

# どこでも安全に使える IoTを目指して

～さくらインターネットのIoTへの取り組み～

<https://www.sakura.ad.jp/>

DAY

2018/12/3

COMPANY

さくらインターネット株式会社

DEPARTMENT

コミュニティマネージャー

NAME

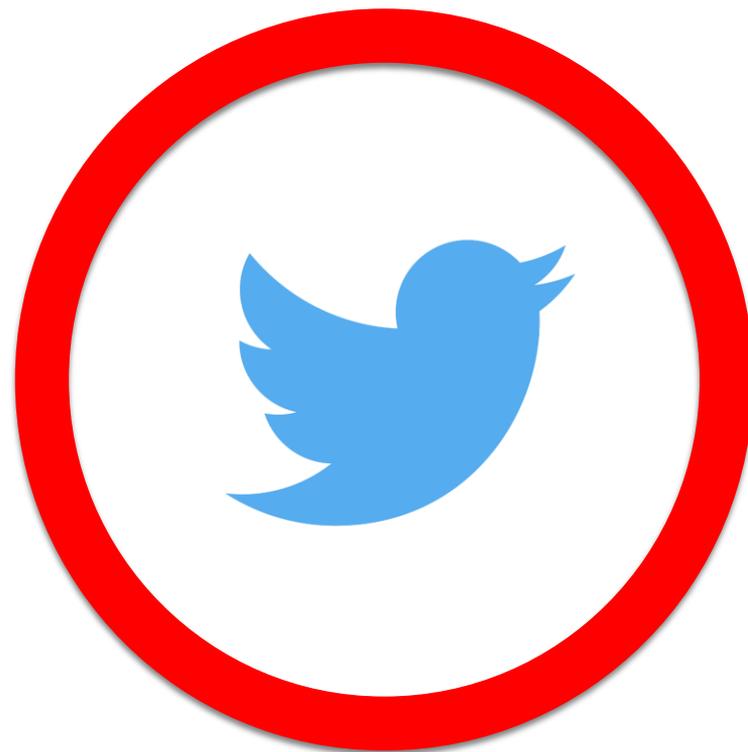
法林 浩之

本日の資料はこちらで公開します

<https://www.slideshare.net/hourin/>

もしくは

「slideshare 法林」で検索



#sakuraio



法林 浩之



@hourin

### どんな人？

- ・フリーランスエンジニア
- ・さくらインターネット コミュニティマネージャー
  - 会社主催イベントの運営
  - 社外イベント対応(協賛/出展/登壇/取材など)
  - さくらのナレッジ 編集長
- ・日本UNIXユーザ会 幹事(元会長)
  - さまざまなコミュニティと共同でイベントを開催
  - 全国各地のイベントで研究会を開催
- ・好きなプロレス団体:みちのくプロレス
- ・くわしくは「法林浩之」で検索

- IoTを取り巻く状況
- IoTサービスを作るときの問題点
- さくらインターネットのIoT向けサービス
  - sakura.io
    - 開発経緯 / サービス概要 / 利用事例
  - さくらのセキュアモバイルコネクト
    - サービス概要
- sakura.ioを使ってみる(デモ)
  - 温度湿度の測定 / LEDの操作



大阪本社  
(梅田/大阪)



東京支社 (西新宿)



福岡オフィス (赤坂)



東証一部上場



10021461(04)



ISMS-SR006JIS Q 27001 (ISO/IEC 27001)

商号	さくらインターネット株式会社 (SAKURA Internet Inc.)
代表取締役	田中 邦裕
設立	1999年8月17日 (サービス開始: 1996年12月23日)
資本金	22億5,692万円
事業内容	インターネットでのサーバの設置およびその管理業務 電気通信事業法に基づく電気通信事業 マルチメディアの企画並びに製作・販売
従業員数	495名 (連結/2017年3月末)
所属団体	特定非営利活動法人 日本データセンター協会 (JDCC) 社団法人 コンピュータソフトウェア協会 (CSAJ) 社団法人 日本ネットワークインフォメーションセンター (JPNIC) 社団法人 インターネットプロバイダー協会 (JAIPA)
グループ会社	アイティーエム株式会社 <a href="https://www.itmanage.co.jp/">https://www.itmanage.co.jp/</a> 株式会社S2i <a href="http://www.s2i.life/">http://www.s2i.life/</a> ゲヒルン株式会社 <a href="https://www.gehirn.co.jp/">https://www.gehirn.co.jp/</a> 株式会社Joe'sクラウドコンピューティング <a href="https://joes.co.jp/">https://joes.co.jp/</a> ビットスター株式会社 <a href="https://bitstar.jp/">https://bitstar.jp/</a> 櫻花移動電信有限公司

## レンタルサーバ



さくらのレンタルサーバ  
さくらのマネージドサーバ

## VPS・クラウド(仮想化基盤)



## 専用サーバ・データセンター



## 新規サービス

IoT 

通信モジュールから提供することで、セキュアな通信環境とデータの保存や処理システムを一体型で提供するIoTプラットフォームサービス

IoT さくらのセキュアモバイルコネク

お使いのデバイスを、SIMからさくらのクラウドのまでインターネットを経由せず、安全に接続できるIoT向けモバイルサービス

AI・人工知能



機械学習、データ解析、高精度シミュレーション用途に特化したGPU搭載の専用サーバサービス

コンテナ

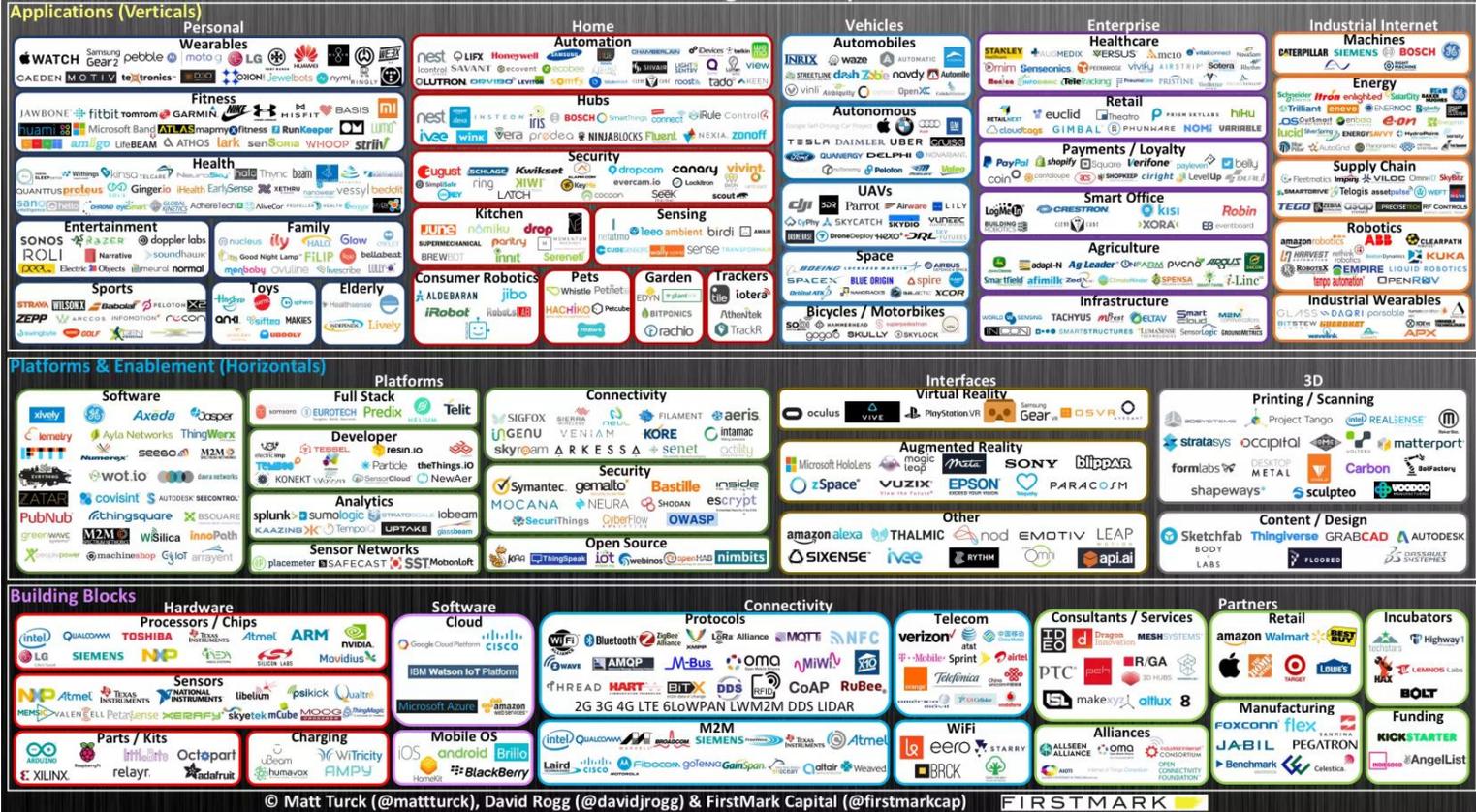


Dockerコンテナのマネージドサービスであり、コンテナを手軽かつシンプルに実行・運用

新しい社会のインフラを支えながら、最先端のサービスを構築

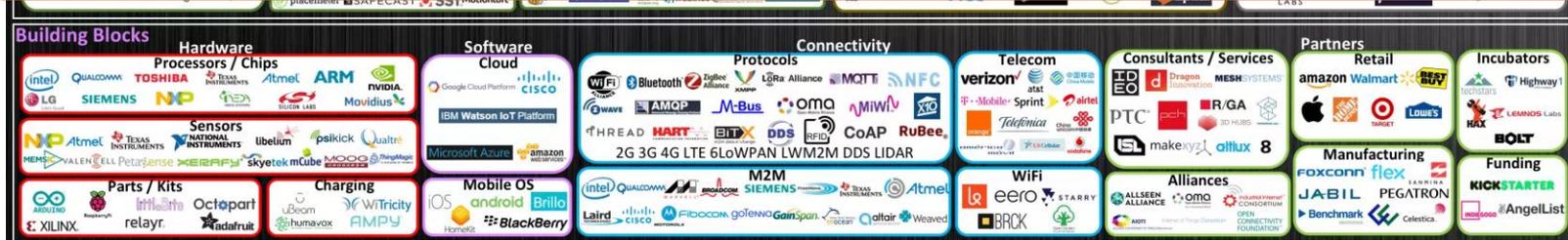
# IoTを取り巻く状況

## Internet of Things Landscape 2016





“インターネット”と同じぐらい広すぎる



Internet of Things Landscape 2016

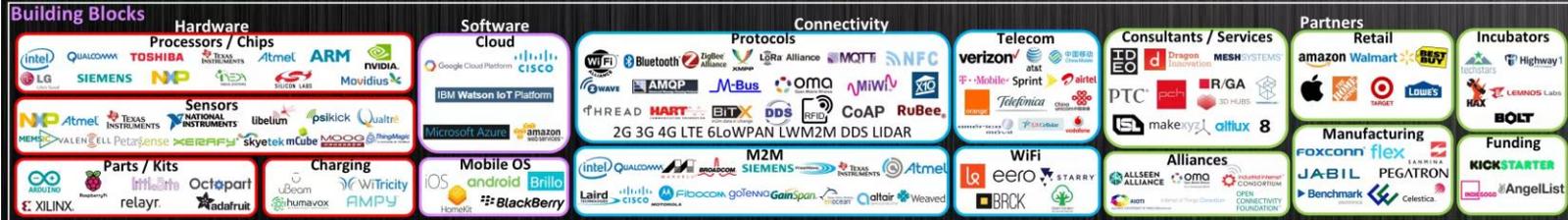


モノとサービスの両方を提供  
(垂直統合型)

Internet of Things Landscape 2016



汎用的な基盤を提供  
(水平統合型)



© Matt Turck (@matturck), David Rogg (@davidjrogg) & FirstMark Capital (@firstmarkcap) FIRSTMARK

Internet of Things Landscape 2016

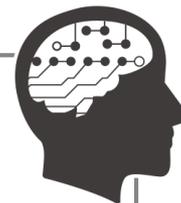


## IoTの領域 → 収集



モノゴトをデバイス経由で  
データ化し収集、蓄積

## AIの領域 → 情報化



データのクレンジング処理  
他情報と組み合わせによる意味付け

情報

## ロボティクスの領域 → フィードバック



情報を活用し具体的な  
アクションに反映

生活

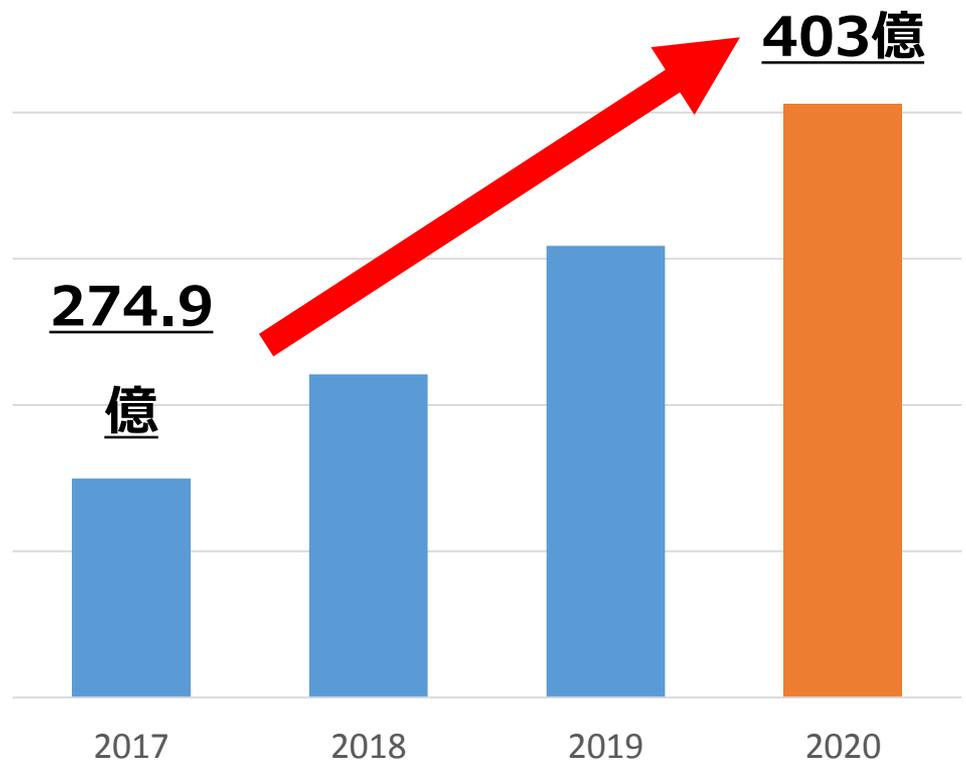


and,more..

世界のIoTデバイス数の推移及び予測

年平均成長率予測

13.6%





繋がっていたもの < 繋がっていなかったもの

## IoTの領域 → 収集



モノゴトをデバイス経由で  
データ化し収集、蓄積

## AIの領域 → 情報化



データのクレンジング処理  
他情報と組み合わせによる意味付け

情報

生活



## ロボティクスの領域 → フィードバック



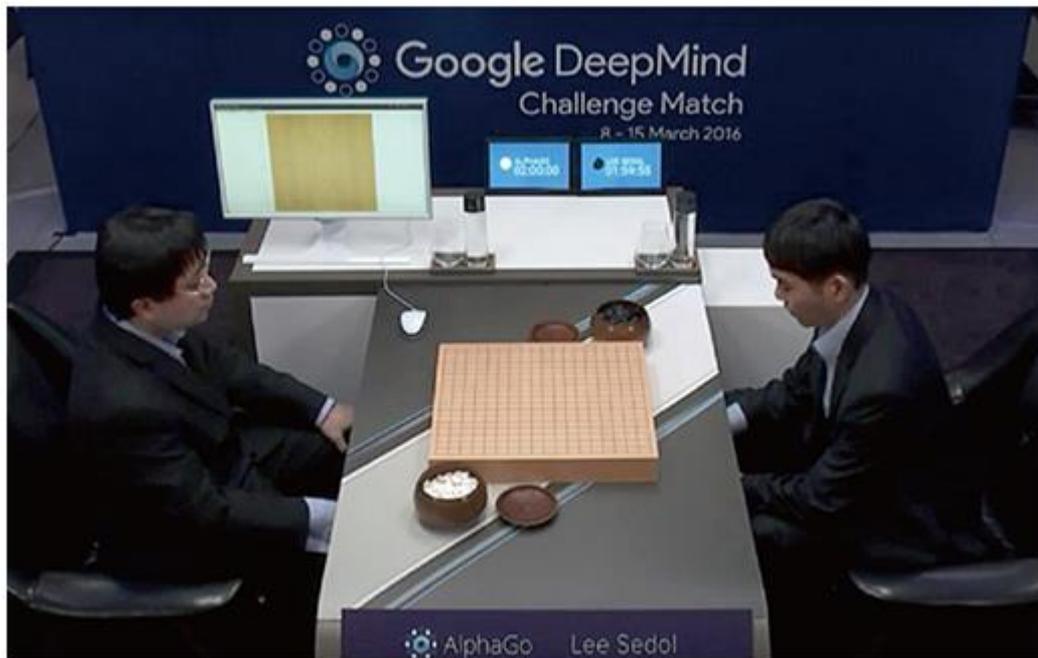
情報を活用し具体的な  
アクションに反映

and,more..



