

Google Cloud IoT Core プロバイダ

Version 1.3.0

ユーザーズ ガイド

Aug 23, 2022

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2020-03-05	初版.
1.1.0	2020-09-01	メッセージの受信に対応.
1.2.0	2021-03-19	TLS1.2 対応. EXE 形式に変更.
1.3.0	2022-08-23	メッセージを受け取る間隔が変更できない問題を修正.

【対応機器】

機種	バージョン	注意事項

目次

1. はじめに.....	4
2. プロバイダの概要.....	5
2.1. 概要.....	5
2.2. メソッド・プロパティ.....	6
2.2.1. CaoWorkspace::AddController メソッド.....	6
2.2.2. CaoController::AddVariable メソッド.....	8
2.2.3. CaoVariable:get_VariableNames プロパティ.....	8
2.2.4. CaoVariable:put_Value プロパティ.....	8
2.2.5. CaoController::OnMessage イベント.....	9
2.2.5.1. イベント受信 (MQTT).....	9
2.2.5.2. イベント受信 (HTTP).....	9
2.3. 変数一覧.....	10
2.3.1. CaoController クラス.....	10
2.4. エラーコード.....	11

1. はじめに

本書は、Google がクラウド上で提供するサービスである Google Cloud Platform(GCP)のうち、Cloud IoT Core とデータの送信を行う CAO プロバイダのユーザーズガイドです。

本書で扱う CAO プロバイダ(CaoProvGoogleCloudIoTCore.exe)を Cloud IoT Core プロバイダと呼びます。第 2 章に Cloud IoT Core プロバイダの概要、変数の詳細を記載しています。

本プロバイダは Google cloud platform との通信にオープンソースのライブラリを使用しています。これらに関しては、以下のサイトを参照ください。

[M2Mqtt のサイトリンク]

URL: <https://github.com/eclipse/paho.mqtt.m2mqtt>

[libcurl.NET の著作権とライセンス]

Copyright (c) 2016 Patrick Quirk

MIT License

<https://github.com/pjquirk/libcurl.NET/blob/master/LICENSE.md>

[Bouncy Castle の著作権とライセンス]

Copyright (c) 2000 - 2019 The Legion of the Bouncy Castle Inc.

MIT License

<https://www.bouncycastle.org>

[jose-jwt の著作権とライセンス]

Copyright (c) 2014-2019 dvsekhvalnov

MIT License

<https://github.com/dvsekhvalnov/jose-jwt/blob/master/LICENSE>

このアプリは Eclipse Public License 1.0 のライセンスで配布されている成果物を含んでいます。

<https://github.com/eclipse/paho.mqtt.m2mqtt/blob/master/LICENSE>

2. プロバイダの概要

2.1. 概要

Google がクラウド上で提供するサービスである Google Cloud Platform(GCP)のうち、Cloud IoT Core とデータの送信を行う CAO プロバイダです。そのファイル形式は EXE であり、CAO エンジンから使用時に動的にロードされます。Cloud IoT Core プロバイダを使用するには表 2-1 の方法で登録を行う必要があります。RegistAsm.bat および UnregistAsm.bat は ORiN2SDK をインストールしたフォルダの下の DotNet¥BAT フォルダにあります。

表 2-1 Cloud IoT Core プロバイダ

ファイル名	CaoProvGoogleCloudIoTCore.exe
ProgID	CaoProv.Google.CloudIoTCore
レジストリ登録	RegistAsm.bat CaoProvGoogleCloudIoTCore.exe
レジストリ登録の抹消	UnregistAsm.bat CaoProvGoogleCloudIoTCore.exe

2.2. メソッド・プロパティ

2.2.1. GaoWorkspace::AddController メソッド

Cloud IoT Core プロバイダは AddController 時に通信用の接続パラメータを参照し、Cloud IoT Core との接続を行います。



```
AddController(<bstrCtrlName:BSTRT>,<bstrProvName:BSTRT>,  
              <bstrPCName:BSTRT>,<bstrOption:BSTRT>))
```

bstrCtrlName : [in] コントローラ名

bstrProvName : [in] プロバイダ名. 固定値 = "CaoProv.Google.CloudIoTCore"

bstrPcName : [in] プロバイダの実行マシン名

bstrOption : [in] オプション文字列

以下にオプション文字列に指定するリストを示します。

表 2-2 CaoWorkspace::AddController のオプション文字列

オプション ¹	説明
@EventDisable=[<クラウドからのメッセージ受信>]	受信有効時:"False", 無効時:"True". デフォルト:False.
BaseUrl=[<BaseUrl>]	GCP の API サービスに接続するための URL. プロトコルに HTTP を選択時のみ有効. (デフォルト: https://cloudiotdevice.googleapis.com/v1)
CaCertificates=[<ルート証明書のパス>]	ルート証明書のフルパス. プロトコルに MQTT を選択時のみ有効.
CloudRegion=<CloudRegion>	Cloud IoT Core で作成した Registry の地域.
DeviceId=<DeviceId>	Cloud IoT Core で作成した Device の ID.
HttpEventInterval=<HttpEventInterval>	メッセージの受け取る間隔(分)を設定. HTTP を選択時のみ有効.
HttpTimeOut=[<HTTP のタイムアウト時間>]	リクエスト時のタイムアウト時間(秒). プロトコルに HTTP を選択時のみ有効. タイムアウト時間を設けない場合は「0」を入力(デフォルト:5)
MqttBridgeHostname=[<MQTT の接続先>]	Cloud IoT Core で使用する MQTT の接続先. プロトコルに MQTT を選択時のみ有効. (デフォルト:mqtt.googleapis.com)
MqttBridgePort=[<MQTT のポート番号>]	Cloud IoT Core で使用する MQTT の接続先のポート番号. プロトコルに MQTT を選択時のみ有効. (デフォルト:8883)
PrivateKeyFile=<PrivateKeyFile のファイルパス>	Device の公開鍵に対応する秘密鍵のファイルパス. 秘密鍵は「RS256」, 「RS256_X509」に対応.
ProjectId=<ProjectId>	Cloud IoT Core で使用しているプロジェクトの ID
Protocol=[<Protocol 番号>]	通信に使用するプロトコル. (1:MQTT, 2:HTTP, デフォルト:1)
QoS=[<QoS レベル>]	QoS レベル. プロトコルに MQTT を選択時のみ有効. (0:レベル 0, 1:レベル 1, デフォルト:0)
RegistryId=<RegistryId>	Cloud IoT Core で作成した Registry の ID.

