

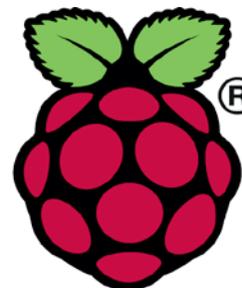
# AWS IoTを使用した電力計キット

～月額無料で電力可視化、無駄を削減～

抜粋版



**AWS IoT**



**Raspberry Pi**

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](mailto:sales@spectrum-tech.co.jp)

# AWS IoT開発キット 目次

## Pi運用マニュアル

1. RaspberryPiについて
2. Linux基本コマンド
3. 基本操作
4. 日常運用(ウイルススキャン、更新)

ページ

3  
4  
5  
6

## AWS IoT開発

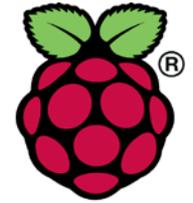
- ① メニュー
- ② AWS設定
- ③ AWS IoT設定
- ④ 電力計
  - 全体構成
  - 接続概要
  - 単体試験
  - AWS IoTルール設定
  - Subscribe設定
  - Publish設定
  - 折り返し試験
  - DynamoDB確認
  - リアルタイムグラフ設定
  - IAMユーザ追加
  - リアルタイム表示
  - DynamoDBからのCSVエクスポート

ページ

8  
9  
16  
29  
31  
36  
39  
45  
46  
48  
49  
51  
52  
55  
57

# 抜粋版

ページは一  
致しません



# Pi運用マニュアル

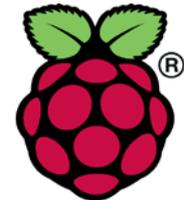
## 1. Raspberry Piについて

既に全世界で1000万台以上販売された手のひらサイズのコンピュータです。  
LinuxベースのRasbianOSで動作しております。

## 2. Linux基本コマンド

### ① システム関係

- 起動: 電源を入れると自動で起動します。
- 再起動: # reboot  
又は、アプリケーション>ログアウト>再起動; 左上のメニューから
- 終了: # shutdown  
又は、アプリケーション>ログアウト>シャットダウン; 左上のメニューから
- ログアウト # logout  
又は、アプリケーション>ログアウト>ログアウト; 左上のメニューから
- **日本語／英語の入力切替**: キーボードのctl+jを同時に押します。又は右上のアイコン(右から7個目)からプルダウンで選択



# Pi運用マニュアル

## 2. Linux基本コマンド

### ② ディレクトリ操作、コピー、移動、削除

root@:~# **cd** /root/Documents      ディレクトリの切り替え  
 root@:/root/Documents# **ls**      ファイルとディレクトリの表示(表示したら操作したいファイルを右クリックでコピーして操作します)

root@:~# **cp** ファイル名 ディレクトリ      配下のディレクトリのファイルを別のディレクトリへコピー  
 root@:~# **mv** ファイル名 ディレクトリ      配下のディレクトリのファイルを別のディレクトリへ移動  
 root@:~# **rm** ファイル名      ファイルの削除

便利な機能      **rm -help**      コマンドのオプションが分からない場合は、ヘルプで問い合わせる。すべてのコマンド共通(マイナスを2個とhelp)

### ③ ユーザ権限、プロセス他

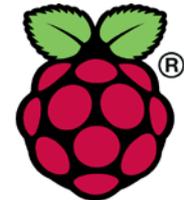
root@:~ \$ **su -**      スーパーユーザ(root)に切り替え、パスワードを入力  
 root@:~# **ps a**      現状の動いているプロセスを表示  
 root@:~# **kill**      特定のプロセスを強制終了

root@:~# **apt-get install pkg**      パッケージのインストールなどに使用  
 root@:~# **date**      日付、時間の設定を行います。

root@:~# **leafpad /etc/network/interfaces**      インタフェースに記述している内容を変更します。Viよりも使いやすいです。

### ④ モジュール、usb、メモリ、HDDなどの表示

root@:~# **lsmod**      linuxのモジュールリスト表示  
 root@:~# **lsusb**      usbのデバイス表示  
 root@:~# **free -mt**      メモリ使用状態表示  
 root@:~# **df -h**      HDD(マイクロSD)の使用状態表示

