

センサーデータの収集から送信まで & ラスト1マイルの解決

コネクト株式会社

本セミナーの構成



IoTとは

Internet of Things (モノのインターネット)

- 様々なモノ(機械やセンサー)が、インターネットを通じてサーバーやクラウドサービスに接続され、相互に情報交換をする仕組みです。

具体的には

- モノの状態を知りたい
- モノの状態を変えたい

なぜ、可能になったか

- センサー/制御機器の進化(小型軽量化)
- インタネット基盤の進化(多様化/無線化)
- サーバ側の進化(高性能化/Cloud化)
- 大量データ処理体制(ビッグデータ/データ解析)

これまでのIoT

大規模なニーズが見込める事案は構築済

- 自動車運行管理: 約15年前にパイロットシステムを開発
 - 車載器(タコメーターから進化)
 - ◆ 位置/加速度/運行時間を採取して安全運行に役立てる
- ガス検針および料金管理: 約13年前に開発
 - オンラン検針システム
 - ワイヤレス検針端末

超大企業から大中企業への拡大/浸透へ

- 小規模(ロット)での対応
- 汎用機器が存在しない
- コストできるだけ抑えたい(パイロットシステムならなおさら)

データ採取方法

エンドノード

IoTのためのコンピュータ

- 汎用性
- 外部機器接続
 - センサー
 - 制御機器操作
 - 通信機能(又はオンボード)
- 省電力
 - 電池駆動可
 - スリープ可
- コンパクト
- 低コスト
- 耐久性

	Raspberry Pi	Arduino
比較モデル名	Raspberry Pi 3 model B	Arduino Uno
CPU	64bit 1GHz	8bit 20MHz
メモリ	1GB	2 KB
ストレージ	SDカード (2~16G)	フラッシュ32KB EEPROM 1KB
OS	あり/マルチタスク	なし/シングルタスク
アナログ入力	要ADコンバータ	可能
PMW信号出力	2チャンネル	6チャンネル
インターフェイス	RJ-45,HDMI, Bluetooth,WiFi	シリアル
価格	4,800円	2,940円
推奨用途	ゲートウェイ	センサーノード

様々なArduino

・ベーシックモデル

USB: 電源/シリアル



電源

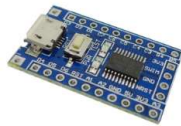
アナログ入出力: 14
デジタル入力: 6
電源供給: 3.3v/5v

ATmega328P
CPU: 8bit 20MHz
Flashメモリ: 32KB
SRAM: 2KB
EEPROM: 1KB

小型モデル

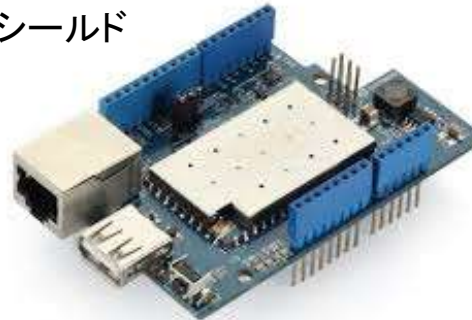


Arduino Nano

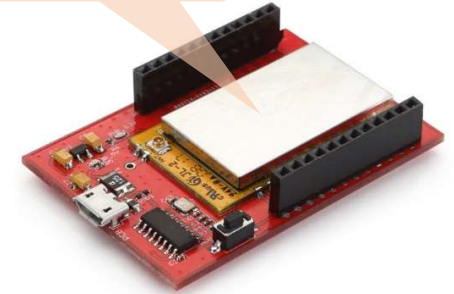


Sduino

拡張シールド



LoRa/LoRaWAN



通信機能搭載型

WiFi



*無線機器を日本国内で使用するには
技適(技術基準適合証明)マークを確認してください

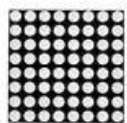
豊富なセンサー & 制御機器

スイッチ/圧力(重量)/振動/チルト(傾き)
温度/湿度/気圧/光(赤外線・紫外線)/距離
方角/加速度/位置(GPS)
ジョイスティック/リモコン/マイク

LED/ディスプレイ
ブザー/スピーカー
リレー(サーボ・ステップモーター)



LED 7 segment 4 7segment



8*8 Matrix



1602 Display



5V stepper Motor



作業手順

デバイス準備

- センサー接続 Arduino + センサ + バッテリー
- コーディング(Arduino IDE)

