

Alibaba Cloud 物#网平台

ベストプラクティス

Document Version20191220

目次

1	デバイスアクセス.....	1
1.1	MQTT.fx を使用して IoT Platform に接続.....	1
1.2	Android Things を Alibaba Cloud IoT Platform に接続.....	9

1 デバイスアクセス

1.1 MQTT.fx を使用して IoT Platform に接続

ここでは、MQTT.fx を例に取り上げ、サードパーティの MQTT クライアントを使用して IoT Platform に接続する方法を説明します。MQTT.fx は Java 言語で記述した Eclipse Paho ベースの MQTT クライアントです。サブスクライブ方式のメッセージングおよびパブリッシュ方式のトピックを介したメッセージングをサポートします。

前提条件

[IoT Platform コンソール](#)でプロダクトとデバイスを作成し、デバイスの **ProductKey**、**DeviceName**、および **DeviceSecret** を入手しました。MQTT.fx の接続パラメーター設定時は、**ProductKey**、**DeviceName**、および **DeviceSecret** の値を使用します。プロダクトおよびデバイスを作成する際のヘルプは、「[#unique_3](#)」、「[#unique_4](#)」、および「[#unique_5](#)」をご参照ください。

手順

1. MQTT.fx ソフトウェアをダウンロードしてインストールします。

Windows 版 MQTT.fx のダウンロード：<http://mqtt.fx.software.informer.com/download/>

Mac 版 MQTT.fx のダウンロード：<http://macdownload.informer.com/mqtt-fx/>

2. MQTT.fx を起動し、[設定] アイコンをクリックします。



3. 接続パラメーターを設定します。

現在、TCP および TLS の 2 種類の接続モードがサポートされています。この 2 つのモードの違いは、クライアント ID と SSL/TLS の設定だけです。

手順は以下のとおりです。

- a. 基本情報を入力します。パラメーターの説明は、以下の表をご参照ください。

[全般] のデフォルトのパラメーターを保持、または必要に応じて値を設定することもできます。

パラメーター	説明
Profile Name	カスタマイズプロファイル名を入力します。
Profile Type	[MQTT ブローカー] を選択します。
Broker Address	$\${YourProductKey}.iot-as-mqtt.$ $\${region}.aliyuncs.com$ の書式で接続ドメインを入力します。この書式で、可変の $\${region}$ は、IoT Platform のサービスリージョンのリージョン ID を示しています。リージョン ID は、「リージョンおよびゾーン」をご参照ください。コーディング例: <code>alPUPCxxxx.iot-as-mqtt.cn-shanghai.aliyuncs.com</code>

パラメーター	説明
Broker Port	1883 に設定します。

パラメーター	説明
Client ID	<p><code>\${clientId} securemode=3,signmethod=hmacsha1 </code> の書式で値を入力します。コーディング例：<code>12345 securemode=3,signmethod=hmacsha1 </code> パラメーターの記述は以下のとおりです。</p> <ul style="list-style-type: none"> ・ <code>\${clientId}</code> は、カスタマイズクライアント ID です。64 文字以内の任意の値にすることができます。クライアント ID の値はお使いのデバイスの MAC アドレスまたは SN コードを使用するよう推奨します。 ・ <code>securemode</code> は接続のセキュリティモードです。TCP モードを使用する場合は、securemode=3 に設定します。TLS モードを使用する場合は、securemode=2 に設定します。 ・ <code>signmethod</code> は、使用する署名メソッドです。IoT Platform では <code>hmacmd5</code> および <code>hmacsha1</code> をサポートしています。 <p> 注： クライアント ID 情報を入力した後は、[全般] をクリックしないようにしましょう。</p>

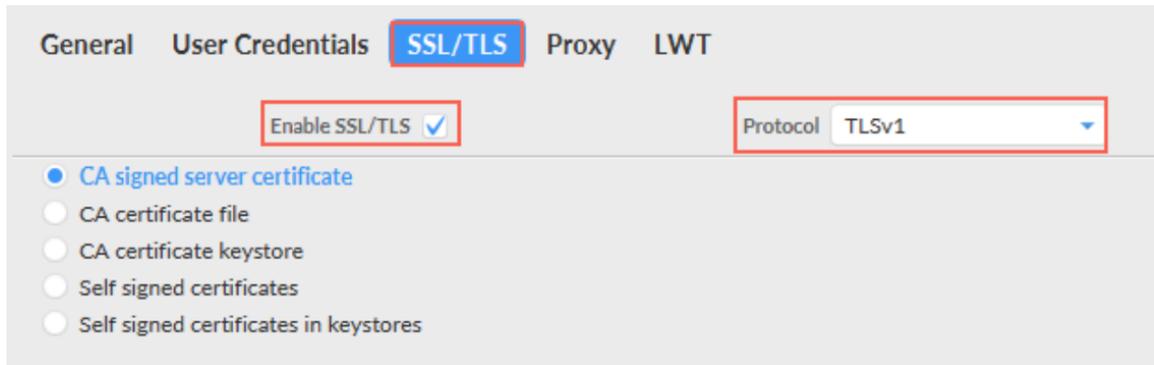
b. [ユーザー資格情報] をクリックして、[ユーザー名] と [パスワード] を入力します。

パラメーター	説明
User Name	デバイス名とプロダクトキーを "&" 文字で直接つないだものを指定する必要があります。書式： <code>\${YourDeviceName}&\${YourPrductKey}</code> 。例： device&fOAt5H5TOWF 。

