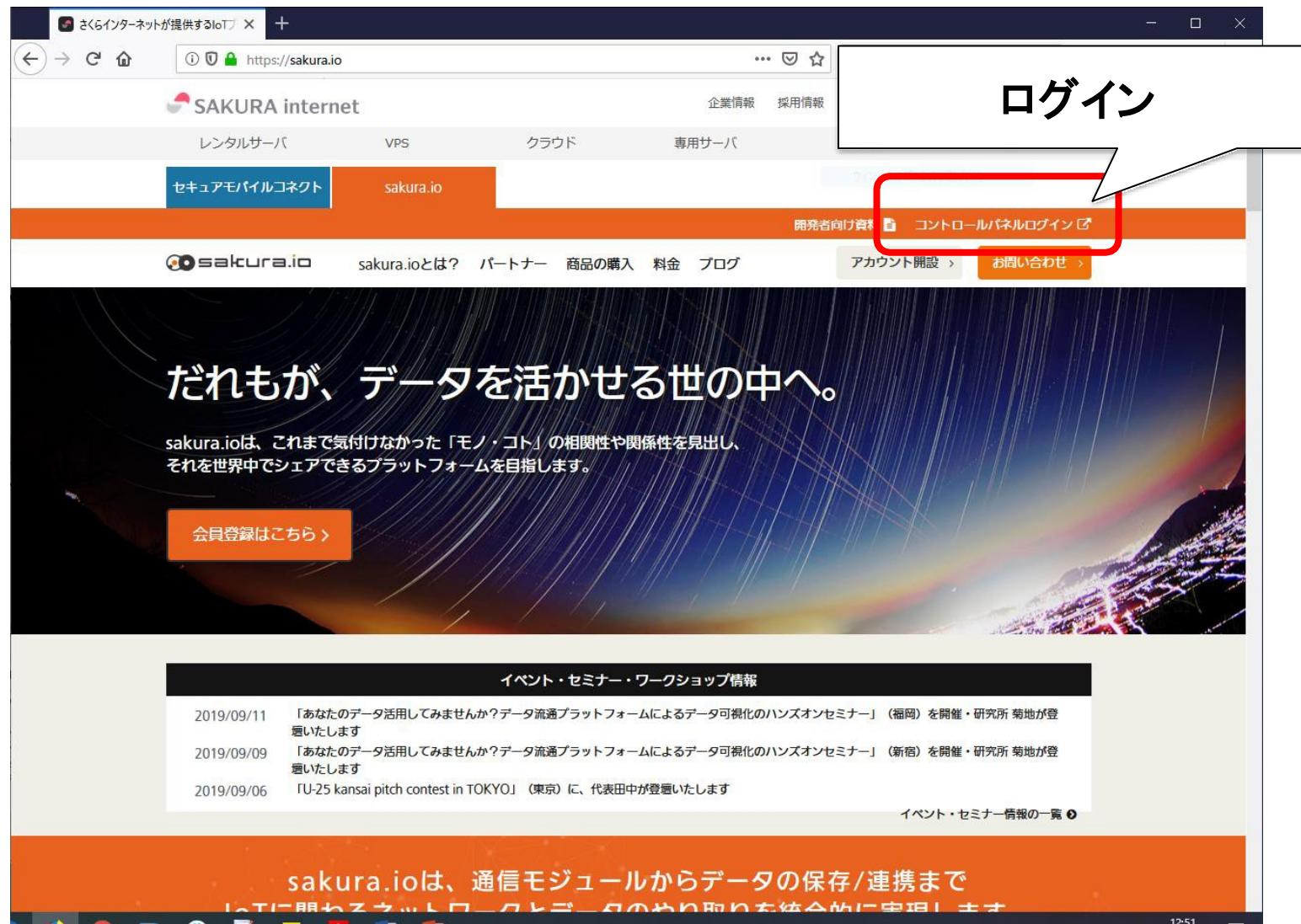
ライブラリ

<https://github.com/sakuraio/SakuraIOArduino>

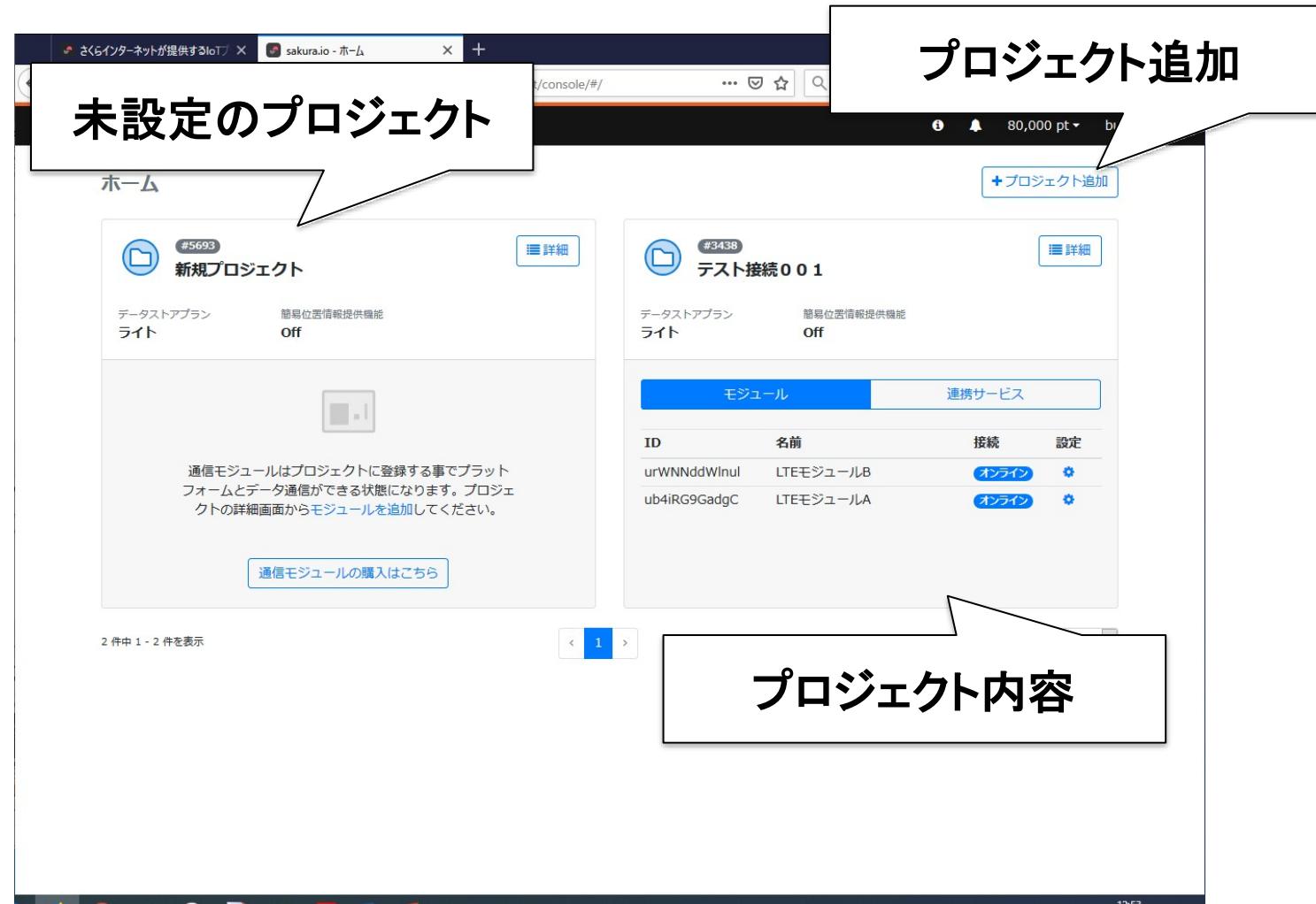
マニュアル

<https://sakura.io/docs/index.html>

7-11. ログインとプロジェクト

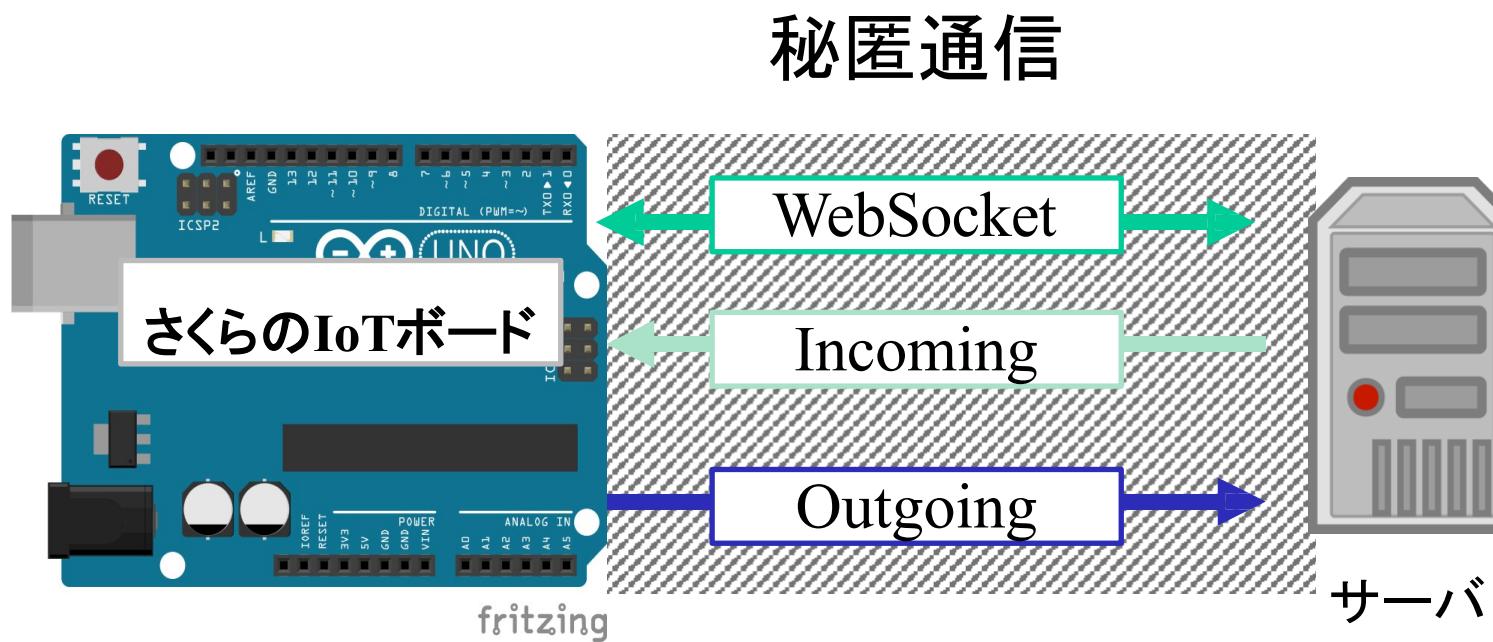


ログインとプロジェクト



7-12. 基本的な考え方

さくらIoTのライブラリを通じてデータの送受信を行う
Arduino側にTCP/IPスタックは必要ない



7-13. コード例

接続

sakuraio.getConnectionStatus()

データ送信キューに貯める

sakuraio.enqueueTx()

データ送信

sakuraio.send()

データ受信

sakuraio.getRxQueueLength()

ライブラリをインポートし、スケッチ例Standardを実行する