サーバ準備

- サーバ構築(Webサーバなど)
- スクリプトコーディング

実行

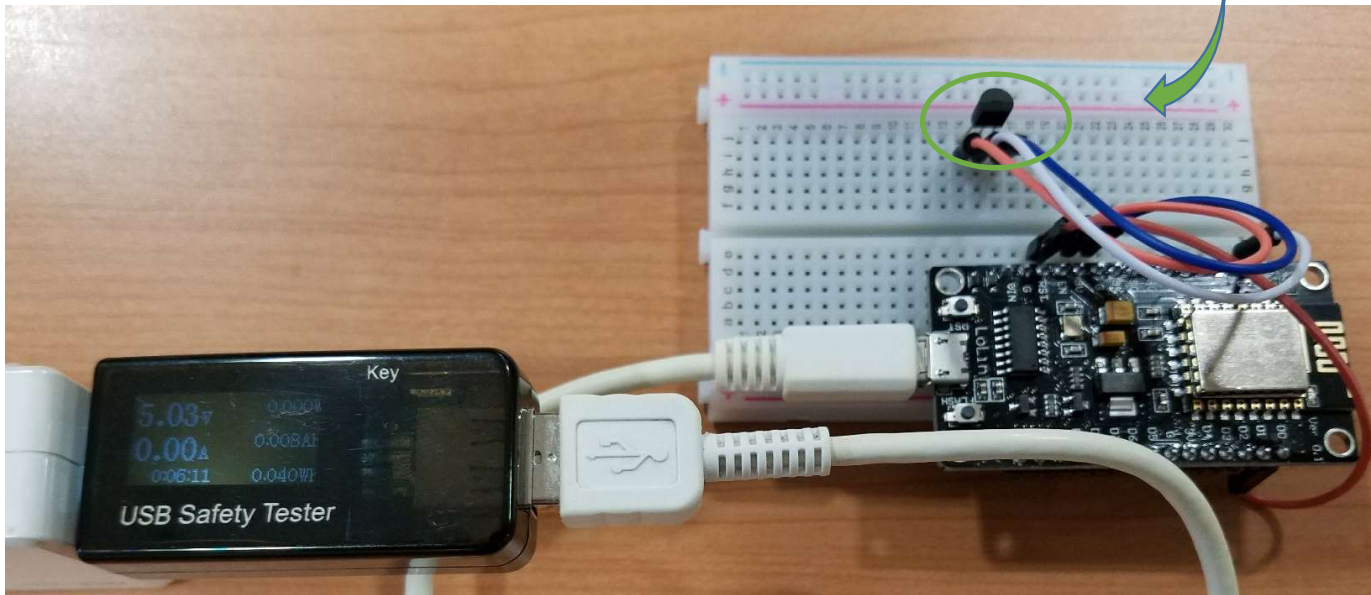
- データ採取
- レポート作成

```
sketch_eps8266_gps $
1 #include <TinyGPS++.h>
2 #include <SoftwareSerial.h>
3 #include <ESP8266WiFi.h>
4
5 static const int gps_RX = 4, gps_TX = 5;
6 SoftwareSerial gps_ss(gps_RX, gps_TX, false, 256);
7 TinyGPSPlus gps;
8
9 const char* ssid = "MyWiFi";
10 const char* password = "abdefg";
11 const char* host = "192.1.2.3";
12
13 void setup() {
14   Serial.begin(115200);
15   delay(10);
16
17   // We start by connecting to a WiFi network
18   Serial.println();
19   Serial.print("Connecting to ");
20   Serial.println(ssid);
21   WiFi.begin(ssid, password);
22
23   board upload progress:
24   最大1044484バイトのフラッシュメモリのうち、スケッチが264800バイト (25%) を使っています。
25   最大81920バイトのRAMのうち、グローバル変数が34220バイト (41%) を使っていて、ローカル変数で47700バイト使うことができます。
26   Uploading 288944 bytes from C:\Users*\*\AppData\Local\Temp\Arduino_build_421930\sketch_eps8266_gps.ino.bin to flash at 0x00000000
27   ..... [ 80% ]
28   ..... [ 80% ]
29   ..... [ 91% ]
30   ..... [ 100% ]
31
32 10 COM5(NodeMCU 1.0 (ESP-12E Module), 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200)
```

今すぐIoTを実現したい①

Arduinoによる温度監視

- WiFi内蔵Arduino(EPS8266) 約1,480円
- 温度センサー(LM35DZ) 約110円
- 電源: USBバッテリー



①ソース

```
1. #include <ESP8266WiFi.h>
2. const char* ssid = "MyWiFi";
3. const char* password = "abcdefg";
4. const char* host = "192.1.2.3";
5. const String docroot = "/test.php";
6. const int httpPort = 80;

• void loop() {
• }

7. void setup() {
8.   WiFi.begin(ssid, password);
9.   while (WiFi.status() != WL_CONNECTED) {
10.    delay(500);
11.   }
12.   WiFiClient client;
13.   if (!client.connect(host, httpPort)) {
14.    return;
15.   }
16.   int sensorValue=analogRead(A0);
17.   String url=docroot;
18.   url = url + "?temp=" + String(sensorValue*100.0 /0.3125/1024*1.00 + "&Mac=" + WiFi.macAddress());
19.   client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" + "Connection: close\r\n\r\n"); int timeout = millis() + 5000;
20.   while (client.available() == 0) {
21.    if (timeout - millis() < 0) {
22.     client.stop();
23.     return;
24.    }
25.   }
26.   while(client.available()){
27.    String line = client.readStringUntil('\r');
28.   }
29.   ESP.deepSleep(57e6); // 62e2 is 60 seconds
30. }
```

今すぐIoTを実現したい①

センサーノード

- 1分間隔で温度とMacアドレスを送信

受信サーバ

- Apache httpd
 - PHPでデータシステムへ格納

アクセスログ

```
192.1.2.3 -- [03/May/2018:07:46:43 +0900] "GET /test.php?temp=26.56&Mac=BC:DD:C2:FD:C6:5E HTTP/1.1"
192.1.2.3 -- [03/May/2018:07:47:43 +0900] "GET /test.php?temp=26.87&Mac=BC:DD:C2:FD:C6:5E HTTP/1.1"
192.1.2.3 -- [03/May/2018:07:48:43 +0900] "GET /test.php?temp=26.56&Mac=BC:DD:C2:FD:C6:5E HTTP/1.1"
192.1.2.3 -- [03/May/2018:07:49:43 +0900] "GET /test.php?temp=26.56&Mac=BC:DD:C2:FD:C6:5E HTTP/1.1"
192.1.2.3 -- [03/May/2018:07:50:43 +0900] "GET /test.php?temp=26.56&Mac=BC:DD:C2:FD:C6:5E HTTP/1.1"
192.1.2.3 -- [03/May/2018:07:51:43 +0900] "GET /test.php?temp=26.87&Mac=BC:DD:C2:FD:C6:5E HTTP/1.1"
192.1.2.3 -- [03/May/2018:07:52:44 +0900] "GET /test.php?temp=27.50&Mac=BC:DD:C2:FD:C6:5E HTTP/1.1"
192.1.2.3 -- [03/May/2018:07:53:44 +0900] "GET /test.php?temp=27.81&Mac=BC:DD:C2:FD:C6:5E HTTP/1.1"
```