パラメーター	説明
Password	<p>入力パラメーターの暗号化値を入力する必要があります。 IoT Platform ではパスワードを簡単に生成できる パスワードジェネレータ を提供しています。ご自身で暗号化することもできます。</p> <ul style="list-style-type: none"> パスワードジェネレータのパラメーター： <ul style="list-style-type: none"> productKey : デバイスが属するプロダクトの一意的識別子。この情報はコンソールのデバイス詳細ページで閲覧できます。 deviceName : デバイスの名前 この情報はコンソールのデバイス詳細ページで閲覧できます。 deviceSecret : デバイスシークレット この情報はコンソールのデバイス詳細ページで閲覧できます。 timestamp : (省略可能) 現在のシステム時間のタイムスタンプ clientId : カスタマイズクライアント ID。クライアント ID の <code>clientId</code> と同じ値にする必要があります。 method : 署名アルゴリズム。クライアント ID の <code>signmethod</code> と同じ値にする必要があります。 パスワードを手動で生成します。 <ol style="list-style-type: none"> パラメーターをソートして結合します。 <p>clientId、deviceName、productKey、および timestamp の各パラメーターを辞書式順序でソートし結合します。(タイムスタンプを設定していない場合は、文字列にタイムスタンプを記述しないようにします)。結合した文字列の例：<code>clientId12345deviceNamedeviceproductKeyf0At5H5T0WF</code></p> 暗号化します。 <p>Client ID で定義されている署名アルゴリズムにより結合文字列を暗号化する秘密鍵として、お使いのデバイスの <code>deviceSecret</code> を使用します。</p> <p>デバイスの <code>deviceSecret</code> は <code>abc123</code>、暗号化形式は <code>hmacsha1(abc123,clientId12345deviceNamedeviceproductKeyf0At5H5T0WF)</code> とします。</p>

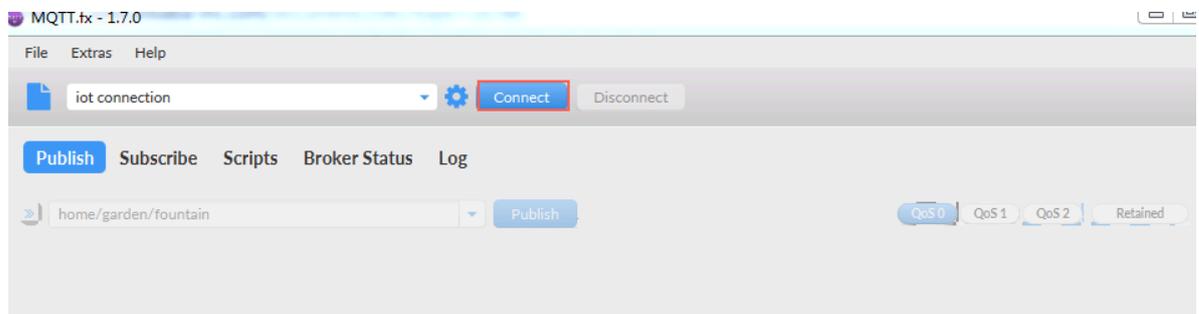
- c. **TLS** 接続モードを使用している場合は、**SSL/TLS** の情報を設定することが必須となります。接続モードが **TCP** の場合、**SSL/TLS** の設定は必須ではありません。

