

抜粋版

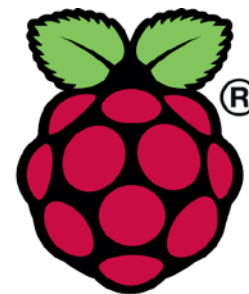
新型肺炎対策：体温検知キット

～顔認識、サーモセンサを使い、自動検温し、AWSにデータ蓄積、
活用～

実践編



AWS IoT Core



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

体温検知キット 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)

AWS IoT開発

- ① メニュー
- ② AWS設定
- ③ AWS IoT設定
- ④ サーモセンサ情報収集、アラーム送付
 - ・ 全体構成
 - ・ サーモセンサ概要
 - ・ サーモセンサRaspberryPi接続
 - ・ サーモセンサ単体試験
 - ・ サーモセンサ画像試験
 - ・ Mqttプログラム設定
 - ・ AWS IoT折り返し試験
 - ・ リアルタイム表示(グラフ)
 - ・ IoTイベント設定
 - ・ IoTイベント試験
 - ・ 遠隔リアルタイムモニタ
 - ・ CSV取り出し
 - ・ CSV画像再生

Opencv開発

体温検知

1. リアルタイム検知
2. データ活用

ページ

[3](#)

[4](#)

[5](#)

[6](#)

ページ

[8](#)

[9](#)

[16](#)

[29](#)

[30](#)

[31](#)

[32](#)

[33](#)

[34](#)

[36](#)

[37](#)

[39](#)

[47](#)

[50](#)

[51](#)

[53](#)

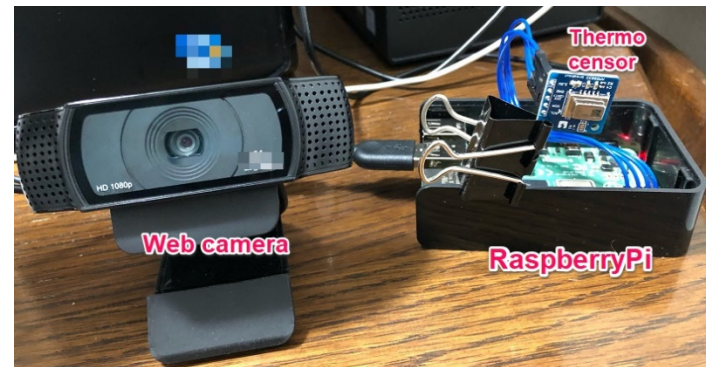
[55](#)

[57](#)

[59](#)

抜粋版

ページは一
致しません



RaspberryPi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

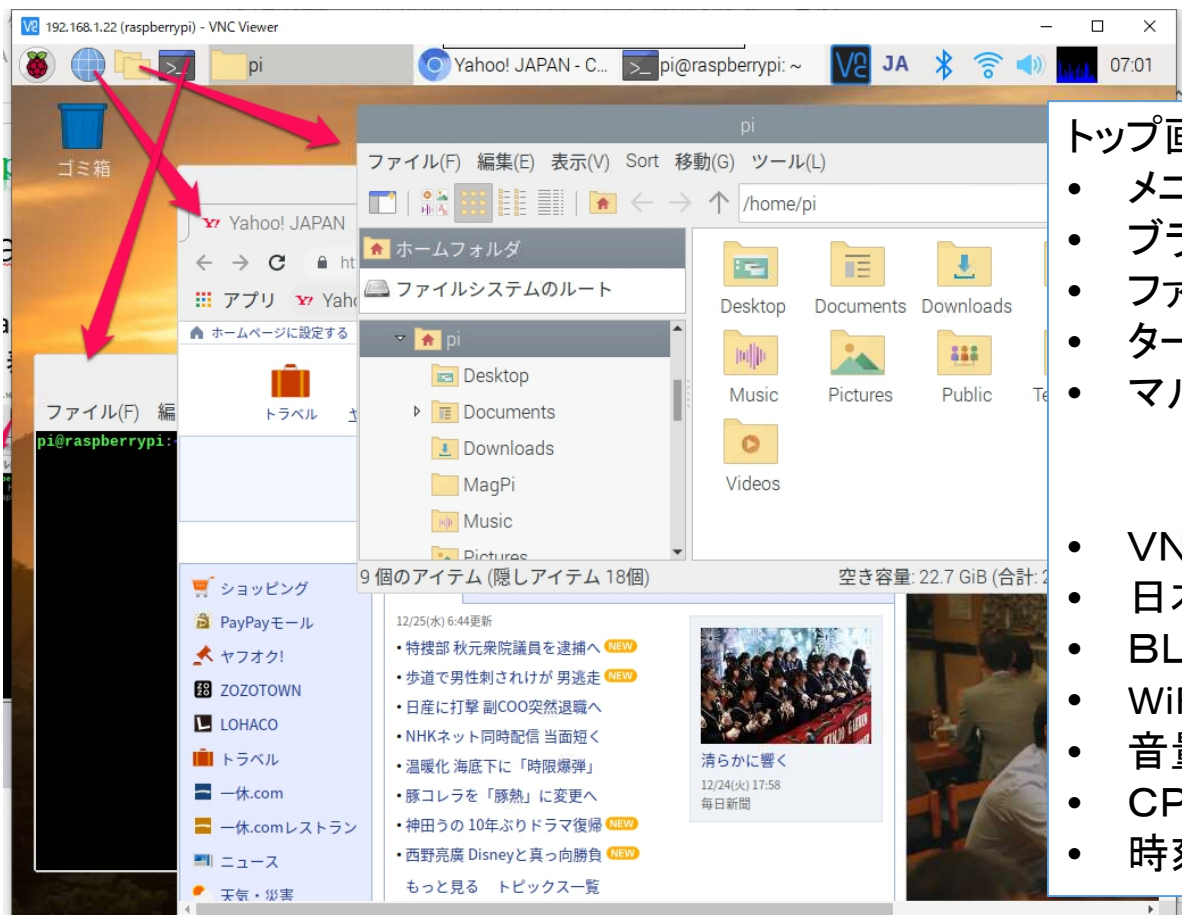
- 起動: 電源を入れると自動で起動します。
- 再起動: `$ reboot`
又は、`menu>shutdown>reboot`; 左上のメニューから
- 終了: `$ shutdown`
又は、`menu>shutdown>shutdown`; 左上のメニューから
- ログアウト `$ exit`
又は、`menu>shutdown>logout`; 左上のメニューから
- **日本語／英語の入力切替**: キーボードの CTL と j を同時に押します (コントロール: 左下とj)

RaspberryPi運用マニュアル

3. Raspberry Piの基本操作

① 表示画面と内容

デスクトップ上によく使うコマンド.txtがあります。
コピーして使ってください



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
- CPU使用率
- 時刻

AWS IoT開発



AWS IoT Core

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
 - 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。