出典：総務省 <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/html/nc123200.html>

IoTにより収集した大量のデータを人工知能で解析することにより  
機器自身が学習し業務改善できるようになる



出典:総務省 / Google DeepMind

<http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/html/nc142110.html>

囲碁の人工知能プログラムが世界のトップ棋士に勝利

プロ棋士 vs コンピュータ将棋  
一騎打ちの頂上決戦

第1期

# 電王戦

100%

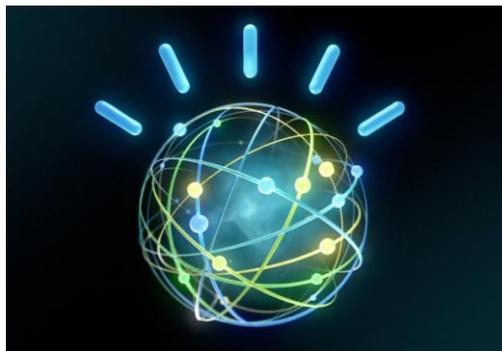
第1局 2016年4月9日(土)・4月10日(日) 関山 中尊寺  
第2局 2016年5月21日(土)・5月22日(日) 比叡山 延暦寺

第1期叡王戦 優勝  
山崎隆之 叡王  
TAKAYUKI YAMASAKI

第3回将棋電王トーナメント 優勝  
PONANZA  
PONANZA

出典: 第1期 電王戦特設サイト  
<http://denou.jp/2016/>

世界コンピュータ将棋選手権や電王戦で連勝  
さくらインターネットの計算機資源を活用



Watson



Siri



Pepper

自然言語を理解・学習し、人間の意思決定を支援する  
サービスが各社から提供され始めている

特化型AI(囲碁/将棋)  
コグニティブコンピューティング



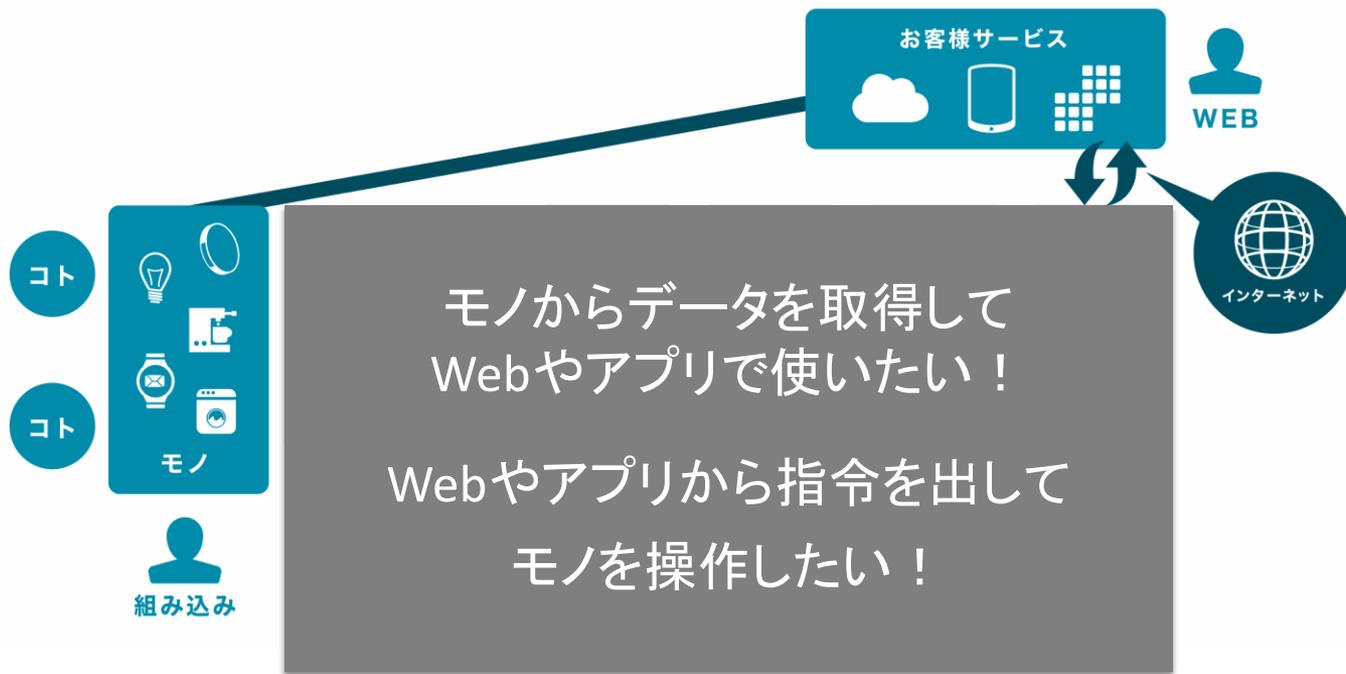
データさえあれば  
何でも分析できます！！

どうやって  
データ  
集めるの？

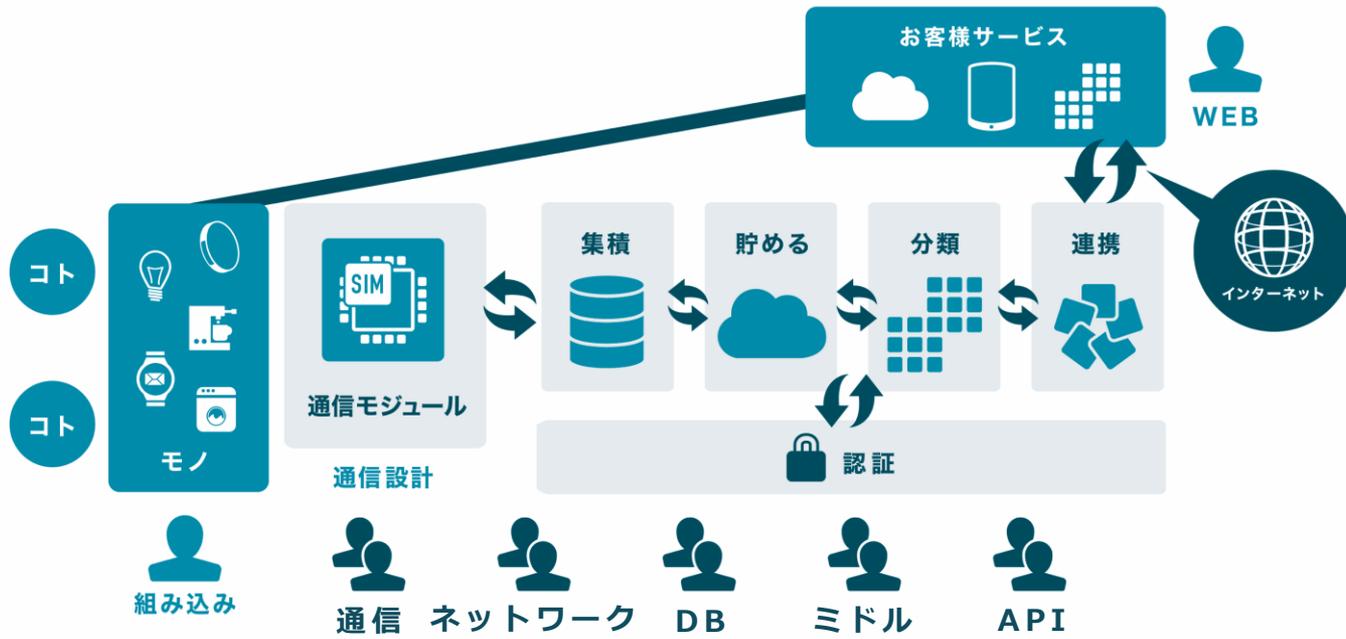
- IoTを名乗るサービスは非常に多い
  - インターネットとほぼ同義
  - 垂直統合型 / 水平統合型 / 要素技術 に分類できる
- IoTと人工知能でビジネスモデルが変わる(と予想されている)
  - IoTで大量のデータを収集
  - 収集したデータを人工知能で分析し意思決定を支援
- しかし良いデータ収集手段はまだ確立されていない



# IoTサービスを 作る時の問題点



モノをインターネットにつないで何かしたい



ハードウェア/ソフトウェア両方の知識が必要

## IoTサービスを作るときに必要な作業

- ハードウェア(モノ作り)
  - 機構設計 / 電気設計 / ファームウェア設計 / 通信設計
- ソフトウェア(フロントエンド)
  - UI設計 / アプリ開発
- ソフトウェア(バックエンド)
  - インフラ設計 / ミドルウェア設計 / DB設計 / API設計
- セキュリティ
  - ユーザ認証 / 機器認証 / 暗号化 / バックアップ

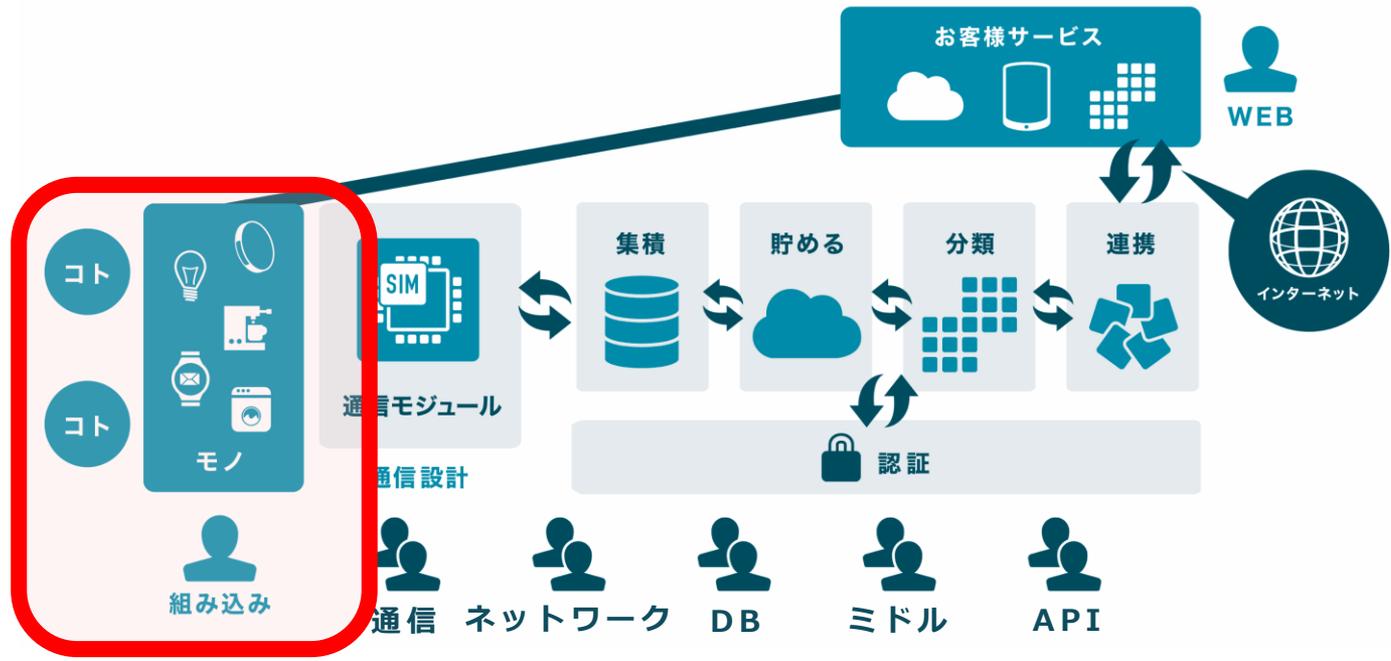
IoTサービスを作るときに必要な作業

ハードウェア(モノ作り)