[SSL/TLS を有効にする] チェックボックスをオンにし、プロトコルとして **TLSv1** を選択します。



d. 必要な情報をすべて入力して、[OK] をクリックします。

4. [接続] をクリックして **IoT Platform** に接続します。

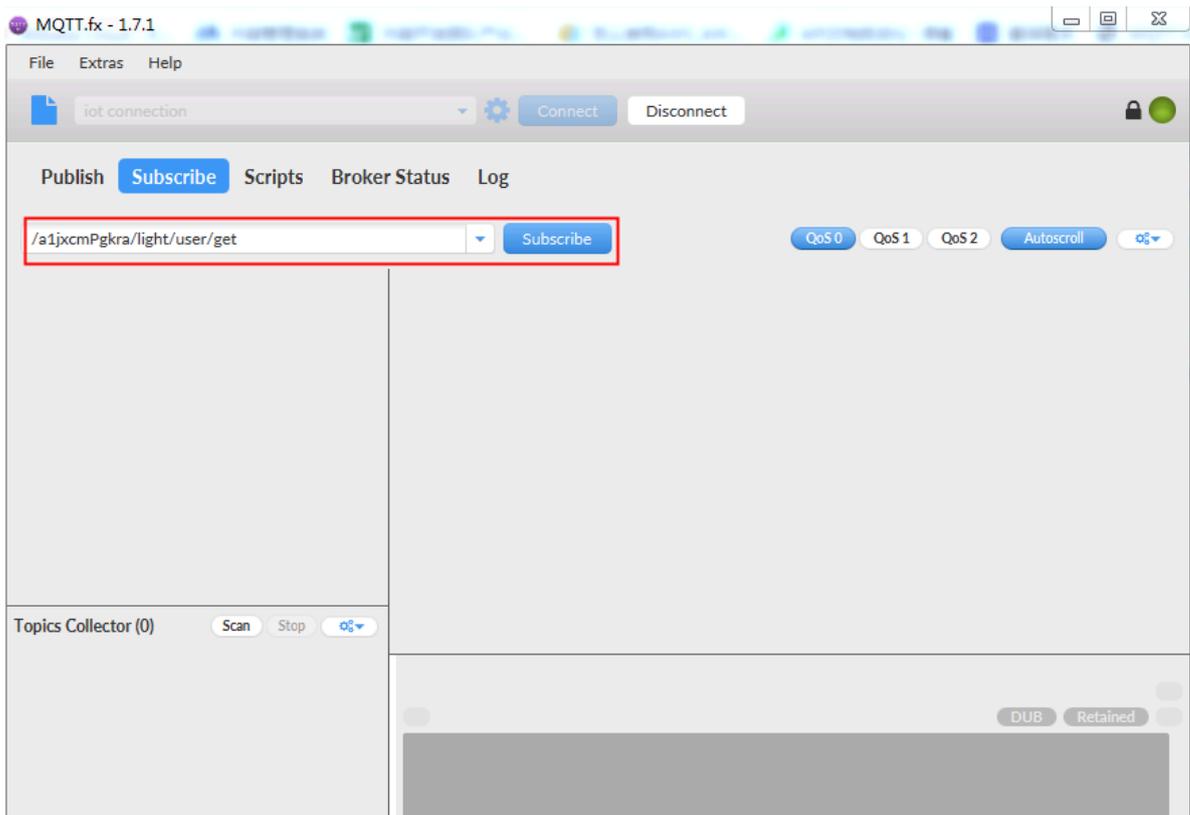


メッセージ通信テスト

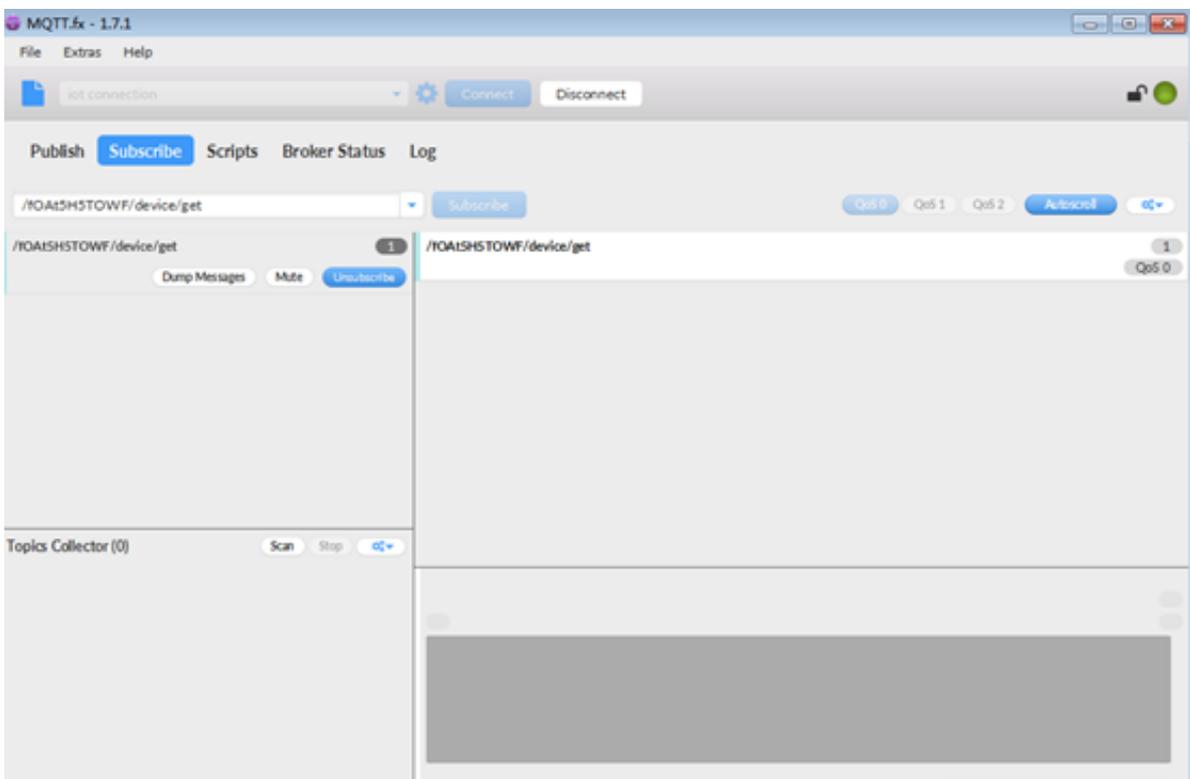
MQTT.fx と **IoT Platform** が正常に接続されているかどうかテストします。