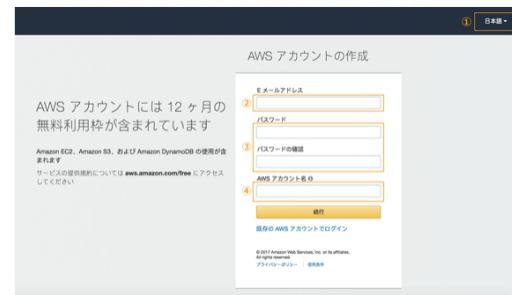


AWS アカウント作成の流れ

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。



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。

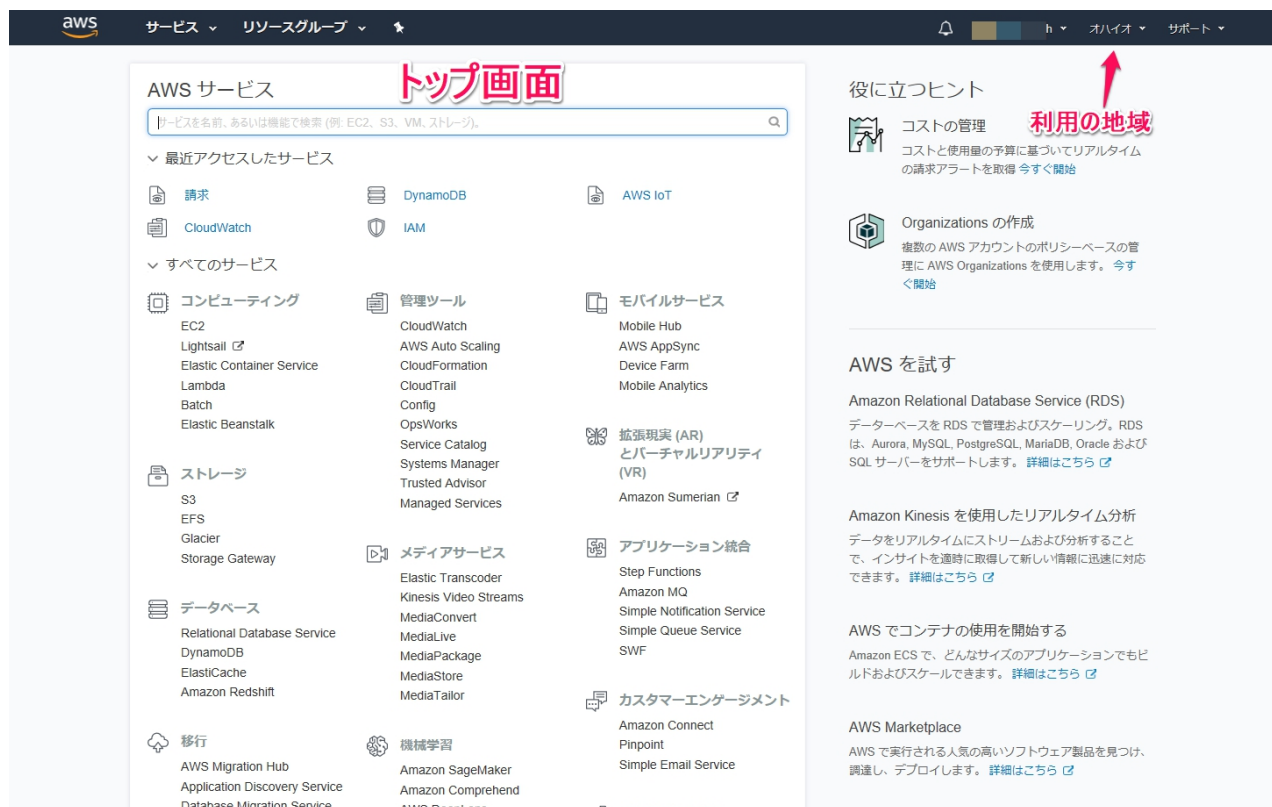
※ご登録いただくメールアドレスは、AWS 側からの通知等にも利用されます。複数の方へ

AWS IoT開発

②. AWS設定

• 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。



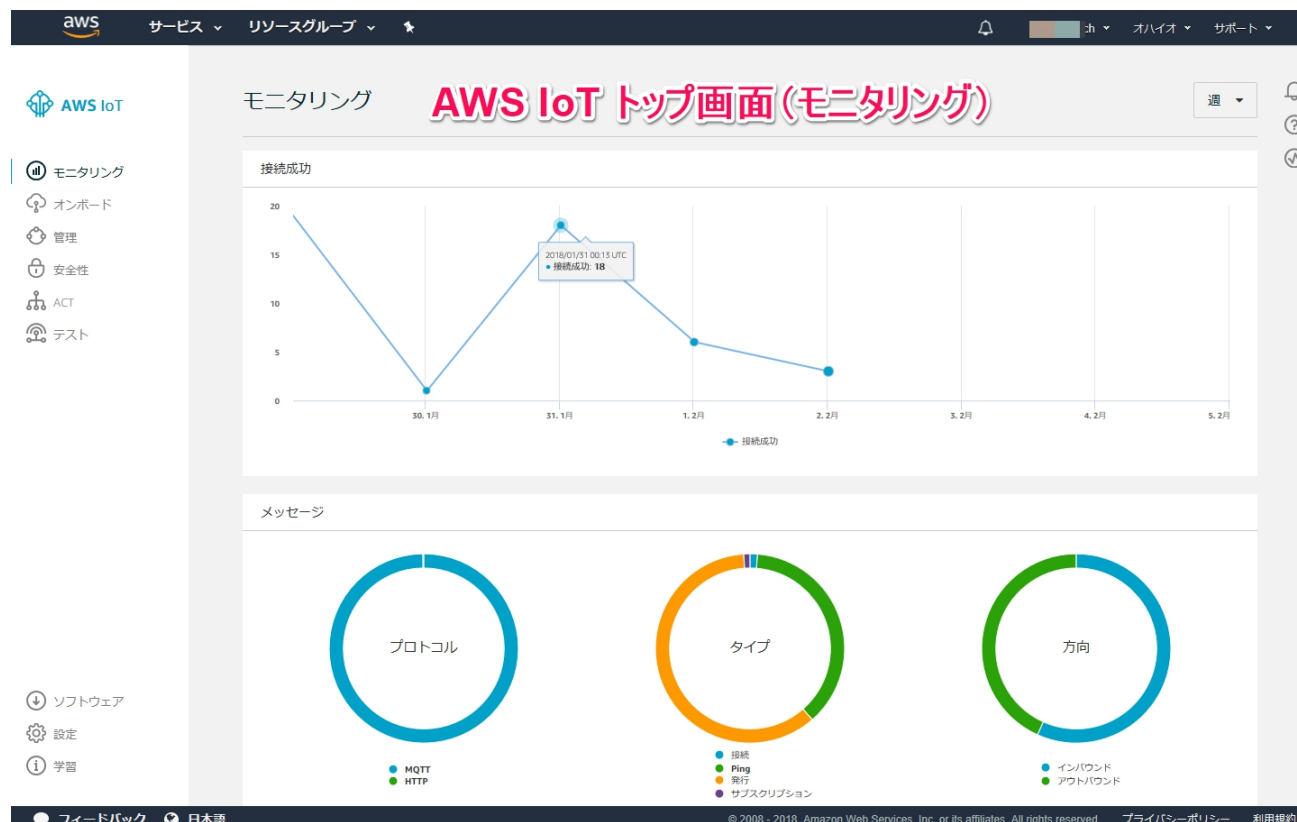
③. AWS IoT設定



AWS IoT Core

• 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



AWS IoT Core

- 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 押して 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

