

記事

[Toshihiko Minamoto](#) · 2020年12月24日 6m read

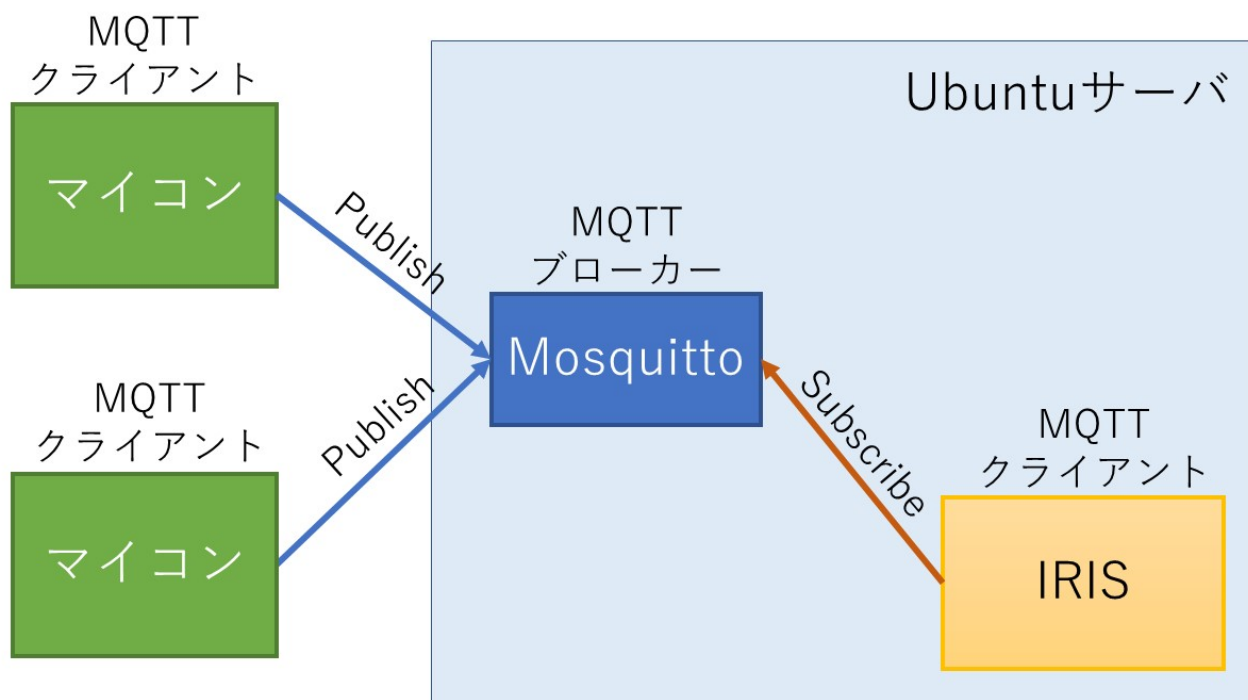
MQTTを使ったIRISとデバイスとの通信

皆さんこんにちは。

IRIS 2020.1からMQTTアダプタが新たに追加されました。MQTTはPublish/Subscribe型のシンプルで軽量なメッセージングプロトコルです。帯域が低いネットワーク環境やArduinoやRaspberry PIといったワンボードマイコンやシングルボードコンピュータなどでも動作できます。今回はクラウド上のUbuntuサーバにIRISをインストールし、MQTTアダプタを使いEsp8266というマイコンで取得した温度、湿度、気圧データをIRIS上のデータベースに登録する方法について説明したいと思います。

アーキテクチャ

今回のアーキテクチャは以下の通りです。



データを送受信するマイコンやアプリケーションにMQTTクライアントを組み込み、MQTTクライアントがMQTTブローカーというメッセージを転送するソフトウェアに接続します。

同じMQTTブローカーにつながっているMQTTクライアントにてメッセージを送信したい場合はPublishメッセージを使用し、メッセージと共にトピックと呼ばれるメッセージの種類を表す文字列を送信します。情報を受信したい場合はSubscribeメソッドを使用し、受信したいトピックを指定します。

IRISのインストール

今回、Ubuntu版のIRIS 2020.1 community editionをインストールしています。お持ちでない方は[こちら](#)からダウンロードをお願いします。インストール方法につきましては[こちら](#)の記事の動画(11分24秒あたりから)をご参照ください。

