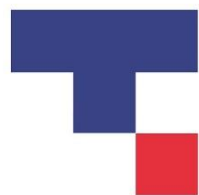


JoyCloudConnect

ユーザーズガイド

Ver1.0.0



TOKYO GAS

内容

1. はじめに.....	4
2. 製品の構成.....	6
3. ライセンスの登録.....	7
4. 起動について.....	8
4.1. 設定モード.....	8
4.2. 実行モード.....	8
5. 実行ツール.....	9
5.1. Broker.....	10
5.1.1. MQTT の設定.....	10
5.1.2. AWS の設定.....	11
5.1.3. Azure の設定.....	11
5.2. Group	12
5.2.1. 設定ダイアログ	13
5.3. アイテム	15
6. 実行ツールのテスト/実行について.....	16
6.1. モニターウインドウについて.....	18
6.1.1. MQTT Log タブ	18
6.1.2. JW Tag タブ	19
7. JccPyEdit.....	20
付録-1 MQTTTest	22
7.1. 共通部	22
7.2. MQTT 用設定	23
7.3. AWS 用設定	23
7.4. Azure 設定.....	24
8. 付録 2 過去データのアップロード.....	25

商標について

本書に記載しているすべての会社名、製品名及び商標は、それぞれの所有者に属します。

【ご注意】

本ソフトウェアの仕様・マニュアルは、予告無く変更することがあります。

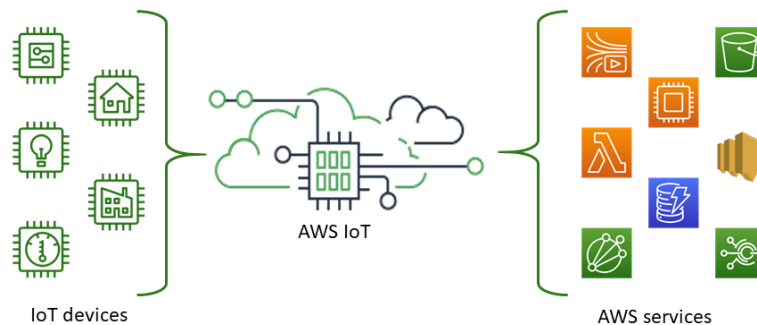
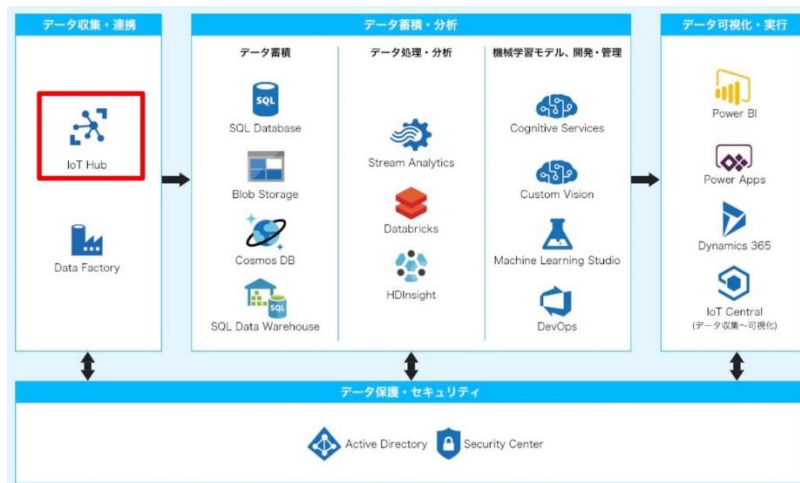
本ソフトウェア及びマニュアルの著作権は、東京ガス株式会社にあります。

許可無く本マニュアルの一部あるいは全てを複製することは禁じられています。

本ソフトウェアとマニュアルを運用した結果については、東京ガスは、一切の責任を負いかねますので、
ご了承下さい

1. はじめに

近年、インターネット技術、クラウド技術の発展により、クラウド側でのデータ処理が注目されています。



クラウドサービスを利用すると、インターネット経由でコンピューティング、データベース、ストレージ、アプリケーションをはじめとした、さまざまな IT リソースが準備されています。これらは、必要な時に必要な分だけ利用することができます。AWS/Azure 等は、そういった 200 を超えるフル機能のサービスを世界中のデータセンターから提供しています。

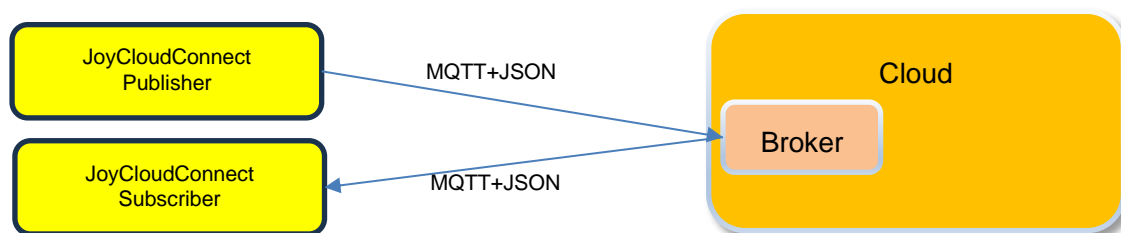
これら、クラウドの現場データの取り込み口として、Amazon や Microsoft が準備しているのが Iot Core や IoT Hub というインターフェースです。

JoyCloudConnect は、JoyWatcherSuite で取得した各種 PLC や制御機器のリアルタイム・タグデータをプログラムレスで各種クラウドにアップロードするためのツールです。