最小限の消費電力で動作

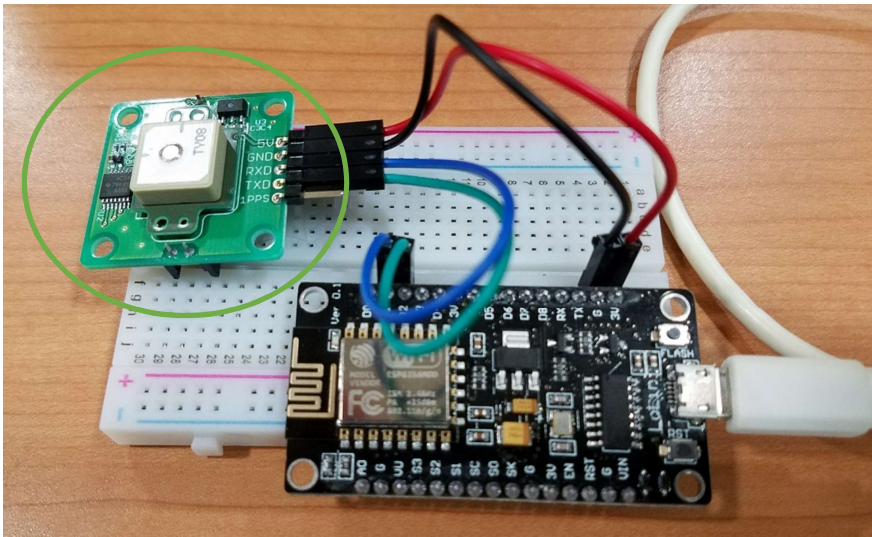
- 動作状態(WiFi使用)
 - それなりの電力を消費します
- スリープ状態
 - タイマー以外をすべて停止



今すぐIoTを実現したい②

Arduinoによる位置監視

- WiFi内蔵Arduino(EPS8266) 約1,480円
- GPSモジュール(GYSFDMAXB) 約2,100円
- 電源: USBバッテリー



APACHE
HTTP SERVER



ソース②

```
1. #include <TinyGPS++.h>
2. #include <SoftwareSerial.h>
3. #include <ESP8266WiFi.h>
4. static const int gps_RX = 4, gps_TX = 5;
5. SoftwareSerial gps_ss(gps_RX, gps_TX, false, 256);
6. TinyGPSPlus gps;
7. const char* ssid = "MyWiFi";
8. const char* password = "abcdefg";
9. const char* host = "192.1.2.3";

10. void setup() {
11.   delay(10);
12.   WiFi.begin(ssid, password);
13.   while (WiFi.status() != WL_CONNECTED) {
14.     delay(500);
15.   }
16.   delay(50);
17. }
```

```
18. void loop() {
19.   WiFiClient client;
20.   const int httpPort = 80;
21.   if (!client.connect(host, httpPort)) return; //connection failed
22.   while (gps_ss.available() > 0)
23.     gps.encode(gps_ss.read());
24.   String url = "/test.php?" + "LAT=" + String(gps.location.lat()) + "&LONG=" + String(gps.location.lng()) + "&ALT=" + String(gps.altitude.meters());
25.   client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" + "Connection: close\r\n\r\n");
26.   int timeout = millis() + 5000;
27.   while (client.available() == 0) {
28.     if (timeout - millis() < 0) {
29.       client.stop(); //>>> Client Timeout !
30.       return;
31.     }
32.   }
33.   while(client.available())
34.     String line = client.readStringUntil('\r');
35.   delay(5000);
36. }
```


今すぐIoTを実現したい②

センサーノード

- 10秒間隔でGPS情報を送信(安定まで40秒ほど)

受信サーバ

- Apache httpd
 - PHPでデータシステムへ格納

アクセスログ

```
192.1.2.3 - - [18/May/2018:17:11:24 +0900] "GET /test.php?LAT=35.68&LONG=139.74&ALT=133.80 HTTP/1.1" 200 1234
192.1.2.3 - - [18/May/2018:17:11:34 +0900] "GET /test.php?LAT=35.68&LONG=139.74&ALT=129.50 HTTP/1.1" 200 1234
192.1.2.3 - - [18/May/2018:17:11:44 +0900] "GET /test.php?LAT=35.68&LONG=139.74&ALT=125.10 HTTP/1.1" 200 1234
192.1.2.3 - - [18/May/2018:17:11:54 +0900] "GET /test.php?LAT=35.68&LONG=139.74&ALT=120.00 HTTP/1.1" 200 1234
192.1.2.3 - - [18/May/2018:17:12:04 +0900] "GET /test.php?LAT=35.68&LONG=139.74&ALT=115.50 HTTP/1.1" 200 1234
192.1.2.3 - - [18/May/2018:17:12:14 +0900] "GET /test.php?LAT=35.68&LONG=139.74&ALT=112.00 HTTP/1.1" 200 1234
192.1.2.3 - - [18/May/2018:17:12:24 +0900] "GET /test.php?LAT=35.68&LONG=139.74&ALT=112.10 HTTP/1.1" 200 1234
192.1.2.3 - - [18/May/2018:17:12:34 +0900] "GET /test.php?LAT=35.68&LONG=139.74&ALT=114.10 HTTP/1.1" 200 1234
```



ラスト1マイルの解決方法

エンドノード&ゲートウェイ

インターネット接続

有線接続

- LANケーブル: 1m~100m(カスケード可能)

無線接続

- WiFi: 1m~50mぐらい(中継器で延長可能)
 - 2.5GHz(申請不要)
 - 5GHz(屋内なら申請不要/屋外でも一部不要)

