

抜粋版

マイコンを使用した換気センサ開発キット
～密閉空間の空気品質を測定、アラーム送信、マイコン、AWS
最大活用～
実践編（ESP版）



AWS IoT Core



ESPRESSIF



スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

sales@spectrum-tech.co.jp

換気センサ開発キット 目次

ESP運用マニュアル

1. ESPについて
2. ESP-devkit
3. Arduino IDE
4. Arduino IDE 設定、使い方
5. ESP起動

AWS IoT開発

- ① メニュー
- ② AWS設定
- ③ AWS IoT設定
- ④ 換気センサ システム概要
 - 全体構成
 - 換気センサ概要
 - 換気センサー、ESP接続
 - 換気センサ単体試験
 - AWS IoT, ESP設定関連図
- ⑤ 換気センサ リアルタイム表示
 - AWS IoT設定
 - ESP32設定
 - リアルタイム表示(グラフ)
- ⑥ 換気センサ メール送信
 - AWS SNS設定
 - AWS IoT設定
 - AWS IoTイベント設定
 - AWS IoTイベント試験

ページ

- [3](#)
- [3](#)
- [3](#)
- [4](#)
- [9](#)

抜粋版のため内容と
ページが違います

ページ

- [10](#)
- [11](#)
- [18](#)
- [32](#)
- [33](#)
- [34](#)
- [35](#)
- [36](#)
- [37](#)
- [44](#)
- [50](#)
- [56](#)
- [59](#)
- [66](#)
- [74](#)



ESP運用マニュアル

1. ESP32について

上海に拠点を置くEspressif Systemsが開発するマイクロコントローラ(MCU)です。WiFi, Bluetoothが搭載されたMCUで安価、高性能なチップです。

2. ESP32-DevKitC

- <https://www.espressif.com/en/products/hardware/esp32-devkitc/overview>
- 開発用キットで、I2Cなどのインタフェースを要しています。AWS IoTの接続実績も豊富で、FreeRTOSの搭載も可能です。今回は未搭載。
- プログラム開発ツールとしては、Arduino IDEを使います。他にESP-IDFがあります。

3. Arduino IDE

- Arduino Uno、NanoなどのMCUを作っているイタリアのメーカーです。そのメーカーの開発ツールを利用します。以下からダウンロードしてください。
- <https://www.arduino.cc/en/Main/Software>



ESP運用マニュアル

WiFiの設定は、
プログラムで実
施します。

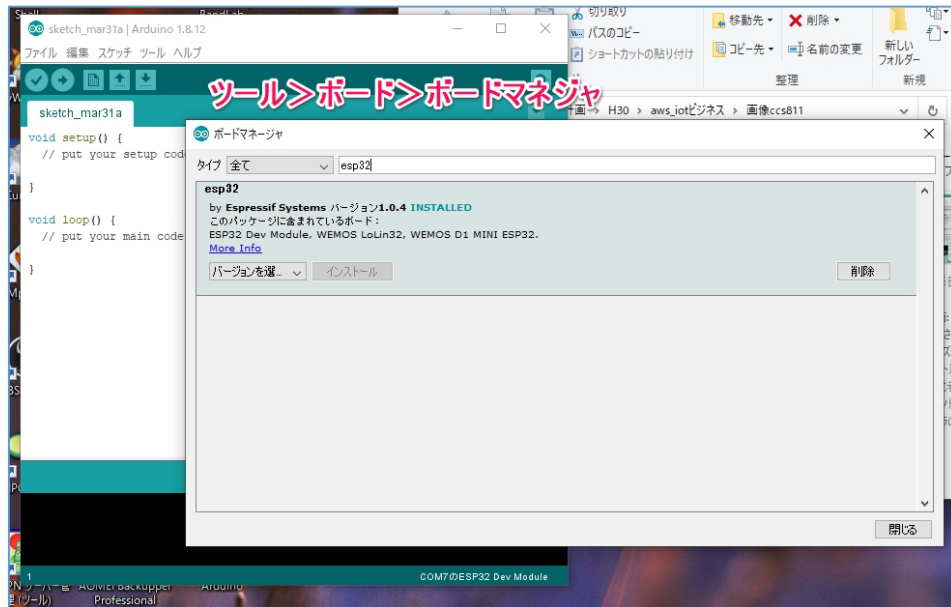
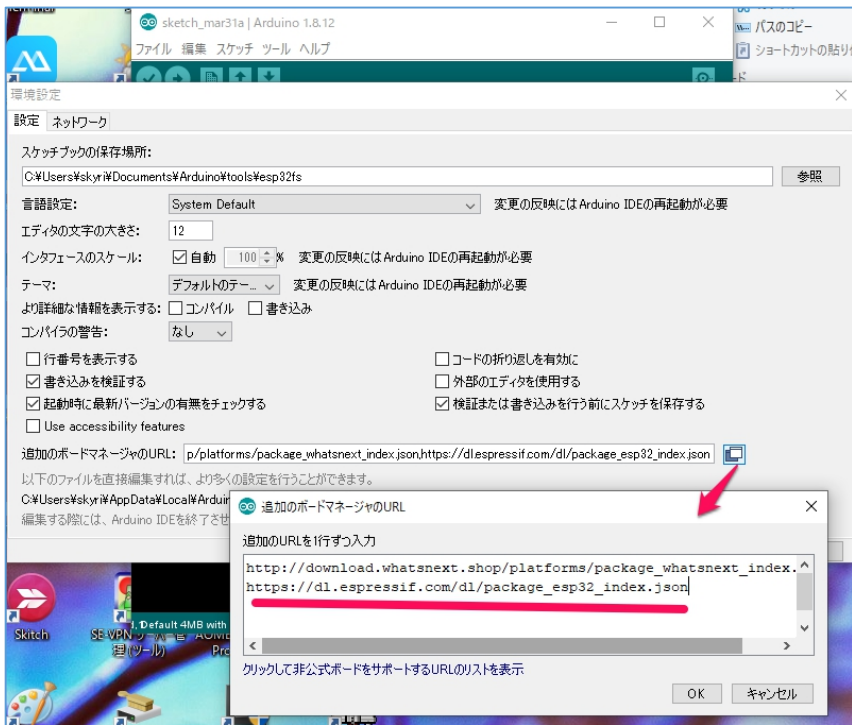
4. Arduino IDEの設定、立ち上げ、操作方法

• 設定(USBケーブルでPCとESPを接続してください)

• 環境設定:ESPのボードをダウンロードします。

- ファイル>環境設定>追加ボードマネージャ、以下のURLを追加
- https://dl.espressif.com/dl/package_esp32_index.json
- ツール>ボード>ボードマネージャ
- Esp32と入力してインストールします。次回、ボード一覧が出たら:ESP32 dev moduleを選択

通常は、ボード一覧にでま
すが、誤ってESPのデー
タを削除した時に使用して
ください





ESP運用マニュアル

4. Arduino IDEの設定、立ち上げ、操作方法

• 設定

- 必要なライブラリをインストールしてゆきます
- スケッチ>ライブラリをインクルード>ライブラリ管理
 - Ccs811と入力し、SparkFun ccs811 Arduino libraryをインストール:環境センサ
 - PubSubclientと入力しインストール:MQTT用
 - なおC++のプログラムをコンパイルしてエラーが出た場合は、必要なライブラリをインストール