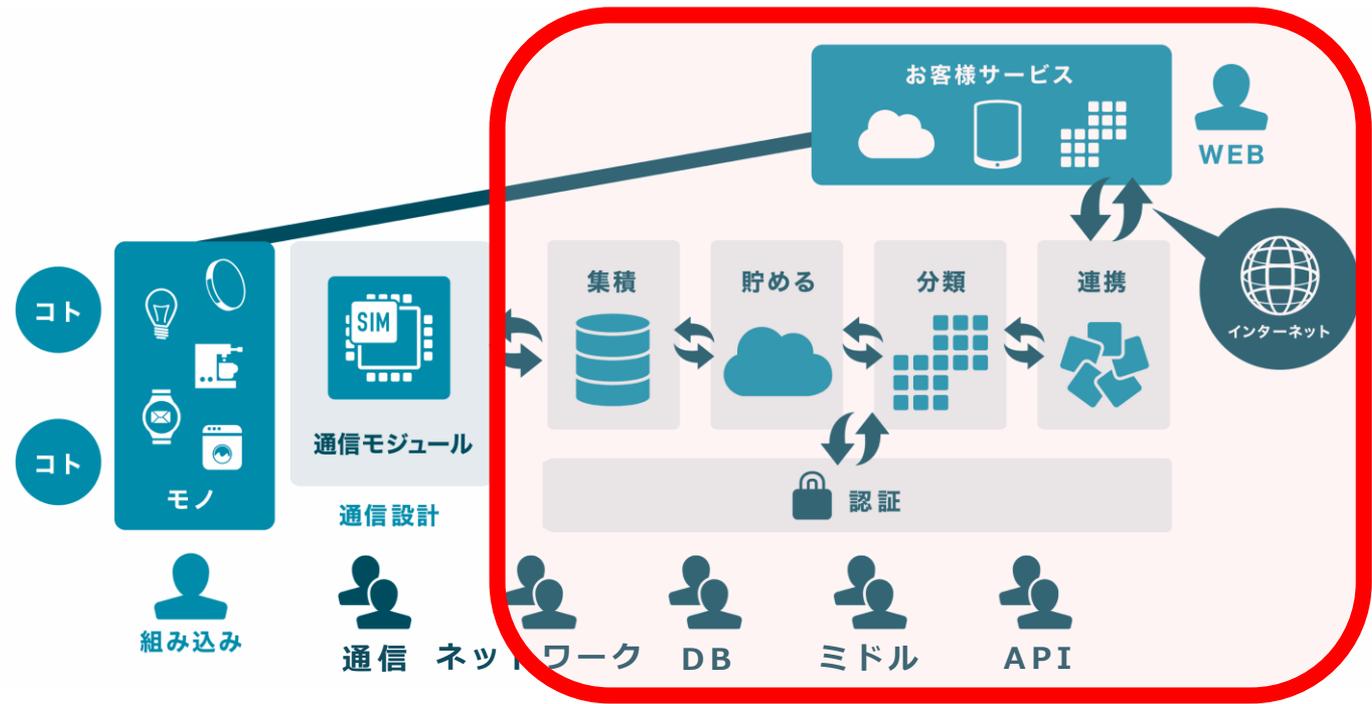
これらを全部できる人は  
ほとんどいない

• セキュリティ

• ユーザ認証 / 機器認証 / 暗号化 / バックアップ



物理的なことや電氣的なことはわかるが  
TCP/IPやHTTPのことはわからない

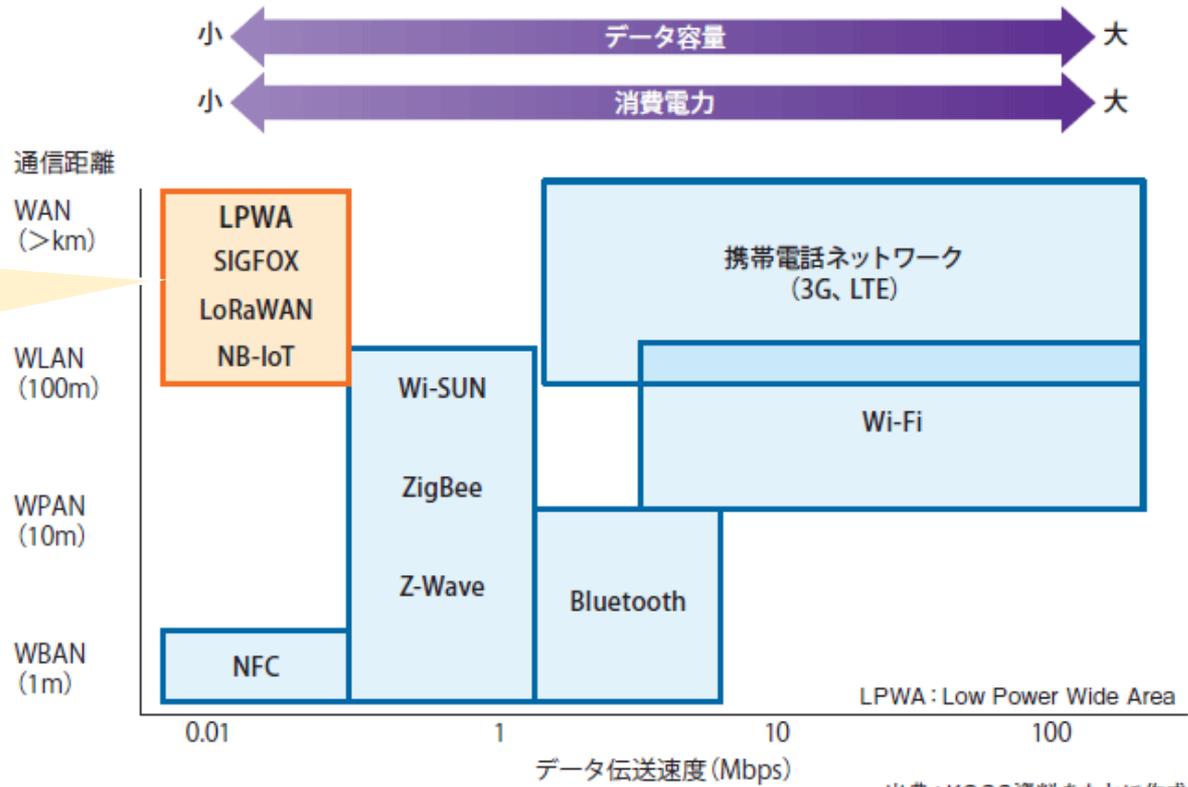


TCP/IPやHTTPのことはわかるが  
物理的なことや電気的なことはわからない

# モノをネットにつなぐ手段は増えている

	TCP/IP	not IP
近くとつながる	 	  
どこでもつながる		 

IoT向けの  
通信方式  
として注目



LPWA : Low Power Wide Area  
出典 : KCCS資料をもとに作成

## どの方式にも弱点がある

- LAN / WiFi: 回線やアクセスポイントが必要
  - モノをつなぐためだけに光ファイバーを敷設？
- LTE
  - 山奥や地中深くでは電波が入らない
  - 空中では使用禁止(電波法により)
- Bluetooth: 近距離でないとならぬ
- その他: 普及していない

- IoTで扱うデータは個人情報が多い
  - 生体情報、位置情報、など
  - データの暗号化は必須
- IoTデバイスはコンピュータとしては非力な場合が多い
  - Linuxでは重いので軽量なリアルタイムOSが普及
  - デバイス側に暗号化の仕組みを実装するのは難しい
- ハードウェアエンジニアにはTCP/IPやSSLの実装は困難
- インターネットにつながると攻撃対象になる
  - 不正アクセスを受けて乗っ取られる
  - DoS攻撃を受けて通信できなくなる

- 多様な環境に適応する必要性
  - WiFiがあるとは限らない
  - スマホがあるとは限らない
- できれば設定や操作をなくしたい
  - デバイスが勝手にデータを送受信すればよい
- 初期投資が大きい
  - モノの製造コストがかかるのが大きな負担
- サービスの継続にコストがかかる
  - デバイスの量産、保守、サポートなど

- ハードウェア/ソフトウェア両方の知識が必要
  - 両方できる人はほとんどいない
- 通信方式の選択が悩ましい
  - どの方式にも一長一短がある
- セキュリティの確保が難しい
  - IoTデバイスをインターネットに接続するのはリスクが高い
- 初期投資が大きい
  - モノの製造コストがかかるのが大きな負担
- サービスの継続にコストがかかる
  - デバイスの量産、保守、サポートなど



# さくらインターネットが考える IoTソリューション

1. どこでも、だれでも
2. 安全なネットワーク
3. ノーコミット&ノーリミット
4. 検証から、量産まで
5. 圧倒的コスパ

## LTEのメリット

1. **高いカバー率** (島嶼、山間)
2. **移動体でも使用可能**
3. **設定不要で繋がる**

LTEなら最終利用者も手軽に利用できる



デバイス⇄クラウド間のリスクポイントを削減

1. 契約期間縛りなし
2. 契約/解約手数料なし
3. 再登録が何度でも可能
4. 速度制限なし

IoT用途での課題となる各種制約や上限を撤廃

さくらクラウド SAKURA CLOUD セキュアモバイルコネクト

## IoT向けSIMの提供

デバイスからコンピューティング  
リソースにセキュアに接続できる  
モバイル回線



カード型SIM

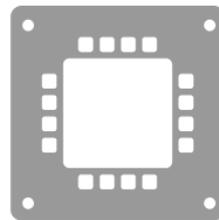


チップ型SIM

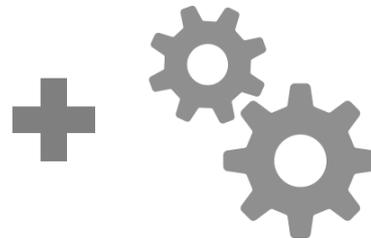
sakura.io

## プラットフォームの提供

モジュール+クラウドの一体提供で  
開発スピードを向上、スケール対応や  
運用保守はさくらで実施



専用モジュール  
(単体方式 or ゲートウェイ方式)



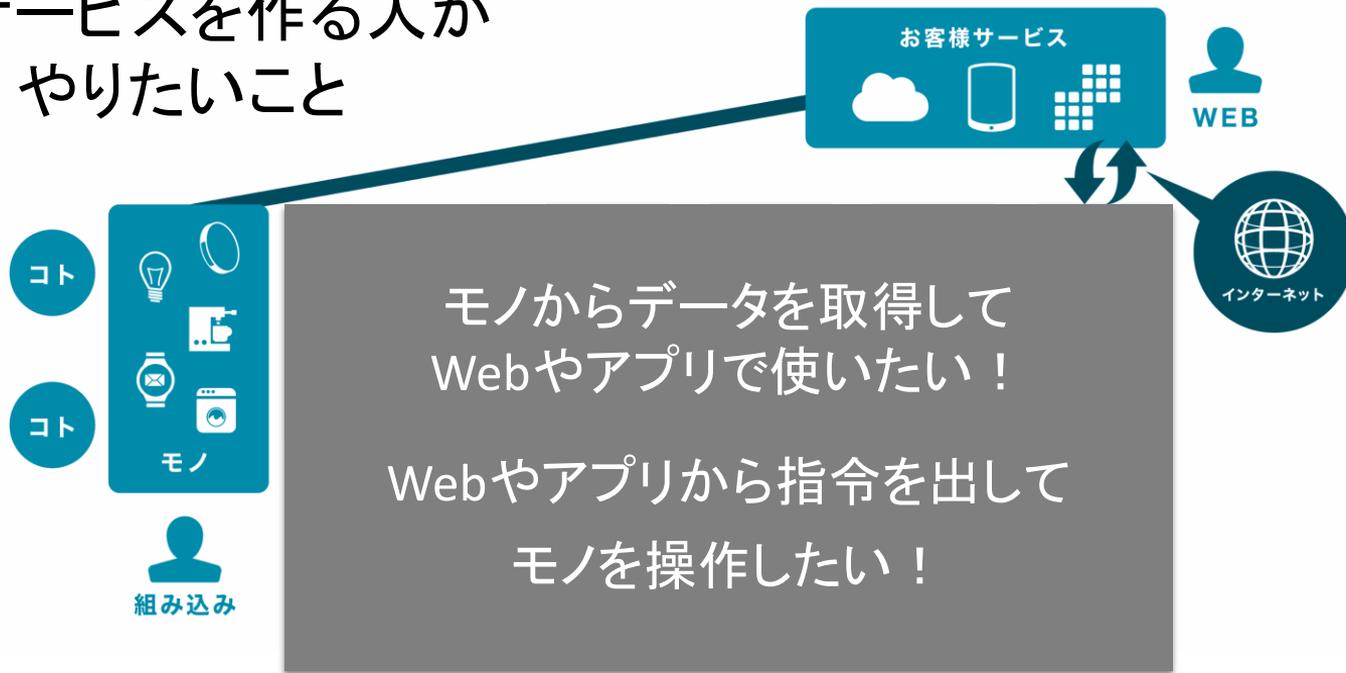
基本機能  
(つなぐ・貯める・連携する)

		セキュアモバイルコネク	sakura.io
デバイス	センサ	-	-
	MCU	-	-
	通信機構	-	○
	回線(SIM)	○	○
クラウド	セキュリティ	○	○
	データの転送	-	○
	データの保管	-	○
	データの連携	-	○

The background of the slide features a long-exposure photograph of a night festival. The sky is filled with numerous white and yellow light trails from stars or distant galaxies. In the foreground, a brightly lit festival float or stage is visible, with various lights and structures. The overall scene is vibrant and festive.

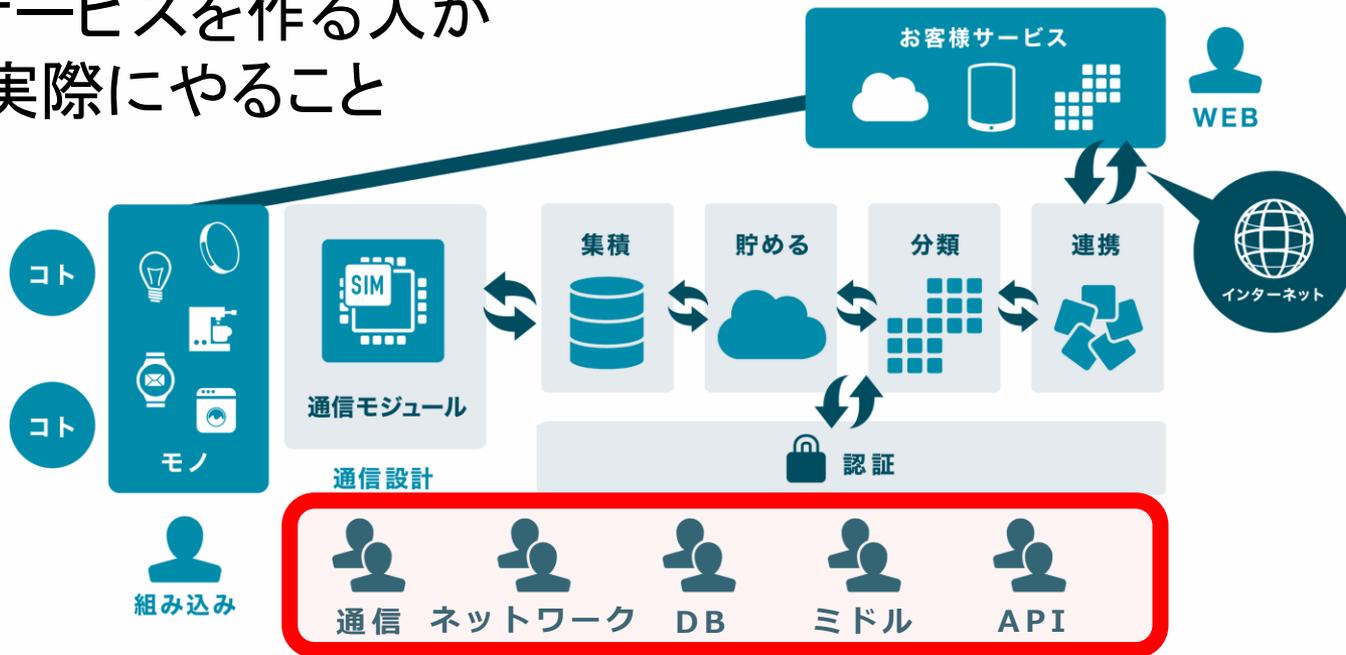
**sakura.io**  
(さくらアイオー)

## IoTサービスを作る人が やりたいこと



モノに関する部分と、Webサービスやアプリの部分だけ作りたい

# IoTサービスを作る人が 実際にやること



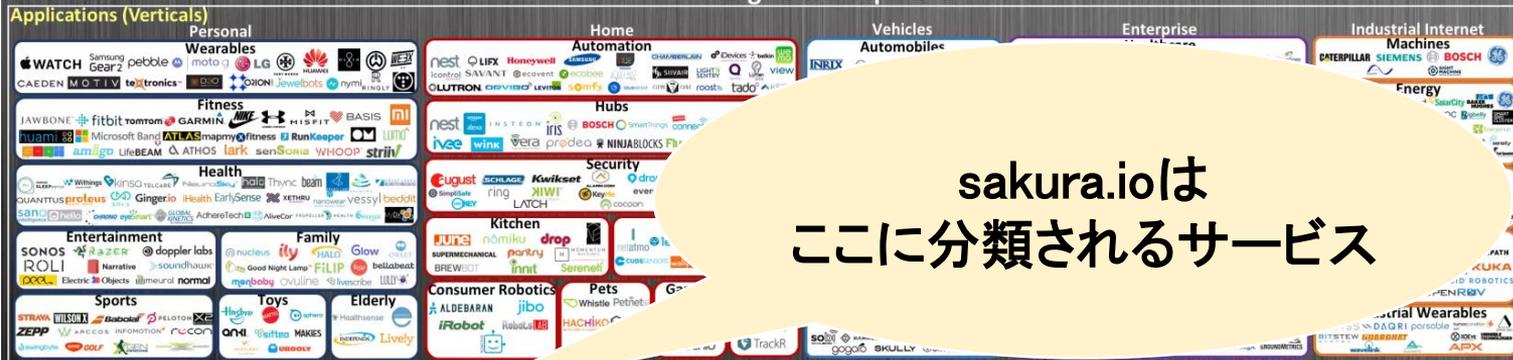
ネットワークとデータをやり取りしたいだけなのに…  
やらなければならないことが多い



モノとWebの間でデータを相互にやりとりするためのプラットフォームサービスを開発



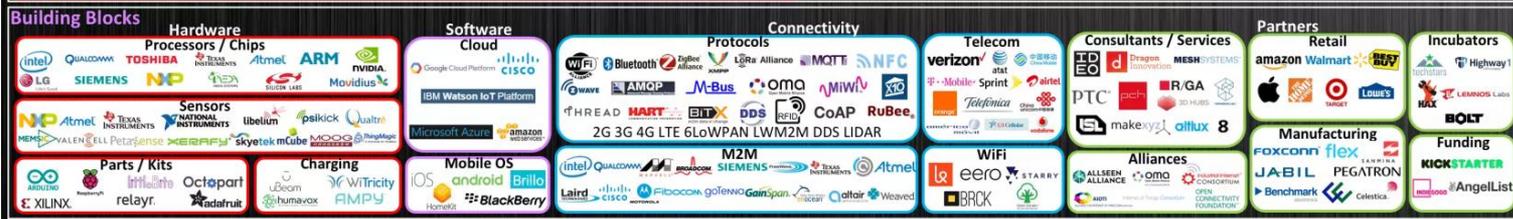
## Internet of Things Landscape 2016



sakura.ioは  
ここに分類されるサービス



ソフトウェア/接続性/分析/セキュリティ  
などのプラットフォームサービス





sakura.io  
サービス概要



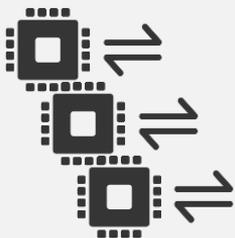
組み込み側はモノとの**電気信号** / Web側はサービスとの**JSONデータ**  
のやり取りに注力可能

		sakura.io	他社IoTプラットフォーム	IoT用MVNO事業者
やりたい	企画・アイデア	<div data-bbox="656 223 1690 420" style="border: 2px solid orange; padding: 10px;"> <p><b>「データを迎えに行く」という発想</b></p> <ul style="list-style-type: none"> <li>・モノからのアウトプットだけでなくモノへのインプットも</li> <li>・電源を入れるだけで利用可能</li> </ul> </div>		
	センサー/チップ			
	デバイス (外装、基板、ファームウェア)			
	サービス (可視化/予測/効率化)		●	
やらねばならぬ	データの送受信手段	●	○	○
	安全な通信経路	●	○	●
	デバイス認証/管理	●	●	●
	プラットフォーム機能 (収集/蓄積/連携)	●	●	○