もっと遠距離(100m以上)の機器と接続するには

ラスト1マイルを解決する方法

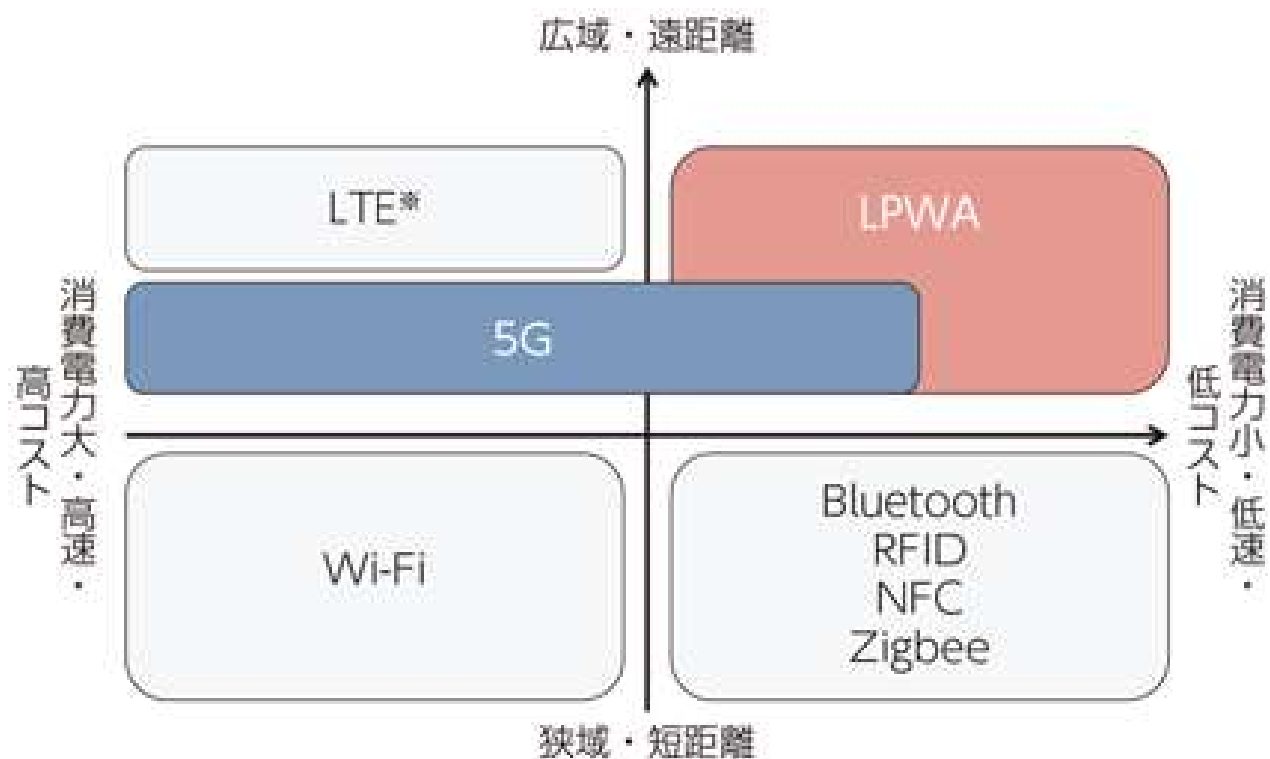
携帯電話網(事業者のみ:各社人口カバー率99%以上)

- 3G(1～1.5GHz)
- LTE(920MHz)

LPWA(Low Power Wide Area) (事業者/自営)

- Sigfox: ~10km(都市では2km程度)
- LoRaWAN: ~10km(都市では2km程度)
- ライセンスバンド

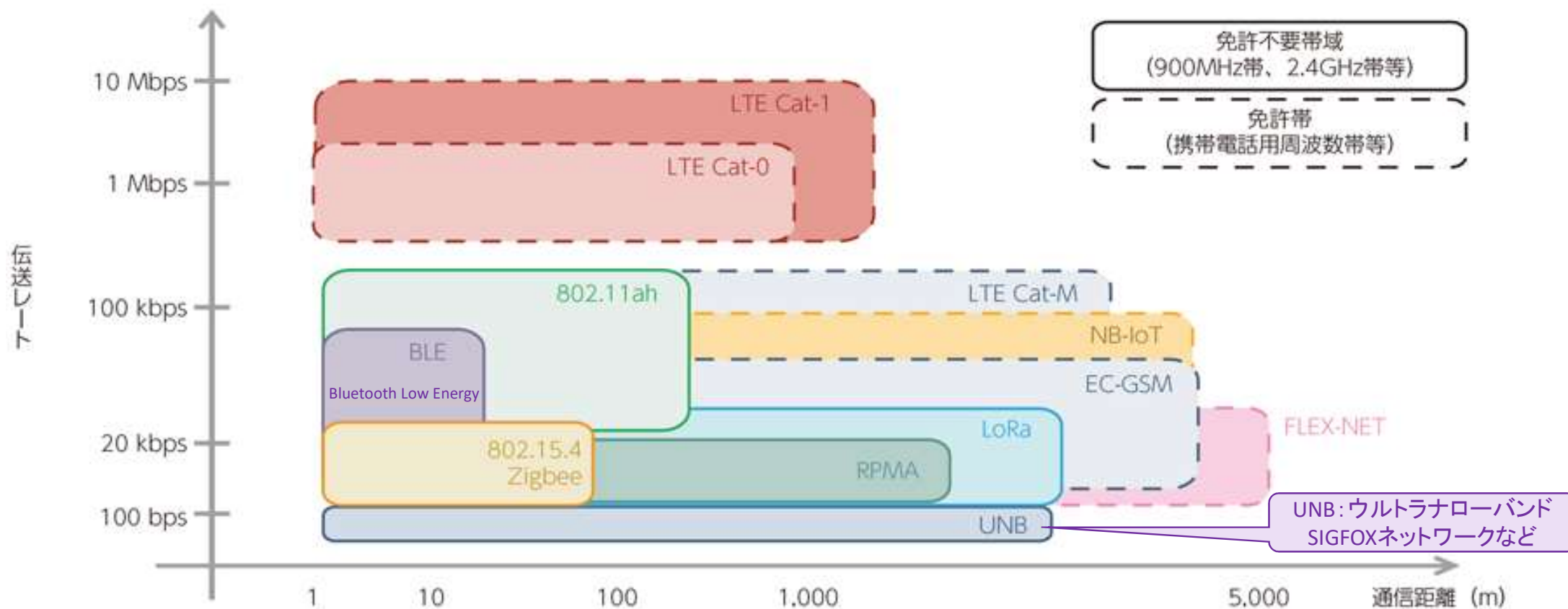
各通信方式の位置付け



※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)

