

## Getting Started with iGS03M

### [Introduction](#)

### [Device Preparation](#)

#### [1. iGS03M](#)

#### [2. Tag/Beacon](#)

### [Connect to the device](#)

#### [1. Join WiFi AP](#)

#### [2. Access WebUI](#)

### [Collect data using TCP client](#)

#### [1. Configuration](#)

#### [2. Connect to the TCP Server](#)

[For iGS03v2:](#)

[For iGS03v3:](#)

#### [3. Report format](#)

#### [4. Set up Beacon filter](#)

### [Transmit data using MQTT protocol and LTE interface](#)

#### [1. LTE Configuration](#)

#### [2. MQTT Configuration](#)

#### [3. Collect \(subscribe\) data from MQTT broker](#)

[Eclipse Mosquitto™](#)

[MQTTLens](#)

[Ingics DemoBoard APP](#)

### [Troubleshooting](#)

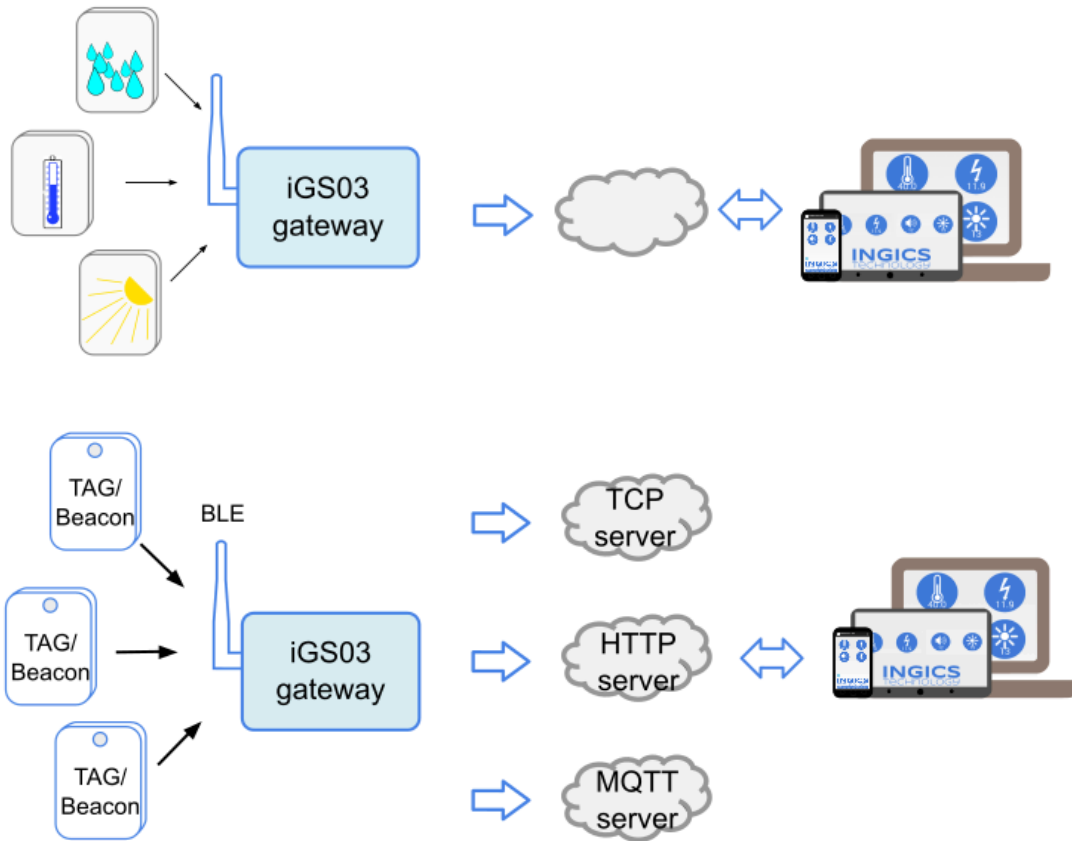
#### [LTE](#)

### [Revision History](#)

# INGICS TECHNOLOGY

## Introduction

The iGS03M device is a BLE-Gateway used to receive BLE advertising data from BLE beacons, tags and sensors, and then send the data to the network server through LTE interface.



In this tutorial, you will learn about how to:

1. set up and configure the iGS03M
2. collect data with a simple TCP client
3. identify the data you want, and set up the beacon filter
4. publish and collect data with a public MQTT broker using LTE interface

## Device Preparation

### 1. iGS03M

To power on the iGS03M, connect the device to the power supply with the USB-C connector. The iGS03M requires a 5V, 2A throughput adapter.







You have to put a **LTE Cat-M1** micro SIM card into the socket of iGS03M to use LTE. Please follow below steps to open the bottom cover and insert the SIM card.