用語について

以下に、JoyCloudConnect を使用する上での基本的な用語の説明を記載します。

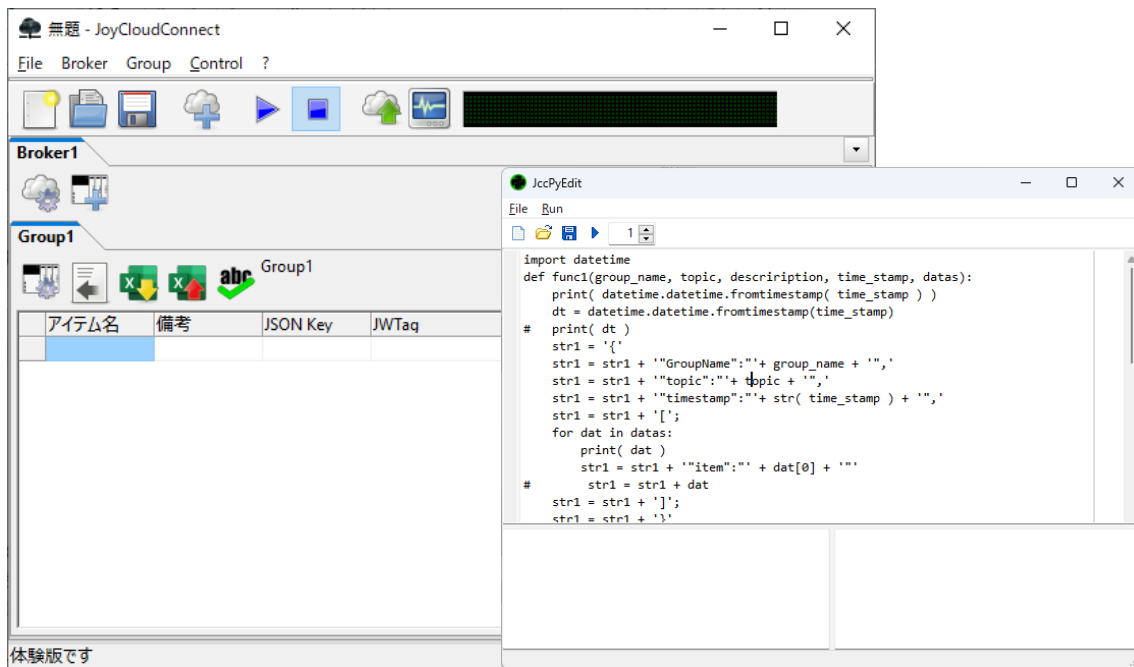
項目名	内容
MQTT	MQTT は、マシン間の通信に使用される標準ベースのメッセージングプロトコル、または一連のルールです。 JoyCloudConnect では、この MQTT プロトコルで現場とクラウドを接続します。
TOPIC	MQTT プロトコルでは、メッセージの講読や発行の対象として MQTT トピックを使用します。言ってみれば、トピックとはひとまとまり電文につけるタイトルのことです。プロトコル上、MQTT トピックは Publisher が多数存在し、それらは階層を持ったトピック名として表現されます。SubScriber は MQTT のトピック名を指定してそのトピックを講読したり、メッセージを発行したりできます。 ※Azure では、TOPIC を使用できません。
Broker	MQTT のサーバーのことを指します。
Publisher	サーバーにデータを送るクライアントです。JoyCloudConnect も基本的にはここに該当します。
Subscriber	サーバーからデータを受信するクライアントです。
JSON	JSONとは「JavaScriptのオブジェクト記法を用いたデータ交換フォーマット」です。一般的な MQTT では、JSON 形式でデータをやり取りします。
アイテム	JoyCloudConnect の用語です。JoyWatcher のタグとそれを同のように送るかをまとめたものです。



2. 製品の構成

JoyCloudConnect は、次の 2 つのプログラムで構成されます。

- JoyCloudConnect 実行用のモジュール
- JccPyEdit 出力する JSON 形式をカスタマイズするための設定ツール



ファイルとしては、拡張子 .json と .py ファイルを使用します。

JccPyEdit を使わないでもクラウドヘデータをアップロードすることは可能です。ただ、クラウド側が送信に使用する JSON の形式を指定する場合があります。そのような場合に JoyCloudConnect では、Python を利用したスクリプトで JSON の形式を自由にカスタマイズすることが可能です。

JSON の形式をカスタマイズするためには、Python を使用したスクリプトを書く必要があります。またその場合には、設定ファイルの JSON と Python のプログラムファイルを拡張子だけ違う同じファイル名で保存してください。

例:

JoyCloudConnect のファイル名を JoyCloud.json とした場合は、Python のファイルも JoyCloud.py として保存してください。

3. ライセンスの登録

JoyCloudConnect を製品版として使用する場合、ライセンス登録プログラムを用いてライセンスの登録が必要になります。

ライセンス登録を行わない場合、ファイルの作成、編集は行えますが、実行時、1 時間で強制的に停止します。

以下に、ライセンス登録プログラムを用いた簡易の登録例を記載します。

※ライセンス登録プログラムの実行には、JoyWatcherSuite のインストールが必要となります。

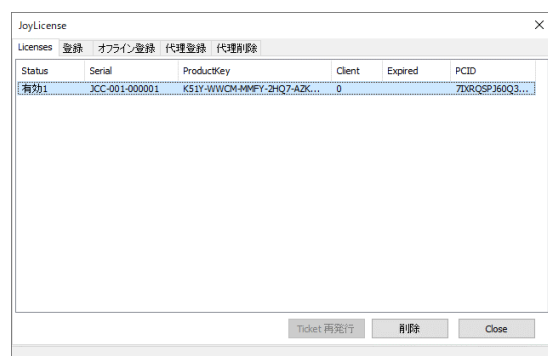
例ではオンライン登録について記載します。オフライン登録を行う場合は、ライセンスを登録する PC のほかに、ライセンスサーバーに接続する PC を別途用意いただく必要があります。