【やりたい】に注力できるプラットフォームとして提供



sakura.io  
サービス詳細



データの収集



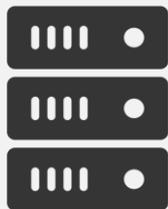
データの蓄積



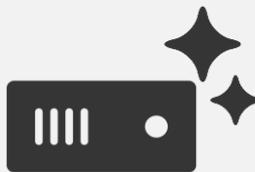
データの連携



運用機能



ラージスケール対応



アップデート



障害切り分け、復旧



セキュリティ

必要な機能・運用をプラットフォームサービスとして提供  
IoTデバイスやサービスごとの基本機能開発や運用設計は不要



LTEモデムチップ

通信制御用MCU

SIMカードコネクタ



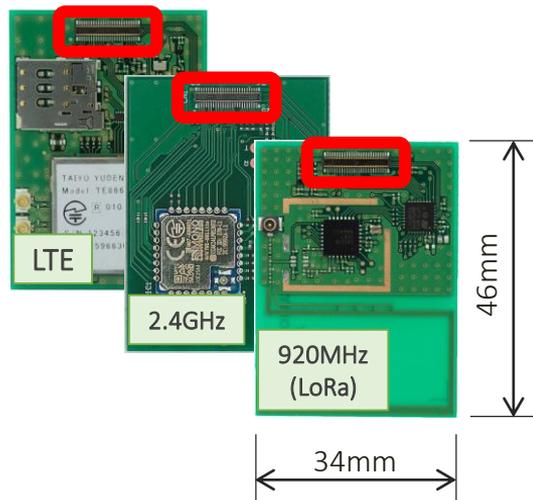
SDカードほぼ2枚分に収まるコンパクトサイズ  
モノ側の通信に必要なすべてを凝縮



IoTデバイス/サービスの開発工数を削減  
「作らなければならないもの」より「作りたいもの」に注力可能

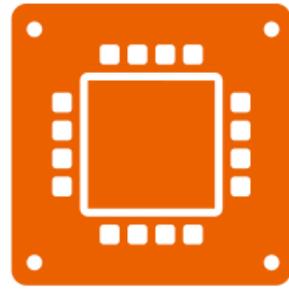


量産性に配慮した  
“基板間コネクタ”を採用



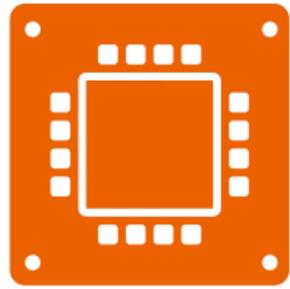
方式	GW	特徴	通信可能レンジ	伝送速度	消費電力
LTE	不要	単独 使用可	キャリア網内 どこでも	速い	大きい
2.4GHz帯	必要	短距離 大容量	数百メートル (最大1km程度)	速い	小さい
920MHz帯 (LoRa)	必要	長距離 小容量	数キロメートル (最大10km程度)	遅い	小さい

共通インターフェースおよび寸法のため  
複数の無線規格への対応が容易



LTE

単体方式 (LTE)



LPWA

ゲートウェイ方式 (LTE+LPWA)



LPWA<->LTE

デバイスは**単体方式**と**ゲートウェイ方式**を提供

# LPWA併用のメリット

1. ラストワンマイル対策
2. デバイスのさらなる省電力化





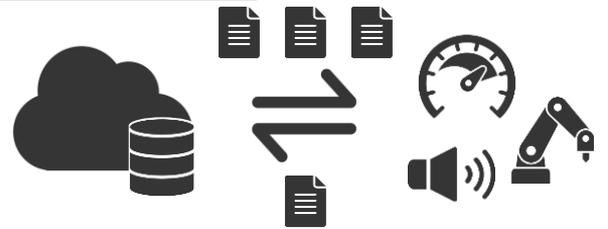
料金	無料	有料(50円/月)*	有料(200円/月)	有料
専有/共有	共有領域	共有領域	共有領域	専有領域
公開有無	公開	非公開	非公開	非公開
閲覧可能期間	40日間	40日間	2年間	制限なし
リリース	未公開	リリース済み	リリース済み	未公開

\*料金は通信モジュール1個あたりの金額となります  
 \*ライトプランは現在無料で提供されています

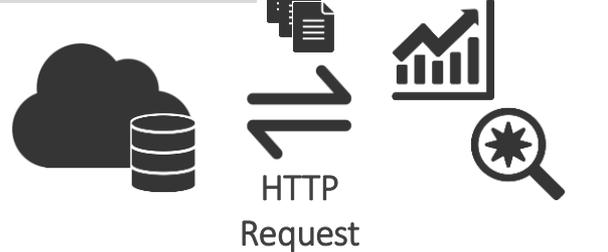
デバイスから送られたデータはポリシーに応じてプラットフォーム内のデータストアに自動で保存される



## リアルタイム連携



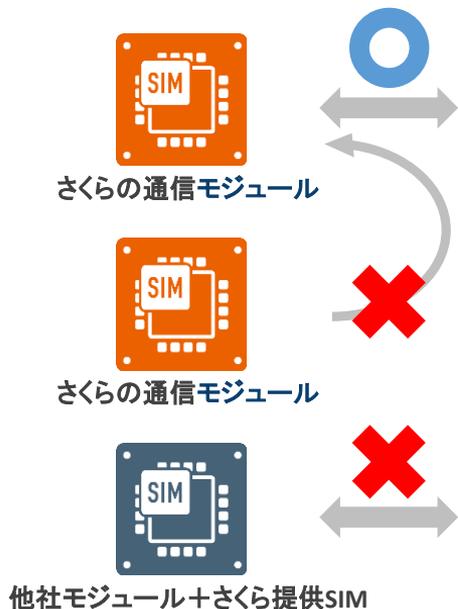
## 一括連携



## JSON形式

```
{
  "module": "XXXXXXXXX",
  "type": "channels",
  "datetime": "2016-06-01T12:21:11.628907163Z",
  "payload": {
    "channels": [{
      "channel": 1,
      "type": "i",
      "value": 1,
      "datetime": "2016-06-01T10:21:11.628907163Z"
    }, {
      "channel": 2,
      ...
    }
  ]
}
```

データ取り出しやデバイス制御はすべてJSONフォーマットで実施  
既存システムや扱いに慣れたクラウドサービスとの接続も可能



デバイスからプラットフォームまでは閉域網経由、他デバイス含め外部アクセスはAPI経由でのみ行うことでセキュリティを担保

- インターネット上のサーバ間の通信はSSLを使用
  - サーバは継続的にアップデート可能
- 組み込み機器との通信は構造上安全にする
  - 閉域網を利用(グローバルネットワークに接続しない)
  - 通信の暗号化と認証は基本的にLTEを利用
  - その上で簡易な暗号化と認証をソフトウェアで実装
  - モノ作りの人にTCP/IPやSSLを実装させない



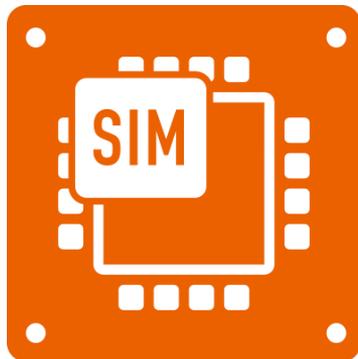
## 時刻提供機能



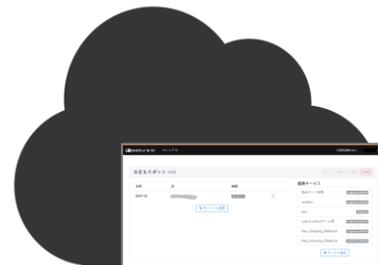
時刻要求



時刻提供



時刻提供



※一部マイコン側での対応が必要です

スケジュールでの一斉動作といった場合でも  
通信モジュールから時刻情報を提供可能



## 簡易位置情報提供機能



※利用には別途月額料金が必要になります

エリアレベルでの大まかな設置場所を収集するような  
ケースであれば通信モジュールの機能として提供



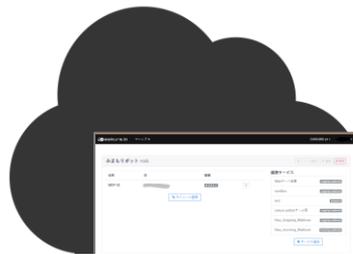
## ファイル配信機能



ファイル要求



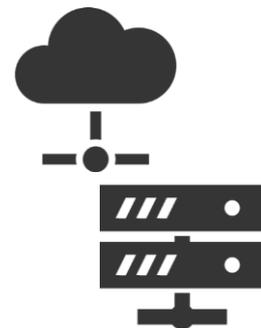
ファイル配信



ファイル要求



ファイル配信



※一部マイコン側での対応が必要です  
※別途通信量に対する課金が発生します

設置済みデバイスに対する遠隔アップデート等  
ソフトウェアな問題への対処を実現

汎用/特定サービスとの連携



世界中で利用できる

Microsoft Azure

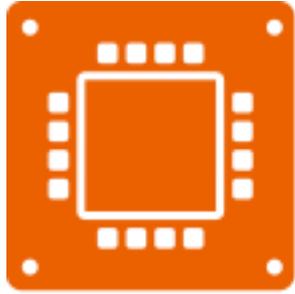


WebSocket



既存システムと連携しやすいフラットな仕組みを  
世界のどこでも使えるように提供

# ご提供価格/方式



つなぐ



貯める



連携

月額1台あたり 60円 / 月 ~

5分に1回の通信は基本料にコミコミ



初期費用 (モジュール購入)

8,000円

月額費用 (回線、プラットフォーム利用)

60円/月 ~

※料金はモジュール単位



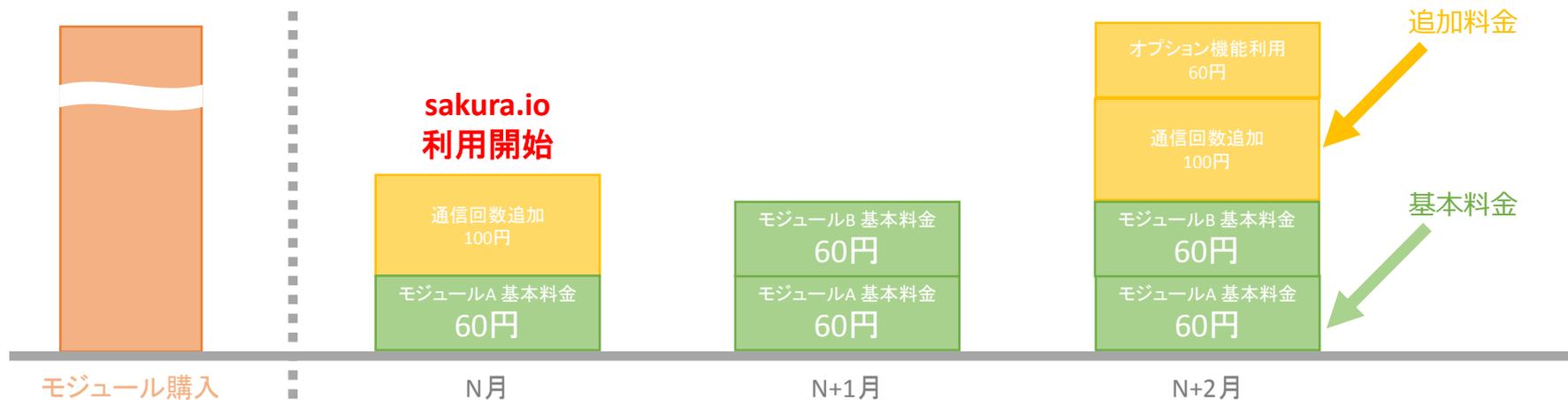
## •基本料金

✓ **60円/月** ※毎月1万回分の通信が可能なポイントを付与

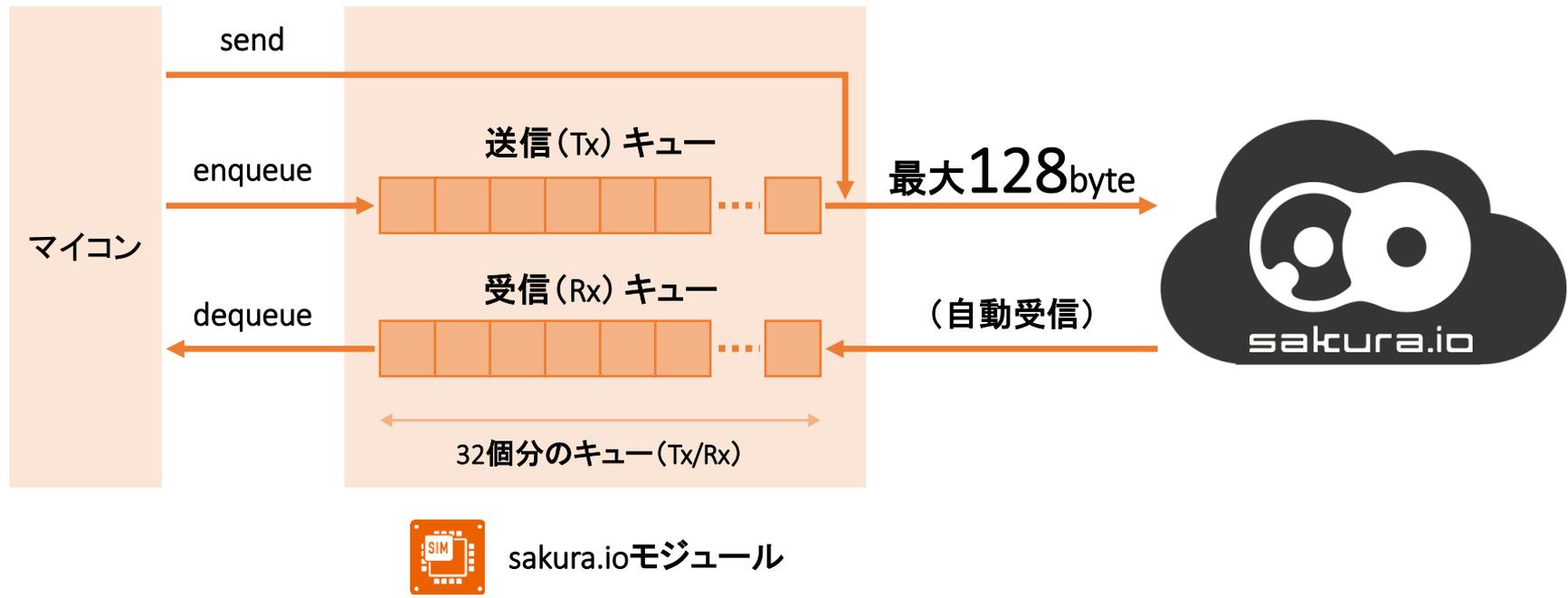
## •追加料金

✓ 通信回数の追加、オプション機能の利用

※特定のオプション機能を利用した場合は別途定額の追加料金がかかります



5分に1回の通信なら、毎月60円で実現  
より幅広いサービスへ適用可能



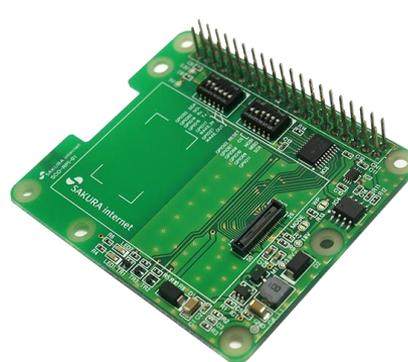
1回の送受信データ(メッセージ)は、最大「128byte (8byte x 16)」  
時間や送信元情報はプラットフォーム受信時に付与、マイコン側対応不要