1. **MQTT.fx** で、[サブスクライブ] をクリックします。

2. デバイスのトピックを入力し、[サブスクライブ] をクリックします。

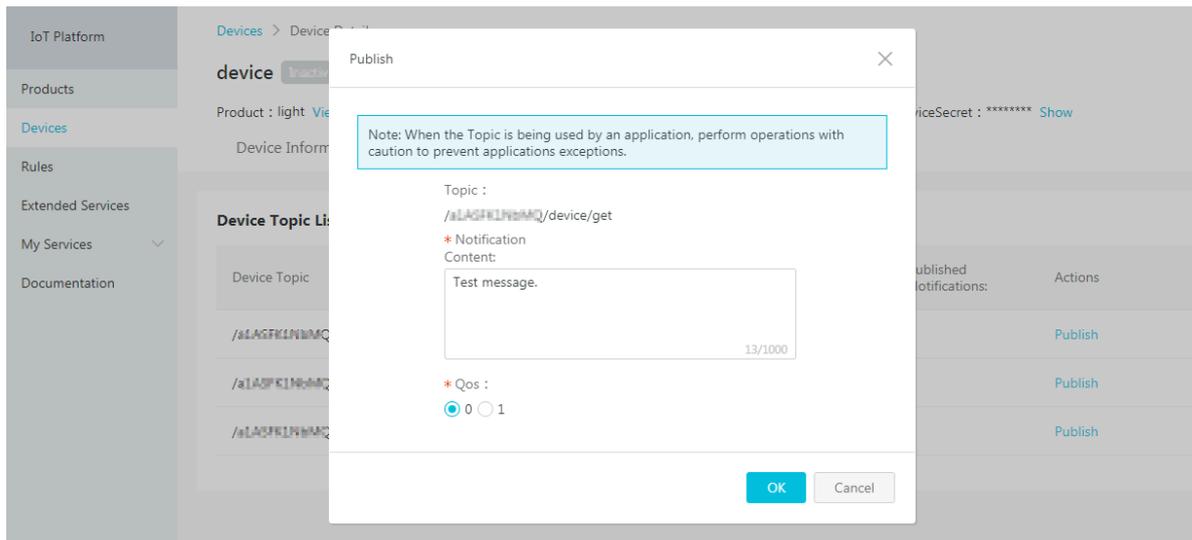


トピックに正常にサブスクライブすると、トピックはトピック一覧に表示されます。

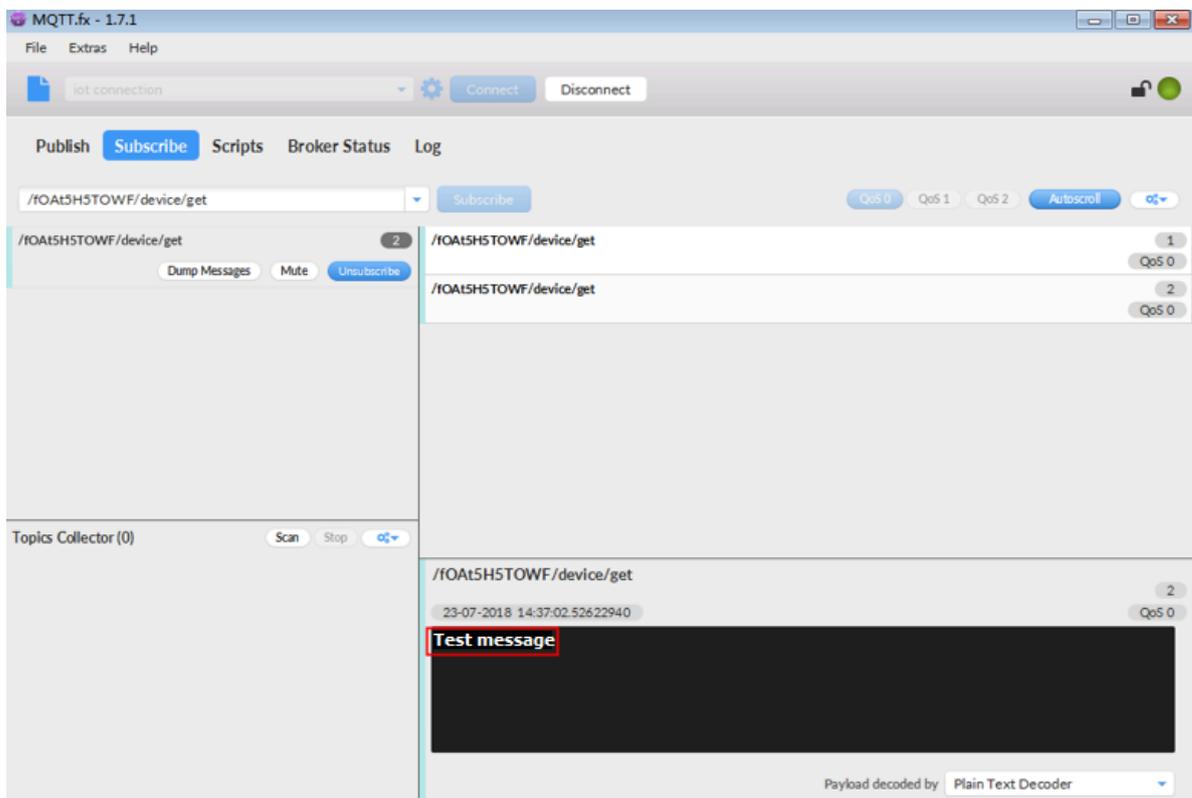


3. *IoT Platform* コンソールの [デバイス詳細] ページの [トピック一覧] で、サブスクライブしたトピックの [パブリッシュ] ボタンをクリックします。