主なLPWA規格の位置付け

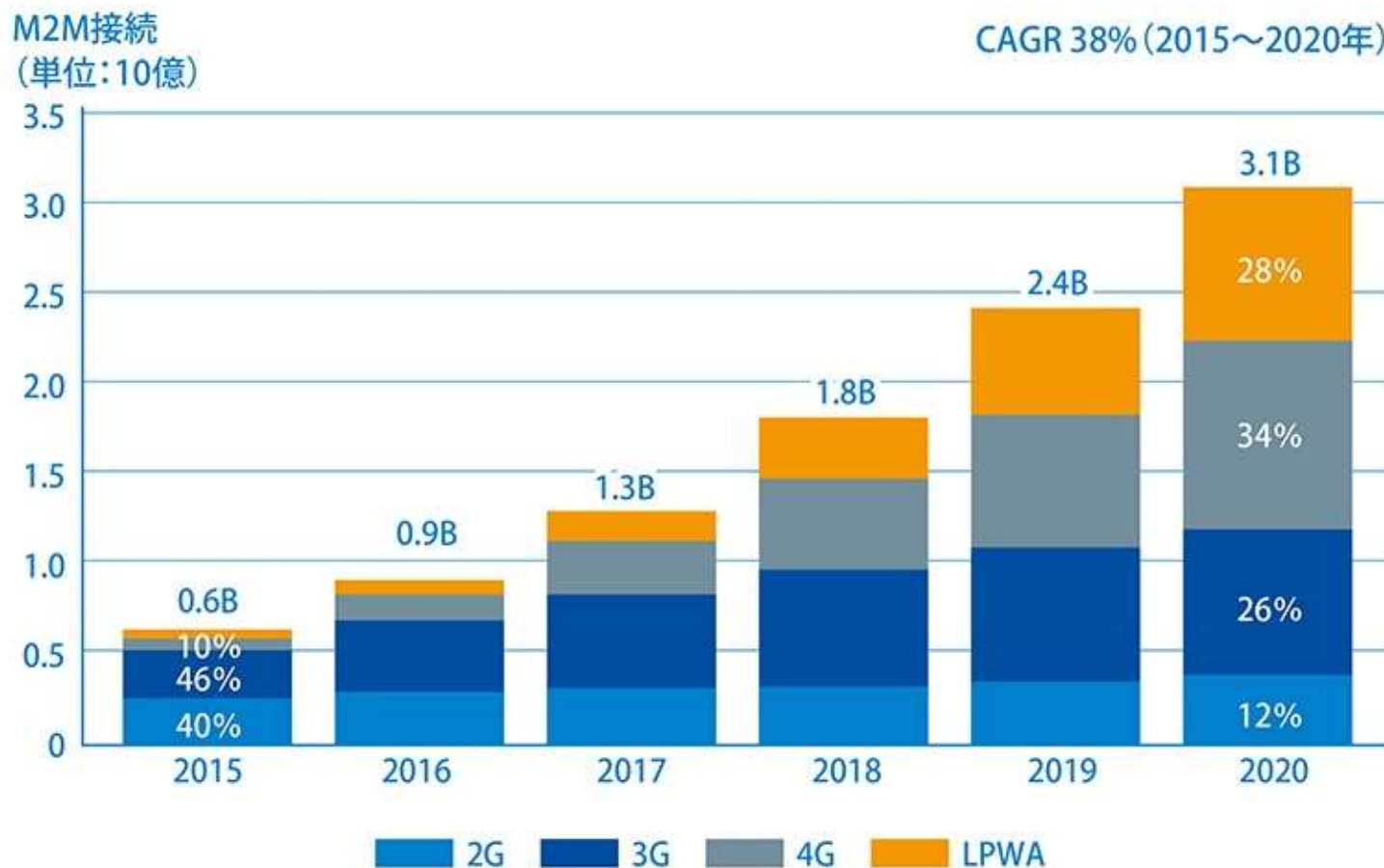


(出典)総務省「第4次産業革命における産業構造分析とIoT・AI等の進展に係る現状及び課題に関する調査研究」(平成29年)

Copyright © konekto Inc. All Rights Reserved.

普及予測

世界の M2M 接続数の推移及び予測



(出典) Cisco VNI Mobile, 2016年

日本での920MHz帯(LoRa)運用

標準規格 ARIB STD-T108 920MHz帯テレメータ用、テレコントロール用及びデータ伝送用無線設備

* 一般社団法人電波産業会(電波利用システムの研究開発・標準化、普及、国際連携)

周波数(200kHzステップ)		CH	キャリアセンス	送信時間	休止時間	総和制限
920.6MHz	923.4MHz	24~38	5ms以上	4s以下	50ms	なし
922.4MHz	923.4MHz	33~38				
922.4MHz	928.0MHz	33~61	128μs以上	200~400ms	送信時間の10倍	1時間あたり360s以下

CSMA/CA(Carrier Sense Multiple Access/Collision Avoidance)

- 送信前に5ミリ秒以上その周波数が使われていないかキャリアセンス
- 使われていなければ4秒間だけ送信できる
 - よって、ペーロードは、11~242バイトとなります。
 - ◆ DR=5(SF=7, BW=125kHz) 242 Byte
 - ◆ DR=2(SF=10, BW=125kHz)で11 Byte
- 送信後に再度送信する場合には50ミリ秒待たなければならない