Step 1. Remove external BLE antenna



# INGICS TECHNOLOGY

<p>Step 2. Remove the screw from bottom cover</p>	
<p>Step 3. Use finger to press and hold the arrow part</p>	
<p>Step 4. Pull out the bottom cover</p>	
<p>Step 5. Insert Micro SIM in to the socket</p>	

## 2. Tag/Beacon

To test the BLE-Gateway functionality, it will require a tag/beacon like Apple iBeacon, Eddystone or any other BLE tag/beacon as the BLE data source. We use an Ingics iBS03T (temperature sensor beacon) as the example in this guide. Follow the beacon's user guide to power on and configure (if

required) the beacon.

## Connect to the device

### 1. Join WiFi AP

The iGS03M has a WiFi device that works as a WiFi access point (AP) by default. After it is powered on, you can scan its native AP and connect it with the WiFi of your NB, PC, Mac, Tablet or Smartphone. Its SSID is just like the below figure with part of the mac address.

For iGS03 v2 firmware, The default key to connect with it is "12345678".

For iGS03 v3 firmware, The default key to connect with it is "apXYZWXYZW" (XYZW is the last 4 digits of wifi MAC).

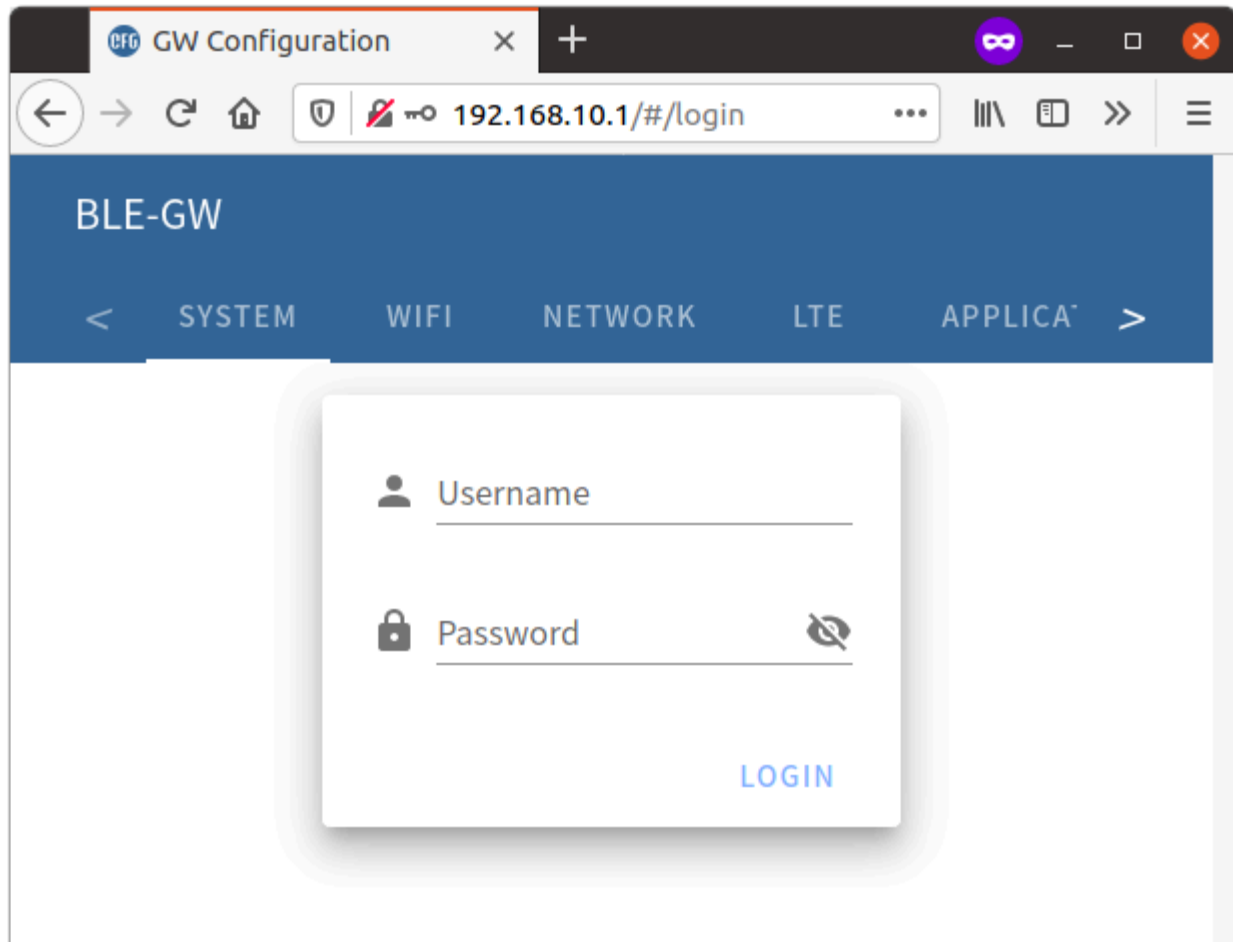


### 2. Access WebUI

After connection, the iGS03M IP address will be 192.168.10.1 by default. You can connect to the IP address on your web browser to access the Web UI. The default account/password are both "admin".