4. メッセージの内容を入力し、[OK] をクリックします。

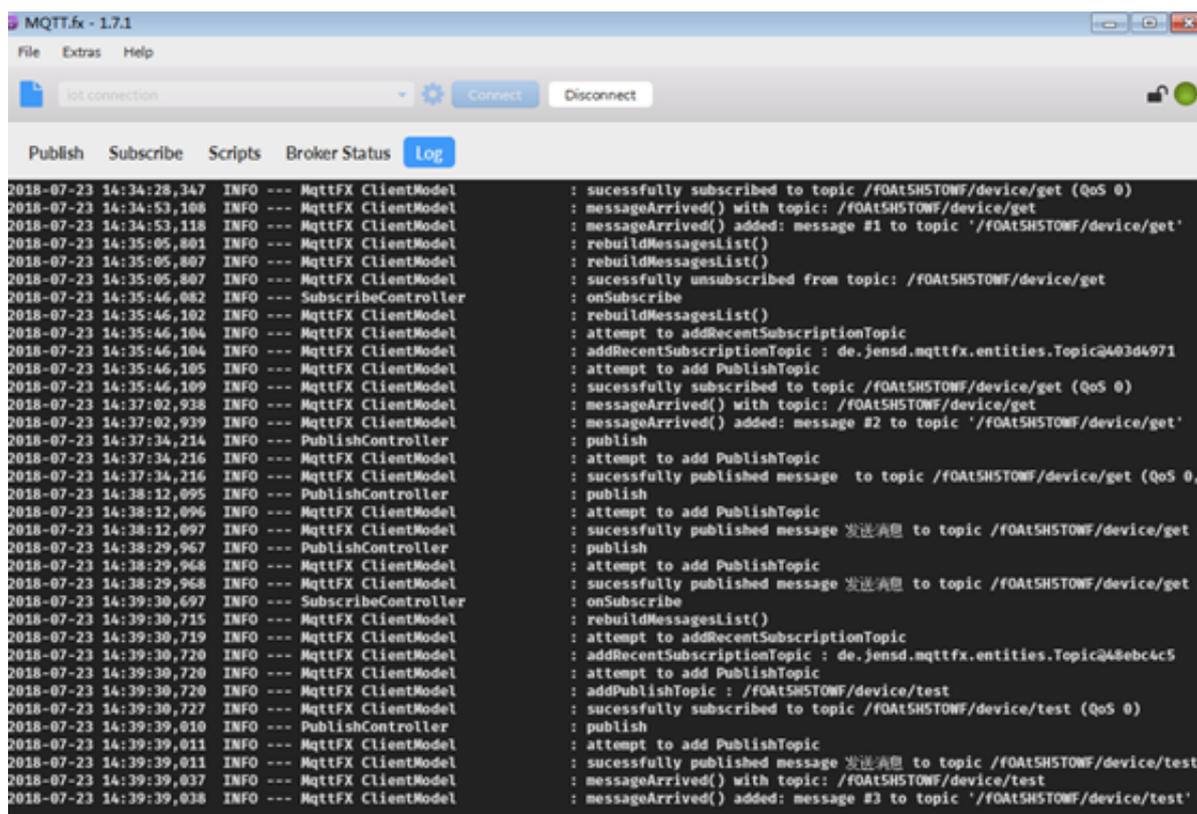


5. MQTT.fx に戻り、メッセージが受信されたかどうか確認します。



ログの表示

MQTT.fx で操作ログとエラーログを閲覧するには、[ログ] をクリックします。



```

MQTT.fx - 1.7.1
File Extras Help
iot.connection [Connect] [Disconnect]
Publish Subscribe Scripts Broker Status Log
2018-07-23 14:34:28,347 INFO --- MqttFX ClientModel : successfully subscribed to topic /FOAT5H5TOWF/device/get (QoS 0)
2018-07-23 14:34:53,108 INFO --- MqttFX ClientModel : messageArrived() with topic: /FOAT5H5TOWF/device/get
2018-07-23 14:34:53,118 INFO --- MqttFX ClientModel : messageArrived() added: message #1 to topic '/FOAT5H5TOWF/device/get'
2018-07-23 14:35:05,801 INFO --- MqttFX ClientModel : rebuildMessagesList()
2018-07-23 14:35:05,807 INFO --- MqttFX ClientModel : rebuildMessagesList()
2018-07-23 14:35:05,807 INFO --- MqttFX ClientModel : successfully unsubscribed from topic: /FOAT5H5TOWF/device/get
2018-07-23 14:35:46,082 INFO --- SubscribeController : onSubscribe
2018-07-23 14:35:46,102 INFO --- MqttFX ClientModel : rebuildMessagesList()
2018-07-23 14:35:46,104 INFO --- MqttFX ClientModel : attempt to addRecentSubscriptionTopic
2018-07-23 14:35:46,104 INFO --- MqttFX ClientModel : addRecentSubscriptionTopic : de.jensd.mqttfx.entities.Topic@403d4971
2018-07-23 14:35:46,105 INFO --- MqttFX ClientModel : attempt to add PublishTopic
2018-07-23 14:35:46,109 INFO --- MqttFX ClientModel : successfully subscribed to topic /FOAT5H5TOWF/device/get (QoS 0)
2018-07-23 14:37:02,938 INFO --- MqttFX ClientModel : messageArrived() with topic: /FOAT5H5TOWF/device/get
2018-07-23 14:37:02,939 INFO --- MqttFX ClientModel : messageArrived() added: message #2 to topic '/FOAT5H5TOWF/device/get'
2018-07-23 14:37:34,214 INFO --- PublishController : publish
2018-07-23 14:37:34,216 INFO --- MqttFX ClientModel : attempt to add PublishTopic
2018-07-23 14:37:34,216 INFO --- MqttFX ClientModel : successfully published message to topic /FOAT5H5TOWF/device/get (QoS 0)
2018-07-23 14:38:12,095 INFO --- PublishController : publish
2018-07-23 14:38:12,096 INFO --- MqttFX ClientModel : attempt to add PublishTopic
2018-07-23 14:38:12,097 INFO --- MqttFX ClientModel : successfully published message 发送消息 to topic /FOAT5H5TOWF/device/get
2018-07-23 14:38:29,967 INFO --- PublishController : publish
2018-07-23 14:38:29,968 INFO --- MqttFX ClientModel : attempt to add PublishTopic
2018-07-23 14:38:29,968 INFO --- MqttFX ClientModel : successfully published message 发送消息 to topic /FOAT5H5TOWF/device/get
2018-07-23 14:39:30,697 INFO --- SubscribeController : onSubscribe
2018-07-23 14:39:30,715 INFO --- MqttFX ClientModel : rebuildMessagesList()
2018-07-23 14:39:30,719 INFO --- MqttFX ClientModel : attempt to addRecentSubscriptionTopic
2018-07-23 14:39:30,720 INFO --- MqttFX ClientModel : addRecentSubscriptionTopic : de.jensd.mqttfx.entities.Topic@48ebc4c5
2018-07-23 14:39:30,720 INFO --- MqttFX ClientModel : attempt to add PublishTopic
2018-07-23 14:39:30,720 INFO --- MqttFX ClientModel : addPublishTopic : /FOAT5H5TOWF/device/test
2018-07-23 14:39:30,727 INFO --- MqttFX ClientModel : successfully subscribed to topic /FOAT5H5TOWF/device/test (QoS 0)
2018-07-23 14:39:39,010 INFO --- PublishController : publish
2018-07-23 14:39:39,011 INFO --- MqttFX ClientModel : attempt to add PublishTopic
2018-07-23 14:39:39,011 INFO --- MqttFX ClientModel : successfully published message 发送消息 to topic /FOAT5H5TOWF/device/test
2018-07-23 14:39:39,037 INFO --- MqttFX ClientModel : messageArrived() with topic: /FOAT5H5TOWF/device/test
2018-07-23 14:39:39,038 INFO --- MqttFX ClientModel : messageArrived() added: message #3 to topic '/FOAT5H5TOWF/device/test'

