

Introduction

This application note provides a guide to connect Google Cloud IoT Core with iGS02E via mqtt bridge.

Get Started

The first step is ensuring that you have a [Google Cloud IoT Core](#) account set up with IoT core.

Follow the [IoT Core Quick start](#) to create a Cloud IoT Core device registry and register a device.

After following the instructions in Quickstart guide, you should have PROJECT_ID, REGION, REGISTRY_ID and DEVICE_ID settings. These settings will be used to config iGS02E.

We suggest users to test your configurations on PC first to confirm your settings are correct.

Below shows the gcloud commands for publish and subscribe to verify your settings:
(Your pub/sub topics may be different from the example, please use your settings accordingly)

Publish some data to projects/igs01s-214703/topics/pub

```
$ gcloud pubsub topics publish projects/igs01s-214703/topics/pub --message="TEST1"
```

Then check if you can receive the published data

```
$ gcloud pubsub subscriptions pull --auto-ack projects/igs01s-214703/subscriptions/igs01s --limit=100
```

Configurations on iGS02E

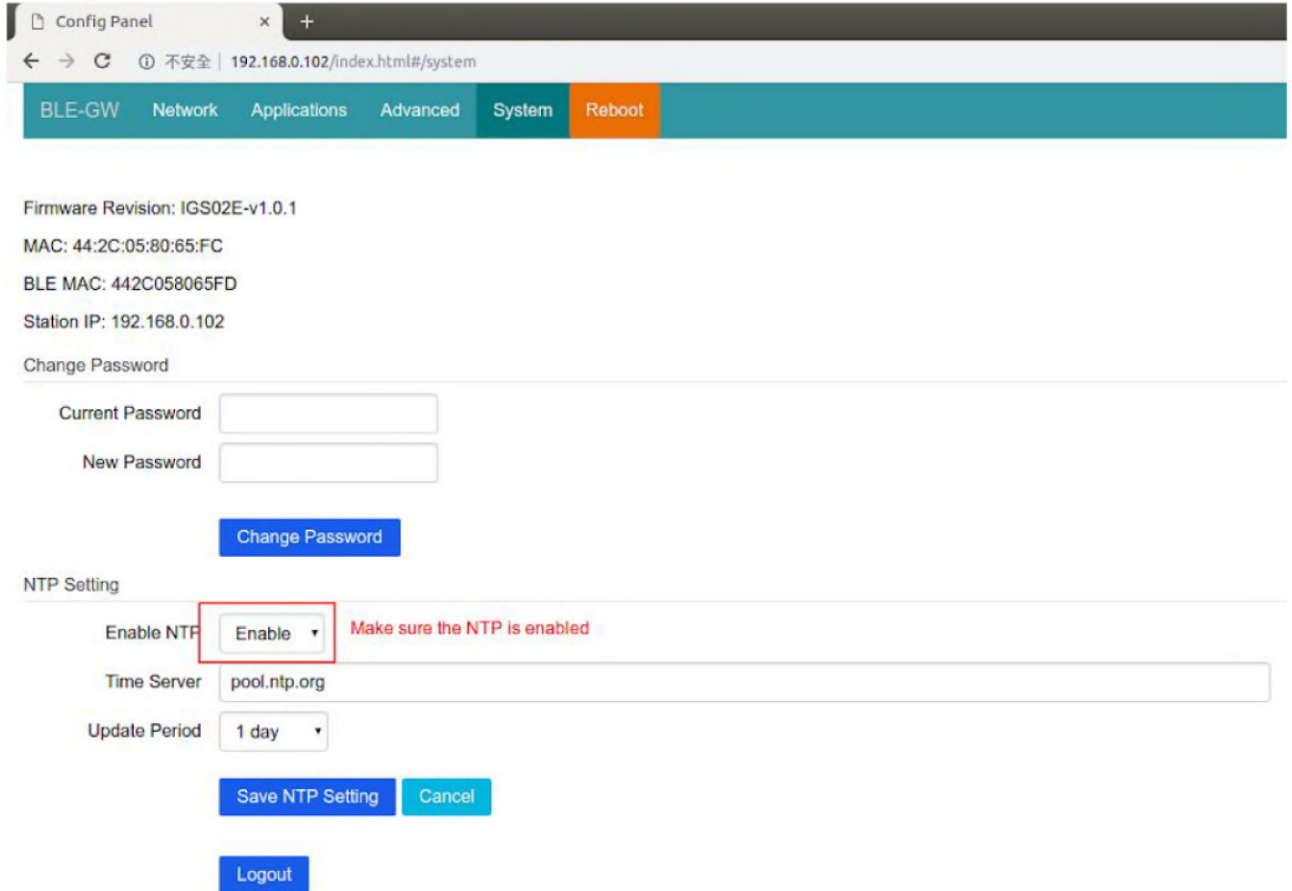
The Google Cloud IoT Core uses JSON Web Tokens (JWT) for authentication. The device uses a private key to sign a JSON Web Token (JWT) for authentication so the user must **upload Private key** to the device. In additional, the JWT requires **enable NTP** setting to get correct expired time.