7-14. 連携サービス

WebSocket

コネクションを維持したままデータ送受信

Outgoing Webhook

モジュールからデータ送信

Incoming Webhook

モジュールへデータ送信

MQTT Client

DataStore API

AWS IoT

Azure IoT Hub (a)

本演習ではWebSocketとIncoming Webhookを行う

7-15. WebSocket

従来のhttp等はコネクションレスの通信プロトコル

WebScoketはコネクションを維持したまま通信可能なプロトコル

さくらのIoTで最も簡単に扱える

10秒に1度keepaliveを送信し、コネクションを維持（keepaliveは課金されない）

PHPでWebSocketを扱うのは容易

ただし、コマンドを都度実行したり、Webブラウザで読み込み続ける必要がある

7-16. データ形式

データ形式はすべてJSON

送信できるデータ形式は決まっている

int型変数は、符号あり32bit整数のint32_tのみ

同じく符号無しのint型変数は、uint32_tのみ

floatやdoubleはそのままでもよい

参照：

<https://sakura.io/docs/pages/platform-specification/message.html>

7-17. JSON例（データが単数）

データの例

```
{  
    "datetime": "2019-08-19T05:25:19.986646718Z",  
    "module": "*****",  
    "payload": {  
        "channels": [  
            {  
                "channel": 0,  
                "type": "channel[0]",  
                "value": 31.864151,  
                "datetime": "2019-08-19T04:56:20.035365877Z"  
            }  
        ],  
        "type": "channels"  
    }  
}
```