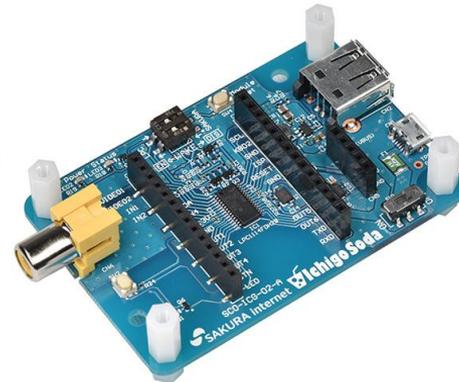
LTE通信モジュール



Arduino用



Raspberry Pi用



IchigoJam用

商品名	必須/オプション	料金
sakura.io モジュール (LTE)	必須	8,000円 登録後 60円/月
sakura.io シールド for Arduino	オプション	5,000円
sakura.io HAT for Raspberry Pi	オプション	5,000円
IchigoSoda / IchigoJam for sakura.io	オプション	5,000円

※モジュール金額は個包装のものです  
 ※すべて税別表記です

提供方式	数量	用途
個包装	1個～	少量生産(数台～十数台) 個人利用/プロトタイプ
トレイ	90個単位	少量～中量生産(数百台) サービス提供開始
ライセンス (プロトコルライセンス)	不問	自由な部品/レイアウト 大量生産製造ライン投入

段階に応じて提供方式を選択可能

# ライセンス方式は通信機能の作り方を提供 「自社ソフトウェアに組み入れたい」に対応

	さくらインターネット	お客さま
開発時	<ul style="list-style-type: none"><li>● プロトコル仕様書</li><li>● ライブラリ／サンプルコード(C言語・Go言語)</li><li>● 製造・検査手順書</li></ul>	<ul style="list-style-type: none"><li>● ファームウェア開発</li><li>● 通信モジュール開発</li><li>● ハードウェア開発</li></ul>
製造時	<ul style="list-style-type: none"><li>● SIM</li></ul>	<ul style="list-style-type: none"><li>● 製造ライン開発</li><li>● 製造</li></ul>



※ ファームウェアのアップデート機構は、お客さまご自身で設計いただく必要があります。

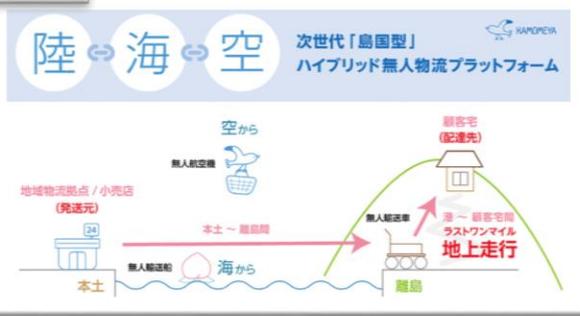
# 電気信号とJSONを相互変換する プラットフォーム



これまでのモノ作り/サービス作りを補完  
現在持っている技術/事業範囲で共創が可能

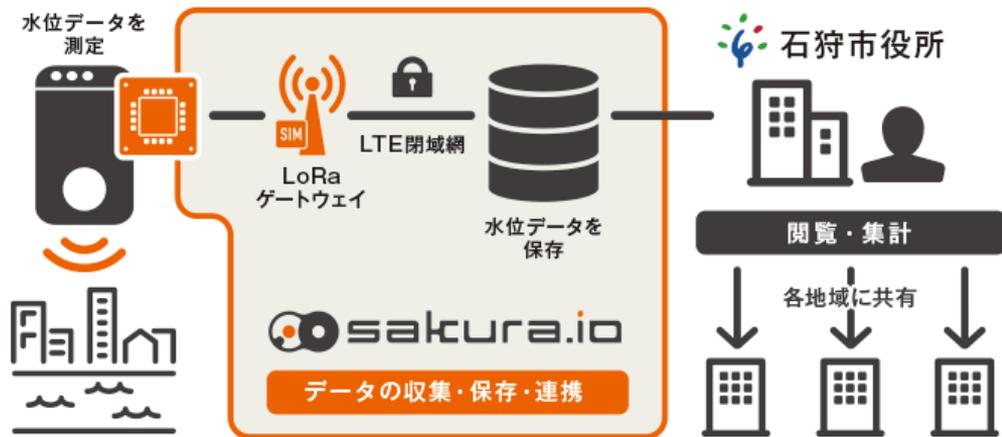
# 利用事例

# 公共 教育 農業 健康 新分野 製造



- スマート水田サービス「Paditch」
- 運営：株式会社 笑農和
- 水門に通信モジュールを装着し遠隔管理
- 水位や時間による水門の開閉を設定可能
- 開発/システム運用コストを重視しsakura.ioを採用
- <http://paditch.com/>





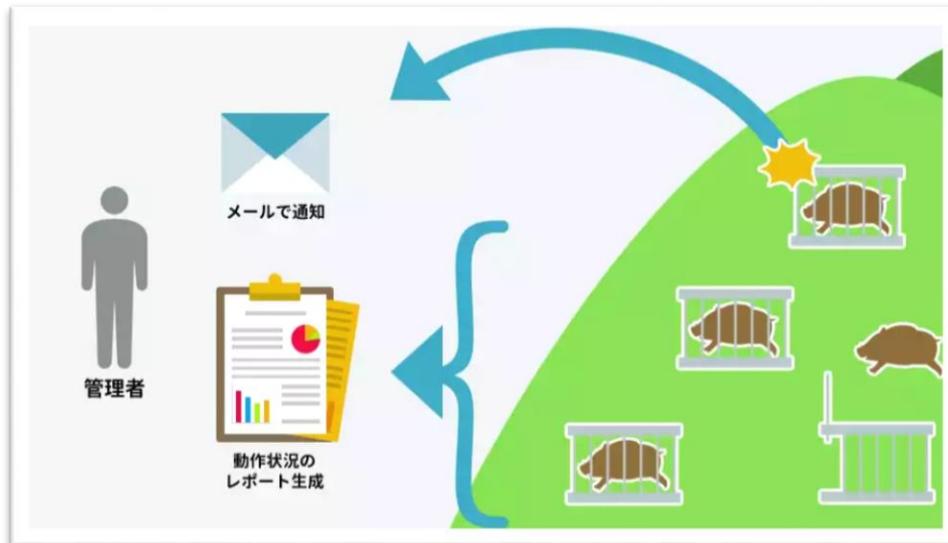
<https://www.sakura.ad.jp/information/pressreleases/2018/03/30/90212/>



<https://sakura.io/blog/2018/10/04/eq-lpwa/>

- 石狩市北部の河川(6か所)の水位計測
- 超音波センサーで測定したデータをsakura.ioに送り可視化
- LPWA採用により低消費電力化(電池で1年間連続稼働予定)

- イノシシやマンゲースの駆除に利用
- 罠に通信モジュールを装着し動作状況を取得
- LTEが不安定な場所ではゲートウェイを利用



<http://pcn.club/katsuyama/azure201703/>

<http://knowledge.sakura.ad.jp/other/7902/>

- 鯖江市のコミュニティバス「つつじバス」
- 運転手の操作盤に通信モジュールを装着し乗客数などを取得
- 集計結果はオープンデータとして活用



[https://www.sakura.ad.jp/press/2017/0330\\_sabae-tsutsujibus/](https://www.sakura.ad.jp/press/2017/0330_sabae-tsutsujibus/)

- シニア向けIoTデバイス「MAGOボタン」
- 制作：MIKAWAYA21株式会社
- 音声入出力とボタンを装備
- 暮らしの情報を音声で通知
- ボタンを押すと安否確認や固定メッセージを送信
- <http://pr.mago-btn.com/>





KAZAMIDORI

sakura.io  
products

リアルタイム気象情報提供サービス

その場所の “いま” をお知らせします。

風向

風速

気温

湿度

気圧

雨量

- 運営：株式会社かもめや
- 無人輸送{機/船/車}に通信モジュールを装着して制御
- 飛行機ではLTEが使えないのでLoRaを使用
- <http://www.kamomeya-inc.com/>





- 3G/LTE搭載鍵デバイス「TiNK」
- 制作：株式会社 tsumug <https://www.tsumug.com/>
- テンキーで暗証番号を入力しsakura.io経由で認証
- 全国の賃貸住宅への導入を目指す

merchari

How to Merchari

Debut!

Port Owners

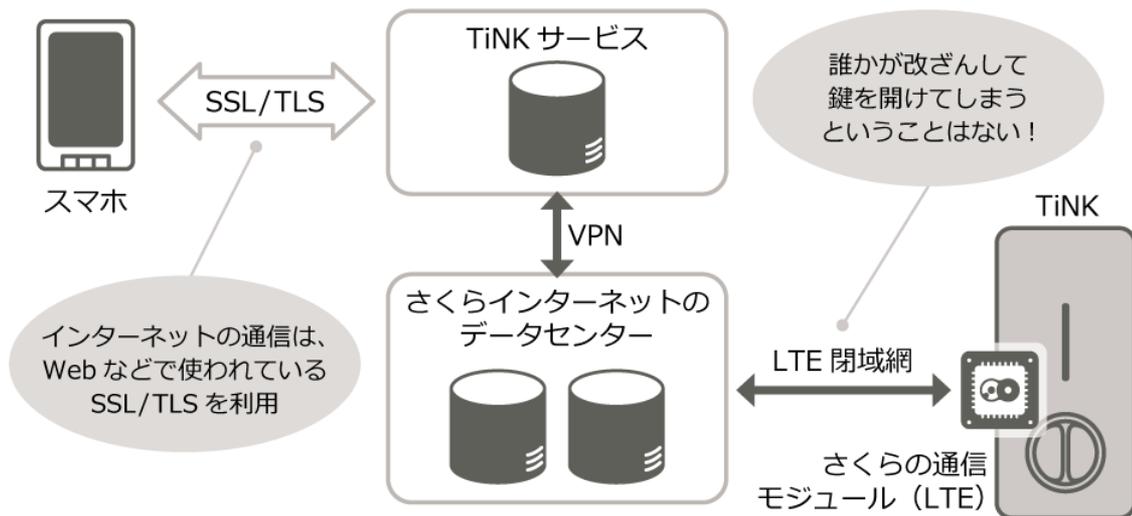
Recr

sakura.io  
products



おまたせしました、メルチャリです。  
いよいよあなたの足元に。

 App Store  
からダウンロード



<http://edge.tsumug.com/entry/tinktech-sakuraio>

- メルカリが運営するシェアサイクルサービス
- アプリで鍵のQRコードを読み取るとsakura.io経由で解錠
- 福岡市内でサービス中

# さくらのセキュアモバイルコネク

		セキュアモバイルコネク	sakura.io
デバイス	センサ	-	-
	MCU	-	-
	通信機構	-	○
	回線(SIM)	○	○
クラウド	セキュリティ	○	○
	データの転送	-	○
	データの保管	-	○
	データの連携	-	○



「さくらのクラウド」のローカルネットワークに直結

## 医療／金融



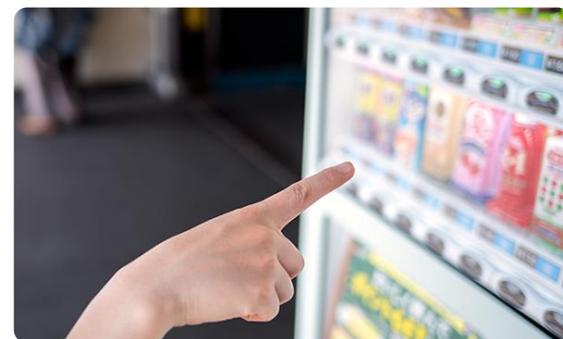
患者や顧客の情報を取り扱う端末も、イントラ内のシステムにのみ安全に接続可能

## ドライブレコーダー



普段は位置情報を閉域網経由で安全に収集、急制動や衝突時のみ映像をアップロード

## 自動販売機



従来の決済情報や在庫情報に加え、利用者の年齢や性別を加味した広告配信にも

# データ使用量に波がある場合に

### 初期費用



SIM本体価格

**2,160** 円/枚



基本利用料

SIM1枚あたり **12** 円/月



モバイルゲートウェイ

**4,320** ※ 円/月

SIMカード最大1万枚まで登録可能

500MBの  
データ通信費  
を含みます



さくらのクラウド  
スイッチ利用料

**2,160** ※ 円/月



データ通信料

**6** 円/MB

100枚の場合：7,680円～ (1枚**76.8円**)

10,000枚の場合：126,480円～ (1枚**12.7円**)

小規模導入でも回線の維持コストは数十円

マルチサイズSIM  
(標準/micro/nano)



スロットサイズに合わせてカット

チップSIM  
(MFF2)



製造ラインへの直接投入

用途に合わせて選択可能



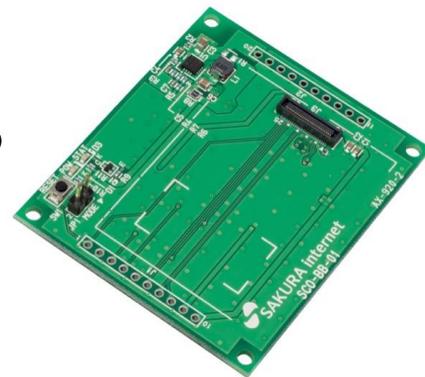
# sakura.ioを 使ってみる

# 電気信号とJSONを相互変換する プラットフォーム



sakura.ioの両端はオープンな通信方式とデータ形式を採用

- I2C/SPIで通信できればよい
- マイコンとはボードで接続
  - Arduino / Raspberry Pi / IchigoJam
- ライブラリやサンプルプログラムも公開
  - <http://www.arduino-libraries.info/libraries/sakura-io>
  - <https://github.com/sakuraio>
- ブレイクアウトボード(検証ボード)もあり



- sakura.ioで用意したAPIによりリクエストを送出
- 対応プロトコル/サービス
  - Websocket
  - Webhook (Incoming / Outgoing)
  - MQTT Client
  - DataStore API
  - AWS IoT
  - Azure IoT Hub / Azure Event Hubs
  - Google Cloud Pub/Sub Publisher
- URLの例
  - `wss://api.sakura.io/ws/v1/(ID)`
  - `https://api.sakura.io/incoming/v1/(ID)`
- JSON形式のデータが返ってくる
- お好みのプログラミング言語でJSONデータを処理
  - 主要な言語はJSONを扱うライブラリあり

```
{
  "module": "XXXXXXXXXX",
  "type": "channels",
  "datetime": "2016-06-01T12:21:11.628907163Z",
  "payload": {
    "channels": [{
      "channel": 1,
      "type": "i",
      "value": 1,
      "datetime": "2016-06-01T10:21:11.628907163Z"
    }, {
      "channel": 2,
      ...
    }
  ]
}
```



# デモその1 温度・湿度の測定

**Draft sakura.io 体験ハンズオン (単独開催版) / sakuraio handson part1**

SAKURA internet

だれもが、データを活かせる世の中へ。

# sakura.io 体験ハンズオン

<https://sakura.io>  
さくらアイオー

<https://www.sakura.ad.jp/>

DAY	COMPANY	DEPARTMENT	NAME
2018/06/19	さくらインターネット株式会社	IoTチーム	西田 有騎

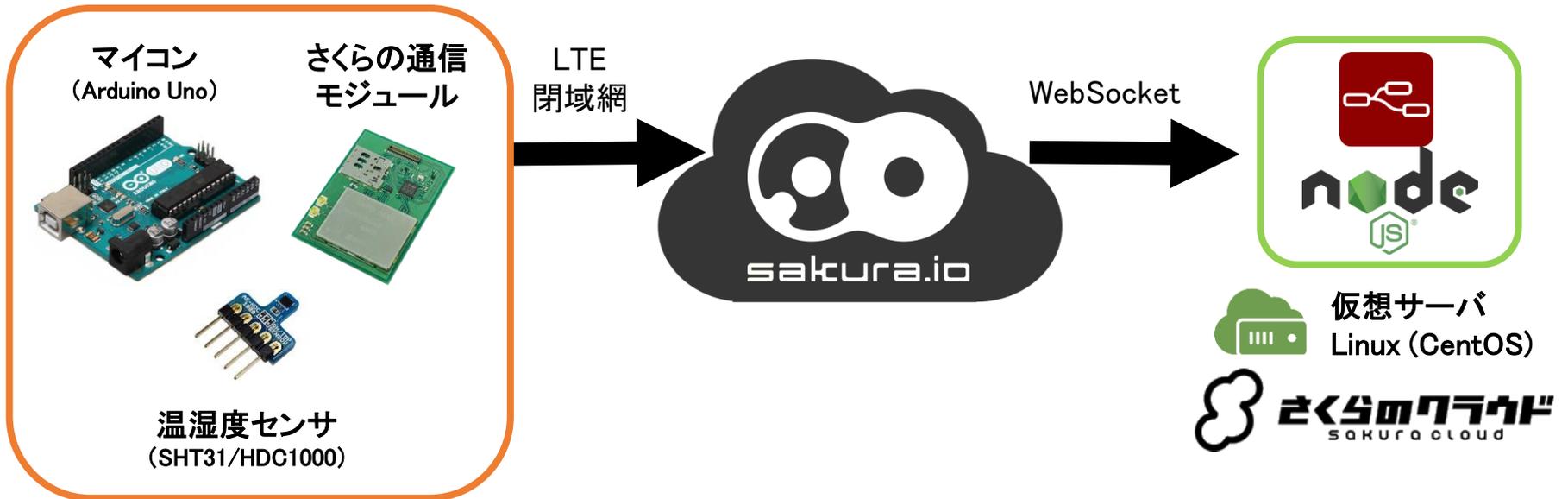
(C) Copyright 1996-2018 SAKURA Internet Inc.