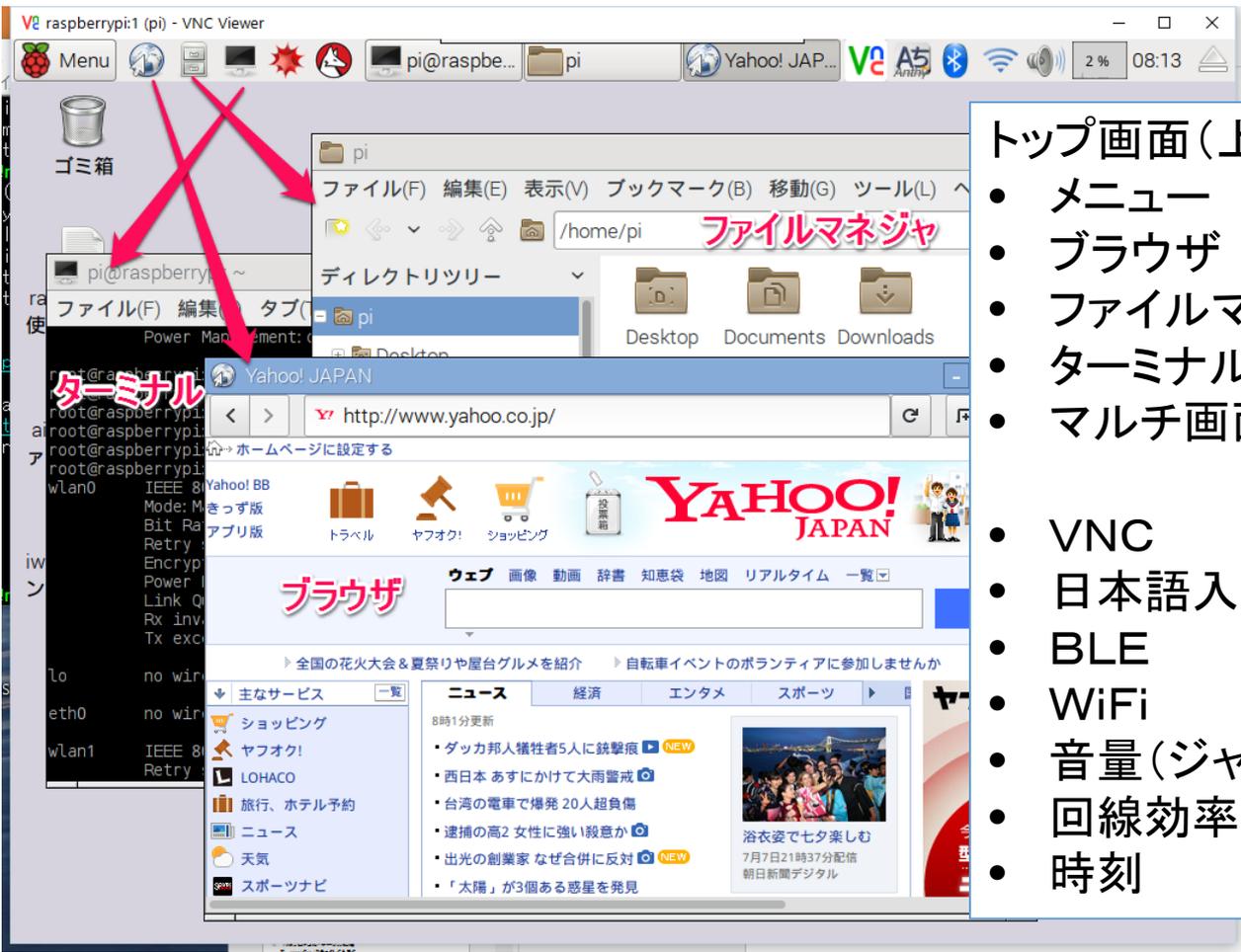


Raspberry Pi

## Pi運用マニュアル

### 3. 基本操作

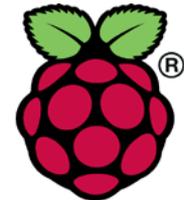
#### ① 表示画面と内容



トップ画面(上段のタスクバーで選択)

- メニュー
- ブラウザ
- ファイルマネージャ
- ターミナル
- マルチ画面選択

- VNC
- 日本語入力
- BLE
- WiFi
- 音量(ジャックで聴けます)
- 回線効率
- 時刻



Raspberry Pi

## Pi運用マニュアル

### 4. 日常運用

- ① セキュリティ対策(アンチウイルス更新、スキャン)
  - アンチウイルス対策として無料のclamAVをインストールしてます。
  - 手動での運用を基本としています。

```
pi@raspberrypi: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
ERROR: /var/log/clamav/freshclam.log is locked by another
ERROR: Problem with internal logger (UpdateLogFile = /var/
og).
root@raspberrypi: ~# leafpad /etc/clamav/freshclam.conf
root@raspberrypi: ~# freshclam
ClamAV update process started at Fri J
main.cvd is up to date (version: 57, sigs: 4216790, f-level: 60, builder: mishh
ammer)
daily.cvd is up to date (version: 21862, sigs: 394456, f-level: 63, builder: neo
)
bytecode.cvd is up to date (version: 283, sigs: 53, f-level: 63, builder: neo)
root@raspberrypi: ~# clamscan --infected --remove --recursive
SCAN SUMMARY
Known viruses: 4607906
Engine version: 0.99.2
Scanned directories: 264
Scanned files: 2063
Infected files: 0
Data scanned: 61.31 MB
Data read: 49.02 MB (ratio 1.25:1)
Time: 71.844 sec (1 m 11 s)
root@raspberrypi: ~#
```

パターンファイル更新  
手動スキャン時に更新されます  
手動でスキャン  
# clamscan --infected --remove --recursive  
自動化可能ですが、バックグラウンドで重くなる可能性大。コマンド入力後約25分位かかります。

手動でスキャン





## 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](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](https://docs.aws.amazon.com/ja_jp/iot/latest/developerguide/what-is-aws-iot.html)
- LED点灯  
AWS IoTのshadowを使って、Raspberry Piに接続したLEDを遠隔から点灯させます
- メール送信  
タクトスイッチを使い、MQTTプロトコルを介して、AWS IoTに接続し、AWS SNSからE-mailの送信を行います。上記同様単体試験などのプログラムを提供
- センサ情報リアルタイム表示  
BMP280センサを使いMQTTプロトコルを介して、AWS IoTに接続し、AWS DynamoDBにデータを保存し、そのデータを外部ツールでリアルタイム表示します。またその過程の、Raspberry Piでのセンサ単体試験、AWS IoT折り返し試験などのプログラムも提供。順序だてて構築して行きます。

(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

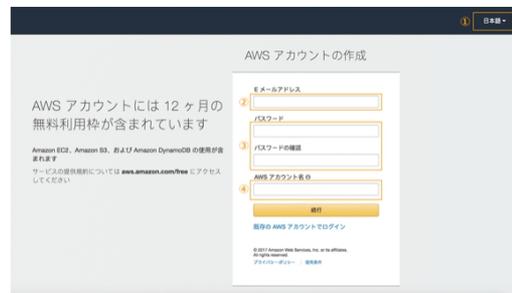
AWS アカウントを作成すると、1年間の無料利用枠はもちろん、AWS クラウドの世界中のリージョンで提供されるすべてのサービスを始めることができます。こちらでは日本のお客様に AWS アカウント作成におけるポイントをご紹介します。