モジュールシリアル
データの型
データの値

7-18. JSON例（データが複数）

※payload部分のみ

```
"payload": {  
    "channels": [  
        {  
            "channel": 0,  
            "type": "f",  
            "value": 47,  
            "datetime": "2019-08-19T04:56:40.190154948Z"  
        },  
        {  
            "channel": 0,  
            "type": "f",  
            "value": 29  
            "datetime": "2019-08-19T04:56:40.190154948Z"  
        }  
    ]  
},
```

7-19. 連携サービスの作成

sakura.ioにログインし、コントロールパネルから作成

ホーム > プロジェクト詳細 > 連携サービスカタログ

外部サービスとsakura.ioを連携し、データのやり取りを行います。
詳しくはドキュメントをご覧ください。 [sakura.ioドキュメント - 連携サービス仕様](#)

WebSocket

Outgoing Webhook

Incoming Webhook

MQTT Client

Datastore API

AWS IoT

Azure IoT Hub(α) : 正式版提供に伴い廃止予定

Google Cloud Pub/Sub Publisher

Azure Event Hubs

Azure IoT Hub

7-20. WebSocketのURLとToken

コントロールパネルで確認可能

外部からアクセスする際は、ここに表示されるURLとTokenが必要

ホーム > プロジェクト詳細 > 連携サービス詳細

The screenshot shows the 'WebSocket' service detail page. At the top, there is a message: 'リアルタイムの双方向通信を行う連携サービスです。詳しくはドキュメントをご覧ください。 [sakura.ioドキュメント - WebSocket](#)'.

The service details are as follows:

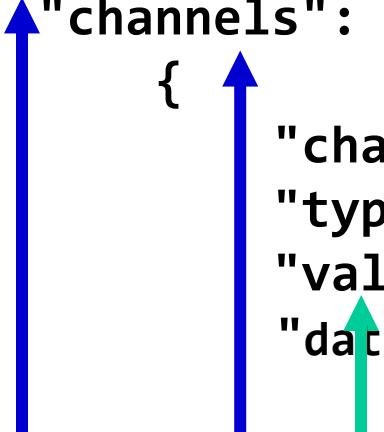
- #14986
- 名前: sakuratest1
- URL: wss://api.sakura.io/ws/v1/d070 [REDACTED]
- Token: d070 [REDACTED]

At the bottom, there are buttons for '編集' (Edit) and '削除' (Delete). Below the service details, there are tabs for '最終到着データ(50件)' (Last 50 arrivals), 'チャンネル別到着データ' (Arrivals by channel), and a '接続' (Connection) button which is currently green and labeled '接続中' (Connected). A message at the bottom says 'データはありません' (No data available).

7-21. JSON例（データが単数）

```
{  
    "datetime": "2019-08-19T04:56:40.190154948Z",  
    "module": "*****",  
    "payload": {  
        "channels": [  
            {  
                "channel": 0,  
                "type": "f",  
                "value": 31.864151,  
                "datetime": "2019-08-19T04:56:40.190154948Z"  
            }  
        ]  
    },  
    "type": "channels"  
}
```