```

1.2 Android Things を Alibaba Cloud IoT Platform に接続

この記事では、室内空気試験プロジェクトを例として **Google Android Things** を **Alibaba Cloud IoT Platform** に接続する方法を説明します。

ハードウェア

- ・ 本プロジェクトのハードウェア一覧

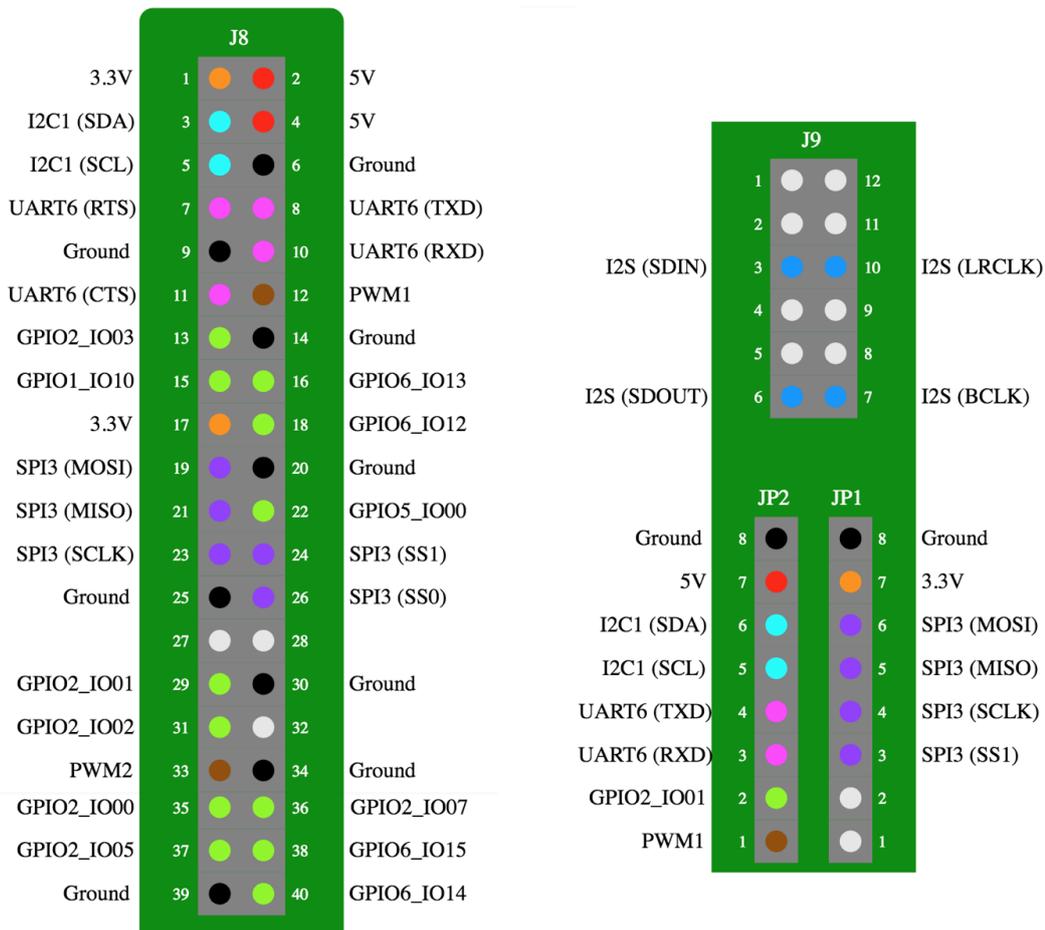
以下の表に、室内空気試験プロジェクトに必要なハードウェアを示します。

ハードウェア	画像	備考
NXP Pico i.MX7D 開発ボード		Android Things 1.0 注： Raspberry Pi も使 用できます。

ハードウェア	画像	備考
<p>DHT12 温湿度センサー</p>	 A blue PCB-based sensor module. It features a blue plastic multi-pin connector on the left side. On the right side, there are three silver pins. The PCB has a circular cutout in the center and some text including 'VCC', 'GND', and '+'. The sensor itself is a small black component mounted on the board.	<p>I2C データ通信方式をサポートします。</p>
<p>ZE08-CH20 ホルムアルデヒド検出センサー</p>	 A green PCB-based sensor module. It features a square grey sensor component in the center. On the bottom right, there is a white multi-pin connector with pins numbered 1 through 5. The PCB has four mounting holes at the corners.	<p>UART データ通信方式をサポートします。</p>