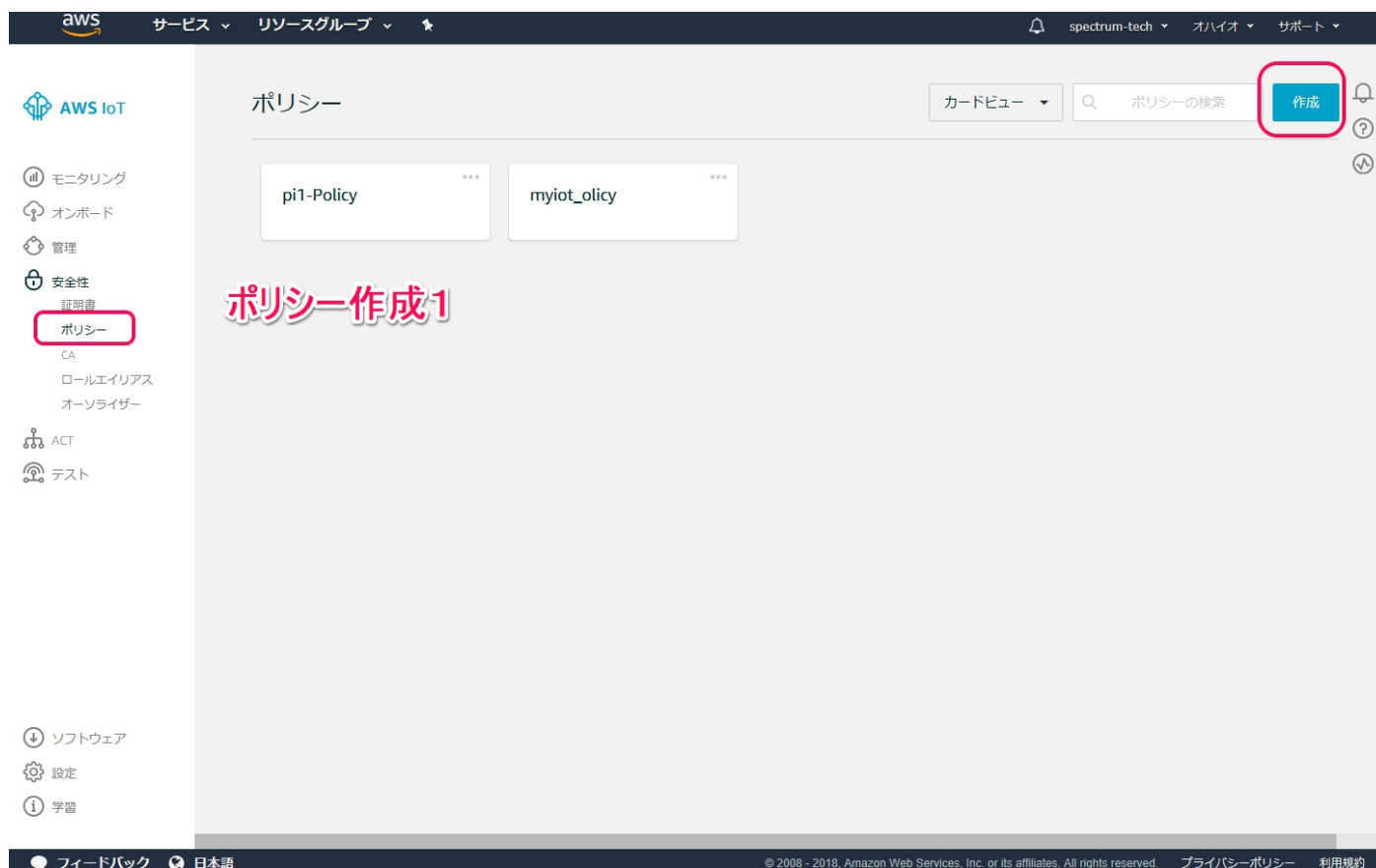
AWS IoTに接続するモノを登録します。

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。



③. AWS IoT設定



AWS IoT Core

• ポリシー作成3

- 安全性>証明書>モノの登録で作成した証明書にポリシーをアタッチします。
- 右クリックして>ポリシーのアタッチを押します

証明書 **ポリシー作成3**

カードビュー

証明書の検索

作成

証明書ID	ステータス
259e1c5de191ddb98...	アクティブ
a4b75954229e44f37...	アクティブ
ef0b8f9919c64310f4...	アクティブ
235c44e7435ab45d8...	アクティブ
c1ab54dd2198105e1...	アクティブ
587e63...	アクティブ