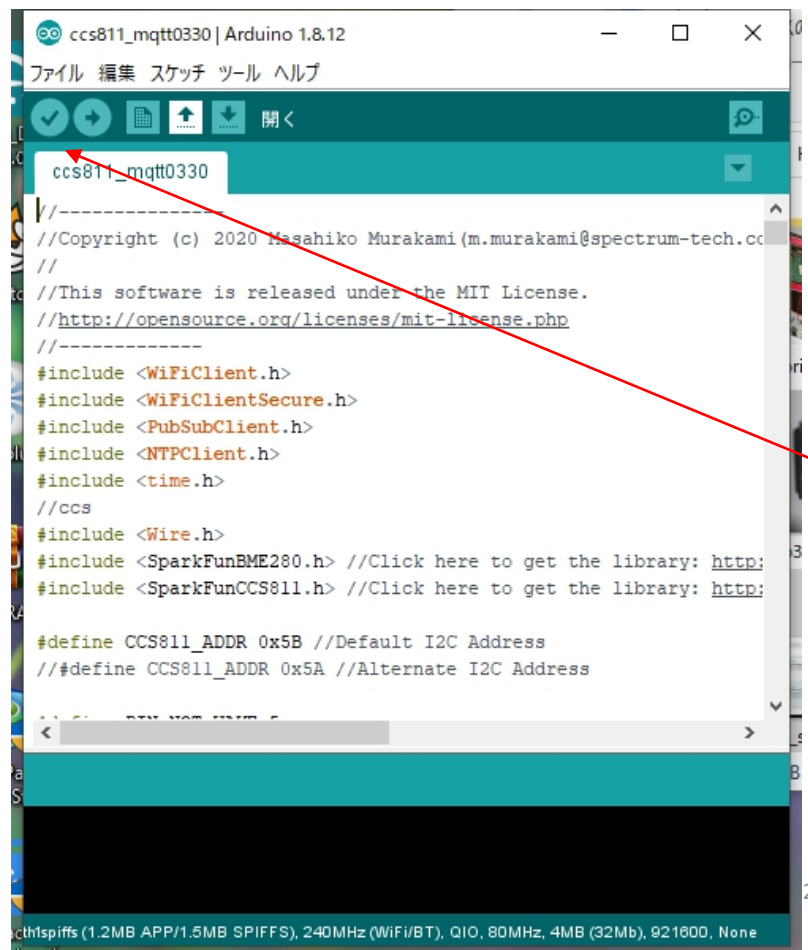
注意

CCS811のインストールバージョンは、1.0.7としてください。最新2.0はエラー

ESP運用マニュアル

4. Arduino IDEの設定、立ち上げ、操作方法

- ファイル>開く
 - 提供のinoファイルを開きます。



左がトップになります。
 Include の右でオレンジになっていない場合は、ライブラリがはいっていないので、前のページのとおり適宜インストールしてください
 各種設定は、AWS設定後に説明します。

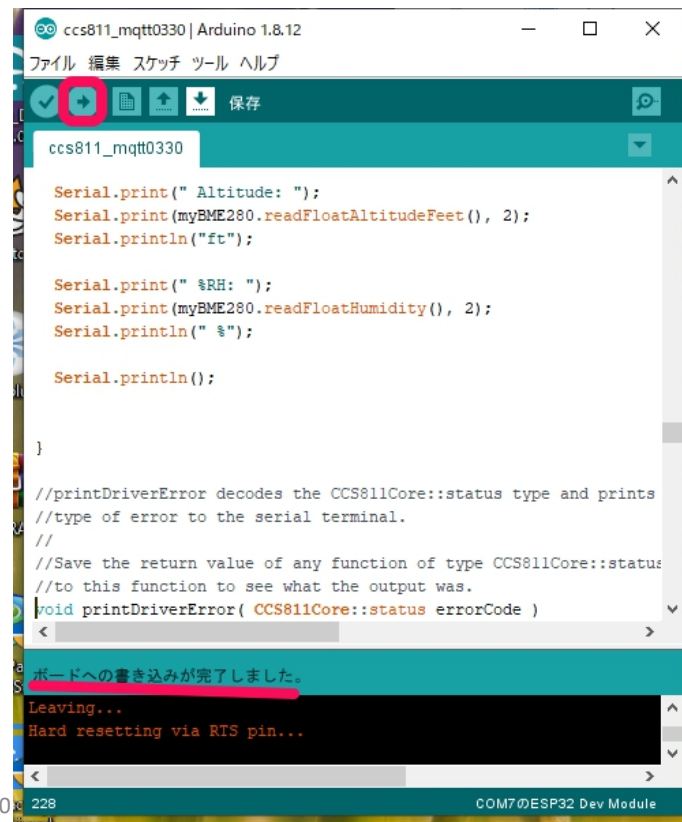
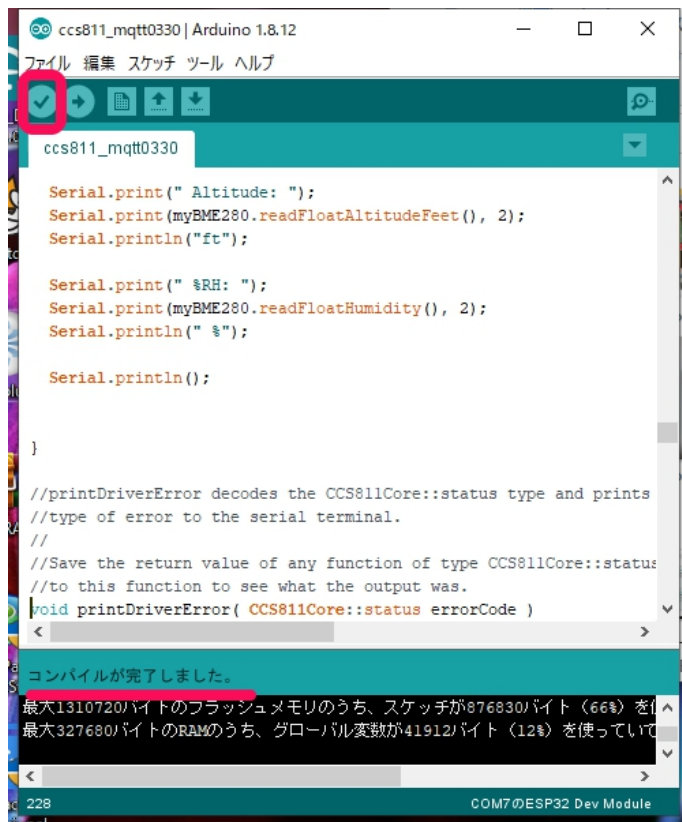
- アイコン説明(左から)
- コンパイル
 - ESPへの書き込み
 - 新規作成
 - 開く
 - 保存