オフライン登録の手順に関しては、別紙『JoyLicense.pdf』をご参照ください。



JoyLicense 登録画面のスクリーンショット。タブは「Licenses 登録」で、サブタブは「オンライン登録」「代理登録」「代理削除」があります。入力欄には「シリアル番号」が「JCC-001-000001」と、「プロダクトキー」が「K51Y-WWCM-MMFY-2HQ7-AZKF-CGCK-NVNZ-QV6Q」が入力されています。下部には「オンライン登録」ボタンがあります。

登録画面で、シリアル番号とプロダクトキーを入力して「オンライン登録」をクリックします。



JoyLicense Licenses 画面のスクリーンショット。タブは「Licenses 登録」「オンライン登録」「代理登録」「代理削除」があります。下部には「Ticket 再発行」「削除」「Close」ボタンがあります。

Status	Serial	ProductKey	Client	Expiry	PCID
有効1	JCC-001-000001	K51Y-WWCM-MMFY-2HQ7-AZKF...	0		7E8RQBP360Q3...

登録されると、Licenses 画面に、Status が「有効 1」の状態に登録されます。

4. 起動について

JoyCloudConnect は、設定を行う設定モードと実際に通信を行う実行モードの 2 つのモードがあります。

4.1. 設定モード

スタートメニューから起動すると、設計モードで起動します。ここでは、設定やテスト実行等が可能です。

4.2. 実行モード

JoyCloudConnect を実行モードに開くには、引数起動を行ってください。/RUN 付きで起動することで、JoyCloudConnect は、実行状態、かつタスクトレイ起動します。

JoyCloudConnect.exe のパス /RUN 設定ファイル名

例 : C:¥Program Files (x86)¥JoyCloudConnect¥JoyCloudConnect.exe /RUN
C:¥JWS¥CloudTest.JSON

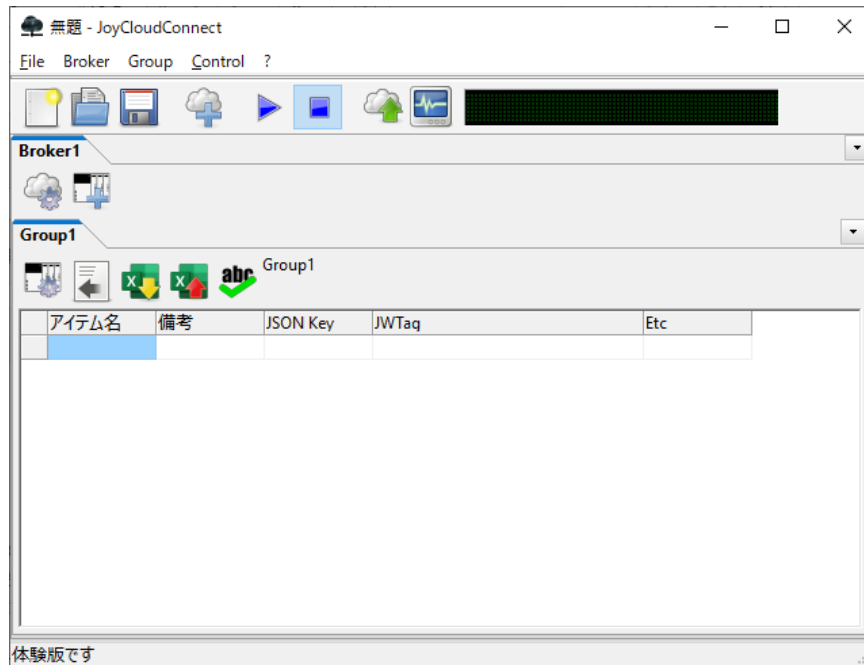
ファイル名は、JSON のファイル名を拡張子ありで、フルパスで指定してください。Python によるカスタマイズを行う時には、JSON と同じ場所にファイルを置いてください。