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

### ②. AWS設定

- 手順2: 連絡先情報を入力します。
  - 住所、電話番号、会社名などを入力します。



の通知が必要な場合は、メールリストのご利用をご検討ください。

#### ステップ 2: 連絡先情報入力

※クリックすると大きな画像でご覧いただけます。

次にお客様情報を登録します。

画面上上の言語選択ボックスが「日本語」でない場合、「日本語」を選択してください。

- お客様情報を入力します。アカウントの種類は、法人利用であれば、「プロフェッショナル」、個人利用であれば「パーソナル」を選択します。
- ①の入力欄に以下の情報は、日本語ではなくすべて「半角アルファベットおよび半角数字」で入力をお願いいたします。
  - フルネーム: お客様のフルネームを入力します。(必須)
  - 会社名: お客様の会社名を入力します。(任意)
  - 電話番号: お客様のお電話番号をハイフン・記号なしで入力します。(必須) (例: 0312345678)
  - 国: 国情報を選択します。(必須)
  - アドレス: お客様のご住所の番地、建物名等を入力します。(必須)
  - 市区町村: お客様のご住所の市区町村名を入力します。(必須)
  - 都道府県または地域: お客様のご住所の都道府県名を入力します。(必須)
  - 郵便番号: お客様のご住所の郵便番号をハイフン付きで入力します。(必須) (例: 153-0064)
- AWS カスタマーアグリーメント (利用規約) に同意の上、②のチェックボックスをクリックしてください。
- すべて入力が終わりましたら、「アカウントを作成して続行」ボタンをクリックしてください。

#### ステップ 3: お支払い情報入力 (クレジットカード情報登録)

[https://aws.amazon.com/jp/getting-started/resource-center/?nc2=h\\_ql\\_gs&awsml=ql-4](https://aws.amazon.com/jp/getting-started/resource-center/?nc2=h_ql_gs&awsml=ql-4)

クレジットカード情報登録画面はこちら

# AWS IoT開発

## ②. AWS設定

- 手順3: クレジットカード情報を入力します。
  - 1年間は無料で利用できますが、支払いが発生した場合のクレジット情報を入力します。

※クリックすると大きな画像でご覧いただけます。

• すべて入力が終わりましたら、「アカウントを作成して続行」ボタンをクリックしてください。

### ステップ 3: お支払い情報入力 (クレジットカード情報登録)

次にクレジットカード情報登録を行います。

※新規お申込みのお客様は 1 年間の無料利用枠をご利用いただけますので、AWS アカウント作成、無料利用枠内のご利用で請求が行われることはありません。

- ①の入力欄にお客様の有効なクレジットカード情報を入力します。
- ②の選択欄で請求先住所を選択します。前のステップでご入力いただいたご住所と同様の場合は、「連絡先住所を使用する」を選択します。アカウント作成時にご入力いただいたご住所と異なる請求先となる場合は、「新しい住所を使用する」を選択して、請求先住所をご入力ください。
- すべて入力が終わりましたら、「次へ」ボタンをクリックします。

※お申し込み時にはクレジットカードが必須となりますが、ご利用いただく間にご利用料金が大きくなっているお客様の場合、請求書による支払いへ変更いただくことができます場合がございます。詳しくはこちらをご覧ください。

※クリックすると大きな画像でご覧いただけます。

### ステップ 4: 日本語自動音声電話によるアカウント認証

新規作成していただいたアカウントのご本人確認として、電話 (自動音声) による認証を行います。

※ ご入力いただいた電話番号に、日本語の自動音声による PIN の入力を求める確認電話が直ちにかかってきます。

- 国コードで日本または会社所在の国を選択し、この場で連絡を受けることができる電話番号を ① にハイフン・記号なしで入力します。(例: 0312345678)
- セキュリティチェックのため、セキュリティチェック文字列として表示された英数字を入力します。

電話による確認

すぐにお客様に自動通話が発信されます。求められたら、AWS ウェブサイトからの 4 桁の番号を電話のキーボードで入力してください。

電話番号の入力  
以下に情報を入力し、[次に連絡を受ける] をクリックしてください。

① 国コード

## AWS IoT開発

### ②. AWS設定

- 手順4: アカウント認証を実施します。
- 電話でPIN通知があり、情報を入力します

**ステップ 4: 日本語自動音声電話によるアカウント認証**

電話による確認

新規作成していただいたアカウントのご本人確認として、電話（自動音声）による認証を行います。

※ ご入力いただいた電話番号に、日本語の自動音声による PIN の入力を求める確認電話が直ちにかかってきます。

- 国コードで日本または会社所在の国を選択し、この場で確認を受けることができる電話番号を ① にハイフンで記号なしで入力します。（例: 0312345678）
- セキュリティチェックのため、セキュリティチェック文字列として表示された英数字を ② に入力します。
- 電話番号をご入力の上、③の「すぐに連絡を受ける」ボタンをクリックします。

※以下の点にご注意ください。

- 携帯電話等で非通知の着信拒否設定を行っている場合は、着信拒否設定の解除が必要です。「すぐに連絡を受ける」ボタンをクリックする前に、必ず電話がかかる状態にしておいてください。
- ③のボタンをクリックすると、即座に電話がかかってきます。すぐに着信を受け取れる電話番号をご入力ください。
- 国番号選択を誤ると電話がかかってきませんので、必ず正しい国コードを選択の上、ハイフンなしで電話番号をご入力ください。

電話による確認

電話認証の際にご入力いただく 4 桁の PIN（明証番号）が画面に表示されます。

指定した番号に電話がかかってきましたら、電話の自動音声に従って、画面に表示された暗証番号を電話のプッシュボタンで入力します。

電話による確認

電話による確認

電話による確認

本人確認が完了しました

## AWS IoT開発

### ②. AWS設定

- 手順5: サポートプランを入力します。
- ひとまず無料のベーシックプランを選択します。

※クリックすると大きな画像でご覧いただけます。

### ステップ 5: AWS サポートプランの選択

サポートプランの選択

AWS では、お客様のニーズに合ったさまざまなサポートプランをご用意しています。お客様の AWS の使用に最も合ったサポートプランを選択してください。詳細はこちら

ベーシックプラン	開発者プラン	ビジネスプラン
無料	29 USD/月	100 USD/月
<ul style="list-style-type: none"> <li>すべてのアカウントに含まれます</li> <li>フォーラムとリソースへの 24 時間 365 日対応のセルフサービスアクセス</li> <li>セキュリティとパフォーマンスを向上させるためのベストプラクティスのチェック</li> <li>ヘルステータスと通知へのアクセス</li> </ul>	<ul style="list-style-type: none"> <li>早期の採用、テスト、開発</li> <li>AWS サポートへの営業時間中の E メールでのアクセス</li> <li>1 人の主な担当官による無制限のサポートケースのオープンが可能</li> <li>非実稼働システムに対する 12 時間の応答時間</li> </ul>	<ul style="list-style-type: none"> <li>実稼働のワークロードおよびビジネスクリティカルな依存関係</li> <li>AWS サポートへの 24 時間 365 日対応のチャット、電話、E メールでのアクセス</li> <li>無制限の担当官による無制限のサポートケースのオープンが可能</li> <li>実稼働システムに対する 1 時間の応答時間</li> </ul>