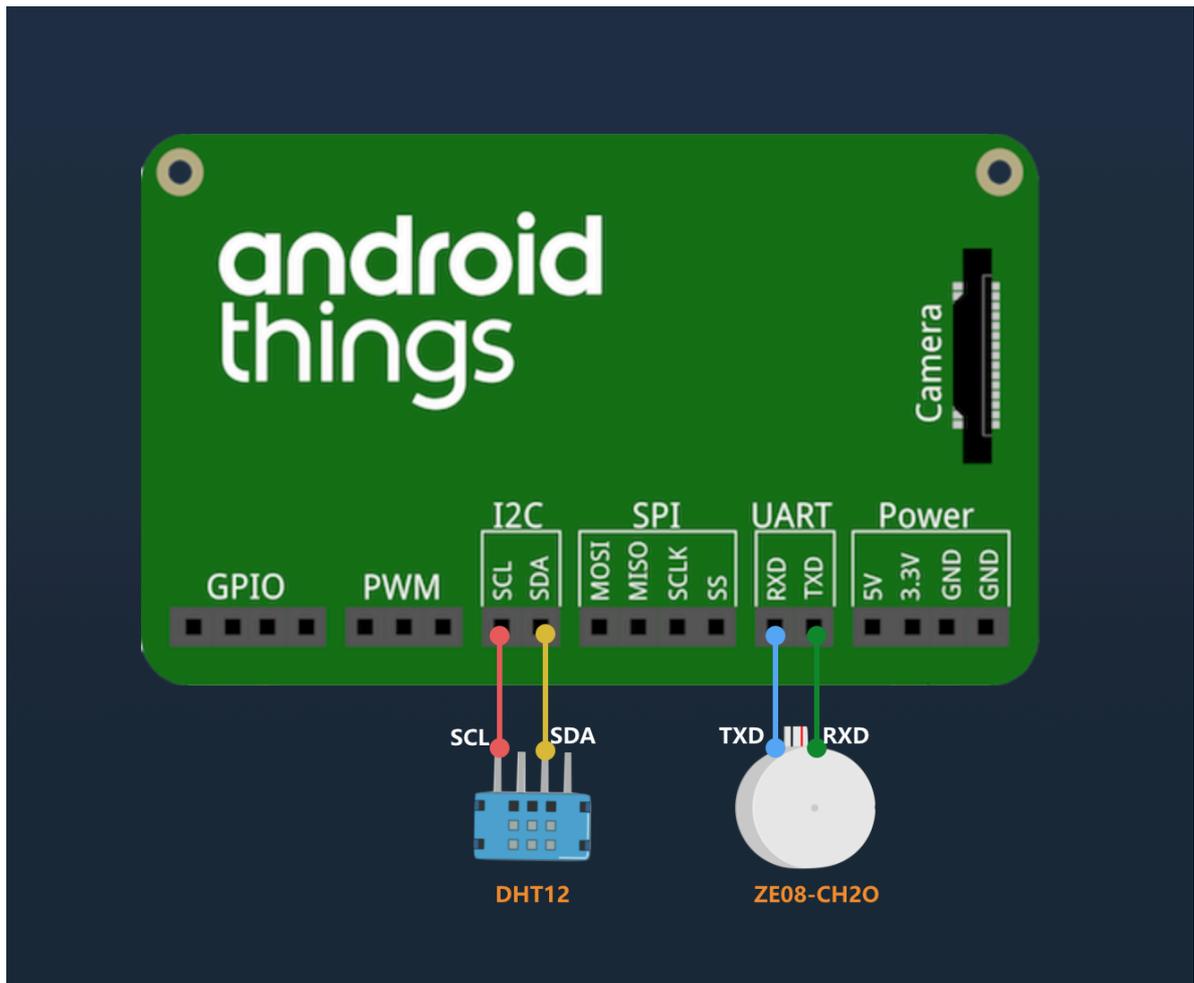
・ NXP i.MX7D 開発ボードピンのレイアウト図

- = 5V
- = 1.8V
- = GPIO
- = I2C
- = SPI
- = 3.3V
- = Ground
- = PWM
- = I2S
- = UART



NXP Pico i.MX7D の詳細は、[『https://developer.android.com/things/hardware/imx7d-pico-io』](https://developer.android.com/things/hardware/imx7d-pico-io) をご参照ください。 .

- ・ ハードウェア接続図



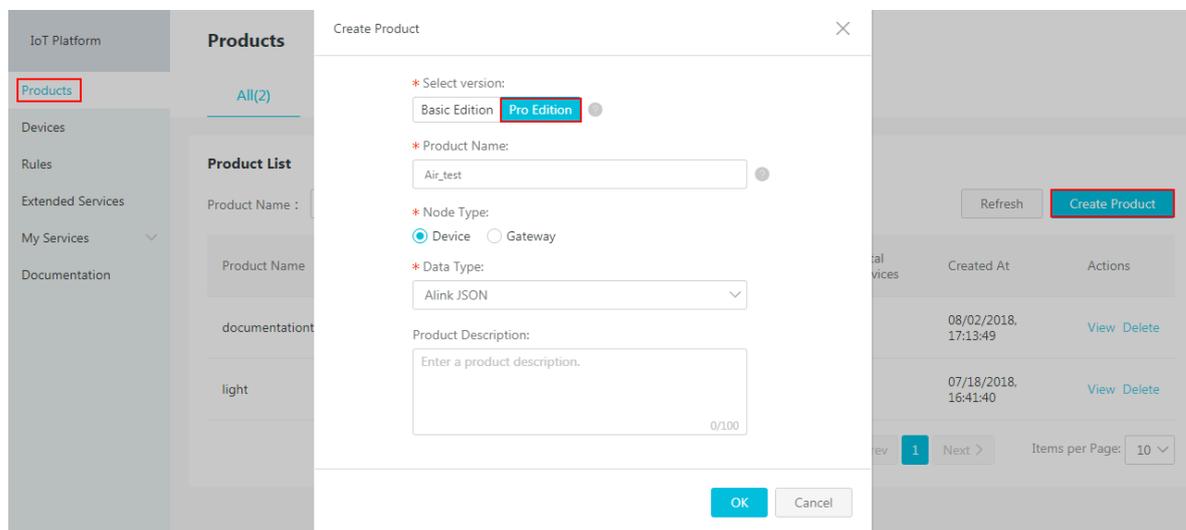
- 温湿度センサー (DHT12) の SCL (クロックライン) ピンと SDA (データライン) ピンを開発ボードの I2C SCL ピンと SDA ピンに接続します。
- ホルムアルデヒド検出センサー (ZE08-CH20) の TXD (データ伝送) ピンを開発ボードの RXD (データ受信) ピンに接続し、ZE08-CH20の RXD ピンを開発ボードの TXD ピンに接続します。