引数起動の中では、.py ファイルの指定は必要ありません。

JoyCloudConnect は 2 重起動を可能ですが、/RUN で起動した時には、同じ設定ファイル名のものを 2 重起動はできないようになっています。

5. 実行ツール

JoyCloudConnect を起動すると次の画面が表示されます。



JoyCloudConnect では、3つの階層でデータを扱います。

1. Broker クラウド側 MQTT の接続先
2. Group 送信するタイミング等の設定
3. アイテム JoyWatcher のタグをどのような名前で送るかの設定

このプログラムでは、これらの階層ごとの設定を行います。

1つのBrokerには、1つ以上のグループを含むことが可能で、1つのグループには1つ以上のアイテムを含むことが可能です。

5.1. Broker

1 段目のタブが Broker の設定になります。Broker ごとにタブが分けられ表示されます。

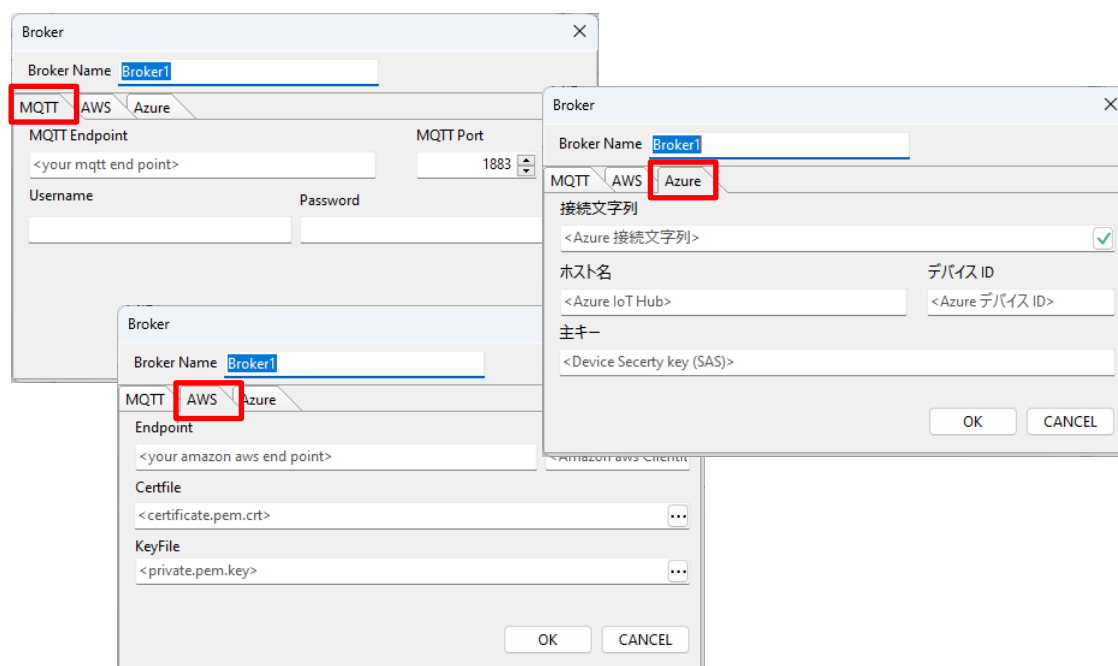
メインメニューまたはツールバーから Broker の追加で Broker を追加できます。複数の Broker が登録済みの場合には、Broker タブに×が表示され、Broker を削除することが可能です。



※Broker の数は 1 つ以下にすることはできません。

Broker の Edit ボタンで設定ダイアログを開くことが可能です。

設定ダイアログは接続する Broker に応じて、MQTT/AWS/Azure の 3 つの設定が行えます。必要に応じてそれぞれのタブで設定を行ってください。最後に開いているタブに選択されている Broker の設定が有効になります。表示中のタブに入力された項目のみが有効です。



5.1.1. MQTT の設定

MQTT の設定を行います。

MQTT Endpoint	接続する MQTT の IP アドレスを設定します。(必須)
MQTT Port	接続する MQTT のポート番号を設定します。(必須)
Username	MQTT でユーザー認証が必要な場合にユーザー名を設定します。(オプション)
Password	MQTT でユーザー認証が必要な場合にパスワードを設定します。(オプション)

5.1.2. AWS の設定

AWS に接続する設定を行います。AWS の場合には、署名ファイルを使用した認証を利用します。

Endpoint	接続する AWS の IP アドレスを設定します。(必須)
CertFile	AWS 側で作成した暗号化ファイルを設定します。(必須)
Keyfile	AWS 側で作成したプライベートのキーファイルを設定します。(必須)

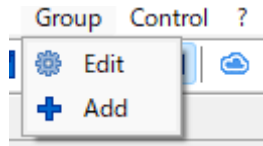
5.1.3. Azure の設定

Azure に接続する設定を行います。Azure の場合には、接続ファイルを入力しての個別項目の転記と、直接個別入力の 2 つが可能です。

接続文字列	Azure で表示される接続文字列を入力します。 横の転記ボタンをクリックすることで、ホスト名、デバイス ID、主キーにデータが転記されます。 この項目は入力補助なので、このフィールドそのものは記録しません。
ホスト名	接続する Azure の IP アドレスを設定します。(必須)
デバイス ID	Azure で設定されているデバイス ID を設定します。(必須)
主キー	Azure で設定されている主キーを設定します。(必須)

5.2. Group

グループは、Borker にどのようなタイミングでどのデータを送信するかの設定を行います。



左の画面のアイコンで、設定と追加を行うことができます。

Broker に複数のグループが登録されていると、タブに×が表示され削除することも可能です。

※グループの数は 1 つ以下にすることはできません。



グループタブのボタンの意味は次の通りです。

	設定ダイアログを表示します。
	JoyWatcher タグのまとめ選択を行います。
	表示されているアイテムを Group 名のシートで EXCEL に転記します。
	アイテムを EXCEL の同名シートから読み込みます。 ※表示している Group 名と同名のシートが無い場合には、読み込みは行われません。
	設定された JoyWatcher タグの存在確認を行います。 存在している場合は OK、存在していない場合は NG が表示されます。
Group1	設定されたトピック名が表示されます。