LPWA製品事例(LoRa) : エンジンスターター

• これまで

- 429MHz(10mW)を使用
- 市街地: ~500m
- 最大: ~2500m

• LoRa採用により

- 920MHz(20mW)を使用
- 市街地: ~2000m 約4倍
- 最大: ~12000m 約5倍



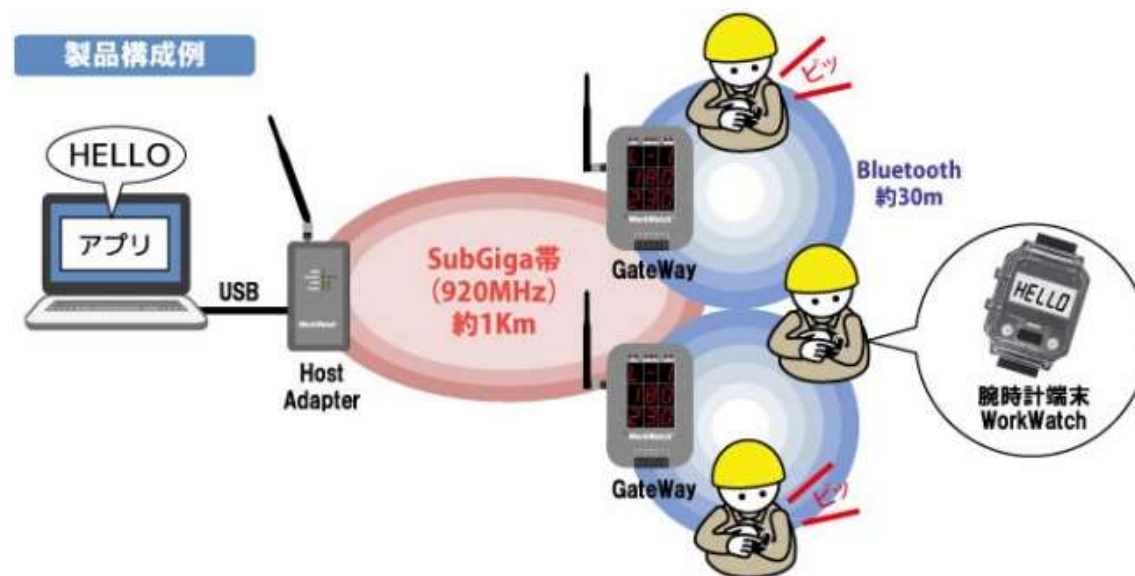
株式会社ユピテル
エンジンスターターVE-E8810st 製品ページ より
https://www.yupiteru.co.jp/products/en_sta/ve-e8810st/

LPWA製品事例：メッセージ送受信

腕時計端末とWindowsPC間でメッセージ送受信をおこなう「WorkWatch Alarm(ワークウォッチアラーム)」

本製品は、腕時計型端末とSubGiga-Bluetooth Gateway、SubGigaホストアダプター、Windows用アプリケーションから構成されるシステムセットです。

腕時計端末を装着した作業員と管理棟のWindows PC間でメッセージの送受信をおこなうコミュニケーションシステムとしてご使用いただけます。WindowsPC用アプリケーション開発キットとしてSDKを提供しておりますので、稼働中の既存システムと連携し、腕時計端末へメッセージ送信やバイブレーションによる通知をおこなうことが可能となります。



ラトックシステム株式会社より
腕時計端末によるメッセージ送受信を既存システムに組み込める
「WorkWatch Alarm(ワークウォッチアラーム)」6月下旬出荷開始
<http://www.ratocsystems.com/info/news/2018/0521.html>

LoRaとLoRaWAN

LoRa(Long Range)



- SEMTECH社が開発した無線通信方式: LoRa変調
 - 低出力で長距離通信が可能
 - 最大ペイロード: 28~50バイト
 - 免許不要: エンドノード、自営ゲートウェイ

LoRaWAN(LoRa+WAN)



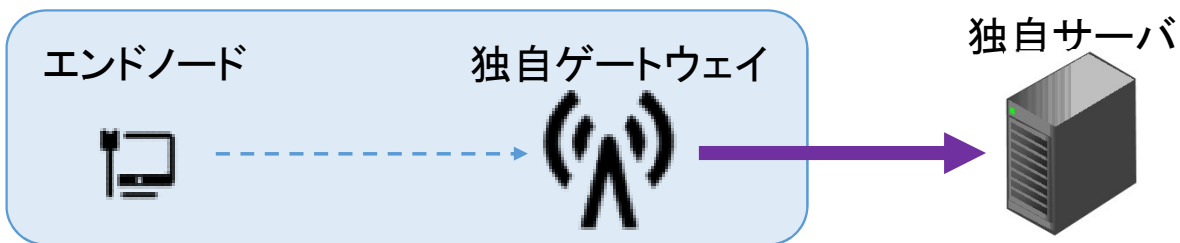
- LoRa Allianceが策定したオープンなネットワーク概念
 - セムテック, STマイクロエレクトロニクス, マイクロチップ・テクノロジーなど300社
- LoRa変調通信+インターネット
 - LoRa変調通信: エンドノード⇔ゲートウェイ
 - インターネット: ゲートウェイ⇔ネットワークサーバ⇔アプリケーションサーバ
 - 最大ペイロード: 11~242バイト
 - 免許不要: エンドノード、自営ゲートウェイ
 - 免許必要: 共用ゲートウェイ