エンタープライズレベルのサポートが必要ですか？  
AWS でのビジネスワークロードおよびミッションクリティカルなワークロードの実行の詳細については、アカウントマネージャーにお問い合わせください (15,000 USD/月)。詳細はこちら

※クリックすると大きな画像でご覧いただけます。

↑ 最初に戻る

- 最後に AWS のサポートプランの選択を行います。
- 特に有償のサポートを現時点で必要としない場合は、無料のボタンをクリックします。

※開発者、ビジネス、エンタープライズのいずれかのプランをご選択した場合、月額最低サポート料金が加入時に請求となります。サポートプラン概要は [AWS サポート](#) をご参照ください。

※AWS サポートの有償プラン（開発者、ビジネス、エンタープライズ）は AWS アカウント作成後にお申込みいただくこともできます。

### AWS アカウントの作成完了です



おめでとうございます！これで、AWS アカウントの作成が完了しました！

## AWS IoT開発

### ②. AWS設定

- 手順6: アカウント作成完了。

メニュー 日本担当チームに問い合わせる 製品 ソリューション 料金 開始方法 詳細 日本語 アカウント [コンソールへログイン](#)

↑ 最初に戻る

### AWS アカウントの作成完了です

おめでとうございます！これで、AWS アカウントの作成が完了しました！

- 数分ほどで、ご登録メールアドレス宛に確認のための E メールが届きます。画面右側の「マネジメントコンソールを起動」ボタンをクリックすると、すぐにアマソンのクラウドを使用開始していただくことができます。

アマゾンウェブ サービスへようこそ

アマゾンウェブ サービス (AWS) アカウントを作成いただいたことありがとうございます。AWS をご利用いただけるように現在アカウントを準備中です。ほとんどの場合、アカウントの準備が完了してすぐにご利用いただけます。アカウントの準備が完了したらご登録のメールアドレスに、確認のメールが届きます。確認が完了し、アカウントが有効になりましたらご登録のメールアドレスでログインしてください。

マネジメントコンソールを起動

サインアップ

今すぐサインアップ

↑ 最初に戻る

AWS の Twitter Facebook Google+ Twitch AWS ブログ 最新情報RSS

アップデートをサブスクライブ

コンソールにサインイン

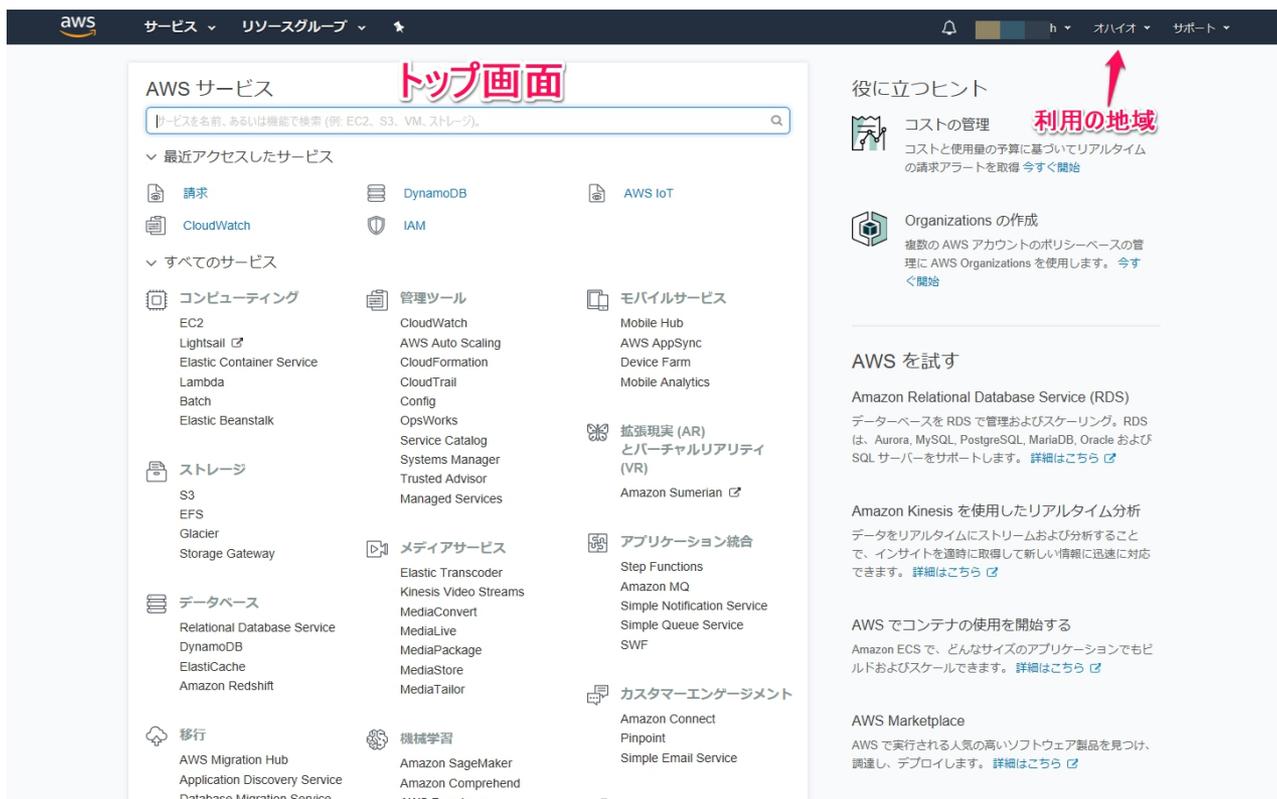
AWS とクラウドコンピューティングクラウドとは	ソリューションウェブサイトとウェブサイトホスティング	リソースとトレーニングクラウド活用資料集AWS での Java	アカウントの管理マネジメントコンソール請求 & コスト管理	アマゾンウェブ サービスでは人材を募集中です。アマゾンウェブ サービス (AWS) は、ダイナミックに成長している
--------------------------	----------------------------	---------------------------------	-------------------------------	---