Yuuki Nishida June 18, 2018 Technology 0 0

<http://bit.ly/sakuraio-handson-part2-01>

温湿度データを取得し  
sakura.ioへ送付

Websocketでデータを入力し  
Node-REDで処理



## 1. sakura.ioの設定

- プロジェクトの作成
- さくらの通信モジュールの登録
- 連携サービス(Websocket)の設定

## 2. 機器の配線とマイコンのプログラム開発

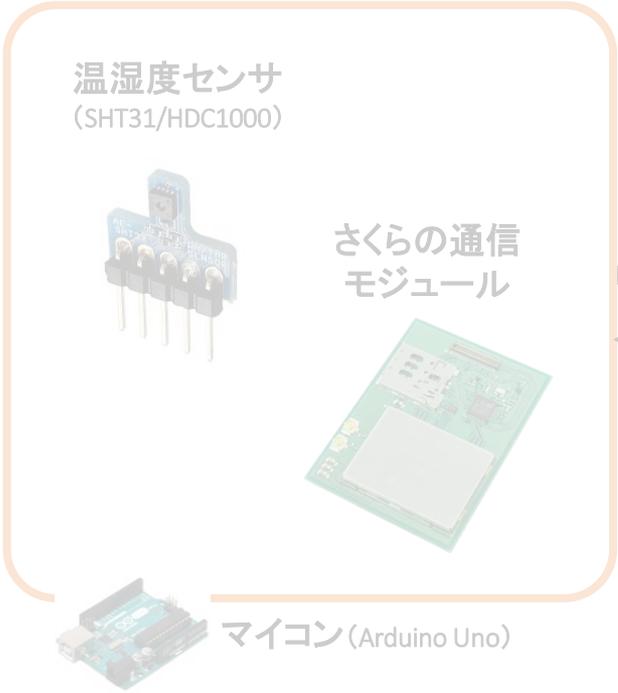
- 通信モジュールとArduinoシールドを接続
- 温湿度センサーからの出力をArduinoに取り込めるよう配線
- 温湿度情報をsakura.ioに出力するArduinoのプログラムを作成

## 3. サーバの作成とデータ加工処理の開発

- Node-REDサーバを作成
- 温湿度データを加工するフローを作成
- WebSocketでデータを入手し、Node-REDで加工してグラフ表示 & Twitterに投稿

②

マイコンおよび  
プログラムの構築



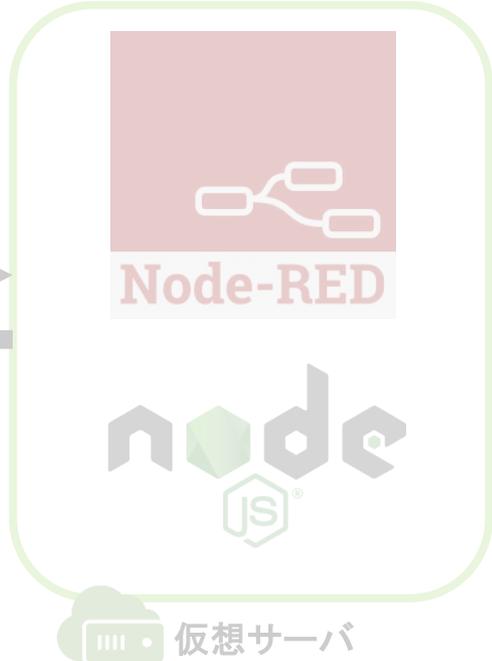
①

sakura.ioの設定



③

Webサービス連携  
(さくらのクラウド)



プロジェクトの登録

New Project #2321

ファイル配信 編集 削除

名称	ID	接続
New Module	[REDACTED]	オンライン

モジュール登録

連携サービス

New Service websocket

+ サービス追加

通信モジュールの登録

連携サービスの登録

連携サービスとして、WebSocketなどのオープンな通信プロトコルや、AWS IoTなどのようなクラウド事業者のサービスを選択することができます。

## 追加サービスの選択

WebSocket

Outgoing Webhook

Incoming Webhook

MQTT Client

DataStore API

AWS IoT

Azure IoT Hub

連携サービスとしてWebSocketを選択すると、WebSocketのURLが設定されます。このURLにアクセスすると、sakura.ioとの間でJSON形式のデータを送受信します。

### サービス連携の編集 WebSocket #3656

名前

New Service

URL

wss://api.sakura.io/ws/v1/

Token

削除

保存

②

## マイコンおよびプログラムの構築

温湿度センサ  
(SHT31/HDC1000)



さくらの通信  
モジュール



マイコン (Arduino Uno)

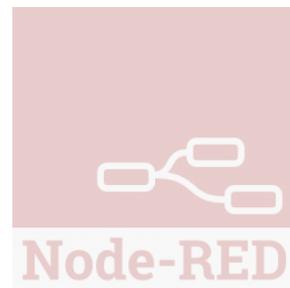
①

sakura.ioの設定



③

Webサービス連携  
(さくらのクラウド)



仮想サーバ

通信モジュールの下にArduinoシールドを敷き、さらにその下にArduinoを接続します。  
通信モジュールにはLTEアンテナを取り付けます。

LTEアンテナ

Arduinoシールド

通信モジュール

Arduino

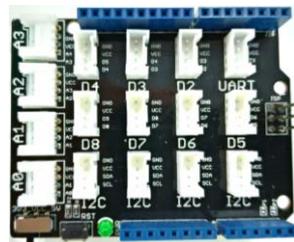


Grove ベースシールドを取り出し、ピン位置に注意しながら組み上げ済みキットに取り付けます。

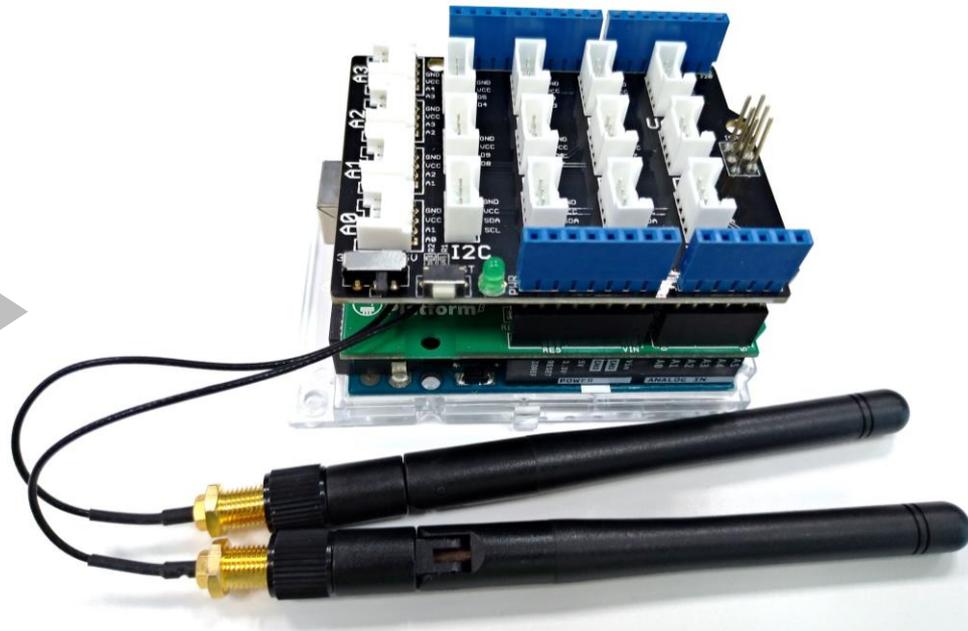


組み上げ済みキット

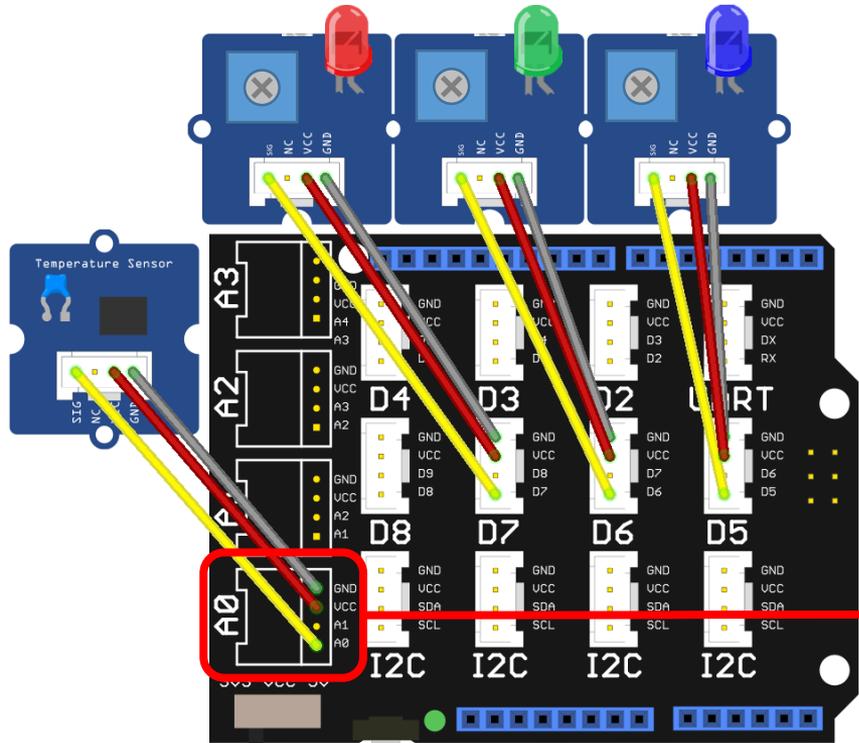
+



Grove ベースシールド

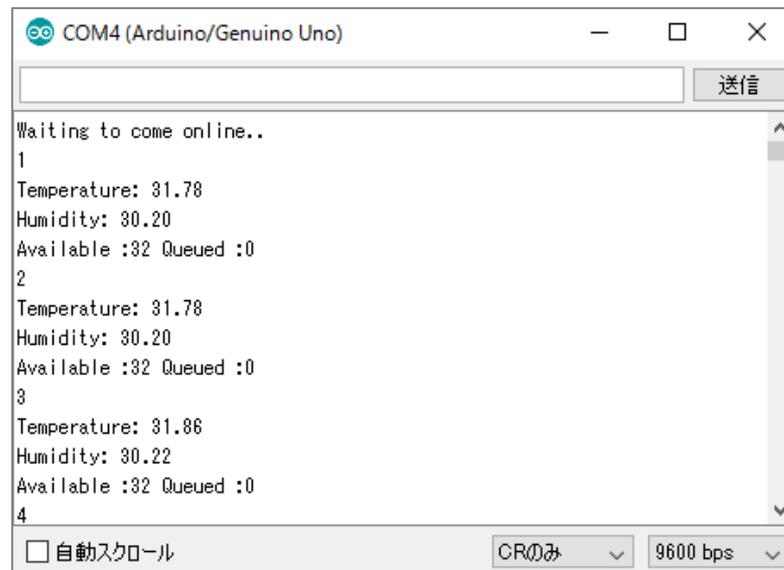
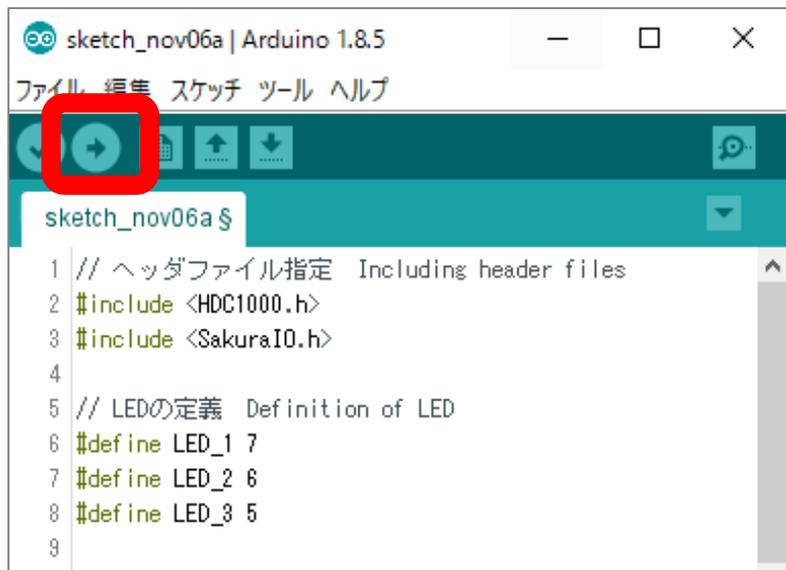


図に従い、Groveケーブルを使用して温度センサを配線します。



A0: 温度センサ

Arduinoにてsakura.ioを制御するためのライブラリをインストールし、  
温湿度センサーの情報を送信するプログラムをArduinoに書き込みます。  
プログラムが動作すると、シリアルモニタに温度・湿度・カウンタ値などが表示されます。



sakura.ioの管理画面では、デバイスから送信されたデータがリアルタイムで表示されます。

時刻:  
データがモジュールのキューに格納された時刻のタイムスタンプ

モジュール:  
データを送信した通信モジュールのID

チャンネル:  
データが格納されたチャンネル番号

型:  
データの型式

値:  
送信された値

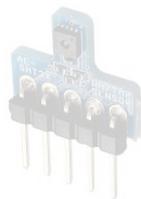
詳細モードに切り替え

時刻	モジュール	チャンネル	型	値	
2017-09-19T03:32:35.760758677Z	[REDACTED]	0	f	27.883957	→ 温度
2017-09-19T03:32:35.782758677Z	[REDACTED]	1	f	43.07469	→ 湿度
2017-09-19T03:32:35.804758677Z	[REDACTED]	2	l	21889	→ カウント値
2017-09-19T03:32:03.852397653Z	[REDACTED]	0	f	27.91066	
2017-09-19T03:32:03.874397653Z	[REDACTED]	1	f	42.899216	
2017-09-19T03:32:03.896397653Z	[REDACTED]	2	l	21888	

②

マイコンおよび  
プログラムの構築

温湿度センサ  
(SHT31/HDC1000)



さくらの通信  
モジュール



マイコン (Arduino Uno)