For iGS03 v2 firmware, Connect to webUI by <http://192.168.10.1>

For iGS03 v3 firmware, Connect to webUI by <https://192.168.10.1>

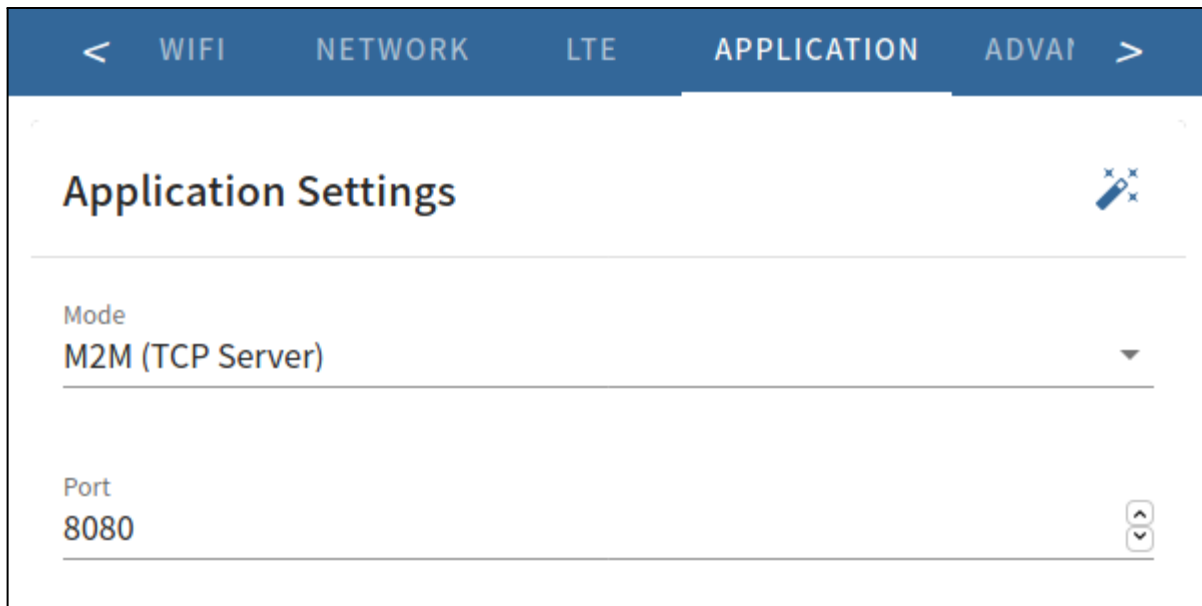


## Collect data using TCP client

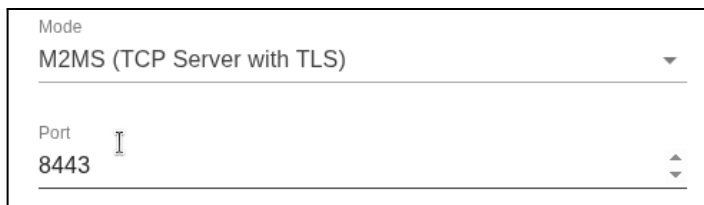
### 1. Configuration

After login to the Web UI, switch to the “APPLICATION” tab. The application means the gateway how-to report data.

For iGS03v2, the default setting is “**M2M (TCP Server)**”, it means the gateway works as a TCP server, you can connect to the server, and the gateway will send the received BLE data to the connection.



For iGS03v3, the default setting is “M2MS (TLS Server)”, it means the gateway works as a TLS server, you can connect to the server, and the gateway will send the received BLE data to the connection.