5.2.1. 設定ダイアログ

設定ダイアログには次のような設定があります。

Group Name	名前を設定します。																		
Description	備考を設定します。																		
Pub/Sub	送受信の形式 ※別章で説明 Publish：送信 JoyCloudConnect 標準 JSON Subscribe：受信 PublishIIIJ：送信 IIIJ 形式の JSON PublishPy1.. PublishPy8：ユーザー定義の JSON ※ 1																		
Topic	送信に使用する MQTT の Topic を設定します。 ※Azure では使用しません																		
Triger	送信タイミング 送信タイミング、送信内容を選択します。 <table><tr><th>項目</th><th>内容</th></tr><tr><td>定周期 秒</td><td>設定した秒の周期で送信します。</td></tr><tr><td>定周期 分</td><td>設定した分の周期で送信します。</td></tr><tr><td>定周期 時</td><td>設定した時の周期で送信します。</td></tr><tr><td>変化 変化アイテムのみ</td><td>アイテムに登録したタグのいずれかが変化したとき、変化した内容のみを送信します。</td></tr><tr><td>変化 全アイテム</td><td>アイテムに登録したタグのいずれかが変化したとき、すべての内容を送信します。</td></tr><tr><td>先頭アイテム変化</td><td>先頭アイテムのタグが変化したとき、すべての内容を送信します。</td></tr><tr><td>先頭アイテム ON</td><td>先頭アイテムのタグが 0 以外になったとき、すべての内容を送信します。</td></tr><tr><td>先 頭 ア イ テ ム ON(自動 off)</td><td>先頭アイテムのタグが 0 以外になったとき、すべての内容を送信します。その後、先頭アイテムのタグを 0 に戻します。</td></tr></table>	項目	内容	定周期 秒	設定した秒の周期で送信します。	定周期 分	設定した分の周期で送信します。	定周期 時	設定した時の周期で送信します。	変化 変化アイテムのみ	アイテムに登録したタグのいずれかが変化したとき、変化した内容のみを送信します。	変化 全アイテム	アイテムに登録したタグのいずれかが変化したとき、すべての内容を送信します。	先頭アイテム変化	先頭アイテムのタグが変化したとき、すべての内容を送信します。	先頭アイテム ON	先頭アイテムのタグが 0 以外になったとき、すべての内容を送信します。	先 頭 ア イ テ ム ON(自動 off)	先頭アイテムのタグが 0 以外になったとき、すべての内容を送信します。その後、先頭アイテムのタグを 0 に戻します。
項目	内容																		
定周期 秒	設定した秒の周期で送信します。																		
定周期 分	設定した分の周期で送信します。																		
定周期 時	設定した時の周期で送信します。																		
変化 変化アイテムのみ	アイテムに登録したタグのいずれかが変化したとき、変化した内容のみを送信します。																		
変化 全アイテム	アイテムに登録したタグのいずれかが変化したとき、すべての内容を送信します。																		
先頭アイテム変化	先頭アイテムのタグが変化したとき、すべての内容を送信します。																		
先頭アイテム ON	先頭アイテムのタグが 0 以外になったとき、すべての内容を送信します。																		
先 頭 ア イ テ ム ON(自動 off)	先頭アイテムのタグが 0 以外になったとき、すべての内容を送信します。その後、先頭アイテムのタグを 0 に戻します。																		
Interval	Triger で定周期を選択した時に間隔を指定します。																		

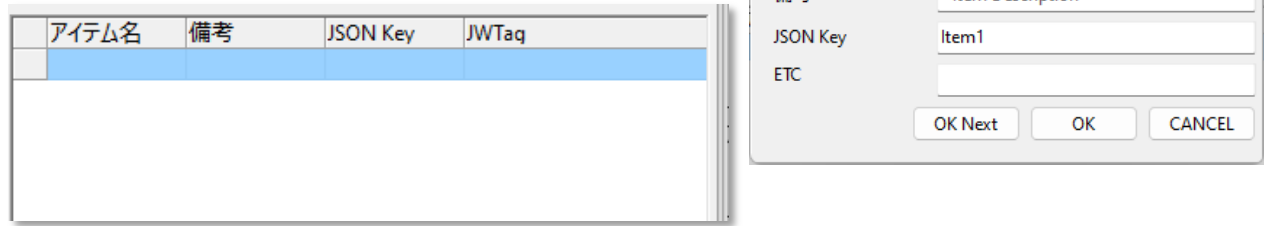
2 番目の行で有効切り替え	2 番目のアイテムを札掛けタグとして使用します。 2 番目のアイテムに登録したタグが 0 の時には送信しません。
起動時に送信する	このチェックボックスにチェックを入れないと、トリガー条件が満たすまで送信しません。

定周期通信

JoyCloudConnect では、定周期 時/分/秒 の場合に、Interval と組み合わせて送信タイミングを設定します。例えば、定周期 時 と Interval 7 の組み合わせでは 7 時間ごとに送信という意味になります。ただし、JoyCloudConnect では、時間の基準を日の 00 :00 :00 にします。先の例の 7 時間間隔では、毎日 00 :00, 07 :00,14 :00,21 :00 の 3 回送信します。

5.3. アイテム

JoyCloudConnect では、実際に送信する、JoyWatcher のタグと、送信する際に使用する情報をまとめてアイテムと呼びます。

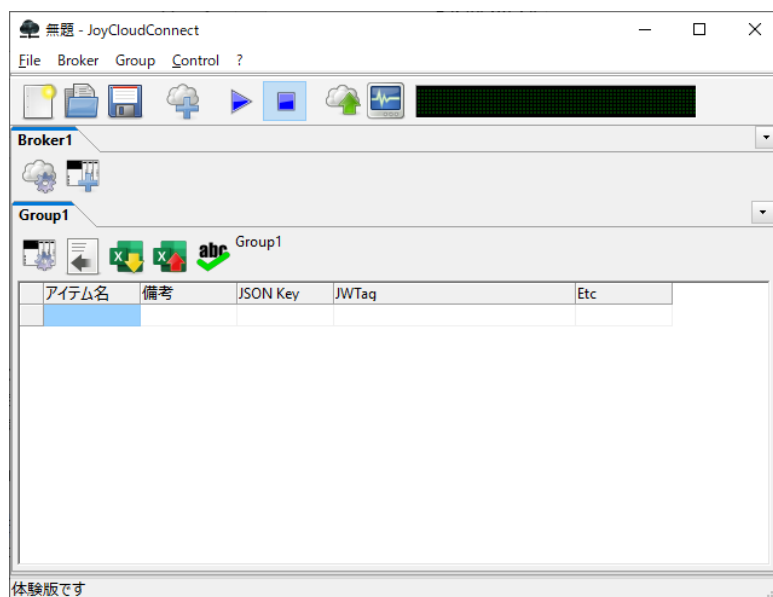


アイテムの設定は、表形式に直接入力、表形式をダブルクリックで設定ダイアログを表示することで行うことができます。まとめて編集したい時には、EXCEL との連携で設定することも可能です。





