

IDG450_0GT0C

Getting Started Guide for AWS IoT Core

Table of Contents

1	<i>Document information</i>	1
2	<i>Overview</i>	1
3	<i>Hardware description</i>	2
4	<i>Set up your development environment</i>	2
5	<i>Set up device hardware</i>	3
6	<i>Setup your AWS account and permissions</i>	5
7	<i>Create resources in AWS IoT</i>	5
8	<i>Provision the device with credentials</i>	5
9	<i>Run the demo</i>	6
10	<i>Verify messages in AWS IoT Core</i>	6
11	<i>Troubleshooting</i>	8

1 Document information

1.1 Document revision history

Version	Date	Author	Description
1.0	2023-07-03	Mike Wang	Creation of the document

1.2 Applicable operating systems for this guide

The operation system is already on AMIT device. AMIT uses Quecopen platform to develop devices.

2 Overview

With this AMIT 5G Modem, IDG450-0GT0C, you have made a great first step in the world of connected Internet of things (IoT) by simply inserting a SIM card from the local mobile carrier into this device to get things connected. This section gives you all the information you need to set up your device.

Main Features:

- Provide 5G WAN connection and is back compatible with 3G/4G.
- Provide one 2.5 Gigabit Ethernet ports for the LAN connection.

- Instinctive Web GUI is used for basic setting and check the cellular status.
- Designed easy-to-mount metal body for business and M2M environment to work with a variety M2M (Machine-to-Machine) applications.

3 Hardware description

3.1 Datasheet

<https://www.amitwireless.com/upload/products/download/Datasheet-Global-IDG450-0GT0C-20230817.pdf>

3.2 Standard kit contents

Device: IDG450-0GT0C 5G Modem*1

Cellular Antenna*4

RJ45 Cable*1

2 Pin Terminal Block*1

12V1A power adaptor*1(optional)

https://www.amitwireless.com/productDetail.php?cate=1009&product_id=1287&pid=1274&pid1=136&pid2=124

3.3 User provided items

None

3.4 3rd party purchasable items

None

4 Set up your development environment

4.1 Tools installation (IDEs, Toolchains, SDKs)

Links to any download pages, user guides, or other documentation for each item.

1. IDE base
<https://github.com/aws-labs/aws-iot-device-client>
2. CLI based
Make/ CMake
3. toolchains and other software utilities : putty/ wget
4. compiler options for mandatory optimizations : Makefile

```

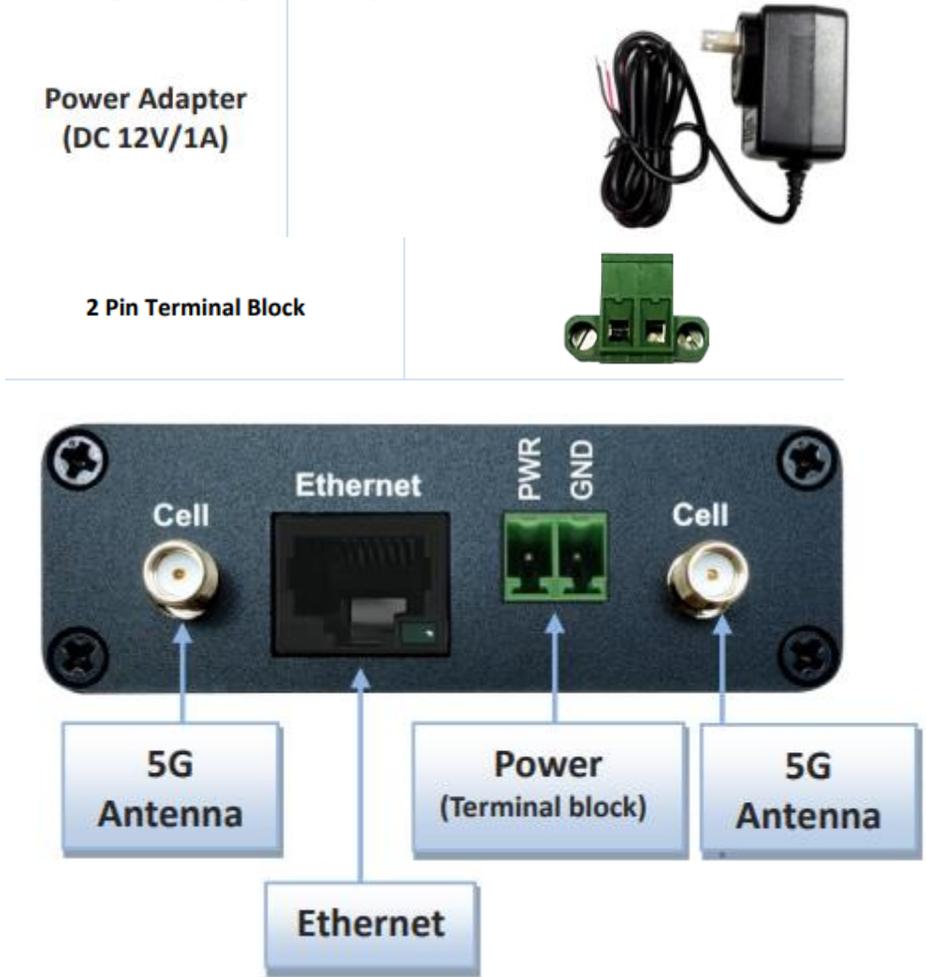
client :
  cd $(CLIENT_SRC_BUILD) \
  && $(SUDO) cmake -E env CFLAGS="$(FILTERED_CFLAGS)"
  CXXFLAGS="$(FILTERED_CFLAGS)" LDFLAGS="$(LDFLAGS)" \
  cmake .. -DCMAKE_SYSROOT=$(QL_SYSROOT_DIR) -DCMAKE_C_COMPILER=
  $(CC) -DCMAKE_CXX_COMPILER=$(CXX) -DCMAKE_AR=$(AR) -
  DCMAKE_SYSTEM_PROCESSOR="arm" -DCMAKE_PREFIX_PATH=
  $(WORKDIR)/amit_libs -DUSE_OPENSSL=ON -DBUILD_SHARED_LIBS=OFF -
  DCMAKE_VERBOSE_MAKEFILE:BOOL=ON \
  && $(SUDO) cmake --build . --target aws-iot-device-client