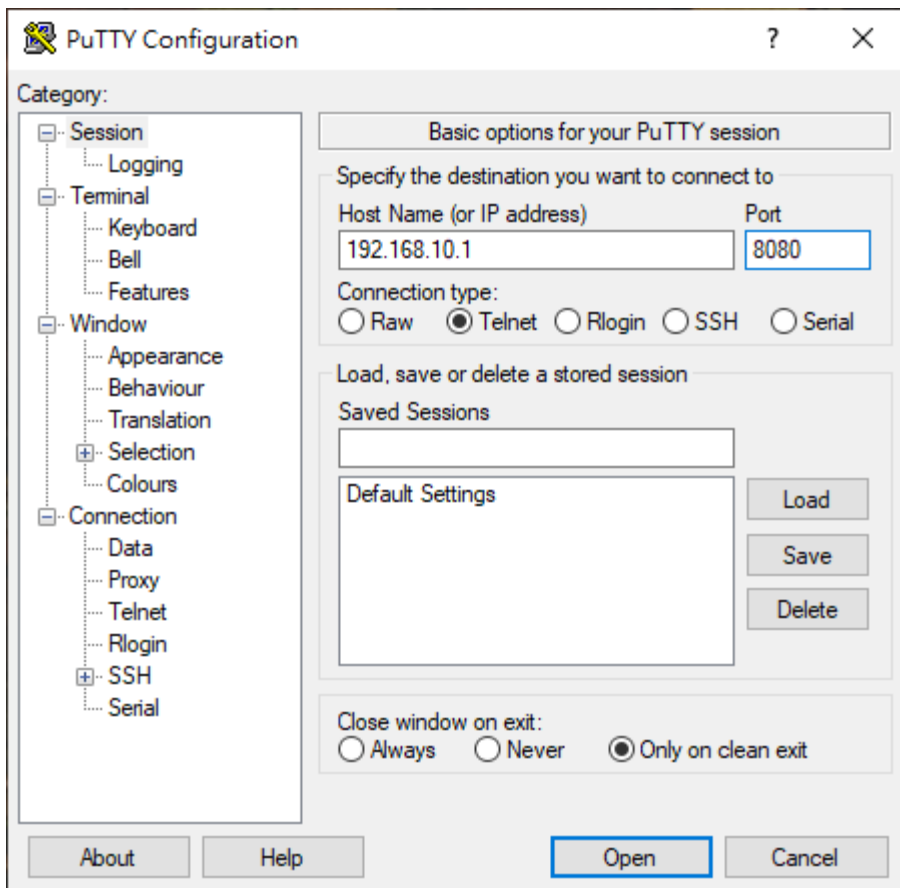
This mode is normally used for testing only. You can check if the gateway can receive the BLE data as your expectation without any server set up and configuration.

## 2. Connect to the TCP Server

For iGS03v2:

As described above, the default setting is running a TCP server at port 8080. You can choose your favorite telnet client to connect to **192.168.10.1** port **8080**.

For example, using PuTTY (<https://www.putty.org/>):



```
192.168.10.1 - PuTTY
$GPRP,C575C4346A1F,F008D1798C5C,-81,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-83,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-82,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-82,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-82,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-63,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-63,02010603031FFE05FF87000BA0
$GPRP,4013F73A4819,F008D1798C5C,-83,02011A020A0C0BFF4C001006511ECD767E90
$GPRP,C575C4346A1F,F008D1798C5C,-64,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-81,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-82,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-82,02010603031FFE05FF87000BA0
$GPRP,0C61CFC14A4E,F008D1798C5C,-29,02010612FF0D0083BC280100B409FFFF000015040000
$GPRP,C575C4346A1F,F008D1798C5C,-81,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-84,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-83,02010603031FFE05FF87000BA0
$GPRP,E0774052C292,F008D1798C5C,-77,0201061BFF570102FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
$GPRP,C575C4346A1F,F008D1798C5C,-83,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-83,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-63,02010603031FFE05FF87000BA0
$GPRP,4013F73A4819,F008D1798C5C,-83,02011A020A0C0BFF4C001006511ECD767E90
$GPRP,C575C4346A1F,F008D1798C5C,-63,02010603031FFE05FF87000BA0
$GPRP,C575C4346A1F,F008D1798C5C,-64,02010603031FFE05FF87000BA0
```

For iGS03v3:

The default setting is running a TLS server at port 8443.

To connect the M2MS (TCP server with TLS) mode on Windows:

1. Install the nmap tool via <https://nmap.org/download.html#windows>
2. Open Windows PowerShell and execute the below command to connect iGS03 M2MS.

```
ncat.exe --ssl 192.168.10.1 8443
```

After connecting, you should see the iGS03M keep reporting data to the connection as in the figure below. That means the iGS03M is working, it can receive the BLE advertisements and report them following the “APPLICATION” settings.

### 3. Report format

Each line reported by iGS03M present one BLE advertisement it received, and the format is:

```
$<report type>,<tag id>,<gateway id>,<rssi>,<raw packet content>,*<unix epoch timestamp>\r\n
```

<report type>	Different report types to distinguish the source of the report.
<tag id>	MAC address or ID of tag/beacon
<gateway id>	MAC address of gateway
<rssi>	RSSI of tag/beacon
<raw packet content>	Raw packet received by the gateway in HEX string
<unix epoch timestamp>	Optional timestamp configured in applications page

You can use the <tag id> to identify the data sent by your beacon. And you can parse the <raw packet content> according to the beacon's payload format.

Below is the example belonging to the test iBS03T beacon. We marked the battery voltage in **green** (2.96V) and temperature in **blue** (25.26) here for easy understanding. You can refer to the iBS03 payload format document for detail.

`$GPRP,0C61CFC14A4E,F008D1798C5C,-29,02010612FF0D0083BC280100D109FFFF000015040000`

And here is an example for Apple iBeacon. You can find the iBeacon payload format on the internet, the UUID marked as **yellow**, the major number in **blue**, and the minor number in **green**.