JoyWatcher Tag	使用する JoyWatcher のタグを設定します。(必須)
アイテム名	アイテムの名前を設定します。(必須)
備考	備考を設定します。(オプション)
JSON KEY	JSON 形式の KEY 部を設定します。(必須)
ETC	備考の予備です。補足情報があれば設定してください。(オプション)
OK Next	設定を終了し、次の項目に移動します。

6. 実行ツールのテスト/実行について

JoyCloudConnect には、設計モードと実行モードがあります。設計モードで一通り設計が終わったなら、実行モードに変更し、テストを行います。テストの際には、最低限 JoyWatcher サーバーの起動が必要になります。



実行ツールのツールバーで実行（テスト）を行うことが可能です。

	クラウドへの実行/停止を管理します。
	クラウド接続/クラウド接続無し（ローカルテスト）を切り替えます。
	モニター画面を表示します。
	体験版で動作させた場合にのみ表示されます。製品版では表示されません。 体験版では実行中にカウントダウンが表示され、1 時間がたつと実行状態が解除されます。

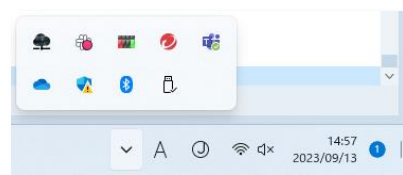
実行モードにすると、JoyWatcher との通信、クラウドへの通信を始めます。

クラウドへの 接続無しは、実行モードにしても、クラウドには接続せず、ローカル側の設定が正しいかどうかのチェックのために使用します。

※クラウド接続なしとした場合は、モニター画面に Status が Fake Send でクラウド送信されるデータが追加されます。

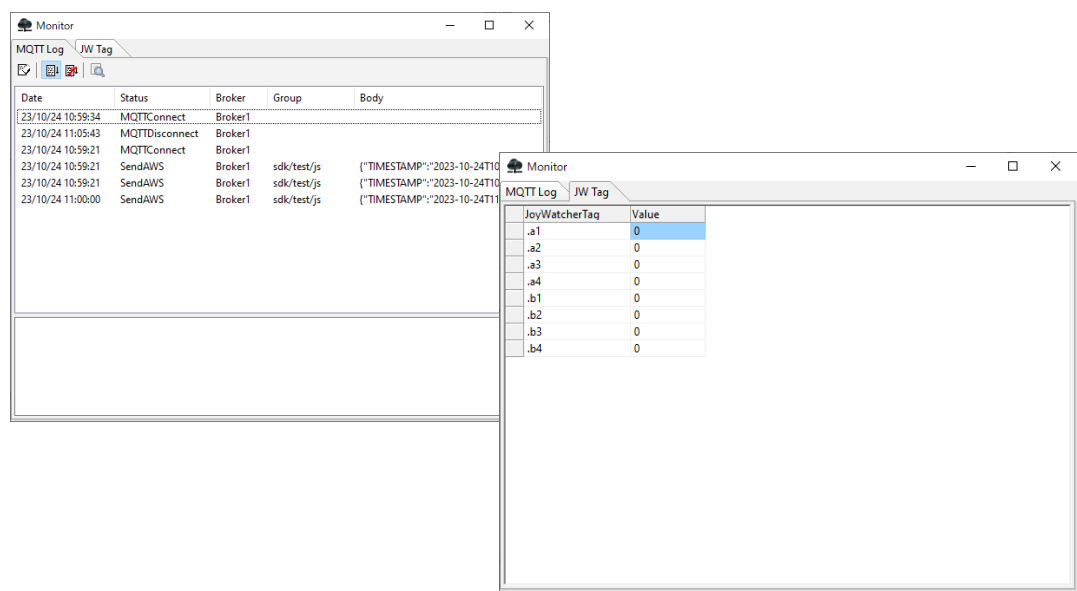
実行中にウインドウの最初化を行うと、タスクバーではなく、右下のタスクトレイに格納します。タスクトレイの JoyCloudConnect のアイコンをダブルクリックすることで表示できます。また右クリックでポップアップメニューが表示されます。

項目	内容
Show	画面を再表示します。
Exit	プログラムを終了します。







6.1. モニターウインドウについて

実行中にモニターウインドウを表示すると次のような画面が表示されます。




6.1.1. MQTT Log タブ

クラウドとの通信の情報を表示します。
上のツールバーのボタンは次の通りです。

項目	内容
	ログの消去を行います。
	ログ自動スクロールモードです。 ログが追加されると自動的に画面がスクロールされます。
	ログ自動スクロールモードを停止します。
	ログの詳細を表示します。 選択行の JSON の内容を表示します。

ログには以下の内容が表示されます。

項目	内容
Date	データの送信日時が表示されます。
Status	状態が表示されます。 例

	接続 : MQTTConnect 切断 : MQTTDisconnect AWS への送信 : SendAWS クラウド送信無効時 : Fake Send
Broker	Broker の情報が表示されます。
Group	Group の情報が表示されます。
Body	JSON の中身が表示されます。  をクリックすることで、詳細が表示されます。