有効化
無効化
取り消し
転送を許可する
転送を拒否する
転送を取り消す

ポリシーのアタッチ
モノをアタッチする
ダウンロード
削除

モノの登録で作成した証明書にポリシーをアタッチします。
右クリック>ポリシーのアタッチ

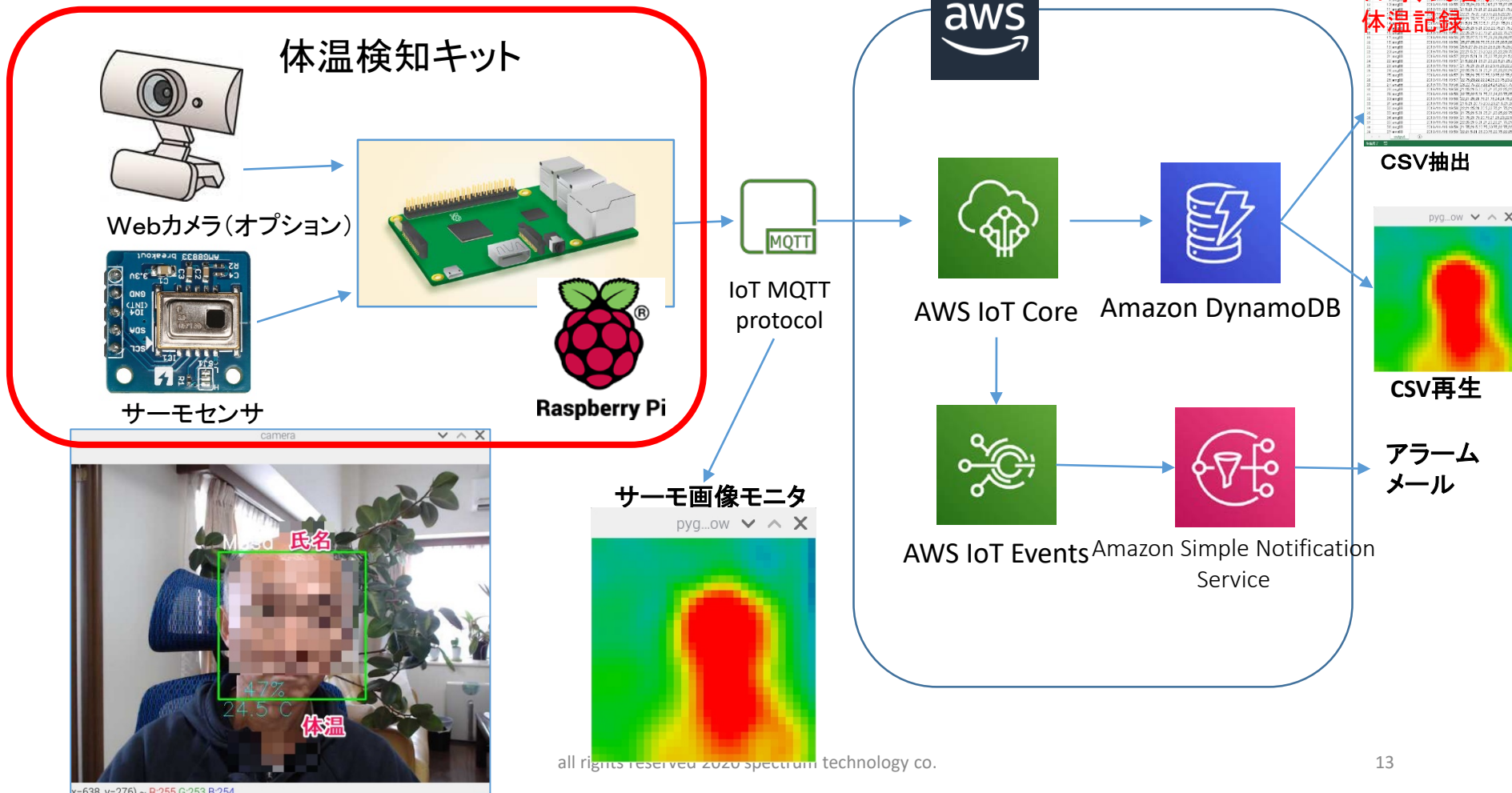
AWS IoT開発

④サーモセンサ情報収集、アラーム送付



AWS IoT Core

全体構成



日時、氏名、
体温記録

CSV抽出

CSV再生

アラーム
メール

AWS IoT開発

④サーモセンサ情報収集、アラーム送付

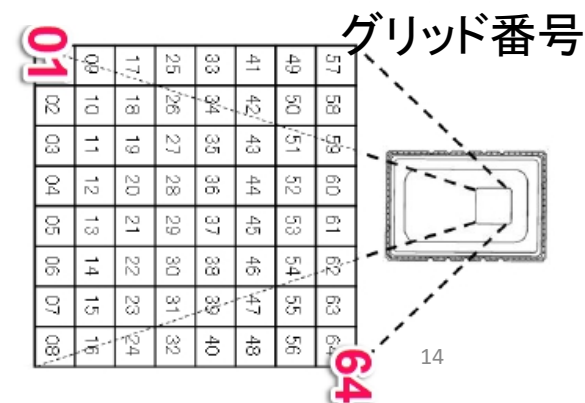
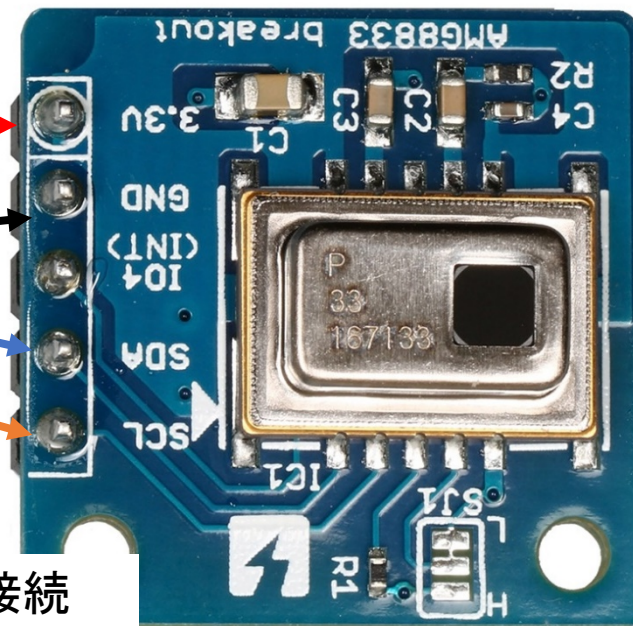
Raspberry PiとAMG8833の接続

上(取付はこの方向)

J8			
3.3V	1	2	3
BCM2	3	4	5
BCM3	5	6	7
BCM4	7	8	9
Ground	9	10	11
BCM17	11	12	13
BCM27	13	14	15
BCM22	15	16	17
3.3V	17	18	19
BCM10	19	20	21
BCM9	21	22	23
BCM11	23	24	25
Ground	25	26	27
	27	28	29
BCM5	29	30	31
BCM6	31	32	33
BCM13	33	34	35
BCM19	35	36	37
BCM26	37	38	39
Ground	39	40	

メスオスのジャンパで接続
Pi ブレッドボード AMG8833

1-----+-----VDD
3-----SDA
5-----SCL
9-----GND





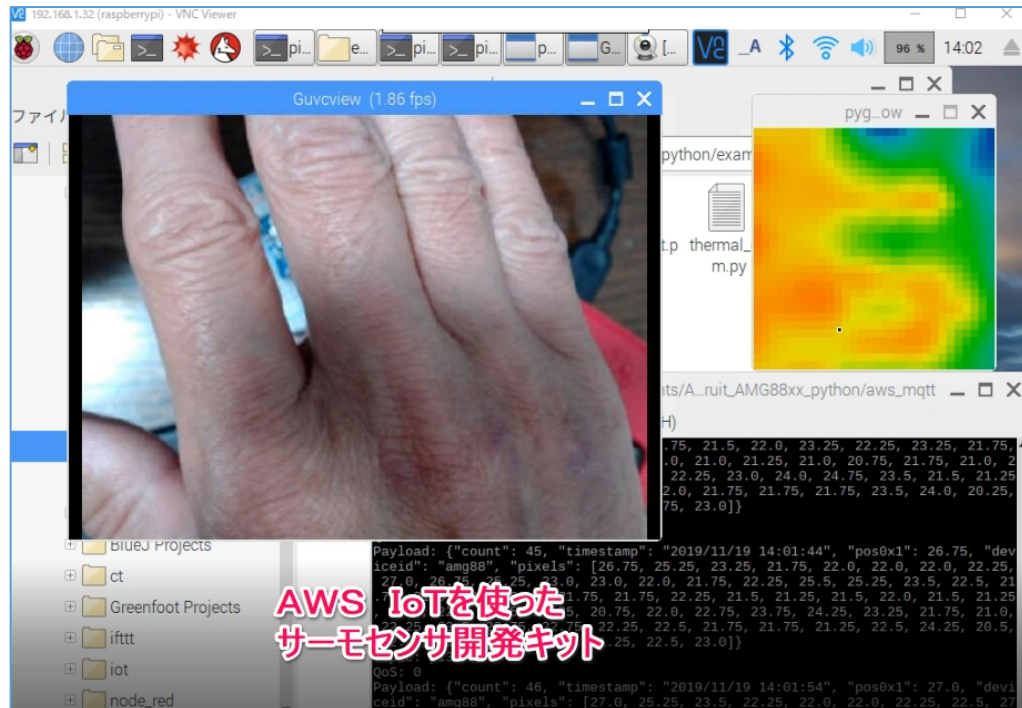
AWS IoT開発

④サーモセンサ情報収集、アラーム

AMG8833 画像試験

- Piコンソールでpython thermal_cam.pyを動作させます。
- 240x240ピクセルの画面で表示されます。
- プログラムで、最低:20度、最高:32度に設定しています。適宜数字を変えてください。設定温度が**実態と乖離していると同じ色**になります。設定を実態と近づけてください。
- 現在の温度は、python thermistor_test.pyでテストしてください。

```
# cd
/home/pi/Documents/Adafruit_AMG88xx_python/examples
# python thermal_cam.py
```



手をかざした
状態

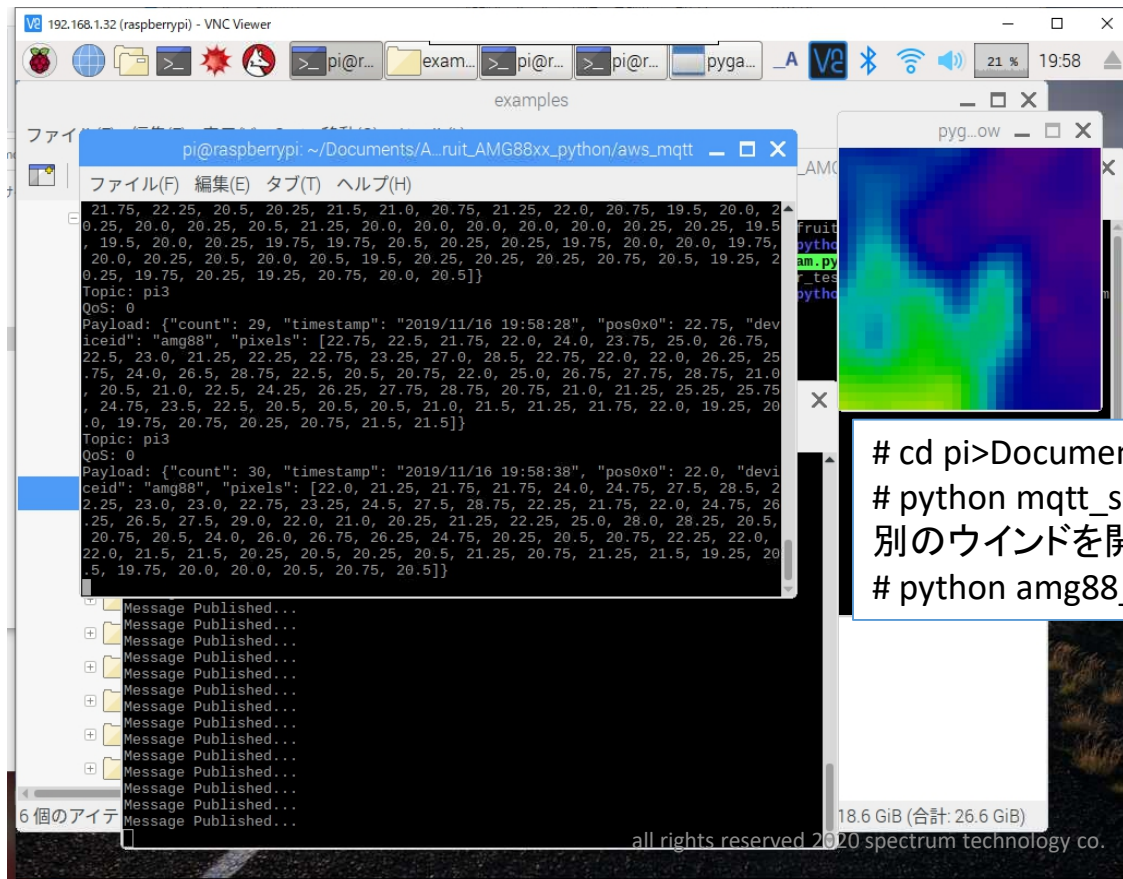
AWS IoTを使った
サーモセンサ開発キット

AWS IoT開発

④サーモセンサ情報収集、アラーム送付

AWS IoT折り返し試験

- Piのコマンド画面からsubscriber, publisherのやりとりを実施
- subscribe側に温度のデータ(リスト形式;64個)が受信できていればOk, なおテスト用に最高温度のデータをグラフ表示のため取り出し