Below shows the steps to config IGS02E to publish data to Google Cloud IoT Core.

1. Enable NTP via system tab of webUI.
2. Upload private key via advanced table of webUI
3. Configure the device as MQTT client with below settings:
 - MQTT HOST `mqtt.googleapis.com`
 - MQTT PORT `8883`
 - MQTT PUBTOPIC `/devices/{device-id}/events`
 - MQTT CLIENTID
`projects/{project-id}/locations/{cloud-region}/registries/{registry-id}/devices/{device-id}`
 - MQTT USERNAME `unused`
 - MQTT PASSWORD `YOUR-PROJECT-ID`
 - Enable MQTTTS
 - Select Google-Cloud-IoT-Core RootCA
 - Disable use certificate

Note, the standard authenticate method is using JWT in mqtt password field. Due to very short expired time of the JWT, we support runtime generate the JWT. So users need to set "project-id" in the password field then the gateway will automatically generate the JWT as password for connecting server.

Below shows the screenshot of IGS02E settings:



The screenshot shows a web browser window with the address bar displaying "192.168.0.102/index.html#/system". The navigation menu includes "BLE-GW", "Network", "Applications", "Advanced", "System", and "Reboot".

System Information:

- Firmware Revision: IGS02E-v1.0.1
- MAC: 44:2C:05:80:65:FC
- BLE MAC: 442C058065FD
- Station IP: 192.168.0.102

Change Password:

Current Password

New Password

NTP Setting:

Enable NTP *Make sure the NTP is enabled*

Time Server

Update Period

Config Panel x +

← → ↻ ⓘ 不安全 | 192.168.0.102/index.html#/advanced

Payload Mask

Payload Pattern 2

Payload Mask 2

Payload Pattern 3

Payload Mask 3

Whitelist

Whitelist MAC

Advanced Filter

Advanced Filter

Device Key/Certificate Update

未選擇任何檔案

Certificate

Existing Brief

```
-----BEGIN PRIVATE KEY-----  
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCbKwggSiAgEAAo  
IBAQC/krPlaMEVsUSB  
CMD6nw97UBuidp9v1msvQPqlzIRzk5n69M ...
```

未選擇任何檔案 **Upload your private key**

Key

Config Panel x +

← → ↻ 不安全 | 192.168.0.102/index.html#/applications

BLE-GW Network Applications Advanced System Reboot

Application

Application: MQTT Client ▾

Host/IP: mqtt.googleapis.com

Port: 8883

Publish Topic: /devices/igs01s/events */devices/{device-id}/events*

Content Type: plain-text ▾

Client ID: projects/igs01s-214703/k *projects/{project-id}/locations/{cloud-region}/registries/{registry-id}/devices/{device-id}*

Username: username

Password: igs01s-214703 *YOUR-PROJECT-ID*

MQTTS: Enable ▾

Root CA: Google Cloud IoT Core ▾

Use Certificate: Disable ▾

Request Interval (in secs): 5

Drop reports while cache full

Throttle Control (filter out redundant records)

Save Cancel

Revision History

DATE	REVISION	CHANGES
Apr 8, 2019	1	Initial release