ESP運用マニュアル

4. Arduino IDEの設定、立ち上げ、操作方法

- 標準的な操作方法

- ① コンパイル
- ② ESPへの書き込み



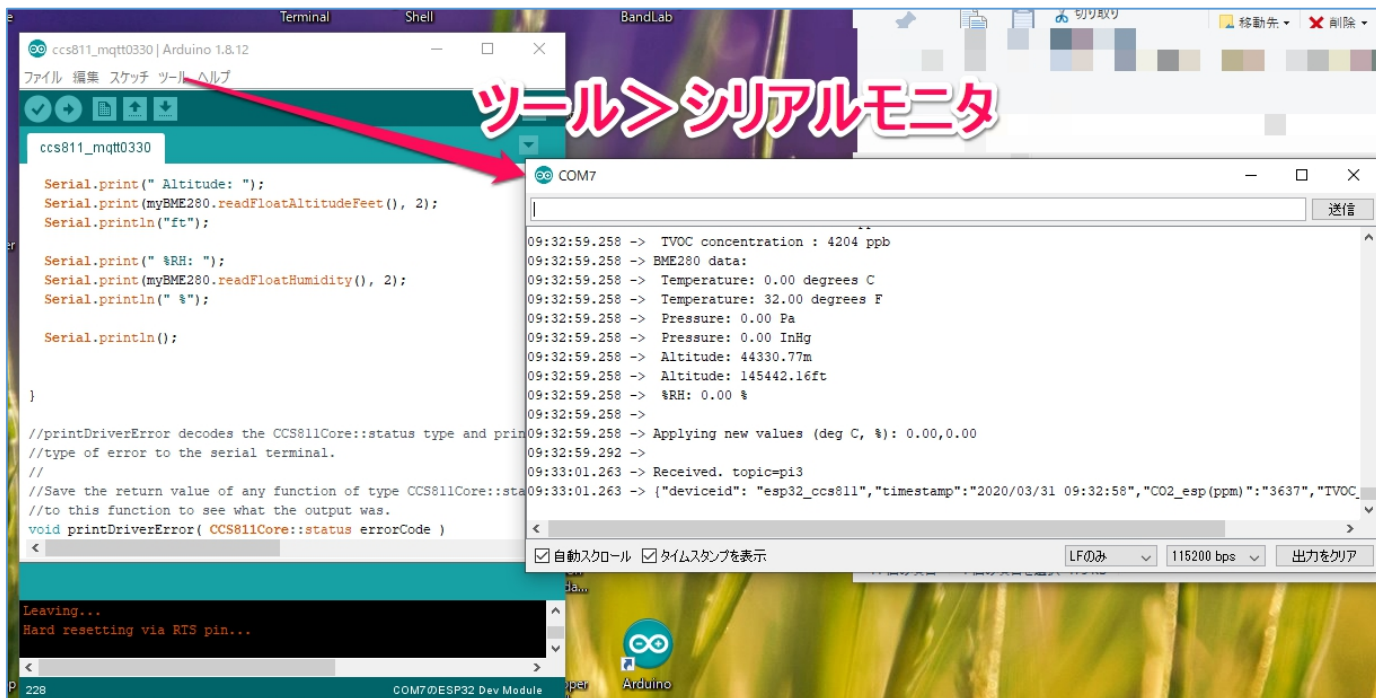
ESP運用マニュアル

4. Arduino IDEの設定、立ち上げ、操作方法

• 標準的な操作方法

- ① コンパイル
- ② ESPへの書き込み
- ③ シリアルモニタで確認(表示されない場合は、再度ESPへの書き込み実施。何故か2回に1回は失敗する)
- ④ 動作確認できたら完了

IDEをインストールしたPCにESPをUSB接続します。
 ツール>シリアルポート>Comポートを選択します





ESP運用マニュアル

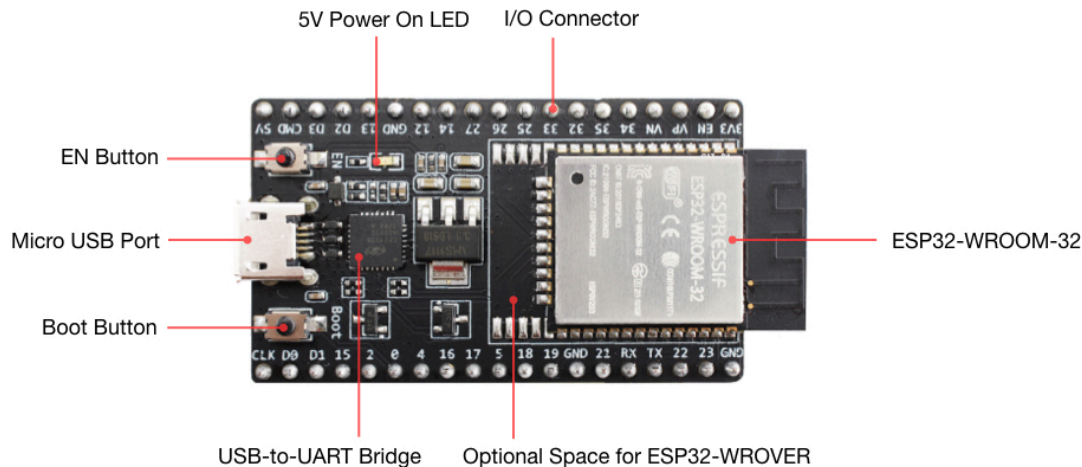
5. ESP起動

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>

- ① ENボタンによる起動: USBケーブル(電源)を接続し、ボタンを押すとリセットされて起動します。ESP単体で動作します。
- ② Bootボタンによる起動: Bootボタンを押したまま、ENボタンを押すと、USBポートからFirmwareをダウンロードします。

6. 消費電力

- 通常: 80mA 5V, 最大1.1A





AWS IoT開発

①. メニュー

- AWS設定
AWS IoTを実施するためのAWSの設定
AWSの詳細説明は以下のURLへ
https://aws.amazon.com/jp/cloud/?nc2=h_l2_cc
- AWS IoT設定
AWS IoTのデバイス、証明書、ポリシー、ルールなどの設定を行います。
開発者ガイドは以下のURLへ
https://docs.aws.amazon.com/ja_jp/iot/latest/developerguide/what-is-aws-iot.html
- AWS SNS設定:メール送信
- AWS IoT event設定: DynamoDBに確保したデータで閾値を設定してメール送信(今回の場合は、CO2 > 1000ppmを超えるとメール
(参考))
- Githubなど
 - <https://github.com/aws/aws-iot-device-sdk-python> (一部のプログラムはTLSエラーで動きません)
 - <https://github.com/eclipse/paho.mqtt.python>
- MQTT
 - <https://www.ibm.com/developerworks/jp/iot/library/iot-mqtt-why-good-for-iot/> (IBMの資料でわかりやすい)

AWS IoT開発

②. AWS設定

- 手順1: アカウントの作成
 - <https://aws.amazon.com/jp/register-flow/>
 - 必要なメールアドレス、パスワードなどを入力します。



AWS アカウント作成の流れ

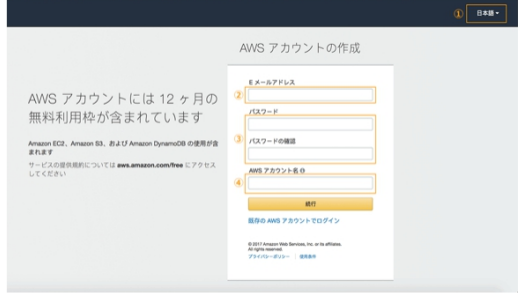
AWS アカウントを作成すると、1年間の無料利用枠はもちろん、AWS クラウドの世界中のリージョンで提供されるすべてのサービスを始めることができます。こちらでは日本のお客様に AWS アカウント作成におけるポイントをご紹介します。

今すぐクラウドを無料でお試ください

[今すぐ無料アカウント作成 >](#)