Mosquittoのインストール

MQTTブローカーとしてMosquittoをインストールします。これには、Ubuntuのターミナルよりaptというパッケージマネージャを使用します。

```
$ sudo apt-get update  
  
$ sudo apt-get install mosquitto
```

これで、Mosquittoのインストールができます。

Mosquittoの設定

この状態でもMosquittoに接続することは可能ですが、誰でも接続できてしまいますので、パスワードを設定することにします。まずは -c オプションを指定し、パスワードファイル(/etc/mosquitto/password)を作成します。

```
sudo mosquitto_passwd -c /etc/mosquitto/password <????> <??????>
```

既にあるパスワードファイルに追加する場合は

```
sudo mosquitto_passwd /etc/mosquitto/password <????> <??????>
```

となります。

今回は、マイコン側からアクセスするユーザ「mqttclient」とIRISがアクセスするユーザ「iris」を作成しました。

mosquitto.confに以下の行を追加し、Anonymousユーザのアクセス禁止とパスワードファイルを設定します。

```
    :  
    allow_anonymous false  
    password_file /etc/mosquitto/password  
    :
```

証明書の作成

MQTTクライアントとMQTTブローカの間は暗号化されていないため、認証時のユーザ名、パスワードが盗聴されてしまいます。

そこでサーバ証明書を発行し、通信を暗号化させることにします。

ここでは、自己認証局ならびにサーバの証明書を作成します。作成にはIRISのPKI機能を利用します。

認証局の秘密鍵の作成

Ubuntuターミナルから認証局の秘密鍵を格納するディレクトリを作成します。

```
$ sudo mkdir -p /usr/irissys/rootCA/private
```

このとき管理ポータルからアクセスできるよう、ディレクトリのオーナーをirisusrに変更します。IRISインストール時に実行ユーザを変更されている場合はその内容に合わせてオーナーを変更してください。

```
$ sudo chown irisusr:irisusr /usr/irissys/rootCA/private
```

管理ポータルを起動し、「システム管理」「セキュリティ」「公開鍵基盤」をクリックします。画面左下の「ローカルの認証局サーバを構成」をクリックしますと、以下のように右側に「ローカルの認証局サーバを構成」と書かれた部分が現れます。ここで認証局のファイル名にファイルの名称、秘密鍵ファイルのディレクトリを入力し「次へ」をクリックします。

InterSystems™
IRIS Data Platform

管理ポータル

開発システム ホーム

サーバ **XXXXXXXXXXXX** ネームスペース %SYS ユーザ _SYSTEM ライセンス先 InterSystems IRIS Community イン

システム > セキュリティ管理 > 公開鍵基盤 - (セキュリティの設定)*

☐ 認証局クライアント

- ▶ 証明書署名リクエストを認証局サーバに送信
- ▶ 認証局サーバから証明書を取得
- ▶ ローカルの認証局クライアントを構成

☐ 認証局サーバ

- ▶ 保留中の証明書署名リクエストを処理する
- ▶ **ローカルの認証局サーバを構成**

ローカルの認証局サーバを構成

認証局の証明書と秘密鍵ファイルのファイル名 (拡張子なし)

cakey

必須です。使用可能な文字: 英数字、ハイフン、アンダースコア

認証局の証明書と秘密鍵ファイルのディレクトリ

/usr/irissys/rootCA/private 参照...

次へ キャンセル

ここで、秘密鍵のディレクトリは右にある「参照...」ボタンでディレクトリを選択することも可能です。また、ファイル名欄には拡張子はいれなくてください。

「次へ」をクリックしますと以下の画面となり、秘密鍵のパスワードや、証明書情報、有効期間を入力し「保存」をクリックします。

ローカルの認証局サーバを構成

認証局秘密鍵ファイルのパスワード	
パスワード確認	
認証局 Subject Distinguished 名:		
属性タイプ	属性値	
Country	JP (2文字の国コードだけを入力してください)	
State or Province	Osaka	
Locality	Kita-ku	
Organization	InterSystems Japan	
Organizational Unit	Osaka office	
Common Name		
* 少なくともひとつの属性値を入力してください		
認証機関の証明書有効期間(日)	3650	
認証局により発行された証明書の有効期間(日)	3650	
メールを設定		
SMTPサーバ	SMTPユーザ名	
SMTPパスワード	パスワード確認	
認証局サーバ管理者のメールアドレス		
戻る	保存	キャンセル