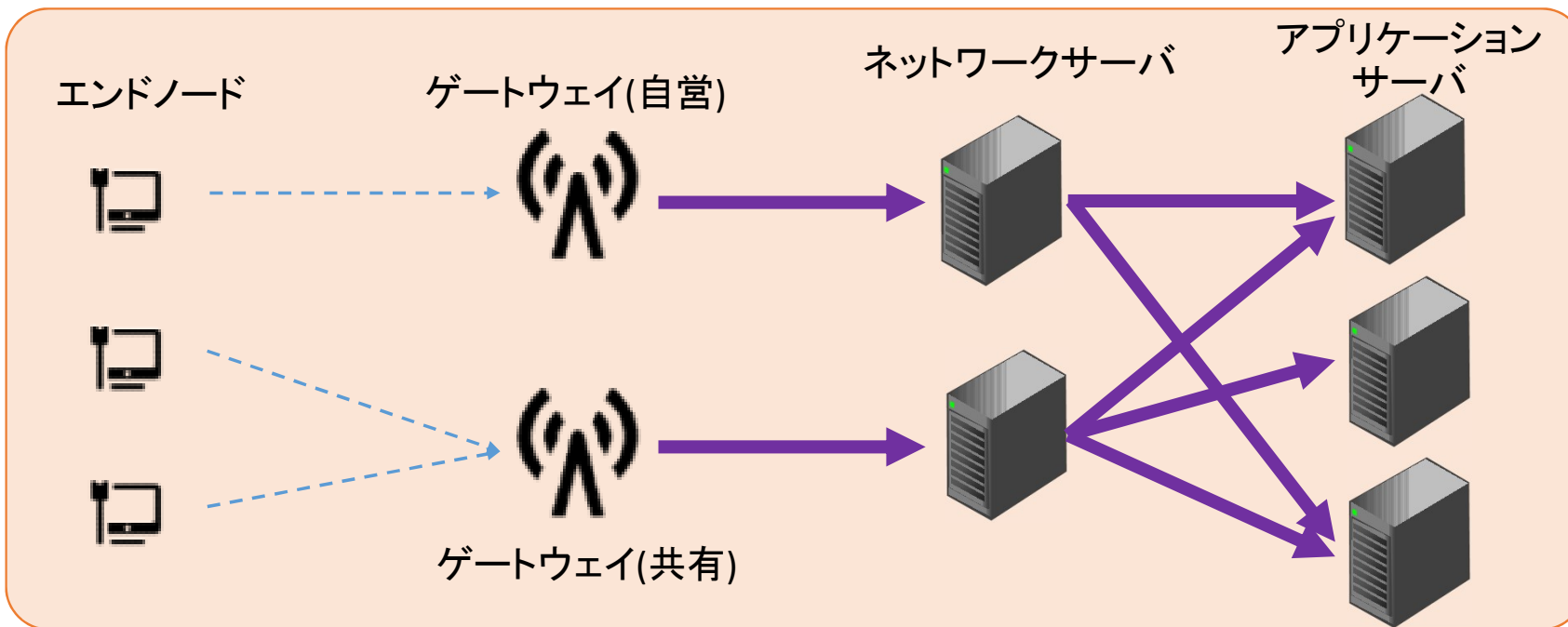
LoRaとLoRaWAN



暗号可能
(独自実装)



暗号化



LoRaWAN普及状況：海外事情と日本

通信事業者がサービスを提供

- 100%網羅：
 - フランス、スペイン、オランダ、ベルギー、スイス、韓国など
- 構築中
 - インド：タタ・コミュニケーションズとヒューレット・パッカード・エンタープライズは共同で、世界最大規模のネットワークを構築すると発表
- 日本でも構築中
 - ソラコム
 - センスウェイ
 - NTT西日本
 - ソフトバンク

Coverage & Operator Map



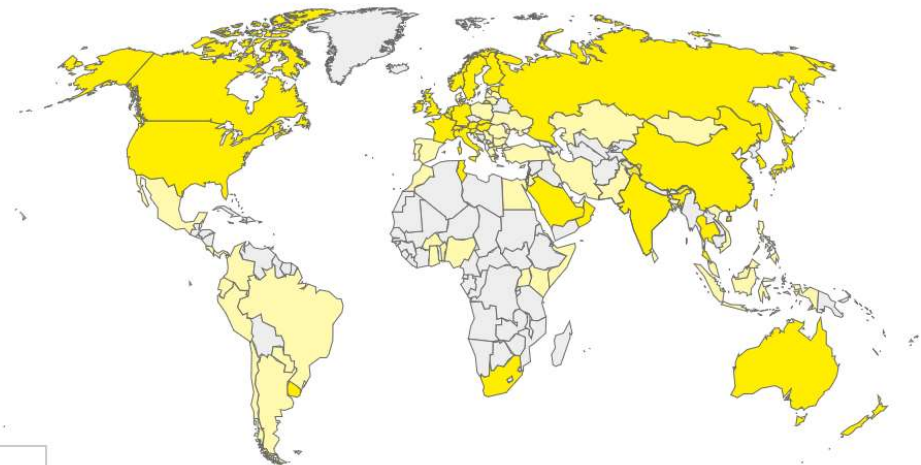
Click on the coverage map to find details of Network Operators by individual country. The extent of LoRaWAN network coverage globally is significant and expanding on a monthly basis. The dark yellow represents LoRa Alliance Member Operators, many of which have extensive network deployments. Light yellow defines smaller scale LoRaWAN activity or networks by Operators who are not yet members. Please contact the LoRa Alliance Operator members for specific information about their networks.

76
Network Operators

57
Alliance Member Operators

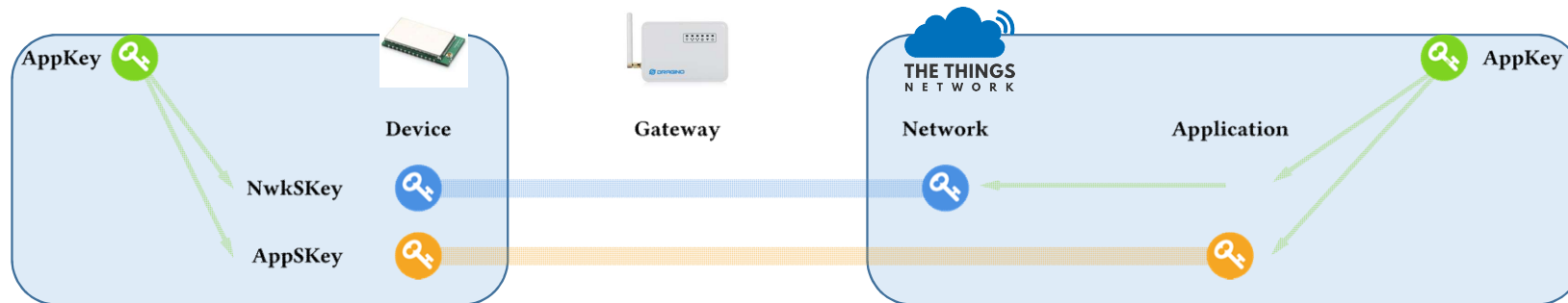
43
Countries operating in

100+
Countries with LoRaWAN Deployments












LoRaWAN: The Things Networkの利用(無償利用可能)