# AWS IoT開発

## ②. AWS設定

- AWSトップ画面

- 利用する場合に、地域を意識して設定してください。地域毎に料金が変わったり、利用できるサービスが限定されている場合があります。



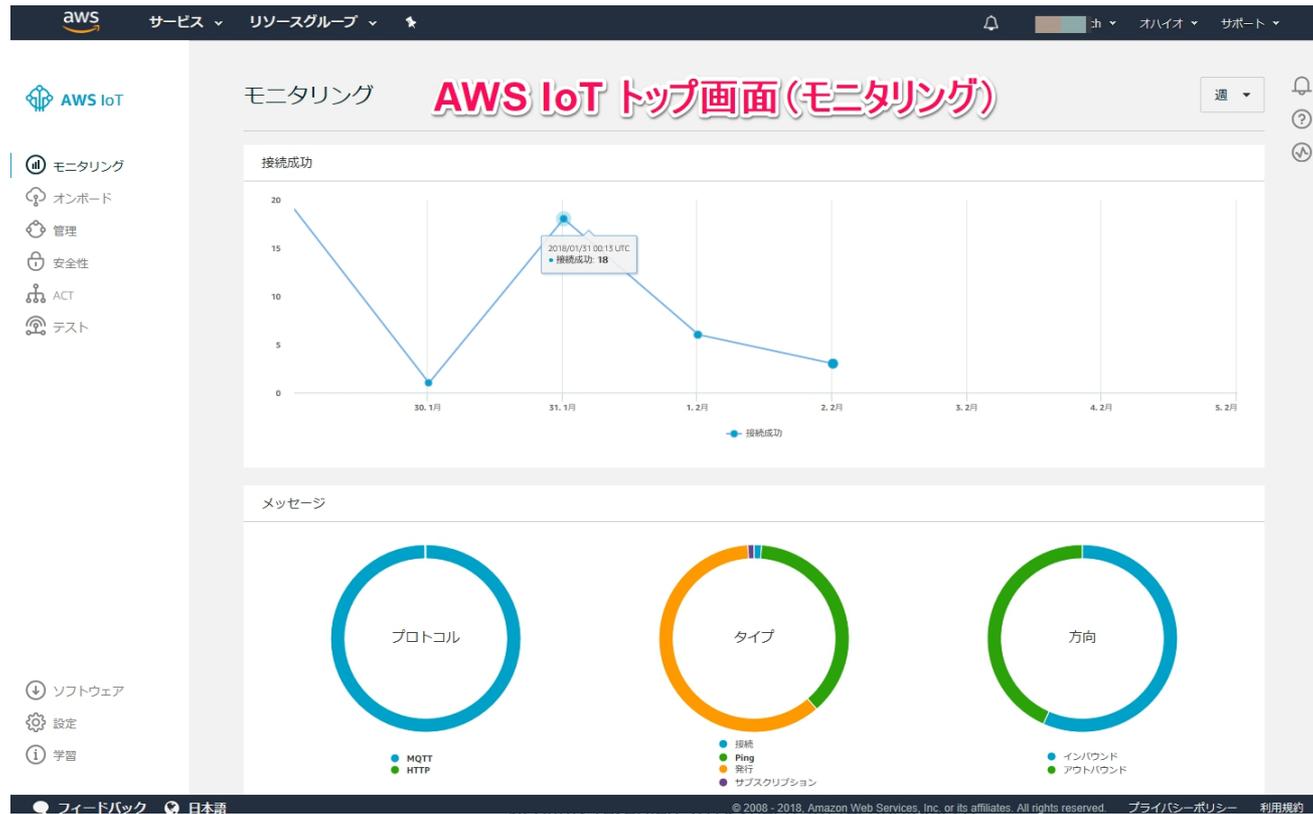


## AWS IoT開発

### 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

The screenshot shows the AWS IoT console interface. On the left sidebar, the 'モノ' (Things) menu item is highlighted with a red box and a red arrow. Another red arrow points to the 'オンボード' (Onboarding) menu item. The main content area displays the 'モノ' (Things) management page with the following text:

**モノ**

basicpi BASIC    pi1 タイプなし

**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。**

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。

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。

今すぐ始める    ボタンの設定

ボタンをお持ちでない場合 購入する



## AWS IoT開発

### ③. AWS IoT設定

- 画面説明3
  - 安全性: 証明書
  - ACT: ルール

**証明書**

モノを接続する時に使用する電子証明書です。  
基本的に通信は、MQTTで暗号化されて行われます。

**ルール**

AWS IoTで受信したMQTTをDynamoDBなどに  
データを確保するなどのルールを設定します。

### ③. AWS IoT設定



AWS IoT

- 画面説明4
  - テスト:MQTTクライアント

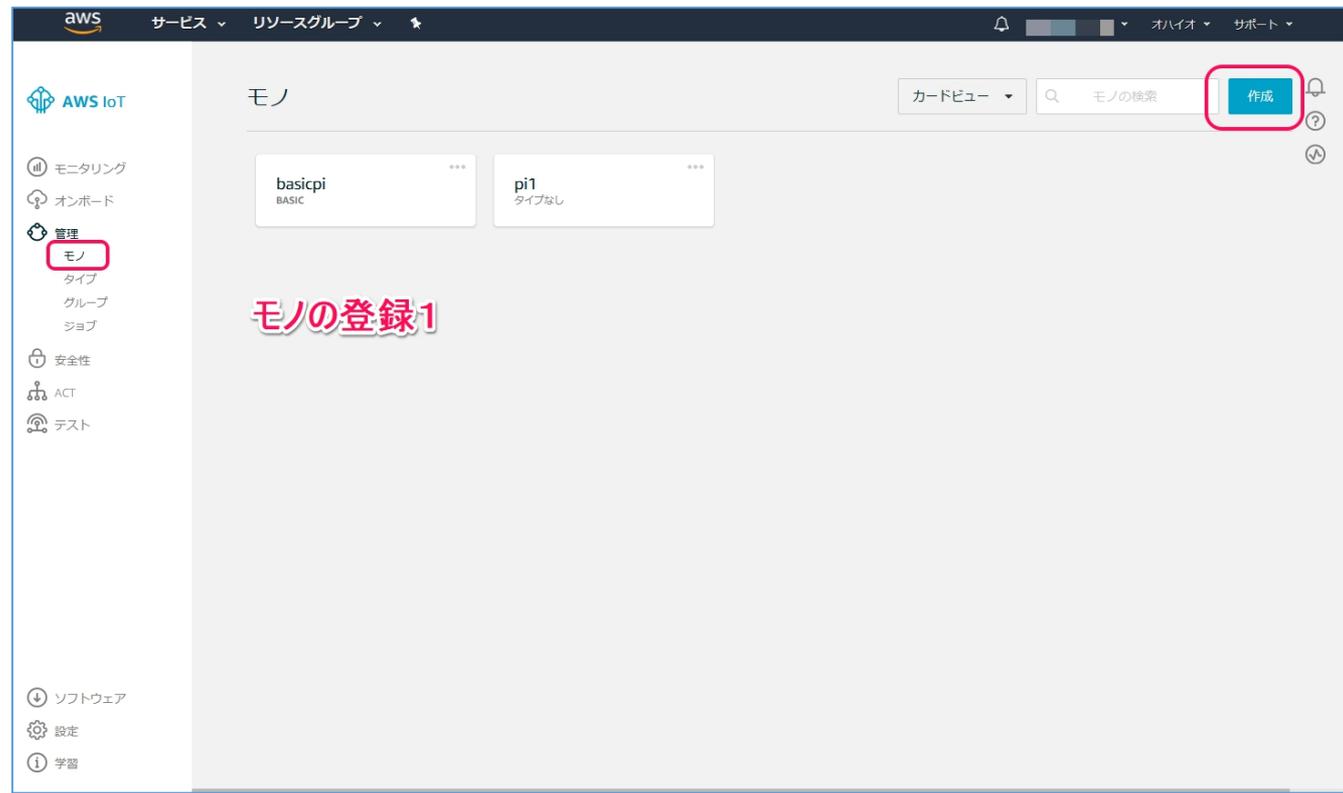
The screenshot shows the AWS IoT console interface for the 'MQTT Client' page. The left-hand navigation menu has the 'テスト' (Test) option highlighted with a red box and a red arrow pointing to it. The main content area features a red banner at the top that reads 'クライアントとしてのテストを行います' (Perform test as a client). Below this, there are two main sections: 'サブスクリプション' (Subscription) and '発行' (Publish). The 'サブスクリプション' section includes a text input field for the topic name (example: myTopic/1) and a 'トピックへのサブスクライブ' (Subscribe to topic) button. The '発行' section includes a text input field for the topic name (example: myTopic/1) and a 'トピックに発行' (Publish to topic) button. The page also contains various configuration options such as 'メッセージキャプチャの最大数' (Maximum number of message captures) set to 100, and 'MQTT ペイロード表示' (MQTT payload display) options.