```

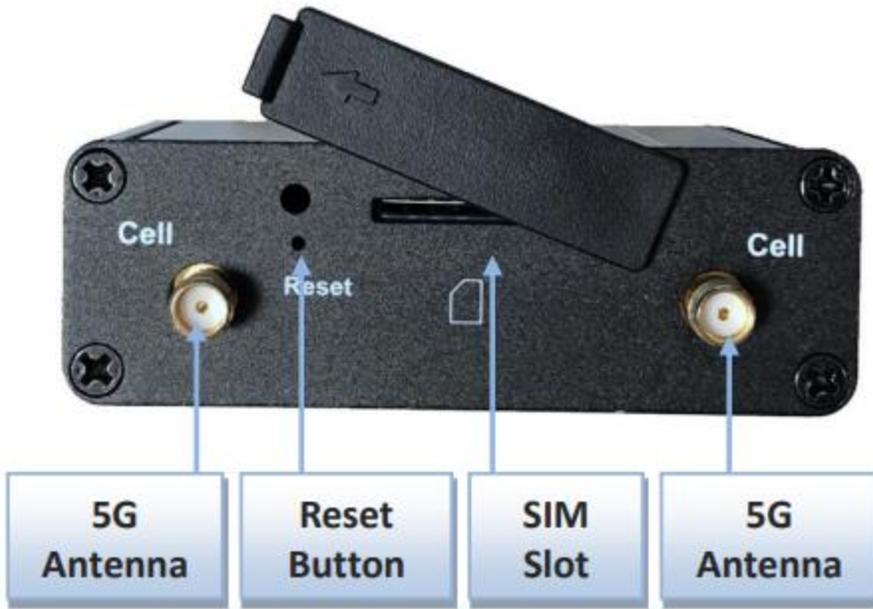
5. None. Device already contains the developed firmware from SDK.

5 Set up device hardware

Provide power by power adaptor with terminal block



Power input: TB 5-32V DC



LED Indication



Indication	LEDColor	Description
 Power	Blue	Steady On: Device power is on Off: Device power is off
 Cell/PDP	Blue Red	Red and Steady On: Cellular is not registered to network. Red and Flash: Cellular is registering to network. Blue and Steady On: The device registers to 5G network. Blue and Slow Flash: The device registers to LTE or 3G network.
 Signal Strength	Blue	Blue and Steady On: Cellular signal is good. Blue and Slow Flash: Cellular signal is weak. OFF: No cellular signal.