The screenshot shows a VNC Viewer window titled "192.168.1.32 (raspberrypi) - VNC Viewer". The terminal window displays the following commands and output:

```
pi@raspberrypi: ~/Documents/A_ruit_AMG88xx_python/aws_mqtt
# cd pi>Documents>Adafruit_AMG88xx_python>aws_mqtt
# python mqtt_subscribe_json.py 受信側を先に立ち上げ
別のウインドを開き
# python amg88_mqtt_publish_json.py 送信側立ち上げ
```

The terminal output shows a list of 64 temperature values (e.g., 21.75, 22.25, 20.5, 20.25, 21.5, 21.0, 20.75, 21.25, 22.0, 20.75, 19.5, 20.0, 20.25, 20.0, 20.25, 20.5, 21.25, 20.0, 20.0, 20.0, 20.0, 20.25, 20.25, 19.5, 19.5, 20.0, 20.25, 19.75, 19.75, 20.5, 20.25, 20.25, 19.75, 20.0, 20.0, 19.75, 20.0, 20.25, 20.5, 20.0, 20.5, 19.5, 20.25, 20.25, 20.25, 20.75, 20.5, 19.25, 20.25, 19.75, 20.25, 19.25, 20.75, 20.0, 20.5]) and a "Message Published..." log.

The heatmap window shows a color-coded temperature distribution, with a color bar on the right indicating temperature values from 18.6 GiB (合計: 26.6 GiB) to 21.8 GiB.

AWS IoT開発

④サーモセンサ情報収集、アラーム送付

リアルタイム表示

- IoT. Kyoto様の利用>グラフ
- Deviceid:amg88(pythonで書き込み済)
- 表示対象をチェックすると自動で表示されます。



8x8=64個のデータの
うち、最大値のデー
タのみ表示。他はリ
スト形式でDBに確
保

AWS IoT開発

④サーモセンサ情報収集、アラーム送付

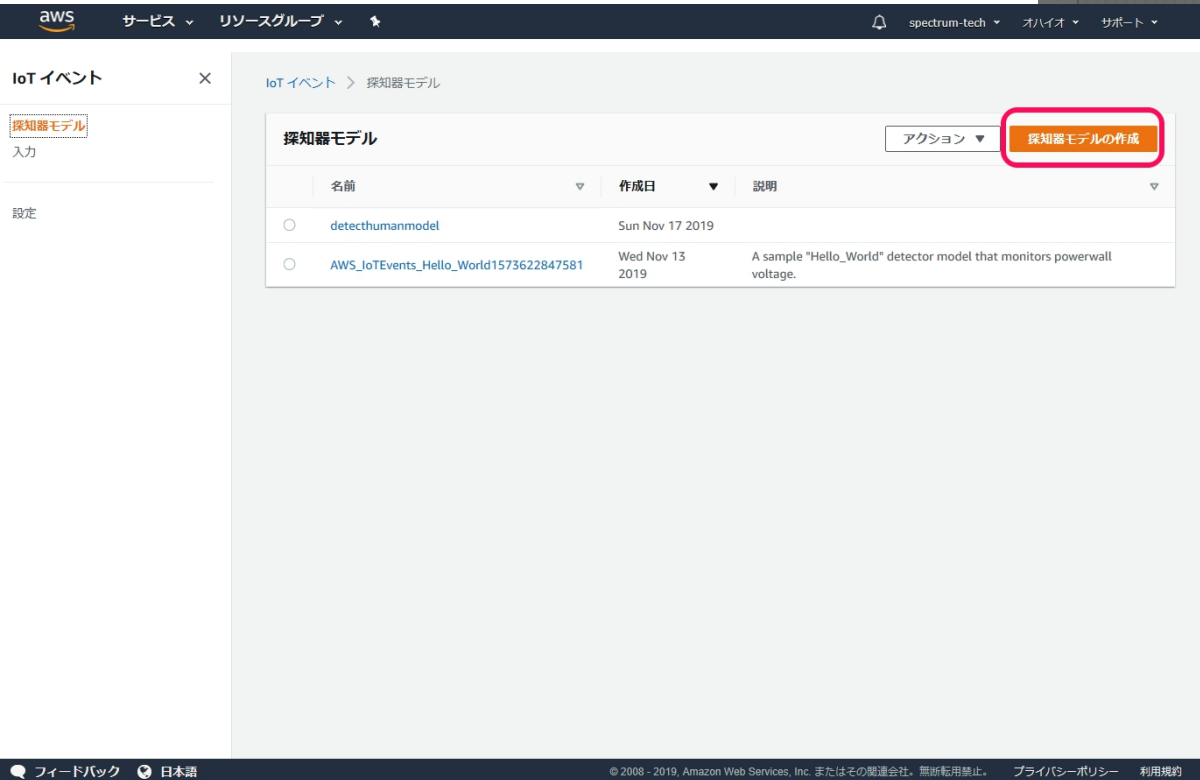
IoTイベント設定

- 指定設定温度で、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



The screenshot shows the AWS IoT console interface. On the left, the 'IoT イベント' (IoT Events) menu is open, and '探知器モデル' (Detector Model) is selected. The main panel displays the '探知器モデル' (Detector Model) page. At the top right of this panel, there is a button labeled '探知器モデルの作成' (Create Detector Model), which is highlighted with a red rectangular box. Below this, there is a table listing existing detector models. The table has columns for '名前' (Name), '作成日' (Created Date), and '説明' (Description). Two models are listed: 'detecthumanmodel' and 'AWS_IoTEvents_Hello_World1573622847581'. To the right of the table, there is a section titled '新しいディテクターモデルを作成' (Create new detector model) with a button labeled '新しく作成する' (Create new), also highlighted with a red rectangular box. Further right, there is a section titled '探知器モデルでの入力の使用' (Use input in detector model) with text explaining that input values are used for monitoring and triggering actions.



The screenshot shows the AWS IoT console interface. On the left, the 'IoT イベント' (IoT Events) menu is open, and '探知器モデル' (Detector Model) is selected. The main panel displays the '探知器モデル' (Detector Model) page. At the top right of this panel, there is a button labeled '探知器モデルの作成' (Create Detector Model), which is highlighted with a red rectangular box. Below this, there is a table listing existing detector models. The table has columns for '名前' (Name), '作成日' (Created Date), and '説明' (Description). Two models are listed: 'detecthumanmodel' and 'AWS_IoTEvents_Hello_World1573622847581'. To the right of the table, there is a section titled '新しいディテクターモデルを作成' (Create new detector model) with a button labeled '新しく作成する' (Create new), also highlighted with a red rectangular box. Further right, there is a section titled '探知器モデルでの入力の使用' (Use input in detector model) with text explaining that input values are used for monitoring and triggering actions.