`$GPRP,0081F985FE8C,F008D1798C5C,-68,0201061AFF4C000215B9A5D27D56CC4E3AAB511F2153  
BCB9670001FE8CD6`

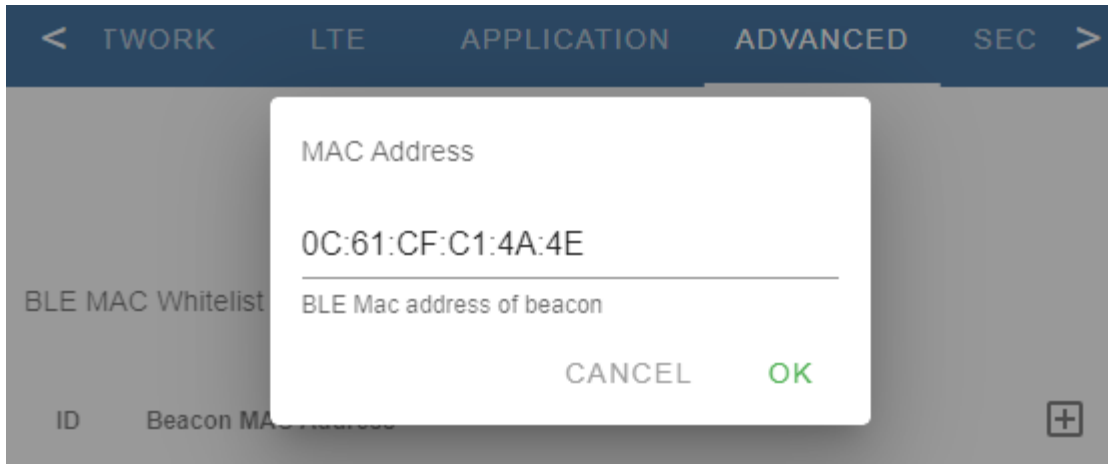
#### 4. Set up Beacon filter

You may notice that there are many BLE devices that broadcast data, and that cause it's hard to identify which data belongs to your beacon. And further, if iGS03M reports those data to the server using the LTE interface, that may take a lot of data usage of your LTE account. So, we strongly recommend that you set up the beacon filter before enabling LTE transmission.

In this example, the test beacon's BLE MAC address is 0C:61:CF:C1:4A:4E. Normally you can find the

# INGICS TECHNOLOGY

MAC address on your beacon's label. Open the Web UI, switch to "ADVANCED" tab, scroll down to the "BLE White List", add a new entry and fill the MAC address.



You need to "SAVE" the change and reboot the device to let the filter setting take effect.

Require reboot for the changes to take effect.

**REBOOT**

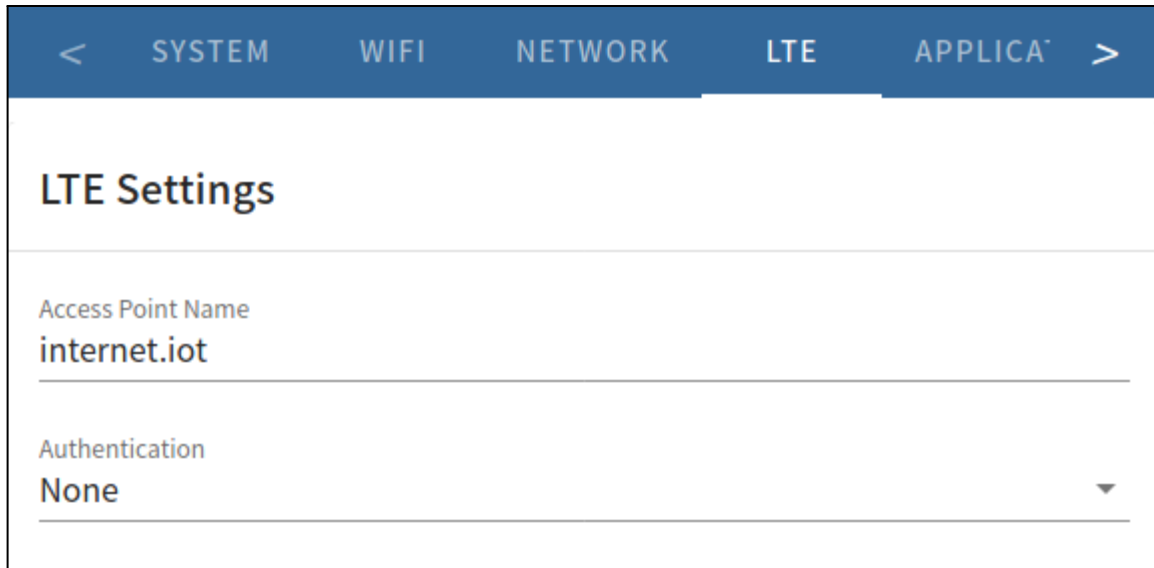
Then you can telnet to the device again, you should see only the data belonging to this beacon will be reported.