12 か月間の AWS 無料利用枠と、24 時間年中無休のカスタマーサービスやサポートフォーラムなどの AWS の基本的なサポート機能を利用できます。

ステップ 1: AWS アカウントの作成



※クリックすると大きな画像でご覧いただけます。

このページの上部タイトルおよび、末尾に設置されているオレンジ色のアカウント作成ボタンよりサインアップ画面へ移動します。

各ページ右上 ① の言語選択ボックスより、「日本語」でない場合「日本語」を選択後、こちらのサインアップ画面へお進みください。

- 最初に AWS アカウントとなる情報を設定します。
- ② の「E メールアドレス」には、AWS へのログインに利用したいメールアドレスを設定します。(※)
 - ③ の「パスワード」および「パスワードの確認」で AWS へのログイン時に使用するパスワードを設定し、さらに確認用にもう一度同じパスワードを入力します。
 - ④ の「AWS アカウント名」テキストボックスに、お客様のお名前を半角アルファベットで入力します。
 - 入力後、「続行」ボタンをクリックします。

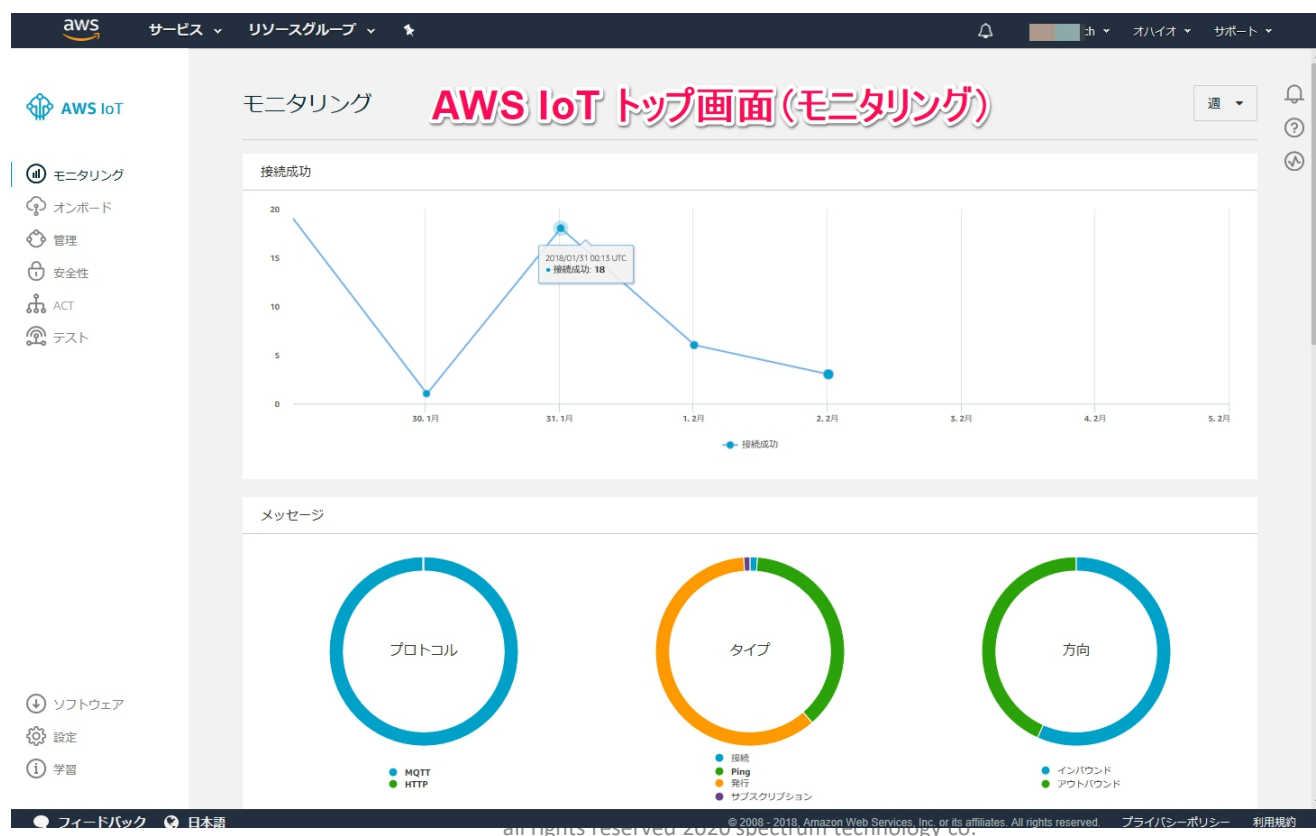
※登録いただくメールアドレスは、AWS 側からの通知等にも利用されます。複数の方へ