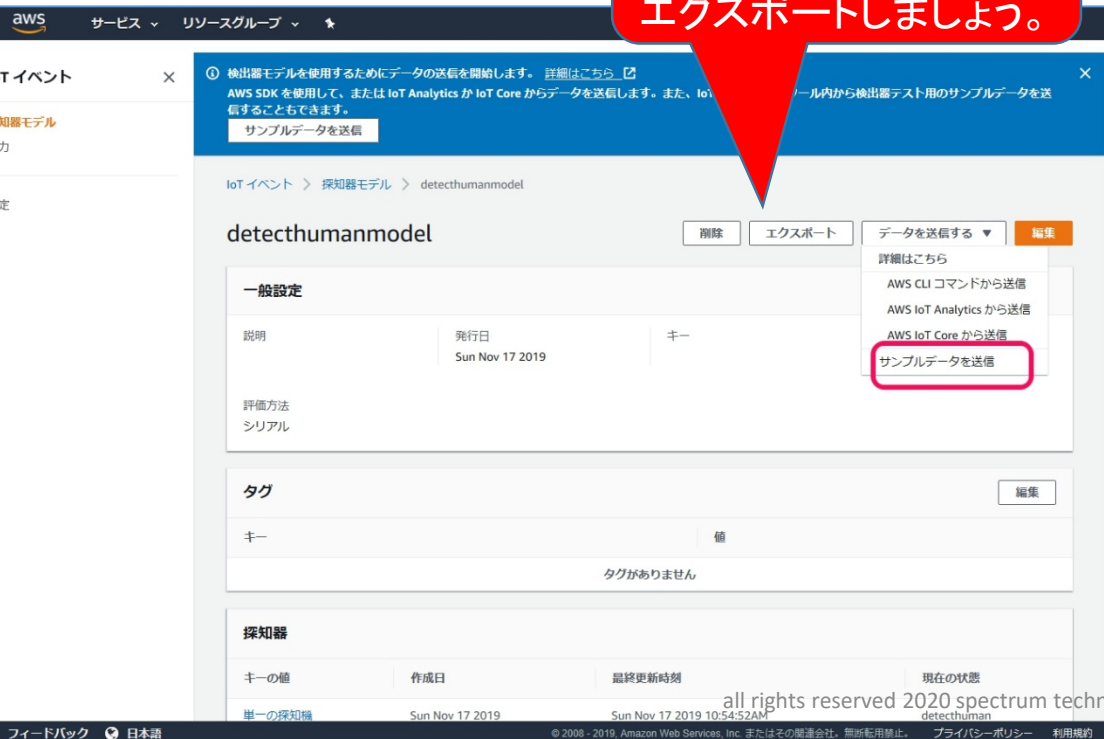
AWS IoT開発

④サーモセンサ情報収集、アラーム送付

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:inputevent
- ・ max_temp:25,番号と入力し、プレビュー
- ・ データの送信

料金の発生を防ぐために、使用しない時は、エクスポートしましょう。



① 検出器モデルを使用するためにデータの送信を開始します。詳細はこちら

AWS SDK を使用して、または IoT Analytics か IoT Core からデータを送信します。また、IoT Core から検出器モデル内の検出器テスト用のサンプルデータを送信することもできます。

サンプルデータを送信

IoT イベント > 探知器モデル > detecthumanmodel

detecthumanmodel

削除 エクスポート データを送信する ▼ 編集

詳細はこちら

- AWS CLI コマンドから送信
- AWS IoT Analytics から送信
- AWS IoT Core から送信
- サンプルデータを送信

一般設定

説明 発行日 キー

Sun Nov 17 2019

キー

評価方法 シリアル

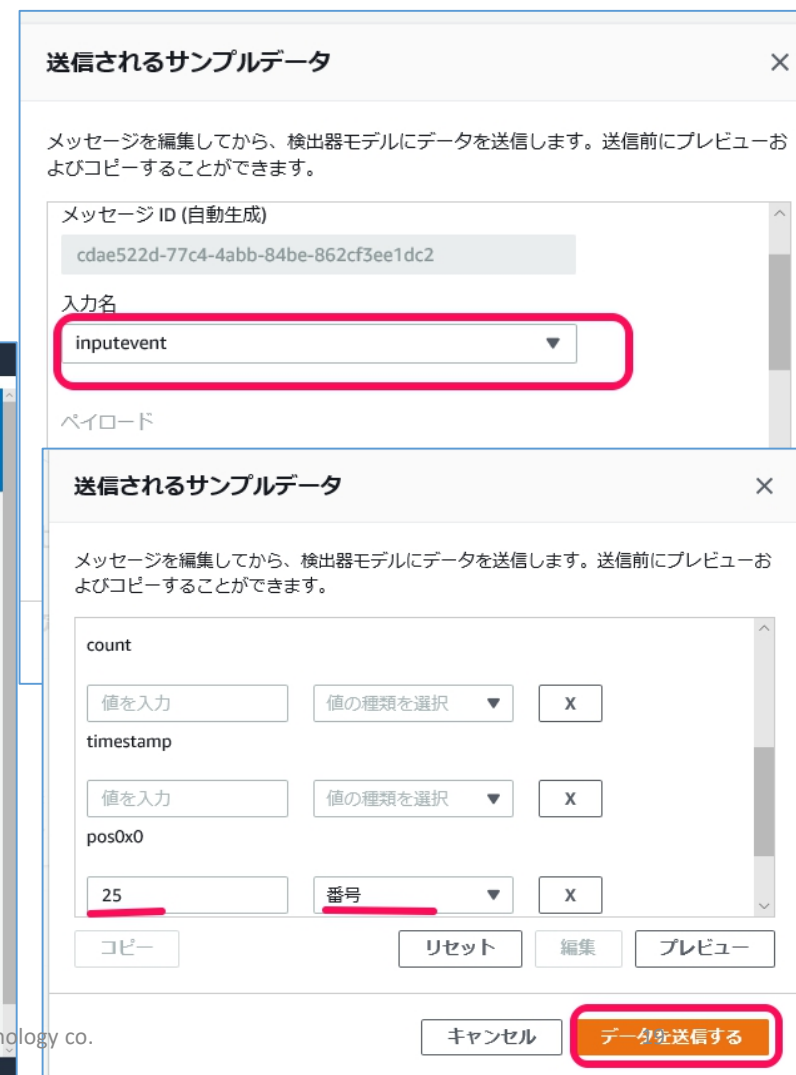
タグ

キー 値

タグがありません

探知器

キーの値	作成日	最終更新時刻	現在の状態
キーの探知機	Sun Nov 17 2019	Sun Nov 17 2019 10:54:52AM	detecthuman



送信されるサンプルデータ

メッセージを編集してから、検出器モデルにデータを送信します。送信前にプレビューおよびコピーすることができます。

メッセージ ID (自動生成)