8.1.2. JW Tag タブ

登録された JoyWatcher のタグの状態を表示します。

項目	内容
JoyWatcher タグ	JoyWatcher タグ名が表示されます。
Value	タグの値が表示されます。

7. JccPyEdit

JoyCloudConnect では 汎用として JSONをサポートしています。システムより、JSON の形式が指定されている場合には、Python を使用したスクリプトを書くことにより柔軟に構造を定義することが可能です。Python は、3.11 形式で対応します。

このための設定ツールとして、JccPyEdit が付属します。

設定の流れ

1. Group の Sub/Pub の設定で PublishPy1 から PublicshPy8 のどれかを選択します。
2. PublishPy1 から PublishPy8 に対応した Python のスクリプトを書いたファイルを設定ファイルと同じ場所に、同名で拡張子だけ変えて配置してください。

使用する Python のルールは次の通りです

```
def func1(group_name, topic, description, time_stamp, datas):  
    str = '{"aaa":123}'  
    return str1
```

SubscribePy1 – SubscribePy8 に対応する関数を func1 から func8 で作成します。
戻り値は、JSON そのものの 文字列とします。

引数は次の通りです

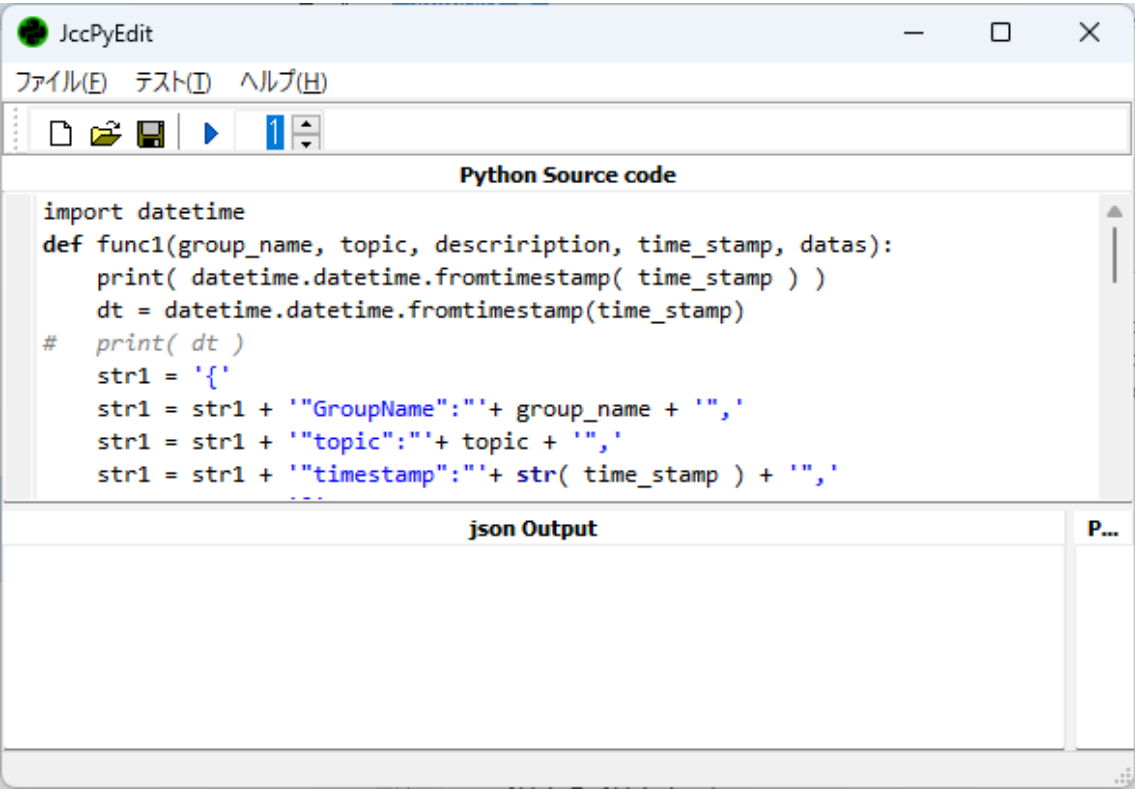
group_name	Group の名前を設定します。
topic	Group に設定した topic を設定します。
description	Group に設定した description を設定します。
time_stamp	変化の日時 UNIX のエポック秒を設定します。
datas	対象のアイテム数分の タプル の配列 アイテム数分下の要素のタプルが渡される アイテム 名前 アイテム Description アイテム JSON KEY 値 数字、BIT の場合には引用符なし、文字列は 文字列として アイテムの ETC





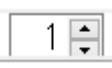
このようなスクリプトを定義するために、JoyPyEdit が付属します。

Python のスクリプトを読み込み、編集、テストする簡易開発環境です。

自作した Python のプログラムの Publish1-Publish8 相当の関数を呼び出し、エラーなく実行できるか、その戻り値の JSON が想定したものと同一かを確認できます。

起動するには、JoyCloudConnect のインストールフォルダ下の JccPyEdit.exe を実行してください。



項目	内容
	Python Source code を初期状態に戻します。
	保存した.py ファイルを開きます。
	.py ファイルを保存します。
	Python Source code の内容を実行し、結果を json output と Python output に出力します。
	実行する func を設定します。初期状態では func1 のみが動作します。