A screenshot of a PuTTY terminal window titled '192.168.10.1 - PuTTY'. The terminal displays a series of lines of data, all starting with '\$GPRP, 0C61CFC14A4E, F008D1798C5C, -'. The data is filtered to show only information related to the beacon with MAC address 0C61CFC14A4E. The lines include various parameters such as signal strength (-28, -37, -39, -34, -49, -43, -42, -41, -40, -36, -54) and other identifiers like '02010612FF0D0083BC280102EE09FFFF000015040000' and '02010612FF0D0083BC280102F509FFFF000015040000'. The terminal window has standard window controls (minimize, maximize, close) in the top right corner.

## Transmit data using MQTT protocol and LTE interface

### 1. LTE Configuration

Switch to 'LTE' tab in Web UI. Fill the APN and authentication (if required) follows the carrier's requirement.



The screenshot shows the 'LTE Settings' page in a web interface. The top navigation bar is blue and contains the following tabs: SYSTEM, WIFI, NETWORK, LTE (which is the active tab), and APPLICA. Below the navigation bar, the page title is 'LTE Settings'. There are two main configuration fields: 'Access Point Name' with the value 'internet.iot' and 'Authentication' with the value 'None'. A dropdown arrow is visible next to the 'None' value.

SAVE the configuration and reboot the device, the network status LED should be light on GREEN. And you can check the LTE information in the "SYSTEM" tab of Web UI.

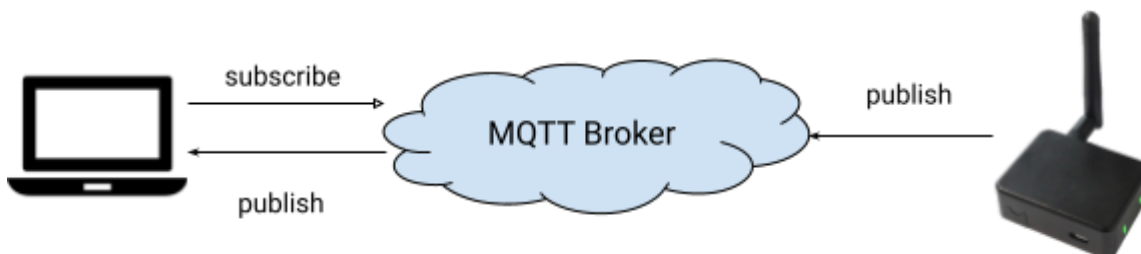
## LTE Information

SIM:	SIM_READY
IMEI:	862785044122532
IMSI:	466924700041387
IP Address:	10.172.78.230
Network Status:	Registered, Chunghwa Telecom, LTE BAND 3, CAT-M1, RUNNING

Please make sure the network status is registered, and the IP address is obtained. If the SIM status is not READY or IMSI is empty, which means the device fails to access the SIM card, please re-install the SIM card and test again. If still having issues, please follow the [LTE troubleshooting](#) section to collect related logs and information and report to us.