cdae522d-77c4-4abb-84be-862cf3ee1dc2

入力名

inputevent

パイロード

送信されるサンプルデータ

メッセージを編集してから、検出器モデルにデータを送信します。送信前にプレビューおよびコピーすることができます。

count

値を入力 値の種類を選択 X

timestamp

値を入力 値の種類を選択 X

pos0x0

25 番号 X

コピー リセット 編集 プレビュー

キャンセル データを送信する

AWS IoT開発

④サーモセンサ情報収集、アラーム送付

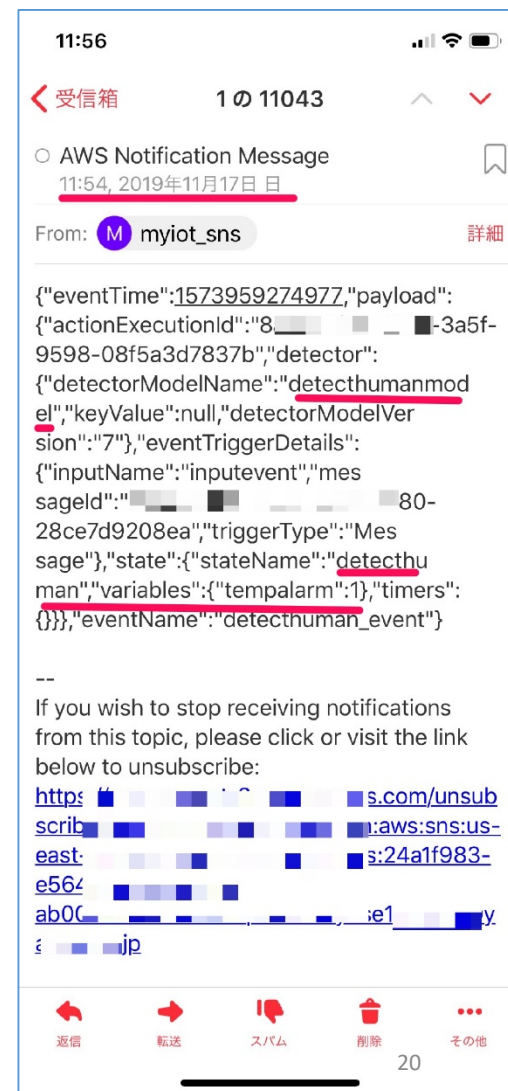
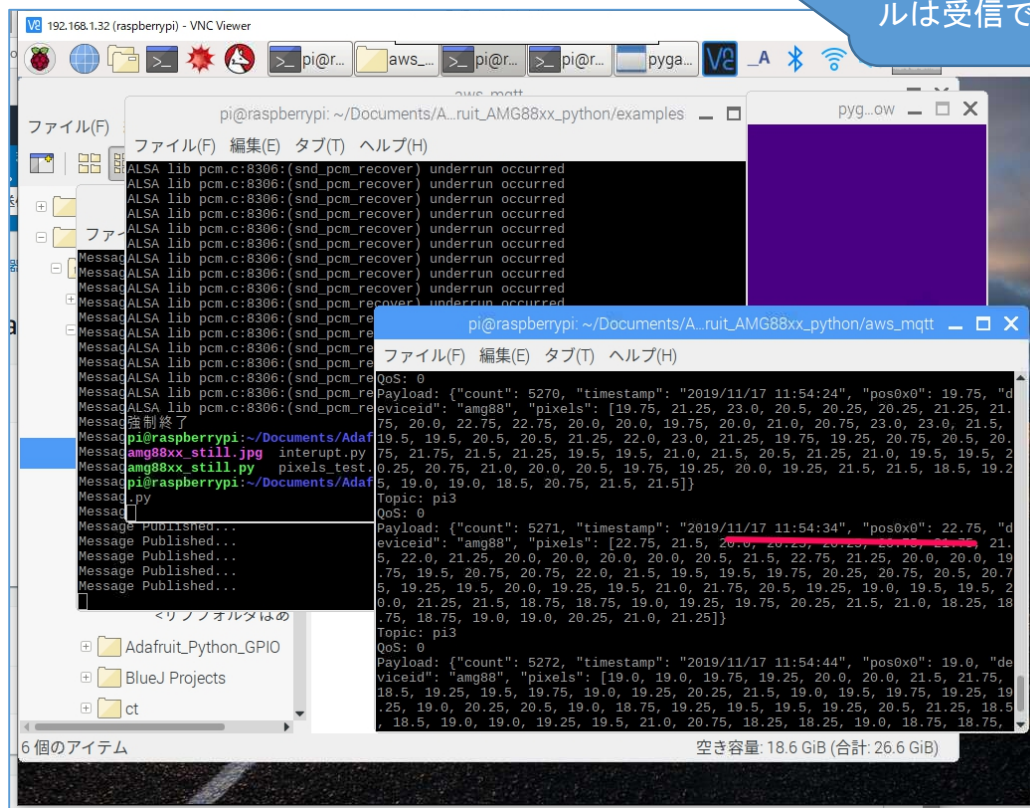


AWS IoT Core

IoTイベント試験

- Amg8833で手をかざして温度を20度以上にします。
- メールの受信を確認

イベントがNormalにもどっていないと、メールは受信できません。



AWS IoT開発

AWS IoT Core

④サーモセンサ情報収集、アラーム送付

```
$ cd  
/home/pi/Documents/Adafruit_AMG88xx_python/aws_mqtt  
$ python mqtt_subscribe_mon.py
```

遠隔リアルタイム・モニタ

- Mqttのsubscribe機能を使って、遠隔地でリアルタイムにサーモセンサの画像のモニタを行います。Mqttの送信タイミングの10秒毎受信
- python mqtt_subscribe_mon.py



AWS IoT開発

④サーモセンサ情報収集、アラーム送付

CSVの画像再生

- DynamoDBのCSVエクスポートしたoutput.csvを画像再生します
- `python thermal_csv.py`

```
$ cd  
/home/pi/Documents/Adafruit_AMG88xx_python/DynamoDB  
toCSV  
$ python thermal_csv.py
```



Opencv開発

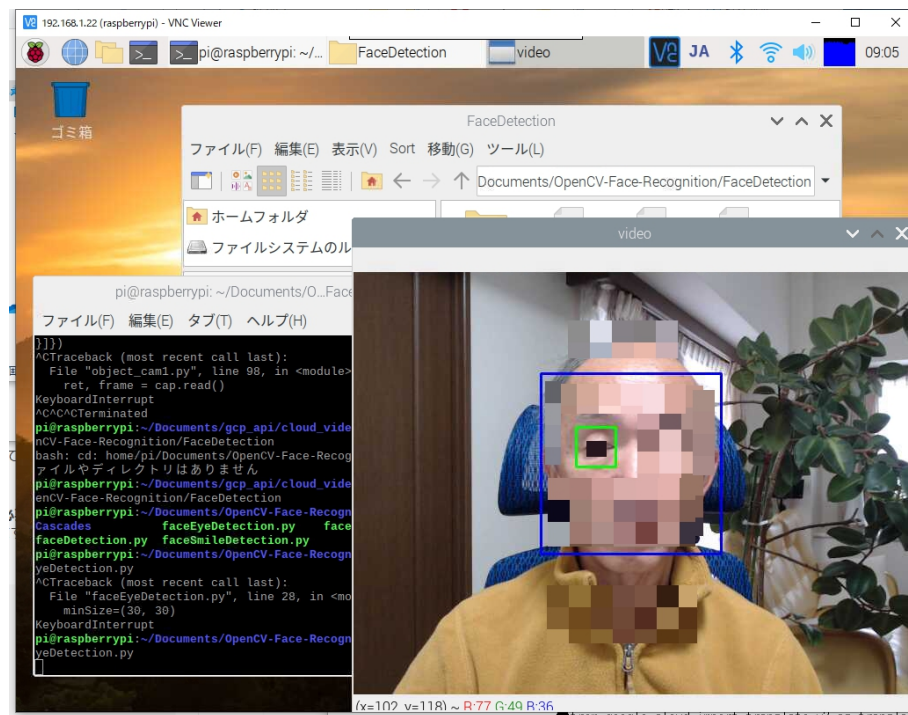
1. Opencv

①顔検出(カメラ)

コマンド

```
$ cd /home/pi/Documents/OpenCV-Face-Recognition/FaceDetection
```

- Opencvを使ったwebカメラでの顔検出です。
- <https://www.instructables.com/id/Real-time-Face-Recognition-an-End-to-end-Project/>
- `python3 faceDetection.py`
- `python3 faceEyedetection.py`



Opencv開発

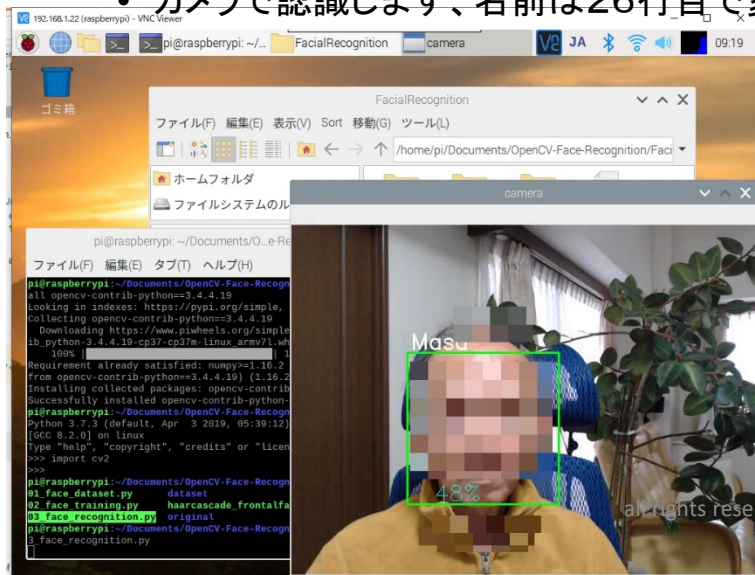
1. Opencv

②顔認識(カメラ)