AWS IoT開発

③. AWS IoT設定

- 画面説明1
 - AWSTップ画面からAWS IoTのサービスを選択します。
 - モニタリング: 接続状況がひとめで分かります





AWS IoT開発

③. AWS IoT設定

- 画面説明2
 - オンボード: 特に使用しません
 - 管理: モノ

オンボード

AWS IoTに接続する

デバイスの設定

AWS IoT Device SDKの接続ウィザードを使用してデバイスまたはコンピュータをAWS IoTに接続します。

今すぐ始める

AWS IoT ボタン

AWS IoT ボタンは、ボタン1押しでAWS IoTにメッセージを送信する単一用途のデバイスです。

ボタンの設定

ボタンをお持ちでない場合 購入する

モノ

AWS IoTに接続するモノを登録します。

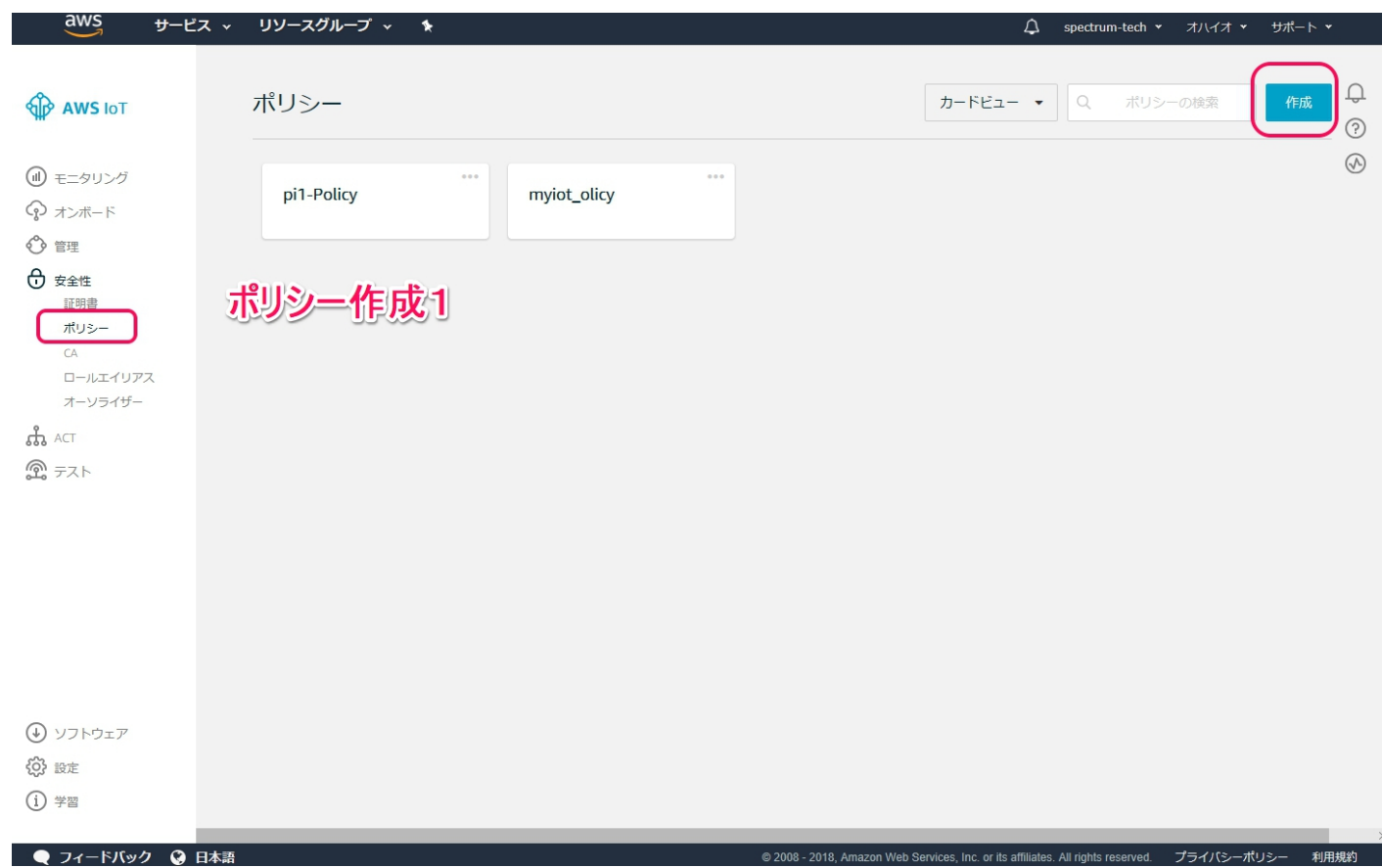
モノに証明書、ポリシーを割り付けます。



AWS IoT開発

③. AWS IoT設定

- ポリシー作成1
 - 安全性 > ポリシーの作成を押します。





AWS IoT開発

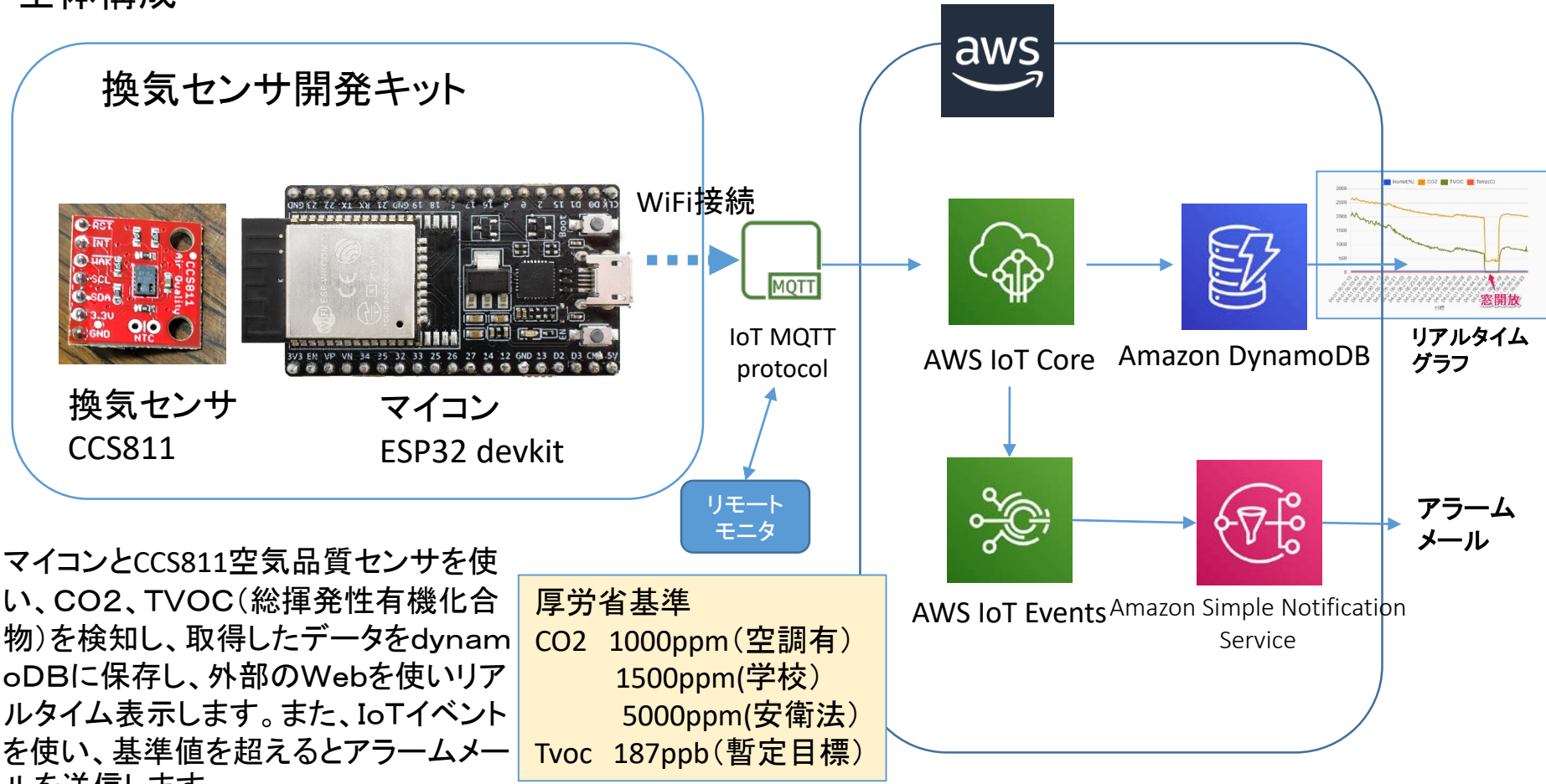
④換気センサ システム概要



AWS IoT開発

④換気センサ システム概要

全体構成



マイコンとCCS811空気品質センサを使い、CO2、TVOC(総揮発性有機化合物)を検知し、取得したデータをdynamoDBに保存し、外部のWebを使いリアルタイム表示します。また、IoTイベントを使い、基準値を超えるとアラームメールを送信します。

厚労省基準

CO2	1000ppm(空調有)
	1500ppm(学校)
	5000ppm(安衛法)
Tvoc	187ppb(暫定目標)



AWS IoT開発

④換気センサ システム概要

換気センサ概要

換気センサ (CCS811)

<https://www.sparkfun.com/products/14193>

Github

https://github.com/sparkfun/CCS811_Air_Quality_Breakout

仕様

- 総揮発性有機化合物 (TVOC) 測定 : 0~1,187 ppb
- eCO2 測定 : 400~8,192 ppm
- 五つの動作モード
- 統合MCU
- オンボード処理
- I²C通信
- 実使用前に48時間のエージングと、20分のコンディショニングの時間が推奨されています (センサがウォームアップして、有効なデータを出力するまでに20分かかります)
- BME280温湿度センサと組み合わせて温度補正が可能です。(プログラムには組み込んでいます)