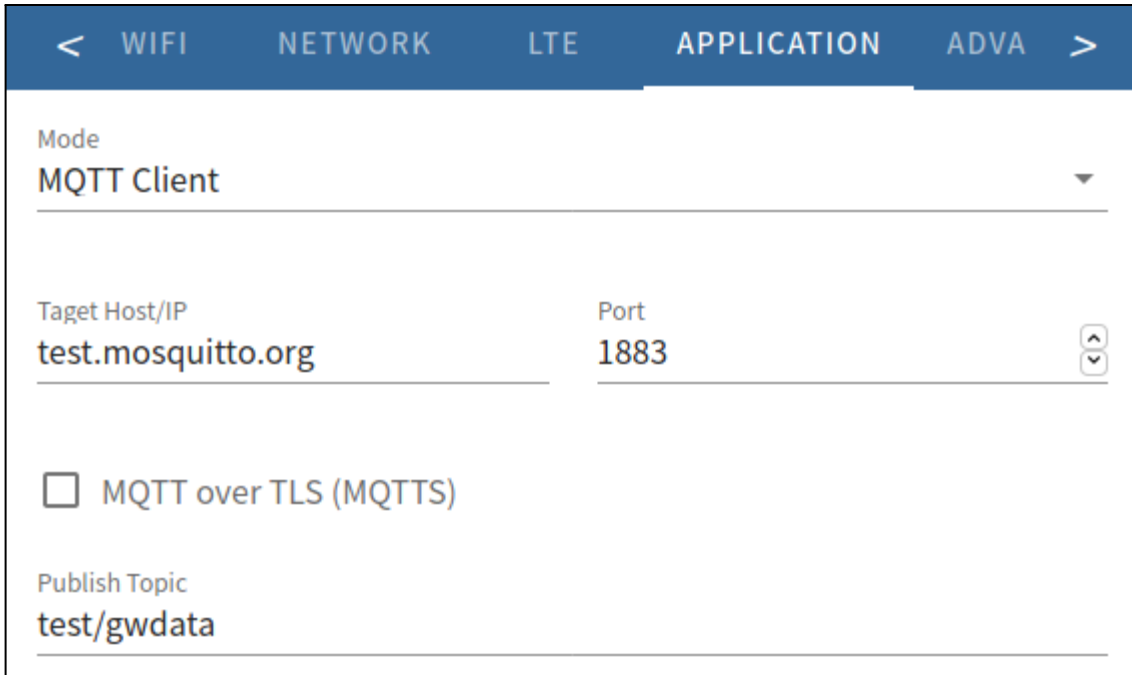
## 2. MQTT Configuration

Message Queuing Telemetry Transport is a message protocol for restricted networks (low bandwidth) and IoT devices with extremely high latency. The iGS03M device will publish data to the MQTT broker, and you can use any MQTT subscriber to subscribe and receive the data from the broker.



In "APPLICATION" tab, set up as below

- Mode: MQTT Client
- Target Host/IP: test.mosquitto.org
- Port: 1883
- Publish Topic: test/gwdata



The “test.mosquitto.org” hosts a publicly available MQTT broker for testing. You should set up your own MQTT broker for your application. We did not invoke the broker set up in this guide, just focus on how-to set up and use MQTT transmission on iGS03M.

Again, to avoid any unexpected data usage on LTE transmission, please scrolling down in the “APPLICATION” tab to the bottom, follow below settings:

- Enable message throttling
- Setup request interval to 60 seconds
- Discard new input data when cache full

Append Timestamp None	<input checked="" type="checkbox"/> Message Throttling
Request Interval 60 seconds	Cache Full Handling Discard new input data

This configuration means the iGS03M publishes only one record for each beacon every 60 seconds.

### 3. Collect (subscribe) data from MQTT broker

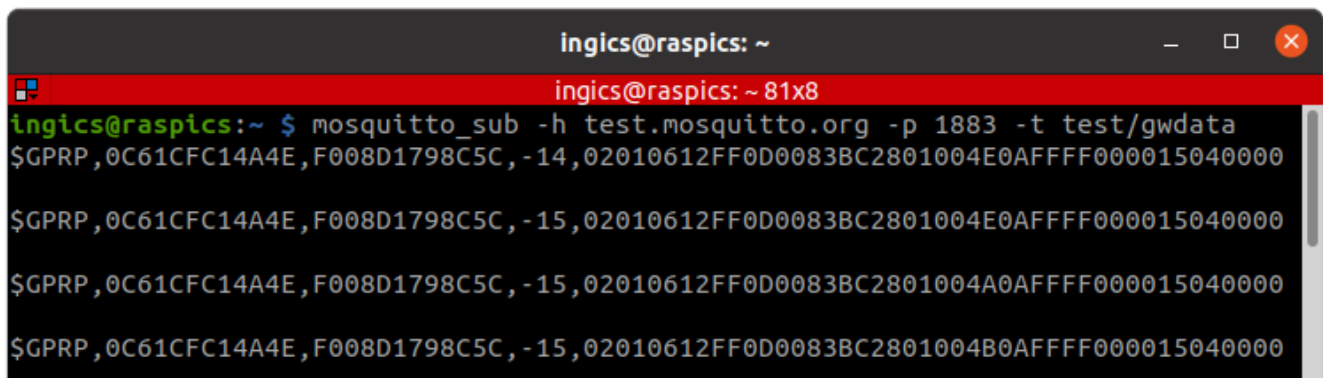
Now we can use a MQTT subscriber to verify the data published by iGS03M. We list three examples here for you to reference.

#### Eclipse Mosquitto™

Eclipse Mosquitto is a project with full function implements of the MQTT protocol. It supports various platforms and operating systems. It includes a MQTT command line client “mosquitto\_sub”, and we can use it to verify the MQTT data. You can download the installer in below URL

<https://mosquitto.org/download/>

```
$ mosquitto_sub -h test.mosquitto.org -p 1883 -t test/gwdata
```



```
ingics@raspics: ~  
ingics@raspics: ~ 81x8  
ingics@raspics:~ $ mosquitto_sub -h test.mosquitto.org -p 1883 -t test/gwdata  
$GPRP,0C61CFC14A4E,F008D1798C5C,-14,02010612FF0D0083BC2801004E0AFFFFF000015040000  
$GPRP,0C61CFC14A4E,F008D1798C5C,-15,02010612FF0D0083BC2801004E0AFFFFF000015040000  
$GPRP,0C61CFC14A4E,F008D1798C5C,-15,02010612FF0D0083BC2801004A0AFFFFF000015040000  
$GPRP,0C61CFC14A4E,F008D1798C5C,-15,02010612FF0D0083BC2801004B0AFFFFF000015040000
```

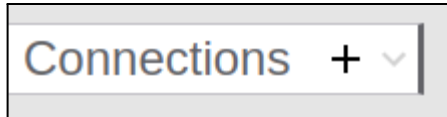
#### MQTTLens

# INGICS TECHNOLOGY

MQTTLens is a Google Chrome Application, you can install and launch it by below URL

<https://chrome.google.com/webstore/detail/mqttlens/hemojaaeigabkbcookmlqmdigohbjm>

Click the “plus” button to create a new connection.