data.payload.channels[0].value

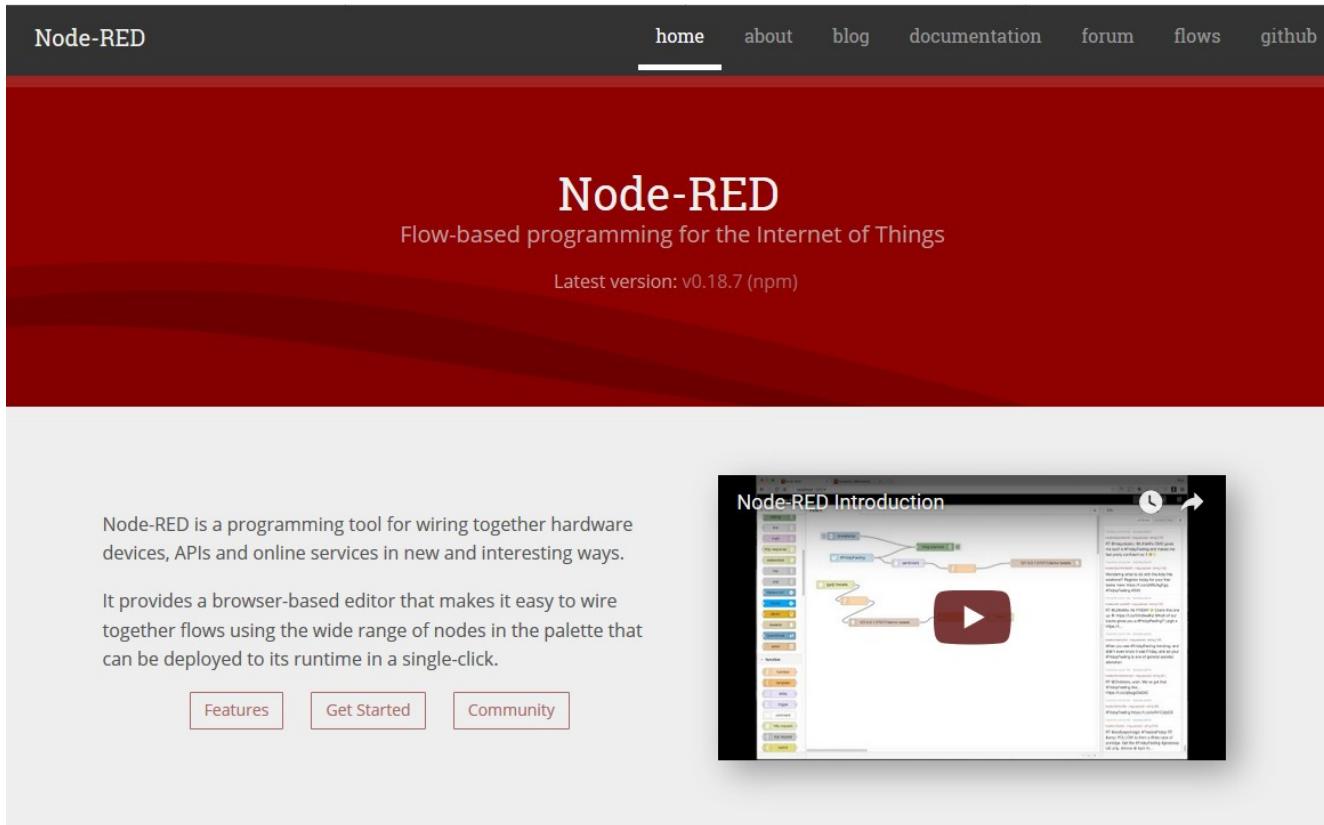


7-22. JSON例（データが複数）

```
"payload": {  
    "channels": [  
        {  
            "channel": 0,      データ1  
            "type": "f",     data.payload.channels[0].value  
            "value": 47,  
            "datetime": "2019-08-19T04:56:40.190154948Z"  
        },  
        {  
            "channel": 1,      データ2  
            "type": "f",     data.payload.channels[1].value  
            "value": 29,  
            "datetime": "2019-08-19T04:56:40.190154948Z"  
        }  
    ]  
},
```

7-23. 開発ツール Node-RED

Flowエディタを使って、プラグイン/モジュールであるノードを視覚的に接続しながら、IoTデバイスとオンラインサービスをつなぐことができる開発ツール



The screenshot shows the Node-RED website's home page. At the top, there is a dark navigation bar with the "Node-RED" logo on the left and links for "home", "about", "blog", "documentation", "forum", "flows", and "github". Below the navigation bar is a large red header section containing the "Node-RED" logo and the tagline "Flow-based programming for the Internet of Things". It also displays the "Latest version: v0.18.7 (npm)". The main content area features a brief introduction to Node-RED, mentioning it as a programming tool for wiring together hardware devices, APIs, and online services. It highlights the browser-based editor and the ability to deploy flows with a single click. Below this text are three buttons: "Features", "Get Started", and "Community". To the right of the text, there is a video player showing a "Node-RED Introduction" video. The video frame shows a screenshot of the Node-RED interface with a flow diagram and a play button.