### ③. AWS IoT設定



AWS IoT

- モノの登録1
  - 管理>モノの作成を押します。



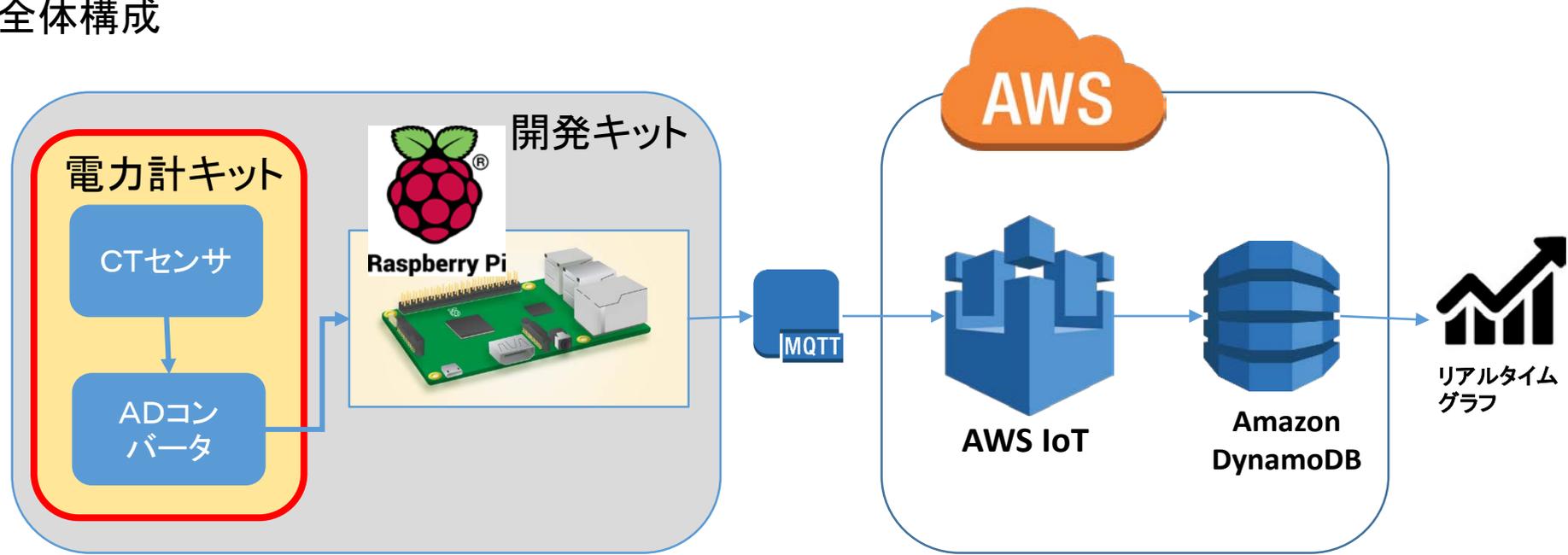


AWS IoT

## AWS IoT開発

### ④. 電力計

全体構成



分電盤の電源ケーブルにCTセンサを取り付けて、その情報をADコンバータを通し、PythonプログラムでAWS IoTに送信します。

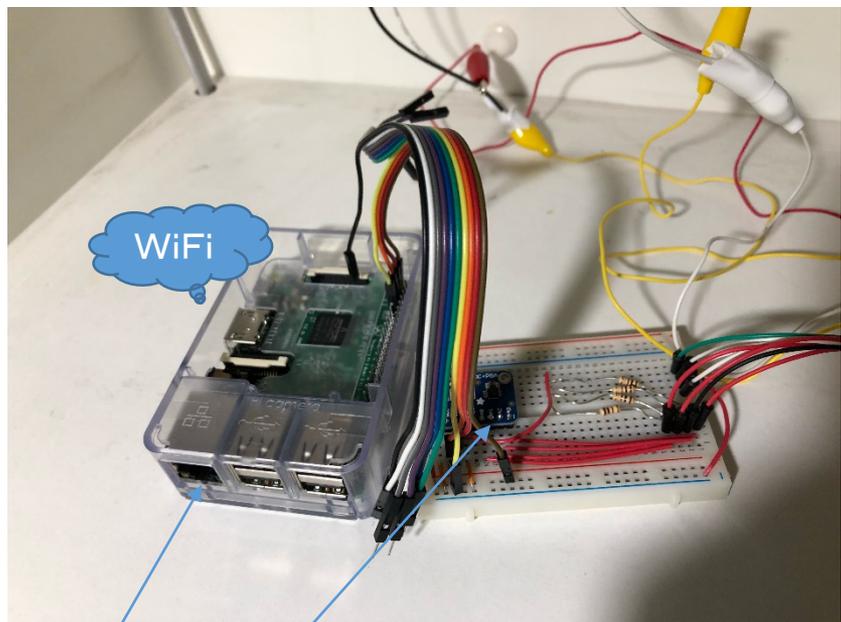
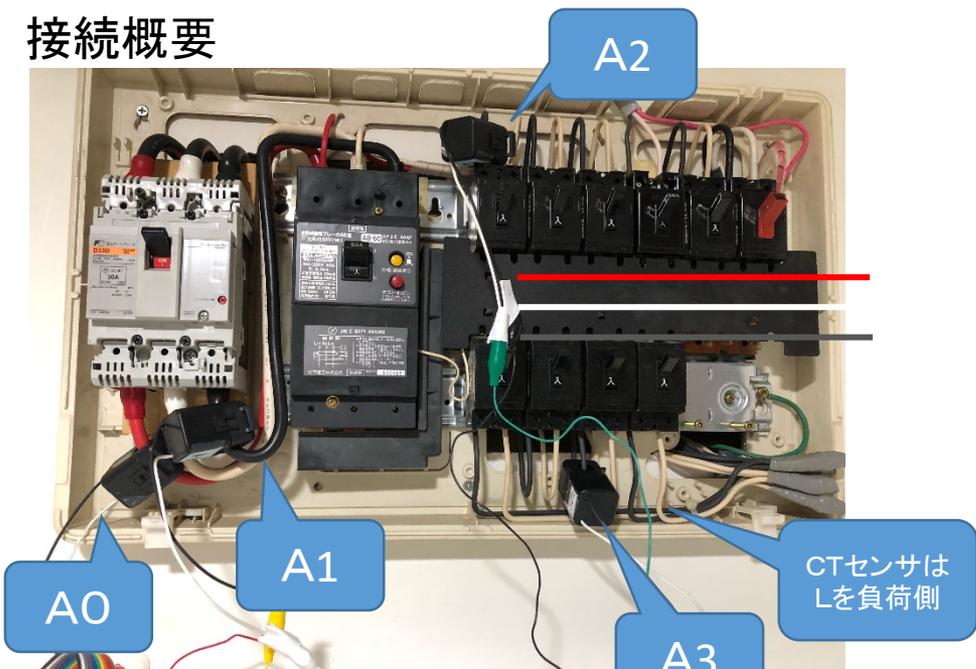
Amazon DynamoDBに保存された電力情報をリアルタイムに表示します。