Alibaba Cloud IoT Platform コンソールでプロダクトとデバイスを作成

1. [IoT Platform コンソール](#)にログインします。

2. IoT Platform Pro でプロダクトを作成します。

[プロダクト] ページで [プロダクトの作成] をクリックします。プロダクトを作成するバージョンとして **Pro Edition** を選択します。詳細は、[#unique_3](#)をご参照ください。



3. 新しく作成したプロダクトの機能を定義します。

[プロダクト詳細] ページで [フィーチャーの定義] > [追加] をクリックし、プロダクトのプロパティを定義します。詳細は、[#unique_7](#)をご参照ください。

表 1-1: 室内空気試験プロジェクトに必要なプロパティ

プロパティ名	識別子	データ型	値の範囲	説明
Temperature	temperature	float	-50~100	DHT12 によって検出されます。
Humidity	humidity	float	0~100	DHT12 によって検出されます。

プロパティ名	識別子	データ型	値の範囲	説明
Formaldehyde concentration	ch2o	double	0~3	ZE08 により検出されます。

IoT Platform > Products > Product Details

Air_test [Pro Edition](#)

ProductKey : XXXXXXXXXX [Copy](#) ProductSecret : ***** [Show](#) Total Devices:0 [Manage](#)

Product Information Notifications **Define Feature** Device Log Online Debugging

Define Feature A standard feature is automatically created based on the device type of the product. You can also add optional features or create your own custom features. [View TSL](#) [Add](#)

Feature Type	Feature Name:	Identifier:	Data Type	Data Definition	Actions
Properties	Temperature	temperature	float	Value Range:-50 - 100	Edit Delete
Properties	Humidity	humidity	float	Value Range:0 - 100	Edit Delete
Properties	Formaldehyde concentration	ch2o	double	Value Range:0 - 3	Edit Delete

4. デバイスを作成

[デバイス] ページで新しく作成したプロダクトの名前を選択し、[デバイスの追加] をクリックしてデバイスを作成します。詳細は、[#unique_4](#)をご参照ください。

IoT Platform > Devices

Air_test(Pro Edit) Total Devices: 0 [Activate Device](#) [Online](#) [Refresh](#)

Device List

Device Name:

DeviceName

[Add Device](#)

Last Online Actions

1 Next > Items per Page: 10

Batch Delete Batch Disable Batch Enable

Android Things クライアントの開発

1. **Android Studio** を使用して **Android Things** プロジェクトを作成し、インターネットの権限を追加します。

```
<uses-permission android:name="android.permission.INTERNET" />
```

2. **eclipse.paho.mqtt** を **Gradle** ファイルに追加します。

```
implementation 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.2.0'
```

3. **DHT12** のデータを **I2C** で読み込む設定をします。

```
private void readDataFromI2C() {
    try {
        byte[] data = new byte[5];
        i2cDevice.readRegBuffer(0x00, data, data.length);

        // check data
        if ((data[0] + data[1] + data[2] + data[3]) % 256 !=
data[4]) {
            humidity = temperature = 0;
            return;
        }
        // humidity data
        humidity = Double.valueOf(String.valueOf(data[0]) + "."
+ String.valueOf(data[1]));
        Log.d(TAG, "humidity: " + humidity);
        // temperature data
        if (data[3] < 128) {
            temperature = Double.valueOf(String.valueOf(data[2])
+ "." + String.valueOf(data[3]));
        } else {
            temperature = Double.valueOf("-" + String.valueOf(
data[2]) + "." + String.valueOf(data[3] - 128));
        }

        Log.d(TAG, "temperature: " + temperature);
    } catch (IOException e) {
        Log.e(TAG, "readDataFromI2C error " + e.getMessage(), e
);
    }
}
```

4. **Ze08-CH20** のデータを **UART** で読み込む設定をします。

```
try {
    // data buffer
    byte[] buffer = new byte[9];

    while (uartDevice.read(buffer, buffer.length) > 0) {
        if (checkSum(buffer)) {
            ppbCh2o = buffer[4] * 256 + buffer[5];
            ch2o = ppbCh2o / 66.64 * 0.08;
        }
    }
}
```

```

        } else {
            ch2o = ppbCh2o = 0;
        }
        Log.d(TAG, "ch2o: " + ch2o);
    }

    } catch (IOException e) {
        Log.e(TAG, "Ze08CH20 read data error " + e.
        getMessage(), e);
    }
}

```

5. Alibaba Cloud IoT Platform とクライアントを接続し、データを報告します。

```

/*
Payload format
{
  "id": 123243,
  "params": {
    "temperature": 25.6,
    "humidity": 60.3,
    "ch2o": 0.048
  },
  "method": "thing.event.property.post"
}
*/
MqttMessage message = new MqttMessage(payload.getBytes("utf-8"));
message.setQos(1);

String pubTopYourPc = "/sys/${YourProductKey}/${YourDeviceName}/
thing/event/property/post";

mqttClient.publish(pubTopic, message);

```

デバイスのリアルタイムデータを表示

デバイスを有効にすると、**IoT Platform** コンソールの [デバイスの詳細] ページの [ステータス] カラムからデバイスのリアルタイムデータを閲覧できます。

The screenshot shows the 'Status' tab of the IoT Platform console for a device named 'Temperature-sensor'. The status card displays the following data:

Formaldehyde concentration	Temperature	Humidity
0.03mg/m ³	10C°	27%
Last update: 2018/07/19 15:50:16	Last update: 2018/07/19 15:50:16	Last update: 2018/07/19 15:50:16
View logs	View logs	View logs