Browser-based flow editing

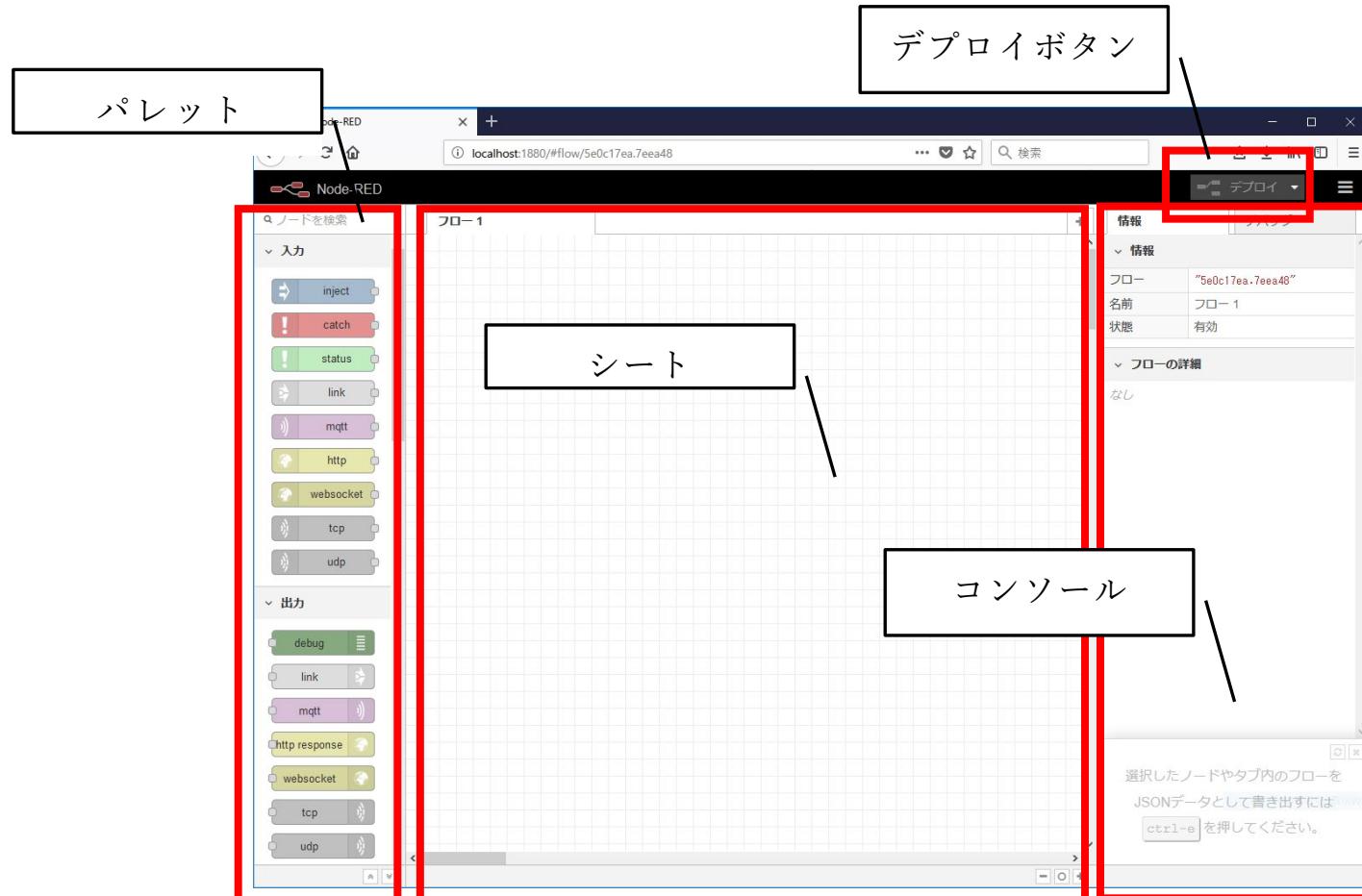
Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette.

Node-RED

<https://nodered.org/>

開発ツール Node-RED

Flowエディタを使って、プラグイン/モジュールであるノードを視覚的に接続しながら、IoTデバイスとオンラインサービスをつなぐことができる開発ツール



演習3 さくらLTEモジュールの回路設計と利活用

さくらのIoTコントロールパネルで確認

WebSocketをJavaScriptで取得して表示

Node-REDを使ったデータ通信

演習4 総合演習

これまで学んだものに基づいて各自のIoTシステムを構築

【必須】

- ・ IoTデバイスに任意のセンサを利用する
- ・ 取得したセンサーの値をsakura.ioにアップロードする

【任意】

- A. センサを複数にする／センサにアクチュエータをつける
- B. sakura.ioに集めたデータをNode-Redを使って可視化する
- C. IoTデバイスへのフィードバック機能を任意につける
- D. その他