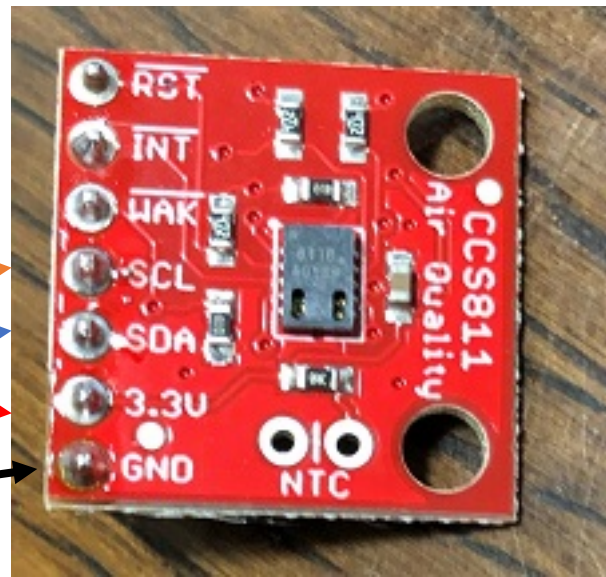
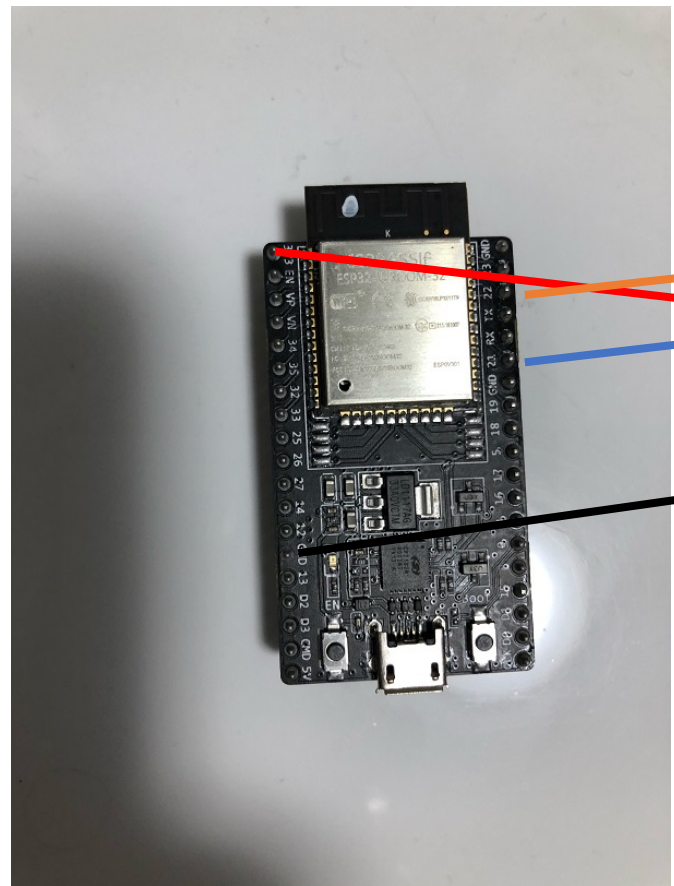
厚生労働省の基準

- <https://www.mhlw.go.jp/bunya/kenkou/seikatsu-eisei10/>
- <https://www.tokyo-co2down.jp/2016/06/24/30472/index.html>

AWS IoT開発

④換気センサ システム概要

ESP32とCCS811の接続



メス-メスのジャンパで接続

ESP32	CCS811
1-----	3.3V
14-----	GND
21-----	SDA
22-----	SCL

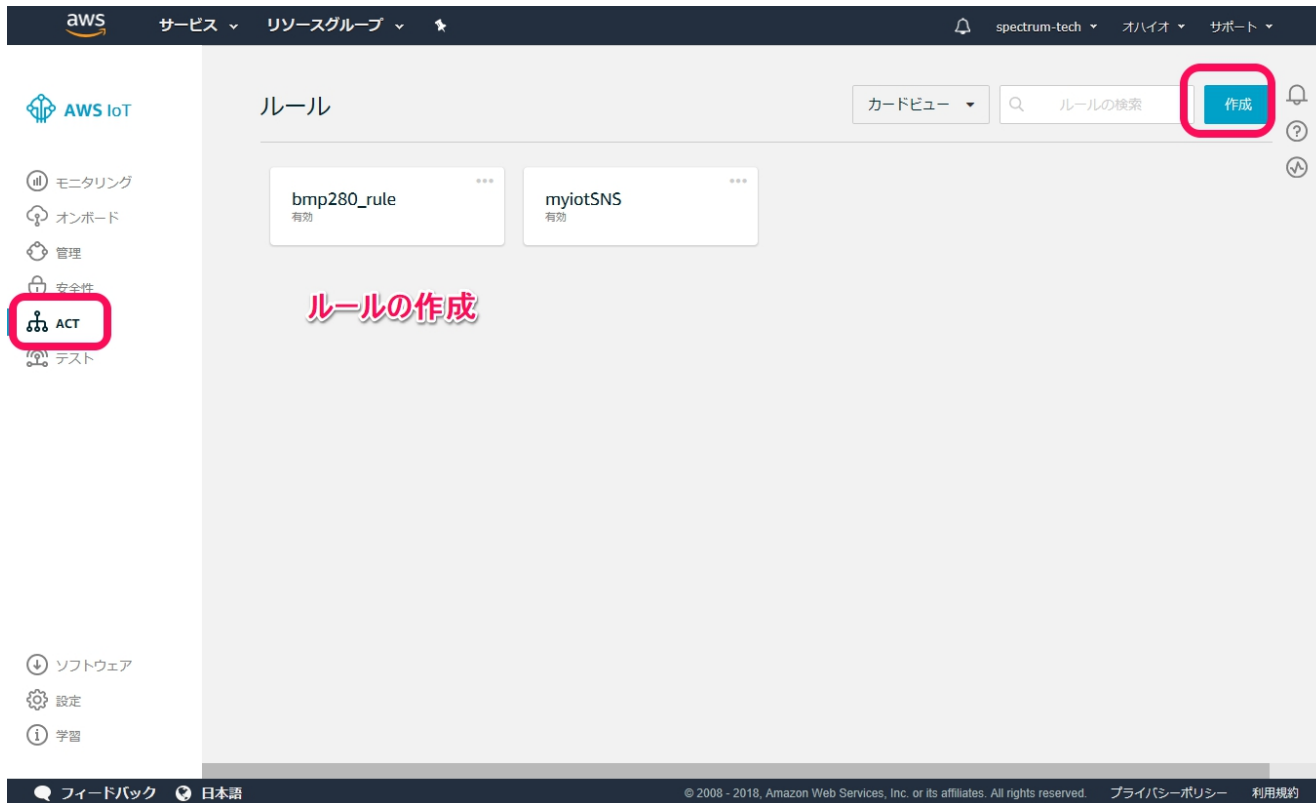


AWS IoT開発

⑤換気センサ リアルタイム表示

AWS IoTのルール設定

- AWS IoTのコンソールからACT>ルール
- ルール作成を押します。(メール送信と同様)



AWS IoT開発

⑤換気センサ リアルタイム表示

ESP32設定

- A) WiFi
 - SSID, パスワード入力
- B) エンドポイント
 - AWSの接続先入力
- C) トピック
 - Publish, Subscribeとも同じもの
- D) 電子証明書
 - ルートCA, Cert, Privateをコピーして入力
 - 各証明書を右クリック>プログラムで開く>サクラエディタなどでひらいてください

```

#define PIN_NOT_WAKE 5

//Global sensor objects
CCS811 myCCS811(CCS811_ADDR);
BME280 myBME280;

char *ssid = "m..._2G";
char *password = "uv...A)";

const char *endpoint = "a...2.amazonaws.com
// Example: xxxxxxxxxxxxxxxx...ap-northeast-1.amazonaws.com
const int port = 8883;
char *pubTopic = "pi3";
char *subTopic(C)pi3";

const char* rootCA = "-----BEGIN CERTIFICATE-----\n\"
"MIIEOzCCA7ugAwIBAgIQGNrRniZ96LtKIVjNzGs7SjANBgkqhkiG9w0BAQUFADCB\
"yJELMAkGAlUEBhMCMCVVMxVzZAVBgNVBAoTD1Zlcm1TaWduLCBjb20uMR8wHQYDVQQL\
"
Leaving...
Hard resetting via RTS pin...
    
```