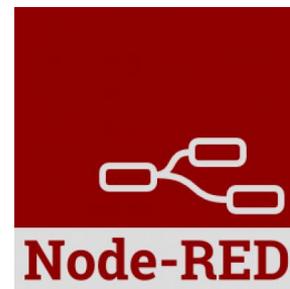
①

sakura.ioの設定

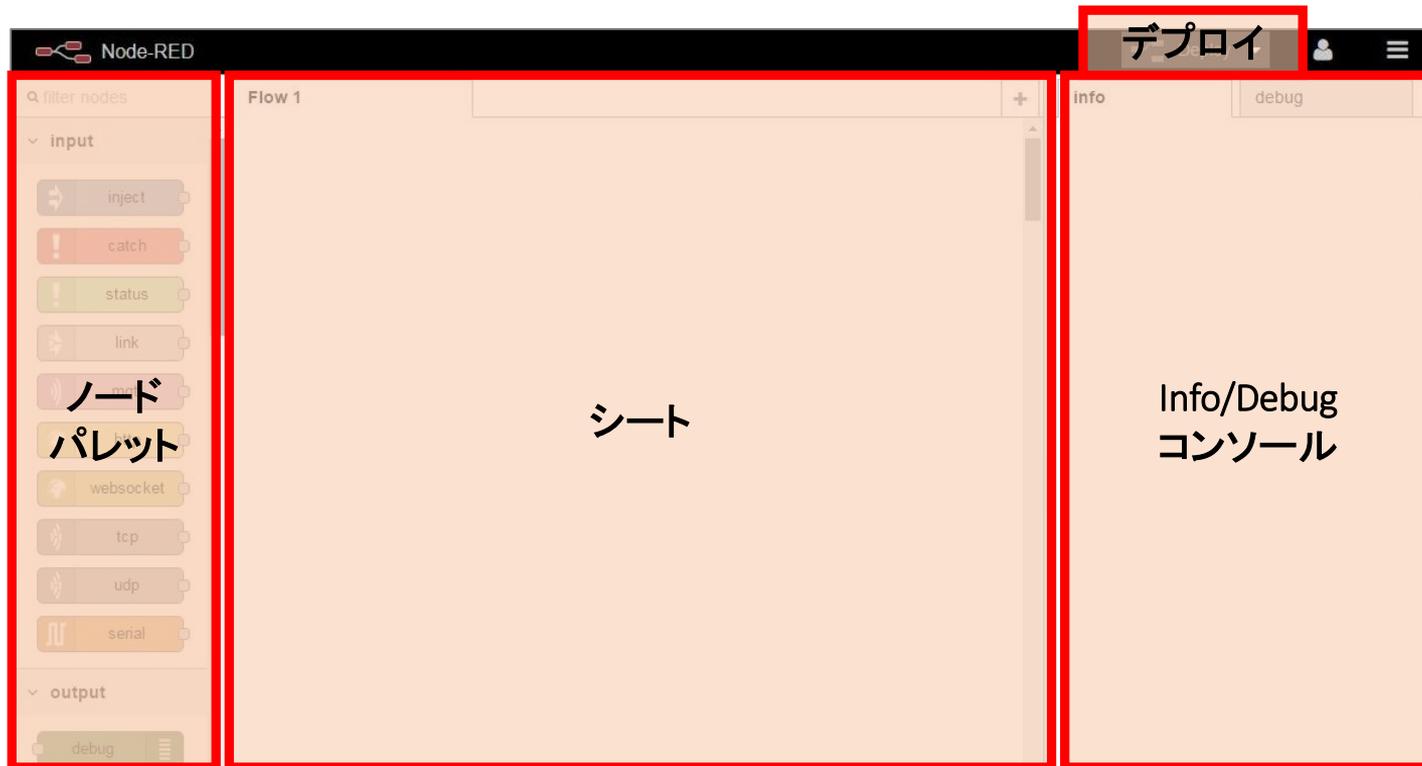


③

Webサービス連携  
(さくらのクラウド)



Node-REDは「ノード」と呼ばれる機能の固まりをシート上で組み合わせ、ひとつの「フロー」にすることで、ほとんどプログラミングを知らない人でもプログラムを構築することができるツールです。





## 作業概要

- サーバを作成 (ここではCentOSを使用)
- Node関連プログラムのリポジトリを登録
- Node.jsのインストール
- Node-REDのインストール
- Node-REDの自動起動設定

さくらのクラウドには、サーバ作成時に任意のスクリプトを自動実行する「スタートアップスクリプト」機能があります。  
スタートアップスクリプトにNode-REDを指定することにより、前ページに掲げた作業がすべて自動的に実行され、Node-REDサーバを簡単に作ることができます。



スタートアップスクリプト

なし  shell  yaml\_cloud\_config

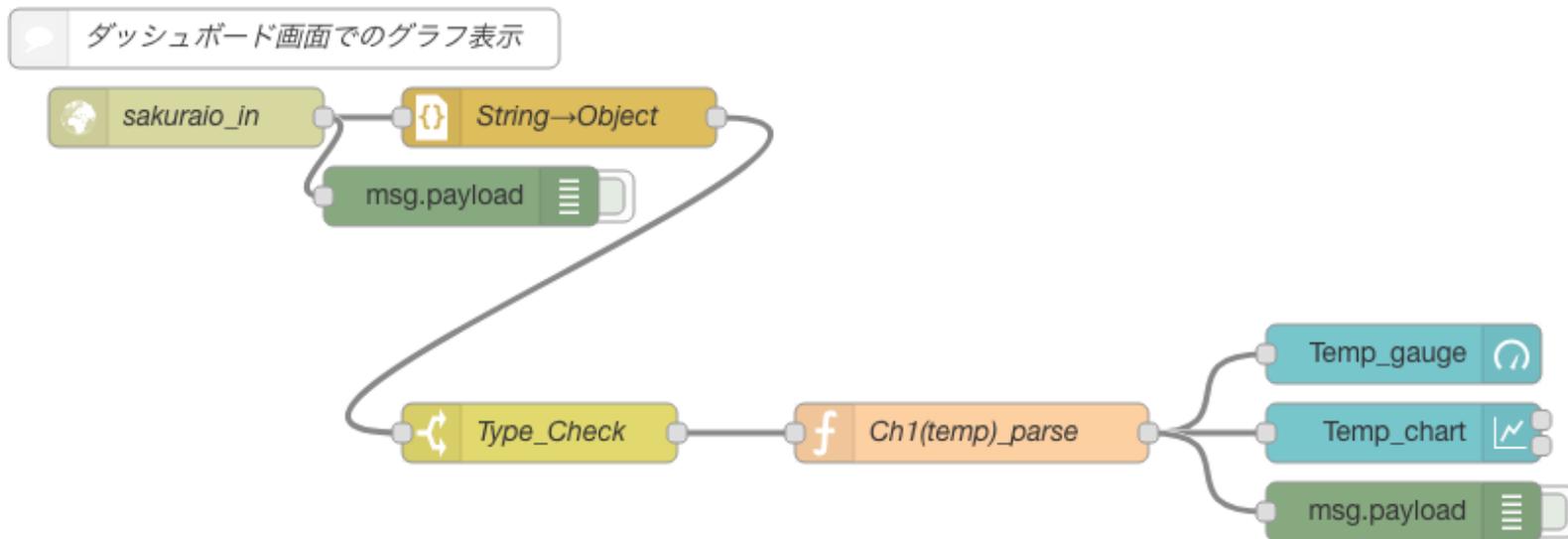
詳細は[技術仕様](#)をご確認ください

配置する スタートアップスクリプト

[public] Node-RED #112900523333

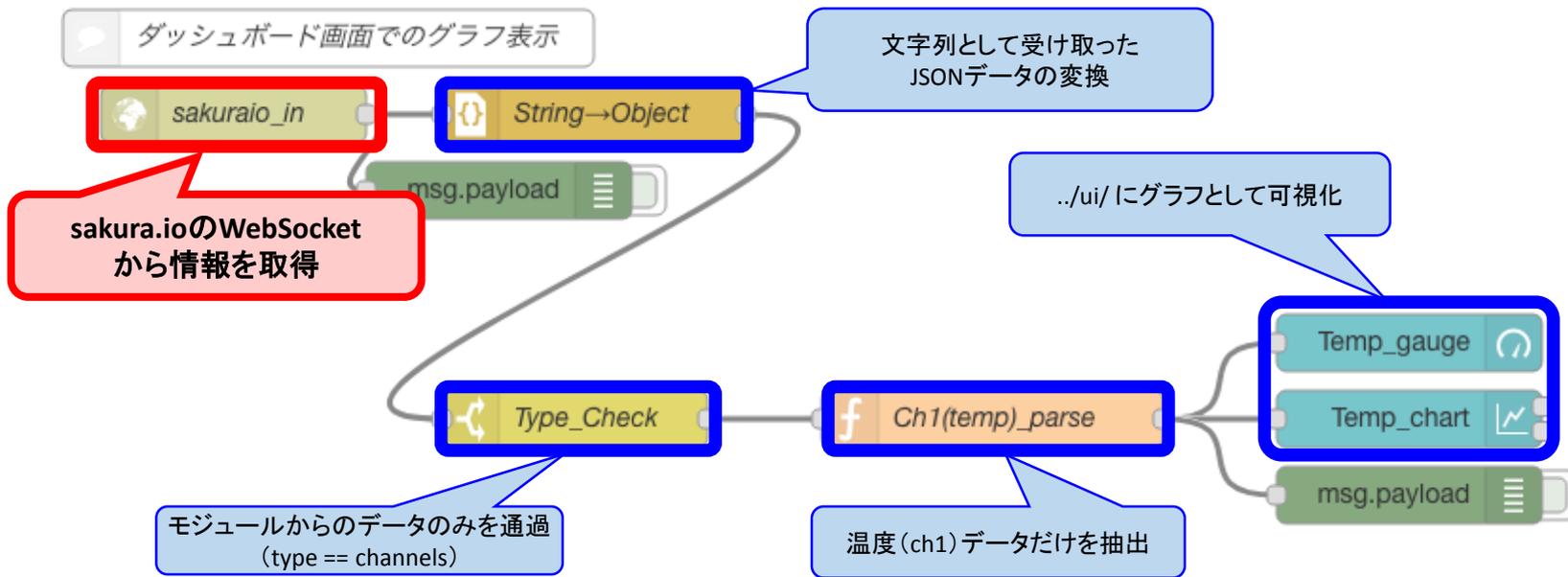
• NVM/Node.js/Node-REDのインストールを実行します。  
このスクリプトは、CentOS 7.xでのみ動作します。  
完了後「http://<IPアドレス>:1880/」にWebブラウザからアクセスできます。  
UIポート番号を指定した場合は、指定したポート番号でアクセスできます。  
Node-Redのログを確認するには「pm2 logs node-red」コマンドを実行します。

sakura.ioからWebSocketで温湿度データを手し、加工して表示するフローを作成します。  
WebSocketのURLを設定し、デプロイすると動作します。

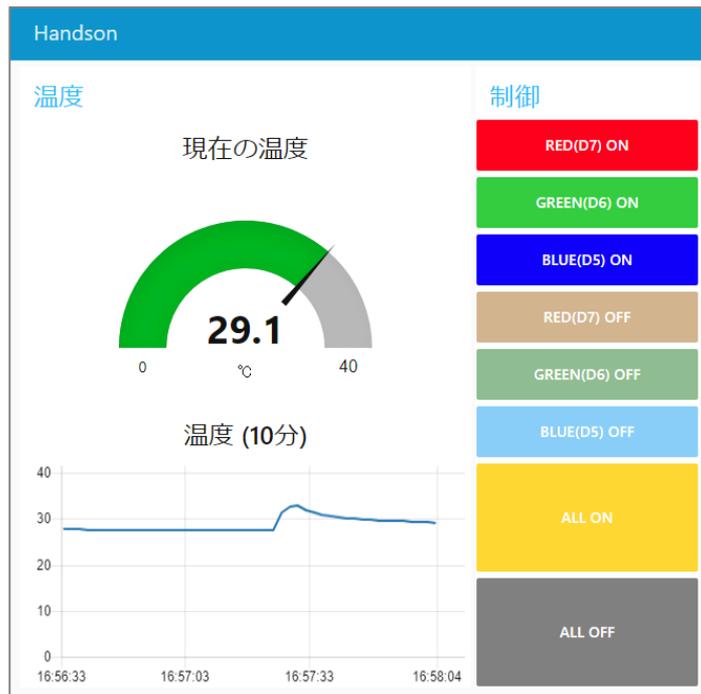


<https://github.com/sakuraio/handson-sample/blob/reform/firststep/part2/nodered-firststep-part2.json>

フローの内部ではこのような処理をしています。



【 <http://<サーバのIPアドレス>:<指定したWeb UIポート番号>/ui/> 】にアクセスすると、取得した情報に応じて動的にグラフが生成されることを確認できます。





# デモその2 LEDの操作

sakura.ioからデータを受信し  
3色のLEDを点灯/消灯

Node-REDのGUIでLEDを操作し  
WebSocketにてデータを送信

抵抗入りLED



さくらの通信  
モジュール

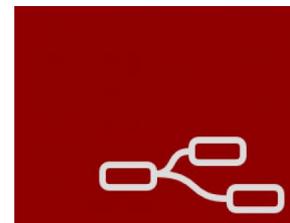


マイコン (Arduino Uno)

LTE  
閉域網



WebSocket



Node-RED



仮想サーバ

## 1. sakura.ioの設定

- デモその1と同じ

## 2. 機器の配線とマイコンのプログラム開発

- 通信モジュールとArduinoシールドを接続
- Arduinoからの信号がLEDに伝わるように配線
- sakura.ioから入手したデータによりLEDを操作するArduinoのプログラムを作成

## 3. サーバの作成とLED操作フローの開発

- Node-REDサーバを作成
- LEDを操作するフローを作成しデプロイ
- Node-REDのGUIを操作しLEDを点灯/消灯

②

マイコンおよび  
プログラムの構築

抵抗入りLED



さくらの通信  
モジュール



マイコン (Arduino Uno)

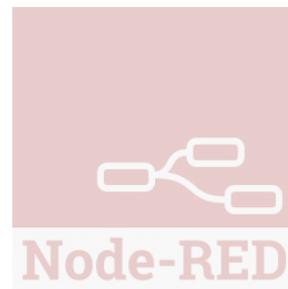
①

sakura.ioの設定



③

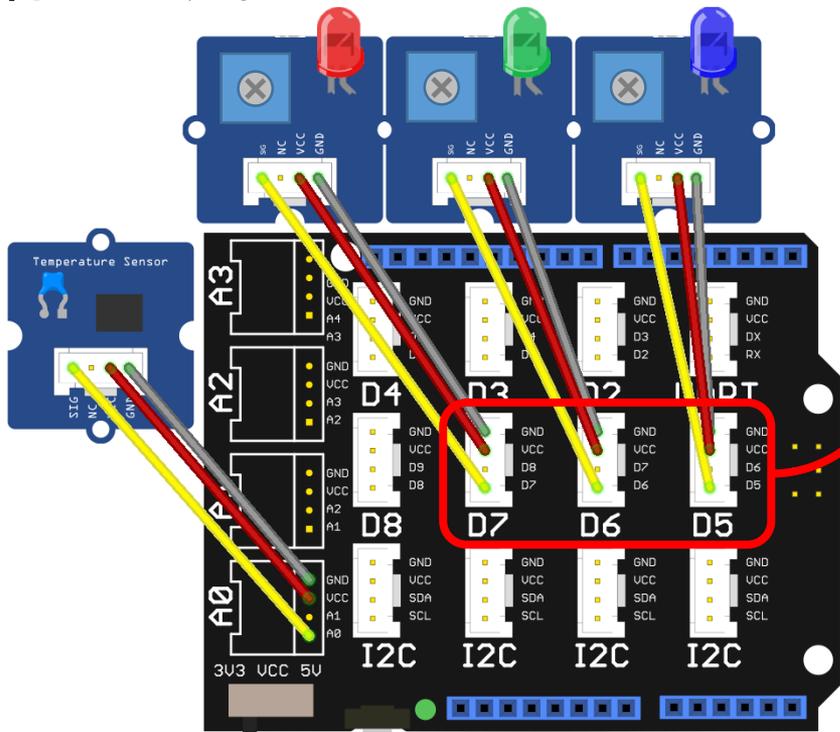
Webサービス連携  
(さくらのクラウド)