電力計キットは、開発キットのオプションになります。電力計キット単独では動作しません。

## AWS IoT開発

### ④. 電力計

#### 接続概要



マンションの引き込み: 単相3線式

CTセンサ: 電源ケーブルの電流値を読み取り電圧で出力します。

- A0: 赤 幹線
  - A1: 黒 幹線
  - A2: 居間
  - A3: エアコン
- } 合計値が総電力

ADコンバータ

CTセンサで取得した電圧を読み取り、数値に変換します。

RaspberryPi

ADコンバータとGPIOで接続し、内部で処理し、WiFiを経由してMQTT送信します。

電源は、モバイルバッテリー又はAC100VからUSB電源アダプタを使用します。

## AWS IoT開発

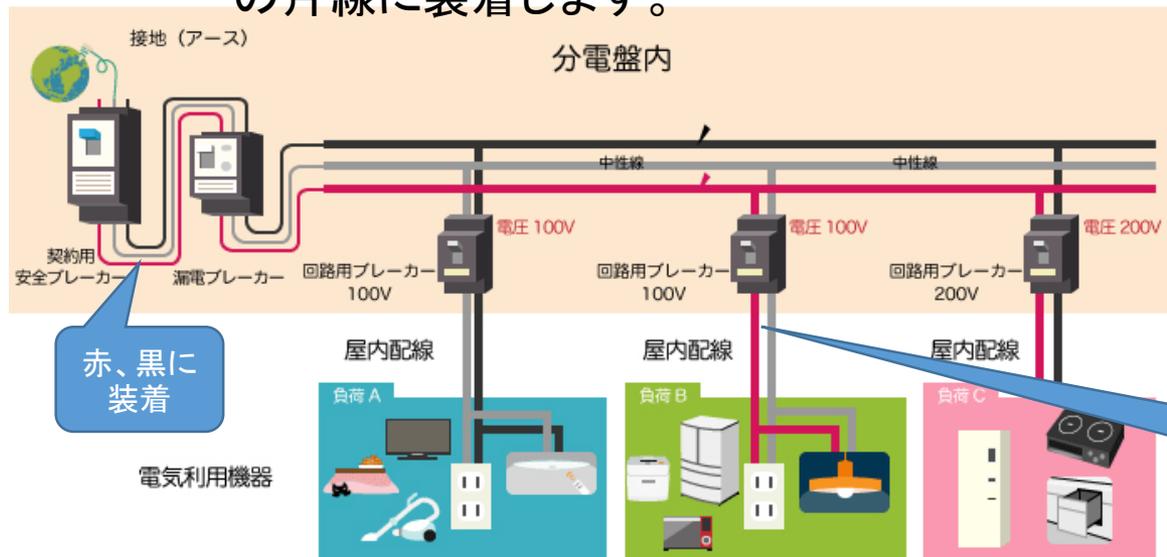
### ④. 電力計

#### CTセンサの装着方法

- **分電盤の工事は、電気工事士の資格が必要です。** CTセンサの取付は資格不要です。
- 単相3線式(家庭用)の場合
  - 主幹の赤、黒にCTセンサを装着します。
  - 白は、中性線のため装着しません
  - 個別のブレーカの場合は、黒または赤の片線に装着します。



- CTセンサには方向があります
- L: 負荷側
- K: 電源側



赤、黒に装着

赤または黒に装着

## AWS IoT開発

### ④. 電力計

#### 単体試験

- CTセンサ取付、Raspberry Pi接続完了。
- 単体試験を実施します。

rootで立ち上げ  
以下のフォルダに移動し単体試験の実施  
# cd /home/pi/Documents/ct  
# python ct\_test.py

```
pi@raspberrypi:~$ cd /home/pi/Documents/ct
pi@raspberrypi:~/Documents/ct$ ls
Adafruit_Python_ADS1x15  ct_mqtt_subscribe_json.py  pi1.private.key
DynamoDBtoCSV           ct_test.py                 pi1.public.key
ct_mqtt_publish_json.py  pi1.cert.pem              root-CA.crt
pi@raspberrypi:~/Documents/ct$ python ct_test.py
Reading ADS1x15 voltage, press ctrl+c to quit
time | 0 | 1 | 2 | 3 |
-----|-----|-----|-----|-----|
1 2018/03/27 07:45:07.549225 | -42 | 55 | 17 | -2 |
1 291.0 2018/03/27 07:45:07.549225 | 126.0 | 165.0 | 51.0 | 6.0 |
2 2018/03/27 07:45:09.558360 | 48 | -44 | -102 | 1 |
2 575.713522185 2018/03/27 07:45:09.558360 | 135.3 | 149.4 | 219.4 | 4.7 |
3 2018/03/27 07:45:11.567415 | -8 | 6 | 121 | -3 |
3 809.488586742 2018/03/27 07:45:11.567415 | 111.3 | 122.4 | 275.7 | 6.5 |
4 2018/03/27 07:45:13.575331 | -53 | 10 | -8 | 0 |
4 1041.54749839 2018/03/27 07:45:13.575331 | 125.0 | 107.1 | 239.1 | 5.6 |
5 2018/03/27 07:45:15.584271 | 67 | -17 | -16 | -2 |
5 1283.44607092 2018/03/27 07:45:15.584271 | 143.4 | 98.5 | 214.9 | 5.7 |
6 2018/03/27 07:45:17.593247 | -193 | 7 | 20 | 1 |
6 1643.95662185 2018/03/27 07:45:17.593247 | 270.2 | 90.3 | 197.7 | 5.3 |
Traceback (most recent call last):
```

プリント文: 1行目は、ADSの値  
2行目は、累積Watt





## AWS IoT開発

### ④. 電力計

AWS DynamoDBでの確認

- テーブル > 設定したテーブル > 項目
- やりとりのデータが入っていればOK

The screenshot shows the AWS DynamoDB console interface. On the left sidebar, the 'DynamoDB' section is expanded, and the 'テーブル' (Tables) option is highlighted with a red box. Below it, the table 'ct1\_db' is selected, also highlighted with a red box. The main content area shows the 'ct1\_db' table details, with the '項目' (Items) tab selected and highlighted with a red box. The table has two primary keys: 'deviceid' and 'timestamp'. Below the table details, there is a 'スキャン' (Scan) section with a dropdown menu set to '[テーブル] ct1\_db: deviceid, timestamp'. A red text overlay 'データが入力されている' (Data is being input) is placed over the data table.

deviceid	timestamp	payload
ct	2018/03/23 17:36:18	{"A0": {"N": "480"}, "A1": {"N": "36"}, "A2": {"N": "..."}}
ct	2018/03/23 17:36:28	{"A0": {"N": "344.9412993539625"}, "A1": {"N": "..."}}
ct	2018/03/23 17:36:38	{"A0": {"N": "283.98239382046205"}, "A1": {"N": "..."}}
ct	2018/03/23 17:36:48	{"A0": {"N": "271.7071033300381"}, "A1": {"N": "..."}}
ct	2018/03/23 17:36:58	{"A0": {"N": "320.7251159482213"}, "A1": {"N": "..."}}
ct	2018/03/23 17:37:08	{"A0": {"N": "430.77430285475475"}, "A1": {"N": "..."}}
ct	2018/03/23 17:37:18	{"A0": {"N": "482.1807544894342"}, "A1": {"N": "..."}}
ct	2018/03/23 17:37:28	{"A0": {"N": "478.6304628834232"}, "A1": {"N": "..."}}
ct	2018/03/23 17:37:38	{"A0": {"N": "451.7930942367313"}, "A1": {"N": "..."}}
ct	2018/03/23 17:37:48	{"A0": {"N": "430.7293813985761"}, "A1": {"N": "..."}}
ct	2018/03/23 17:37:58	{"A0": {"N": "410.6933162348762"}, "A1": {"N": "..."}}



## AWS IoT開発

### ④. 電力計

リアルタイム表示

- IoT. Kyoto様の利用>グラフ
- Deviceid:ct
- CTセンサ主幹赤のWatt推移

<https://vis.iot.kyoto/>





AWS IoT

## AWS IoT開発

### ④. 電力計

リアルタイム表示

- IoT. Kyoto様の利用>グラフ
- Deviceid: ct
- 集計したkwh

<https://vis.iot.kyoto/>