AWS IoT開発

⑤換気センサ リアルタイム表示

ESP32コンパイル、書き込み、確認

A) コンパイル

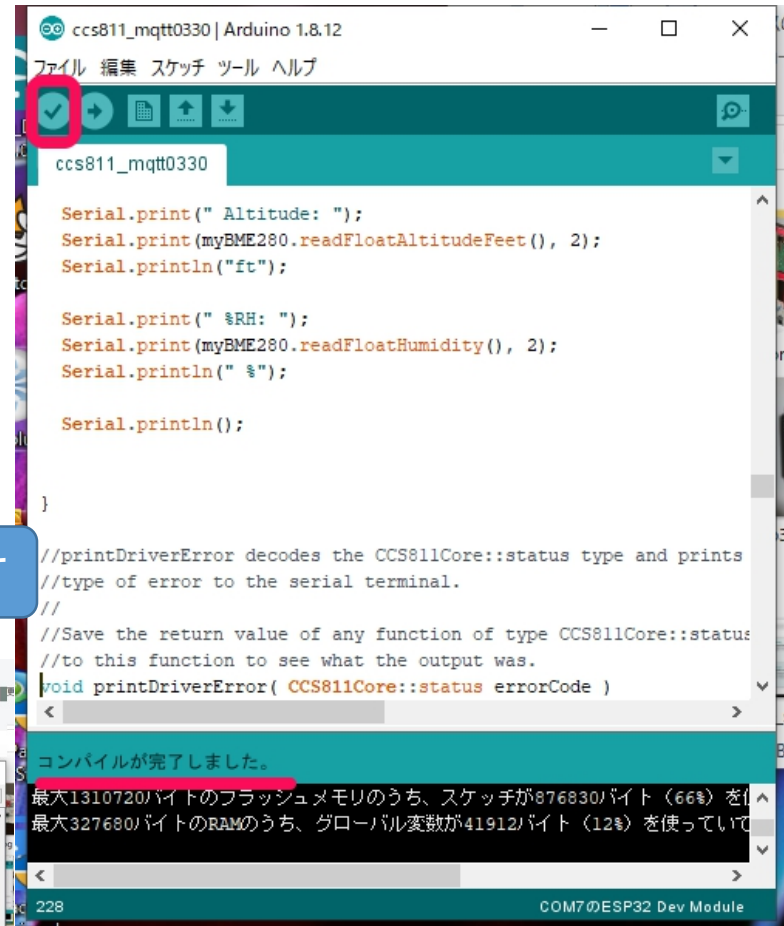
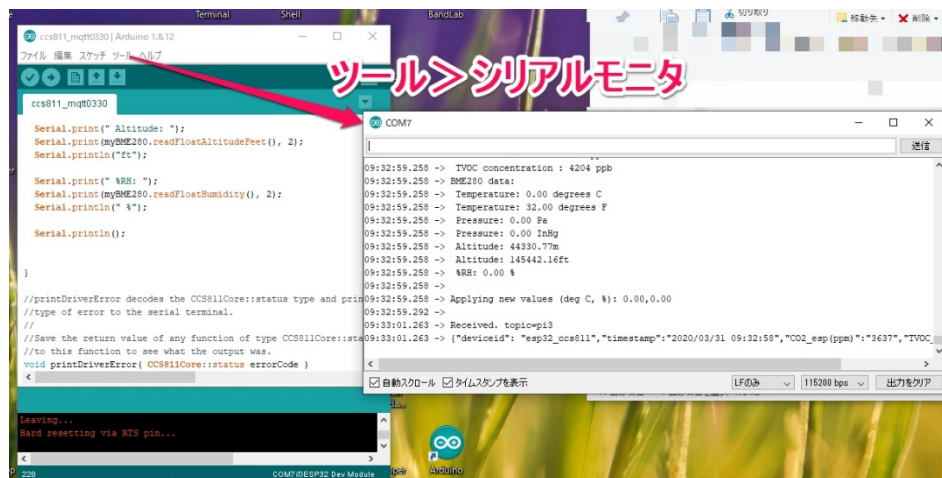
- エラーが出たら修正

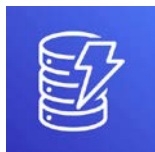
B) ESP書き込み

C) シリアルモニタ確認

- 正常表示されない場合は、書き込みを再度実施
- 最初WiFi接続
- その後MQTT送信

完了です





AWS IoT開発

⑤換気センサ リアルタイム表示

AWS DynamoDBでの確認

- IoTでその実施した内容をDynamoDBで確認
- AWSのトップから>DynamoDBを選択
- テーブル>設定したテーブル>概要

The screenshot shows the AWS IAM console interface. On the left, the navigation pane shows 'DynamoDB' selected. The main content area is divided into three sections:

- Table Creation:** Shows 'Amazon DynamoDB は、完全マネージド型のノンリプレースメント型で予測可能なパフォーマンスとシームレスな拡張性を提供する。' and a 'テーブルの作成' button.
- Table List:** A table with one entry: 'pi1_db'.
- Table Summary (pi1_db):** Shows the table's configuration details.

Red boxes highlight the 'テーブル' link in the navigation pane, the 'pi1_db' table entry, and the '概要' (Summary) tab in the table details view.

現在のステータス	詳細
✓ Amazon DynamoDB (Ohio)	サービス

項目	値
テーブル名	pi1_db
プライマリパーティションキー	deviceid (文字列)
プライマリソートキー	timestamp (文字列)
有効期限 (TTL) 属性	無効 TTL の管理
テーブルの状態	有効
作成日	2018年1月31日 13:50:01 UTC+9
プロビジョニングされた読み込みキャパシティユニット	5 (Auto Scaling 無効)
プロビジョニングされた書き込みキャパシティユニット	5 (Auto Scaling 無効)
最後の減少時刻	-
最後の増加時刻	-
ストレージ容量 (バイト単位)	28.95 KB
項目数	211
リージョン	US East (Ohio)
Amazon リソースネーム (ARN)	arn:aws:dynamodb:us-east-2:123456789012:table/pi1_db

AWSのトップ画面からDynamoDBへ



AWS IoT開発

⑤換気センサ リアルタイム表示

リアルタイム表示

- IoT. Kyoto様の利用>グラフ
- Deviceid: esp32_ccs811
- 表示対象をチェックすると自動で表示されます。



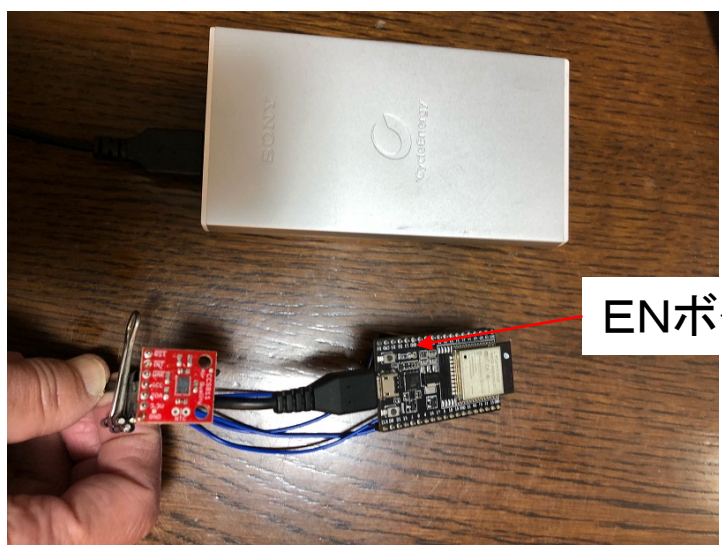
温度で補正した場合の表示
48時間エイジングが必要と最初の20分は異常値がでます