仮想サーバ

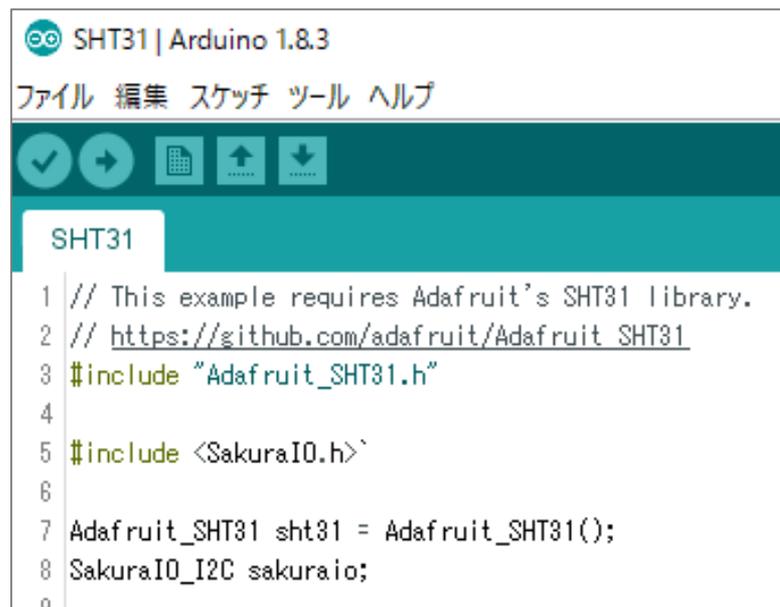


図に従い、Groveケーブルを使用してLED Socket Kitを配線します。

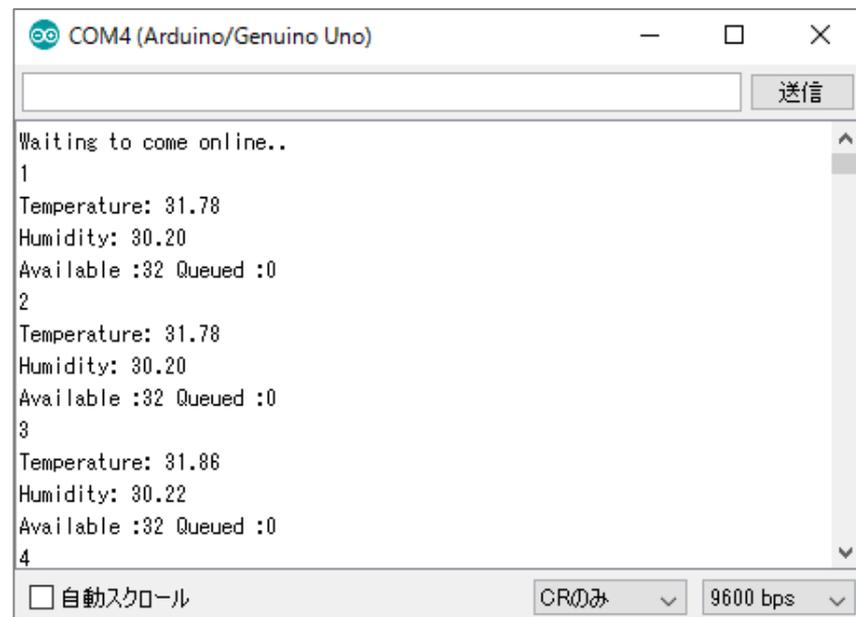


- D7: LED (赤)
- D6: LED (緑)
- D5: LED (青)

sakura.ioから受信したデータに応じてLEDに信号を送るプログラムをArduinoに書き込みます。  
プログラムが動作すると、シリアルモニタに結果が表示されます。  
(LEDに関する情報は出ません)



```
SHT31 | Arduino 1.8.3
ファイル 編集 スケッチ ツール ヘルプ
SHT31
1 // This example requires Adafruit's SHT31 library.
2 // https://github.com/adafruit/Adafruit_SHT31
3 #include "Adafruit_SHT31.h"
4
5 #include <SakuraIO.h>
6
7 Adafruit_SHT31 sht31 = Adafruit_SHT31();
8 SakuraIO_I2C sakuraio;
9
```



```
COM4 (Arduino/Genuino Uno)
送信
Waiting to come online..
1
Temperature: 31.78
Humidity: 30.20
Available :32 Queued :0
2
Temperature: 31.78
Humidity: 30.20
Available :32 Queued :0
3
Temperature: 31.86
Humidity: 30.22
Available :32 Queued :0
4
 自動スクロール
CRのみ
9600 bps
```

②

マイコンおよび  
プログラムの構築



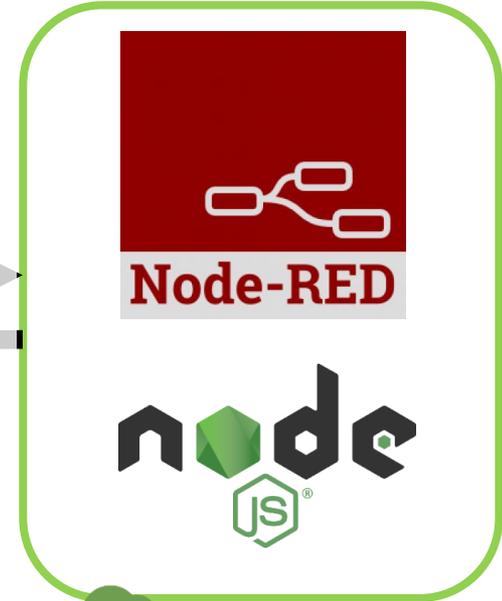
①

sakura.ioの設定



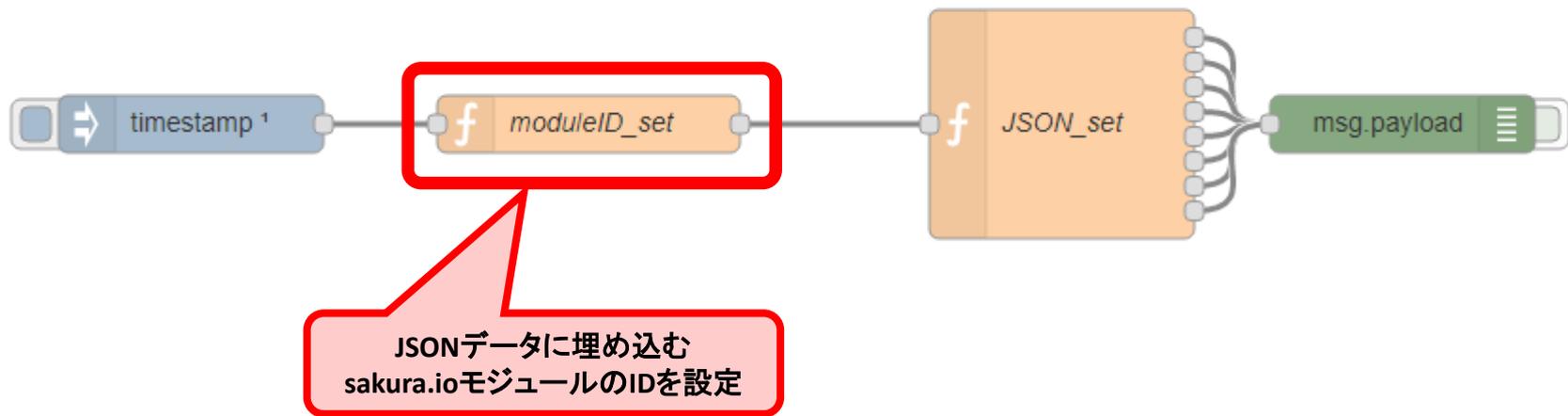
③

Webサービス連携  
(さくらのクラウド)



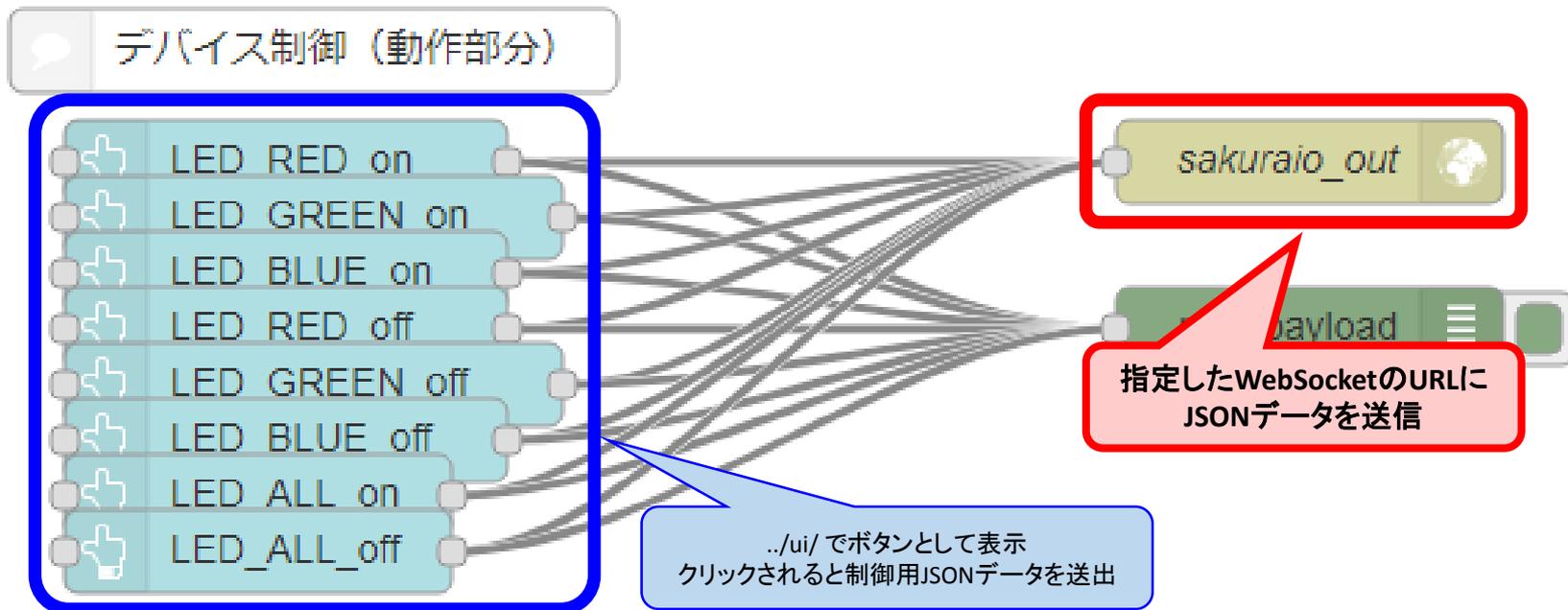
- LEDが接続されているsakura.ioモジュールのIDを指定します。
- このフローはデプロイされた時に1回だけ動作し、用途に応じた制御用JSONデータを8個作成し、それらを変数にセットします。

デバイス制御 (モジュールIDの指定と制御用JSONデータの定義)

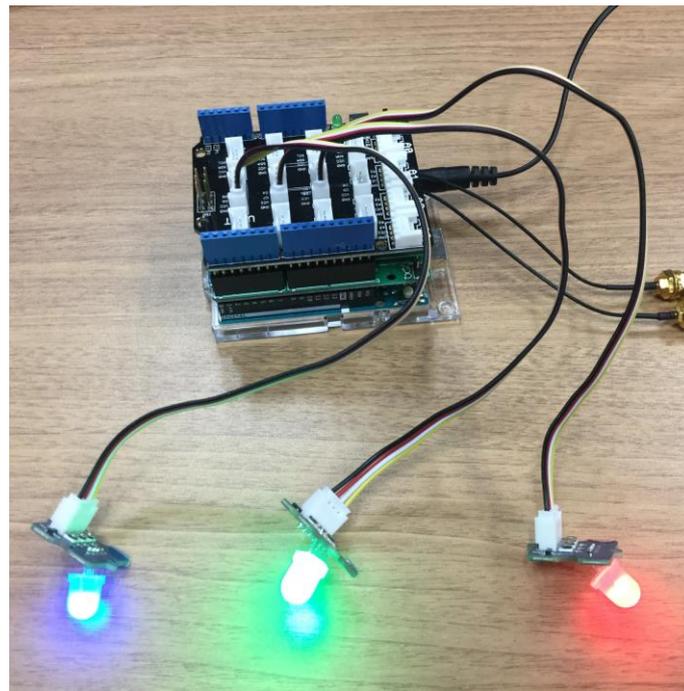
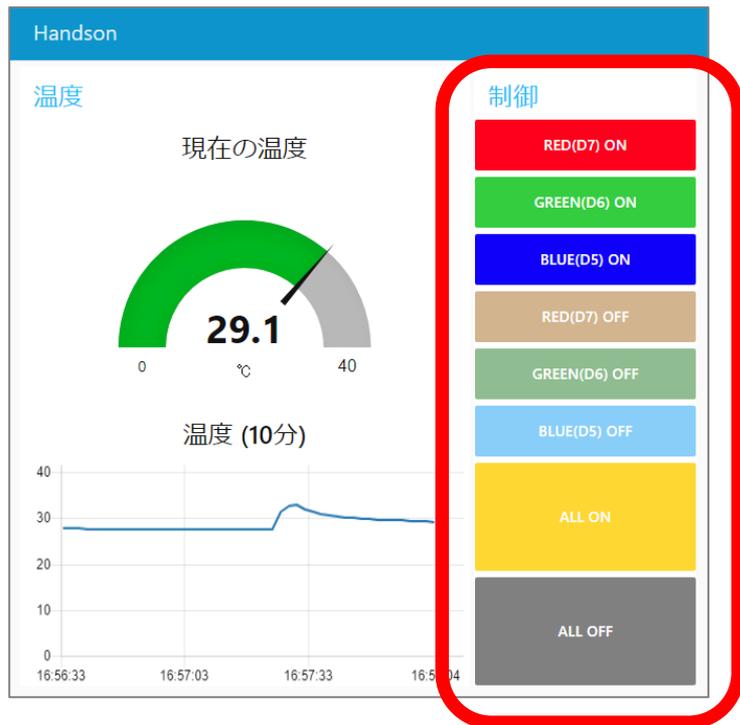




- sakura.ioからJSONデータを送信するWebSocketのURLを指定します。
- ダッシュボード画面でボタンが押下されると、制御用JSONデータ呼び出し、指定したWebSocketに送信します。



【 <http://<サーバのIPアドレス>:<指定したWeb UIポート番号>/ui/> 】にアクセスし、最右列のボタンをクリックすると、それに応じてLEDが点灯/消灯します。





# デモその3 パトランプの操作

sakura.ioからデータを受信し  
パトランプを点灯/消灯

パトランプ



sakura.io  
モジュール



LTE  
閉域網



WebSocket

GUIでボタンを操作し  
WebSocketにてデータを送信

sakura.io 積層信号灯制御

WebSocket 接続状態 : Open

接続 切断

青 黄 赤 緑

No.1

jQuery JS

NGINX

仮想サーバ

# まとめ

- IoTを取り巻く状況
  - データ収集方法が課題
- IoTサービスを作るときの問題点
  - 作業範囲が広範 / 通信方式が悩ましい / セキュリティの確保が難しい
- sakura.ioについて
  - 開発経緯 / サービス概要 / 利用事例
  - デモ (温度湿度の測定/LED操作)
- セキュアモバイルコネクトについて
  - サービス概要



既存の事業領域/スキルセットの大幅な変更なく  
モノ/サービスづくり、連携に注力可能

- IoTは見るよりもやってみた方がいい
  - 道具はある
  - 体験することで皆さんの業務に活用するヒントが得られれば
- IoTサービスはひとりで(1社で)作るのは難しい
  - ハードウェアの人とソフトウェアの人が協業するとよい
- IoTは費用はかかるけど可能性は無限
  - ハードウェアが必要なので初期投資が高い
  - サービスを作れるかどうかはアイデア次第
  - あらゆる業種に適用できる可能性がある

# 参考情報

- 今日の講演のデモ内容(温度湿度測定/LED操作)を実習するハンズオン
- 東京と大阪では定期的開催中
- パートナーとの共催ハンズオンも実施中
  - Twilio / Azure / Bluemix / 駅すぱあと / AWS など
- イベントページはDoorkeeperやconnpassを検索
- 秋田でも開催したい！
  - 全国約20都道府県で開催したが東北地方は仙台のみ
  - 協力者求む！



- sakura.ioを使ったIoT工作の解説書
- 2018年2月発売
- 主な内容
  - sakura.ioの使い方
  - Arduinoでsakura.ioを使う
  - Raspberry Piとsakura.ioの接続
  - LinuxやPythonでセンサーの値を処理
  - sakura.ioから受信したデータの処理 (JavaScript、MQTT、データストアなど)



- IoT全般の解説書
- 2017年10月刊行
- 目次
  - IoTの現状と取り巻く環境
  - IoTアーキテクチャの構成要素
  - さまざまなデータソース
  - IoTデータの収集/蓄積
  - データの活用を見据えた分析
  - IoTシステムの運用
  - IoTの安全性
  - サービス展開をするシステムの考え方

- さくらのイベントを全国で開催したい！
  - sakura.ioのハンズオン
  - さくらのクラウドなど各種サービスのハンズオン
  - さくらのタベ / さくらクラブ など…
- 協力者求む！
  - 会場の提供
  - 参加者集め
  - 地元コミュニティとの共催も可
  - 連絡先 : sakura-club@sakura.ad.jp

そこに、さくら