Fill the connect configuration

- Hostname: test.mosquitto.org
- Port: 1883

### Connection Details

<b>Connection name</b>	<input type="text" value="test.mosquitto.org"/>	<b>Connection color scheme</b>	<input type="color" value="#76b82a"/>
<b>Hostname</b>	<input type="text" value="tcp://"/> <input type="text" value="test.mosquitto.org"/>	<b>Port</b>	<input type="text" value="1883"/>
<b>Client ID</b>	<input type="text" value="lens_46ydWAGWeUp1yGROZC2JMJCnZbl"/>	<input type="button" value="Generate a random ID"/>	
<b>Session</b>	<input checked="" type="checkbox"/> Clean Session	<b>Automatic Connection</b>	<input checked="" type="checkbox"/> Automatic Connection
		<b>Keep Alive</b>	<input type="text" value="120"/> seconds


After connection created, fill the subscribe topic: test/igs03m

Connection: test.mosquitto.org

### Subscribe

# INGICS TECHNOLOGY

Click the "SUBSCRIBE" button, you should see the published data like below.



The screenshot shows a MQTT client interface with a light green background. At the top, it says "Subscriptions". Below that, it displays "Topic: 'test/gwdata'" with "Showing the last 5 messages" and a trash icon. To the right, it says "Messages: 0/21" with an upward arrow. The interface lists two messages:

#	Time	Topic	QoS	Info
16	2:02:35	test/gwdata	0	i
Message: \$GPRP,0C61CFC14A4E,F008D1798C5C,-15,02010612FF0D0083BC280100480AFF000015040000				
17	2:02:45	test/gwdata	0	i
Message: \$GPRP,0C61CFC14A4E,F008D1798C5C,-14,02010612FF0D0083BC280100460AFF000015040000				

## Ingics DemoBoard APP

It is an application for Ingics Beacon Gateway testing & demonstration. The application includes a payload parser for most of INGICS BLE beacons and some public payload format (for example, iBeacon, Eddystone, ..., etc). You can download the latest release in the URL below.

<https://github.com/ingics/ingics-blegw-demoboard/releases>

Click the "plus" button to create a new Gateway.

- Protocol: MQTT
- Host: test.mosquitto.org
- Port: 1883
- Subscribe Topic: test/gwdata

### Gateway Configuration

Protocol	MQTT
Name	test.mosquitto.org
Host	test.mosquitto.org
Port	1883
Subscribe Topic	test/gwdata

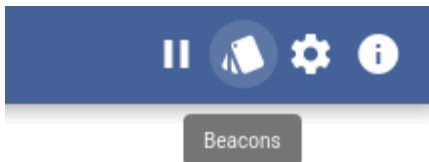
CANCEL SAVE

Click the gateway card to start receiving data.


### Logs

11/26/2020 2:04:35 PM	\$GPRP, 0C61CFC14A4E, F008D1798C5C, -15, 02010612FF0D0083BC280100440AFFFF000015040000
11/26/2020 2:04:30 PM	\$GPRP, 0C61CFC14A4E, F008D1798C5C, -14, 02010612FF0D0083BC280100440AFFFF000015040000
11/26/2020 2:04:25 PM	\$GPRP, 0C61CFC14A4E, F008D1798C5C, -15, 02010612FF0D0083BC280100420AFFFF000015040000

Click the “Beacon” icon on the right of the header to switch to the beacon view.



The beacon view will display the newest status (parsed from latest log) of beacon.

	MAC	Last Update	RSSI
 IBS03T	0C61CFC14A4E	11/26/2020 2:09:00 PM	-14 
Advertisement: 02010612FF0D0083BC280100370AFFFF000015040000 Flags: 0x6 Company: Ingics Type: IBS03T Battery: 2.96V Temperature: 26.15°C Events: 0x0 Manufacture Data: 0D0083BC280100370AFFFF000015040000			

## Troubleshooting

### LTE

For LTE-M connection related issue, please provide us below information:

- Which carrier is used
- Which SIM card is used (APN/auth/user/password settings). Please check with the carrier to make sure the SIM card is LTE-M capable.
- Telnet to device IP and get below command output
  - SYS
  - LTE INFO
  - LTE LOG

## Revision History

DATE	REVISION	CHANGES
Nov 26, 2020	1	Initial release

# INGICS TECHNOLOGY