AWS IoT開発

⑤換気センサ リアルタイム表示

ESP32起動

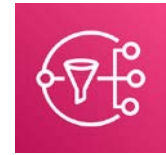
- ESP32単独で、電池(モバイルバッテリー)を使用して起動します。
- ENボタンを押します。
- リアルタイム表示を確認します。



ENボタンです



⑥換気センサ メール送信



Amazon SNS設定

- AWSのトップ画面でSNSを選択
- SNS>トピック>新しいトピック作成

The screenshot shows the AWS console interface. On the left, the 'AWS サービス' (AWS Services) page is visible, with 'Simple Notification Service' highlighted in a red box under the 'データベース' (Database) category. On the right, the 'トピック' (Topic) page is shown, with '新しいトピックの作成' (Create new topic) highlighted in a red box. Below the navigation pane, a table lists existing topics:

名前	ARN
dynamodb	arn:aws:sns:ap-northeast-1:123456789012:dynamodb
myiot_sns	arn:aws:sns:ap-northeast-1:123456789012:myiot_sns



AWS IoT開発

⑥換気センサ メール送信

AWS IoTのルール設定

- ルール作成
- アクション追加 > SNSプッシュ通信としてメッセージを送信する

The image shows two screenshots from the AWS IoT console. The left screenshot displays the 'Rule Editor' for a rule named 'topic/mysns'. The 'Actions' section is highlighted with a red box, and a blue arrow points to the right screenshot. The right screenshot shows the 'Select an action' dialog box with 'SNS プッシュ通知としてメッセージを送信する' (Send message as SNS push notification) selected and highlighted with a red box.



AWS IoT開発 ⑥換気センサ メール送信

IAMの確認

- AWSのトップ画面からIAMを選択
- IAMで、ユーザ、ロールなどアクセス権が一覧でわかります。

The screenshot shows the AWS IAM console interface. On the left, the navigation menu is visible with 'IAM' highlighted in a red box. A blue arrow points from this box to the main content area. The main content area displays the 'Identity and Access Management' page, showing a list of IAM resources including users, roles, and policies. A red text overlay at the bottom of the screenshot reads 'AWSへのアクセス権が一覧でわかります'.



AWS IoT開発

⑥換気センサ メール送信

IoTイベント設定

- 指定設定値で、メール送信するため、IoTイベントを設定します。
- 探知器モデル > 探知器モデルの作成
- 新しく作成する

IoT eventは料金が発生します。適宜請求額を確認

aws サービス リソースグループ spectrum-tech オハイオ サポート

IoT イベント

探知器モデル

探知器モデル

アクション 探知器モデルの作成

名前	作成日	説明
detecthumanmodel	Sun Nov 17 2019	
AWS_IoTEvents_Hello_World1573622847581	Wed Nov 13 2019	A sample "Hello_World" detector model that monitors powerwall voltage.

フィードバック 日本語

© 2008 - 2019, Amazon Web Services, Inc. またはその関連会社。無断転用禁止。 プライバシーポリシー 利用規約

aws サービス リソースグループ

タイトルなし 編集

+

AWS IoT イベント

台方法

るには、デバイスの状態 探知器モデルを作成しま

作成する

新しいディテクターモデルを作成

新しく作成する

テンプレート

業界固有のレートを選

テンプレートし、洗濯

探知器モデルでの入力の使用

入力は、探知器モデルで監視して使用するテレメトリデータ(す。入力値は、アクションを起動したり状態を変更したりする入力値は、左のナビゲーションから利用できる【入力】ページで

日本語

© 2008 - 2019, Amazon Web



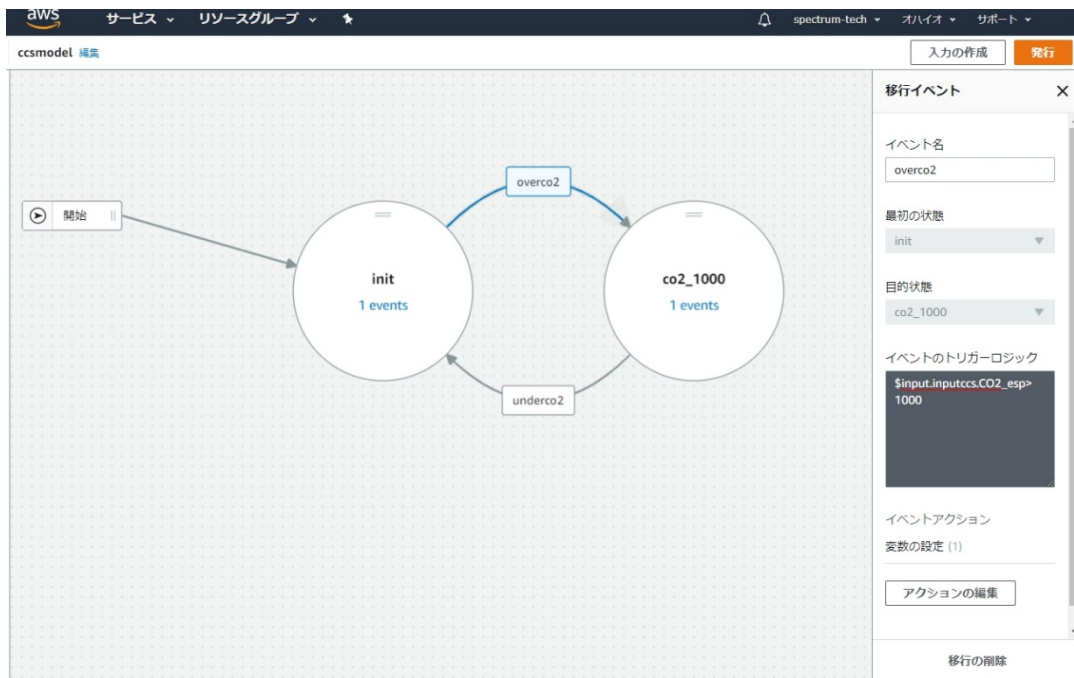
AWS IoT開発

⑥換気センサ メール送信

IoTイベント設定

- 移行イベントの作成
- イベント名 : overco2
- トリガーロジック : `$input.inputccs.CO2_esp > 1000`
- 変数の設定 > 値の割り当て
- 変数名 : co2alarm 値 : 1

Initからドラッグすると移行イベントが設定できます





AWS IoT開発

⑥換気センサ メール送信

IoTイベント試験

- 探知器モデル>サンプルデータの送信
- 入力名:inputccs
- Co2_esp:1000,番号と入力し、プレビュー
- データの送信
- メールの確認
 - AWS Notification Messageが受信できます
- 実データでメールを行うには一度状態をinitにサンプルデータで戻します。