コマンド

```
$ cd /home/pi/Documents/OpenCV-Face-Recognition/FacialRecognition
```

- Opencvを使った、webカメラでの登録した顔を認識します。
- <https://www.instructables.com/id/Real-time-Face-Recognition-an-End-to-end-Project/>
- 顔の登録
 - python3 01_face_dataset.py
 - Face id:0-3までを入力してください。
 - 自動で顔のスキャンが始まります。数分で終了します。
 - python3 02_face_training.py
 - python3 03_face_recognition.py
 - カメラで認識します、名前は26行目で変更可能



体温検知

1. リアルタイム検出

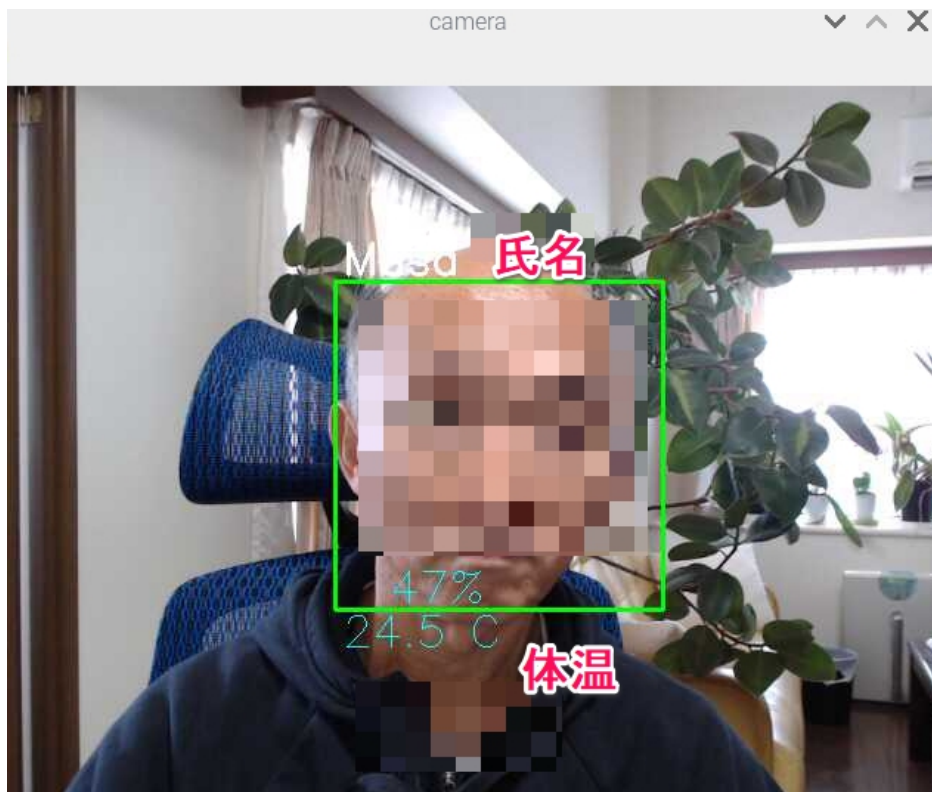
①顔、体温検出

コマンド

```
$ cd /home/pi/Documents/OpenCV-Face-Recognition/FacialRecognition
```

AWS IoT Core

- webカメラでの登録した顔、体温を検出し、AWSに送信します。
- python3 04_face_recognition_thermo.py
- 同じフォルダにtest.aviで動画保存。保存しない場合は、83行をコメントアウト。



体温は、サーモセンサの最高温度

体温検知

1. リアルタイム検出

② MQTT送信、モニタ

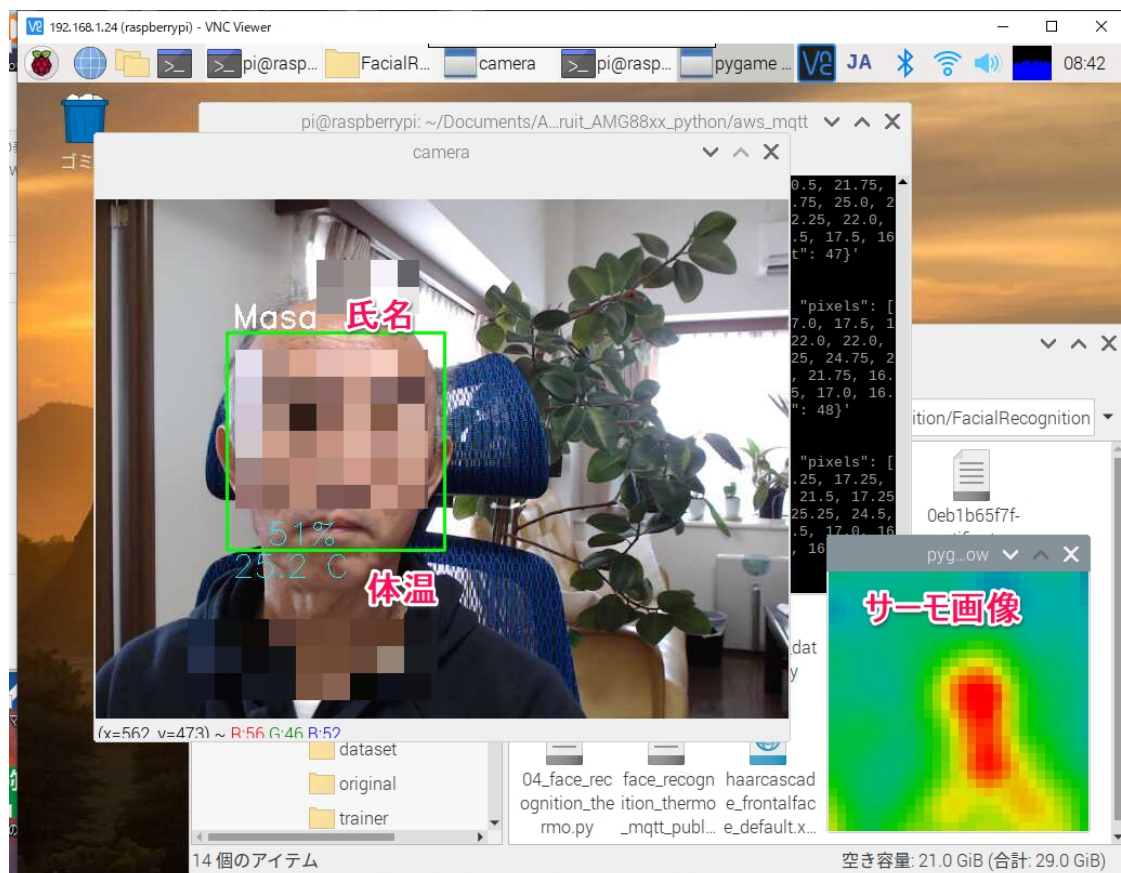
コマンド

```
$ cd /home/pi/Documents/OpenCV-Face-Recognition/FacialRecognition
```

AWS IoT Core

- webカメラでの登録した顔、体温を検出し、AWSに送信します。
- python3 face_recognition_thermo_mqtt_publish.py
 - 約10秒毎にデータをAWS IoTに送信します。

34ページを参考に証明書の情報などを記入し、フォルダに
関係の証明書をコピーしてください



```
$ cd
pi>Documents>Adafruit_AMG88xx_
python>aws_mqtt
$ python3 mqtt_subscribe_mon.py
受信側を先に立ち上げ
```


体温検知

2. データ活用

① CSV抽出、サーモ画像再生

- DynamoDBに蓄積したデータをCSVで抽出：[52ページ](#)参照
- サーモ画像再生：[54ページ](#)参照



AWS IoT Core