秘密鍵(cakey.key)、証明書(cakey.cer)が指定されたディレクトリに作成されます。



サーバ ████████████████████ ネームスペース %SYS ユーザ _SYSTEM ライセンス先 InterSystems IRIS Community インスタンス IRIS-
システム > セキュリティ管理 > 公開鍵基盤 - (セキュリティの設定)

☐ 認証局クライアント

- ▶ 証明書署名リクエストを認証局サーバに送信
- ▶ 認証局サーバから証明書を取得
- ▶ ローカルの認証局クライアントを構成

認証局サーバを正常に構成しました。作成した新しいファイル: /usr/irissys/rootCA/private/cakey.cer, .key, .srl

☐ 認証局サーバ

- ▶ 保留中の証明書署名リクエストを処理する
- ▶ ローカルの認証局サーバを構成

認証局クライアントの設定

先ほど作成した認証局への接続設定を行います。

以下のメニューで「ローカルの認証局クライアントを構成」をクリックします。



サーバ ████████████████████ ネームスペース %SYS ユーザ _SYSTEM

システム > セキュリティ管理 > 公開鍵基盤 - (セキュリティの設定)

☐ 認証局クライアント

- ▶ 証明書署名リクエストを認証局サーバに送信
- ▶ 認証局サーバから証明書を取得
- ▶ **ローカルの認証局クライアントを構成**

☐ 認証局サーバ

- ▶ 保留中の証明書署名リクエストを処理する
- ▶ ローカルの認証局サーバを構成

以下の画面が表示されます。同じ環境ですので、そのまま「保存」をクリックします。

ローカルの認証局クライアントを構成

認証局サーバのホスト名	<input type="text" value="localhost"/>
	必須です。
認証局ウェブサーバのポート番号	<input type="text" value="52773"/>
	必須です。
認証局サーバのパス	<input type="text" value="/isc/pki/PKI.CAServer.cls"/>
	必須です。
ローカルの技術窓口	
名前	<input type="text" value="minamoto"/>
	必須です。
電話番号	<input type="text"/>
メールアドレス	<input type="text"/>
<input type="button" value="保存"/>	<input type="button" value="キャンセル"/>

サーバ認証の作成

続いて暗号化通信を行うサーバ証明書を作成します。

以下のメニューで「証明書署名リクエストを認証局サーバに送信」をクリック



サーバ **XXXXXXXXXXXX** ネームスペース %SYS ユーザ _SYSTEM

システム > セキュリティ管理 > 公開鍵基盤 - (セキュリティの設定)

☐ 認証局クライアント

▶ 証明書署名リクエストを認証局サーバに送信

▶ 認証局サーバから証明書を取得

▶ ローカルの認証局クライアントを構成

☐ 認証局サーバ

▶ 保留中の証明書署名リクエストを処理する

▶ ローカルの認証局サーバを構成

以下の画面が表示されるので、ファイル名、証明情報を入力し、「送信」をクリックします。
パスワードは設定しないでください。

証明書署名リクエストを認証局サーバに送信

ローカルの証明書と秘密鍵ファイルのファイル名 (拡張子なし)

必須です。使用可能な文字: 英数字、ハイフン、アンダースコア

秘密鍵ファイルのパスワード

パスワード確認

Subject Distinguished 名:

属性タイプ	属性値
Country	JP (2文字の国コードだけを入力してください)
State or Province	Osaka
Locality	Osaka
Organization	InterSystems Japan
Organizational Unit	
Common Name	MQTTServer

*少なくともひとつの属性値を入力してください

送信をクリックしますと、以下のように「保留中の証明書署名リクエストを処理する」が選択できるようになりますので、それをクリックします。

サーバ **XXXXXXXXXX** ネームスペース %SYS ユーザ _S'

システム > セキュリティ管理 > 公開鍵基盤 - (セキュリティの設定)