¹ 角括弧("[]")内は省略可能を示します.

2.2.2. CaoController::AddVariable メソッド

CaoController クラスの AddVariable メソッドは、それぞれのプロバイダがの変数オブジェクトを作成するためのメソッドです。変数名には、2.3.1 章の変数のみ使用することができます。



AddVariable(<bstrVariableName:VT_BSTR>[,<bstrOption:VT_BSTR>])

<bstrVariableName> : [in] 変数名

<bstrOption> : [in] オプション文字列

オプション文字列には以下のものを用います。

表 2-3 CaoController::AddVariable のオプション文字列

オプション	意味
MessageType=[<MessageType>]	送信メッセージのタイプ。 (1:EVENT, 2:STATE, デフォルト:1)

2.2.3. CaoVariable:get_VariableNames プロパティ

2.3.1 章の変数を取得します。

2.2.4. CaoVariable:put_Value プロパティ

変数に対応する情報を設定します。各変数の実装状況および設定データについては、2.3.1 章を参照して下さい。使用できる変数の中でも Cloud IoT Core と接続するものは通信を行うため、処理が完了するまで時間がかかるものがあります。

2.2.5. CaoController::OnMessage イベント

以下の契機で CaoController クラスの OnMessage イベントが発生します。

表 2-4 メッセージ種別

	メッセージ種別	発生契機
1	Cloud IoT Core データ 受信	Cloud IoT Core からデータ受信が行われた際に発生します。

2.2.5.1. イベント受信(MQTT)

クラウドからのメッセージで得られるデータ形式を以下に示します。

受信できるメッセージは Config と Command, Error です。

Number : 受信メッセージの番号
Value : 受信したメッセージの内容
DateTime : タイムスタンプ
Description : 受信時のトピック名

2.2.5.2. イベント受信(HTTP)

クラウドからのメッセージで得られるデータ形式を以下に示します。

受信できるメッセージは Config です。

Number : 受信メッセージの番号
Value : 受信したメッセージの内容
DateTime : タイムスタンプ
Description : -

2.3. 変数一覧

2.3.1. CaoController クラス

表 2-5 CaoController クラス ユーザ変数一覧

変数名	データ型	説明	属性		オプション
			get	put	MessageType
*	VT_BSTR	Cloud IoT Core に JSON 文字列を送信します。	-	○	○

2.4. エラーコード

Cloud IoT Core プロバイダでは、以下の独自エラーコードが定義されています。ORiN2 共通エラーについては、「ORiN2 プログラミングガイド」のエラーコードの章を参照してください。

表 2-6 独自エラーコード一覧

エラー名	エラー番号	説明
CloudRegionNull	0x80100001	AddController オプション「CloudRegion」が未入力です。
DeviceIdNull	0x80100002	AddController オプション「DeviceId」が未入力です。
PrivateKeyFileNull	0x80100004	AddController オプション「PrivateKeyFile」が未入力です。
PrivateKeyFileNotExist	0x80100005	PrivateKeyFile が見つかりません。
PrivateKeyFileIllegal	0x80100014	PrivateKeyFile が不正です。「-----BEGIN PRIVATE KEY-----」から始まっていることを確認してください。
ProjectIdNull	0x80100006	AddController オプション「ProjectId」が未入力です。
ProtocolIncorrect	0x80100007	AddController オプション「Protocol」が不正です。
registryIdNull	0x80100008	AddController オプション「RegistryId」が未入力です。
CaCertNull	0x80100009	AddController オプション「CaCertificates」が未入力です。
CaCertNotExist	0x8010000A	CaCertificates が見つかりません。
MqttBridgePortIncorrect	0x8010000B	AddController オプション「MqttBridgePort」が不正です。
MqttBridgePortOutOfRange	0x8010000C	AddController オプション「MqttBridgePort」が範囲外です。
QoSOutOfRange	0x8010000D	AddController オプション「QoS」が未入力です。
RequestTimeOutIncorrect	0x8010000E	AddController オプション「RequestTimeOut」が不正です。
RequestTimeOutOutOfRange	0x8010000F	AddController オプション「RequestTimeOut」が未入力です。
ConnectFailed	0x80100010	接続に失敗しました。

HttpConnectionFailed	0x80100016	Http による接続が失敗しました。表示される Http のエラーコードをもとに対処してください。
MessageTypeIncorrect	0x80100011	AddVariable オプション「MessageType」が不正です。
PublishMessageFailed	0x80100012	メッセージの送信に失敗しました。
FailedToCreateJWT	0x80100015	JWT の作成しに失敗しました。PrivateKeyFile が正しいか確認してください。
MqttClientDisconnect	0x80100013	MQTT のクライアントが見つかりません。再度 AddController を行ってください。