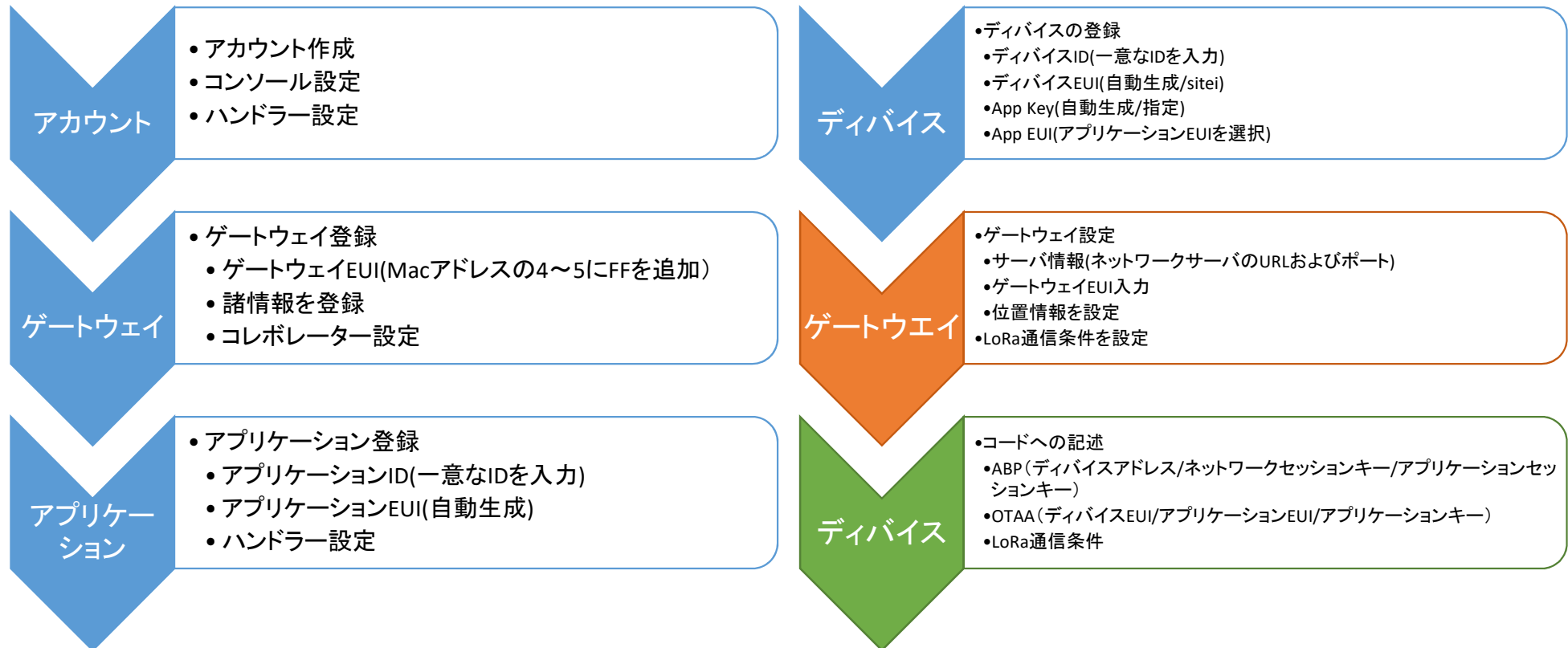
The Things Networkはオランダのスタートアップベンチャー企業によるクラウドファウンディングから始まった世界最大のLoRaWANコミュニティです。



The Things Networkは、ネットワークサーバになります。LoRaエンドノード、LoRaゲートウェイ、アプリケーションサーバと組み合わせて使います。

エンドノード	ゲートウェイ	ネットワークサーバ	アプリケーションサーバ
			        

LoRaWAN設定手順(The Things Network)



LoRa IoT スターターキット

• LoRa IoT スターターキット-I

LoRa ゲートウェイ LG01-JP x 1

LoRa mini 用開発ボード x 2

温度、湿度センサーDHT11 x 2

59,800 円(税別)

• LoRa IoT スターターキット-II

LoRa ゲートウェイ LG01-JP x 2

LoRa センサーノード LoRa mini x 2

LoRa mini 用アンテナ x 2

LoRa mini 用開発ボード x 5

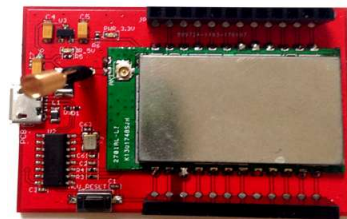
LoRa mini 用テストユーティリティ x 1

温度、湿度センサーDHT11 x 2

158,000 円(税別)



LG01-JP



LoRa mini



株式会社オープンウェーブ

神奈川県横浜市神奈川区西神奈川 1-8-11-103

メール: contact@openwave.co.jp

電話: 045-620-6648 (IoT担当まで)

使用例：温度データ送信/GPSデータ送信



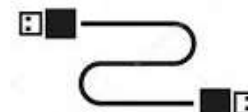
Qiita「LoRaでIoTプラットフォーム(ThingSpeak)にデータを上げる方法」より

<https://qiita.com/openwave-co-jp/items/fc4695f6de36fc9cd398#%E4%BD%BF%E7%94%A8%E3%82%BD%E3%83%95%E3%83%88>

Qiita「LoRaにGPSを付けて、会社の周りを歩いてみた」より

<https://qiita.com/openwave-co-jp/items/8dae802638cf6ac62650>

評価例：固定文字送信



Qiita「LoRaWANでIoTプラットフォーム(The Things Network)にデータを上げる方法」より
<https://qiita.com/openwave-co-jp/items/d1f7aad54a5bba7b81cd#%E6%B3%A8%E6%84%8F>

もっと詳しい内容はこちらへ



konekto

オープンソースで推進するICT

コネクト株式会社
〒102-8578
東京都千代田区紀尾井町4-1
ニューオータニ・ビジネスコート 10F
TEL: 03-6256-8598
FAX: 03-3234-0191



大型オープンソース・ビジネス、日本上陸。

since 2001