付録-1 MQTTTest

MQTT/AZURE/AWS と接続と、Publish と Subscribe の確認用プログラムが付属します。このプログラムは JoyCloudConnect のライセンスを使用せず、実環境でクラウドのサーバーと通信できるかどうかの確認が可能です。

起動するには、JoyCloudConnect のインストールフォルダ下の MQTTTest.exe を実行してください。

別環境で動作させるには、ダウンロードした JoyCloudConnect.zip に格納されている「MQTTTest」フォルダをコピーして、動作をご確認ください。

設定は、通常の MQTT と AWS と Azure の 3 つのタブに分かれています。

接続設定は、それぞれのタブ内で行ってください。

7.1. 共通部

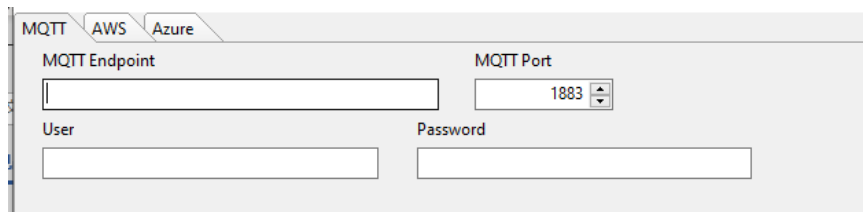
The image shows a screenshot of the MQTTTest application interface. It features a light gray background with several buttons and input fields. On the left, there are five buttons arranged in a 3x2 grid: 'Connect', 'Disconnect', 'Send', 'Subscribe', and 'Unsubscribe'. To the right of these buttons, there are three input fields. The first is labeled 'send topic' in red text and contains the text 'topic1'. The second is labeled 'recv topic' in red text and also contains 'topic1'. The third is labeled 'send msg' in red text and contains the JSON string '{"AA":123}'.

項目	内容
Connect	設定したサーバーに接続します。
Disconnect	接続しているサーバーから切断します。
Send	横にある send topic+send msg(JSON)を送信します。
Subscribe	recv topic の内容が変更された時に、通知が来るようになります。
Unsubscribe	通知設定を解除します。

Topic の内容は、弊社で確認した範囲では日本語も対応していました。

対応していない内容(空白等)を送った場合も、エラーはサーバーから帰ってきません。エラーを送った場合には即座に Disconnect が通知されます。


7.2. MQTT 用設定



項目	内容
MQTT Endpoint	MQTT ブローカーのホスト名を設定します。
MQTT PORT	接続ポート番号を設定します。
User	MQTT ブローカーの認証が必要な場合に、認証ユーザー名を設定します。
Password	MQTT ブローカーの認証が必要な場合に、認証パスワードを設定します。

7.3. AWS 用設定

AWS は、x509 認証必須 + ファイルを AWS で生成されるため、その設定を使用します。
ClientID と Topic は セキュリティポリシーで設定必須となります。



項目	内容
Endpoint	AWS ブローカーのホスト名を設定します。
Client_id	AWS で設定したクライアント ID を設定します。
Cerfile	AWS で作成された Cerfile のパスを設定します。
Private key	AWS で作成された Private key のパスを設定します。

弊社で試験した接続手順の一例を記載します。

※あくまで一例です。実際に接続を行う際には、必要に応じて設定を変更ください。

- 1.AWS で「AWS IoT」を選択
- 2.「1 個のデバイスを接続」-任意名称で新しい「モノ」を作成
- 3.プラットフォームを Windows、Node.js を選択
- 4.接続キットをダウンロードし、解凍した中にある、ps.1 ファイルを読み込む

7.4. Azure 設定

項目	内容
Connection String	Azure 側で作成されたプライマリ接続文字列を設定します。
AzureParse	Connection String に設定された内容から、対応する Host Name、DeviceId、ShardAccessKey を貼り付けます。
Host Name	接続先のブローカー名を設定します。
DeviceId	データを送るデバイス ID を設定します。
ShardAccessKey	生成されたアクセスキーを設定します。

弊社で試験した接続手順の一例を記載します。

※あくまで一例です。実際に接続を行う際には、必要に応じて設定を変更ください。

簡易手順

1. Azure にログイン
2. IoT Hub の作成 XXX
3. Device を作る
4. プライマリ接続文字列を Connection string に入力
5. Azure Parse をクリックすることで、必要な情報が設定される

8. 付録 2 過去データのアップロード

過去データをアップロードする場合、DB データを JoyWatcher タグにコピーする必要があります。

標準機能で実現する場合には、JWDBLinkiTool を使用することで、DB 内のデータを JoyWatcher タグに出力できるため、こちらと連携させて、クラウドへのアップロードを行ってください。

JWDBLinkiTool の使用方法に関しては、別冊『JoyWatcherSuite ユーザーズガイド.pdf』を参照ください。

JoyCloudConnect ユーザーズガイド

2023 年 10 月 25 日 Ver1.0.0 版発行

※本書に関するご質問・お問い合わせについては下記までお願いいたします。

<https://eee.tokyo-gas.co.jp/lp/joy/contact/form.html>

パッケージの概要説明等、営業的なお問い合わせは「パッケージソフトに関する営業的なお問い合わせ」に。

具体的な設定手順等、技術的なお問い合わせは「パッケージソフトに関する技術的なお問い合わせ」にお送りください。

技術的なお問い合わせに関しては、問い合わせ専用サイトを使用しております。

ユーザー申請(無料)後、送られるメールから、問い合わせサイトのログイン情報を設定して、ログインいただければと思います。

※ユーザー申請からメールの送信までは、最大1営業日ほどお時間を頂く場合がございますので、あらかじめご承知おきください。