☐ 認証局クライアント

- ▶ 証明書署名リクエストを認証局サーバに送信
- ▶ 認証局サーバから証明書を取得
- ▶ ローカルの認証局クライアントを構成

☐ 認証局サーバ

保留中の証明書署名リクエストを処理する

- ▶ ローカルの認証局サーバを構成

クリックすると以下の画面が表示されますので、該当するリクエストの「プロセス」をクリックします。

保留中の証明書署名リクエストを処理する

ホスト名	インスタンス	ファイル名	
XXXXXXXXXX	IRISHEALTH	server2	プロセス 削除

すると証明書の詳細が表示されますので、「証明書を発行」ボタンをクリックします。

保留中の証明書署名リクエストを処理する

ホスト名	インスタンス	ファイル名
» [REDACTED]	IRISHEALTH	server2

リクエストのソース

ホスト名: [REDACTED]
インスタンス: IRISHEALTH
ファイル名: server2

リクエストの内容

Subject Distinguished 名:

C=JP
ST=Osaka
L=Osaka
O=InterSystems Japan
CN=MQTTServer

SHA-256 フィンガープリント:

DF:C2:BF:7B:37:D8:63:E6:F7:F7:EB:6C:AF:03:F5:C0:95:85:BF:E8:96:CB:AB

技術窓口

名前: minamoto
電話番号:

証明書使用目的

- TLS/SSL と XML セキュリティ
 中間証明機関
 コード署名

警告!

証明書を発行する前に、上に表示されている技術窓口にご連絡して検証する必要があります:

1. このIDです。
2. この認証局により署名された、Subject Distinguished 名を含む証明書を保持する組織
3. 上記 SHA-256フィンガープリントは、証明書署名要求の提出時に報告されたものと一

証明書を発行

キャンセル

さらに認証局のパスワードを入力画面が下に現れますので、パスワードを入力し、「完了」ボタンをクリックします。

1. このIDです。
2. この認証局により署名された、 Subject Distinguishe
3. 上記 SHA-256フィンガープリントは、証明書署名要

証明書を発行

キャンセル

認証局秘密鍵ファイルのパスワード

.....

完了

作成されたサーバ証明書の取得

作成されたサーバ証明書は以下のメニューにて「認証局サーバから証明書を取得」をクリックします。



管理ポータル

サーバ **XXXXXXXXXX** ネームスペース %SYS ユーザ _S'

システム > セキュリティ管理 > 公開鍵基盤 - (セキュリティの設定)

☐ 認証局クライアント

- ▶ 証明書署名リクエストを認証局サーバに送信
- ▶ **認証局サーバから証明書を取得**
- ▶ ローカルの認証局クライアントを構成

☐ 認証局サーバ

- ▶ 保留中の証明書署名リクエストを処理する
- ▶ ローカルの認証局サーバを構成

以下の画面が表示されますので、「取得」ボタンをクリックしますと/usr/irissys/mgrディレクトリに秘密鍵(<ファイル名>.key)、証明書(<ファイル名>.cer)が作成されます。これらをMosquittoにコピーします。

```
sudo cp /usr/irissys/mgr/server2.key /etc/mosquitto/certs
sudo cp /usr/irissys/mgr/server2.cer /etc/mosquitto/certs
sudo cp /usr/irissys/rootCA/private/cacert.cer /etc/mosquitto/ca_certificates
```

さらにserver2.keyの権限を変更します。(変更しないとエラーが発生します)

```
$ sudo chmod 600 server2.key
```

mosquitto.confに証明書を設定します。

```
:  
cafile /etc/mosquitto/ca_certificates/cacert.cer  
certfile /etc/mosquitto/certs/mqttsrv2.cer  
keyfile /etc/mosquitto/certs/mqttsrv2.key  
:
```

さらにmosquitto.confに以下の行を追加し、ポート番号を8883に変更します。

```
port 8883
```

Mosquittoを再起動し設定を反映させます。

```
/etc/init.d/mosquitto restart
```

以上でMosquittoと他のデバイス間の通信を暗号化し、接続に認証を必要とする設定が完了しました。

次回はマイコン側の配線やプログラムについて説明したいと思います。

[#IoT](#) [#相互運用性](#) [#InterSystems IRIS](#)

ソースURL:

<https://jp.community.intersystems.com/post/mqtt%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%9Firis%E3%81%A8%E3%83%87%E3%83%90%E3%82%A4%E3%82%B9%E3%81%A8%E3%81%AE%E9%80%9A%E4%BF%A1>