UM:

https://www.amitwireless.com/upload/products/download/UM_IDG450-0GT0C-20230502.pdf

6 Setup your AWS account and permissions

If you do not have an existing AWS account and user, refer to the online AWS documentation at [Set up your AWS Account](#). To get started, follow the steps outlined in the sections below:

- [Sign up for an AWS account](#)
- [Create an administrative user](#)
- [Open the AWS IoT console](#)

Pay special attention to the Notes.

7 Create resources in AWS IoT

Refer to the online AWS documentation at [Create AWS IoT Resources](#). Follow the steps outlined in these sections to provision resources for your device:

- [Create an AWS IoT Policy](#)
- [Create a thing object](#)

Pay special attention to the Notes.

8 Provision the device with credentials

Device support wget, use it to transmit credential files into the device.

```

root@sdxlemur:/tmp# mkdir certs
root@sdxlemur:/tmp# cd certs/
root@sdxlemur:/tmp/certs# wget http://192.168.123.8/certificate.pem.crt
--2023-09-12 08:15:26-- http://192.168.123.8/certificate.pem.crt
Connecting to 192.168.123.8:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1224 (1.2K) [application/octet-stream]
Saving to: 'certificate.pem.crt'

certificate.pem.crt  100%[=====>]  1.20K  --.-KB/s  in 0.001s

2023-09-12 08:15:26 (1.37 MB/s) - 'certificate.pem.crt' saved [1224/1224]

root@sdxlemur:/tmp/certs# wget http://192.168.123.8/private.pem.key
--2023-09-12 08:15:32-- http://192.168.123.8/private.pem.key
Connecting to 192.168.123.8:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1675 (1.6K) [application/octet-stream]
Saving to: 'private.pem.key'

private.pem.key  100%[=====>]  1.64K  --.-KB/s  in 0.001s

2023-09-12 08:15:32 (1.08 MB/s) - 'private.pem.key' saved [1675/1675]

```

Use `chmod` to change directory and file permission for `aws-iot-device-client` to work.

```

chmod 700 /tmp/certs
chmod 600 /tmp/certs/private.pem.key
chmod 644 /tmp/certs/certificate.pem.crt

```

9 Run the demo

We ported `aws-iot-device-client` onto the device, provide access to AWS IoT Core, AWS IoT Device Management, and AWS IoT Device Defender features.

Run `aws-iot-device-client` with credential files and custom configure to connect with AWS IoT Core.

```

aws-iot-device-client --endpoint YOUR_ENDPOINT --cert
YOUR_CERTIFICATION_FILE --key YOUR_PRIVATE_KEY_FILE --thing-name
YOUR_THING_NAME --enable-pub-sub true --publish-topic YOUR_PUB_TOPIC --
publish-file YOUR_PUB_FILE --subscribe-topic YOUR_SUB_TOPIC --subscribe-file
YOUR_SUB_FILE

```

10 Verify messages in AWS IoT Core

Configure `aws-iot-device-client` with following content.

```

./aws-iot-device-client --endpoint a23v0kc331fyxj-ats.iot.us-east-1.amazonaws.com --
cert ./certs/certificate.pem.crt --key ./certs/private.pem.key --thing-name
IDG450_0GT0C --enable-pub-sub true --publish-topic IDG450/pubtest --publish-
file ./subpub/pub --subscribe-topic IDG450/subtest --subscribe-file ./subpub/sub

```

Specify file subpub/pub as publish MQTT topic file subscribing topic IDG450/pubtest , with content below

```
root@sdxlemur:/tmp/subpub# cat pub
{"IDG450_0GT0C": "Hello AWS IoT Core!"}
```

And then can see the MQTT test client subscriber received the message that device published.

The screenshot shows the MQTT console interface. At the top, there are two tabs: "Subscribe to a topic" (active) and "Publish to a topic". Under "Subscribe to a topic", there is a "Topic filter" section with an "Info" icon. Below it, a text box contains "#". A "Subscribe" button is visible. Below the "Publish to a topic" section, there is a "Subscriptions" table with columns for topic name and actions (Pause, Clear, Export, Edit). The table shows a subscription for "IDG450/pubtest" with a timestamp of "September 13, 2023, 10:47:10 (UTC+0800)". Below the table, a message is displayed: {"IDG450_0GT0C": "Hello AWS IoT Core!"}. A "Properties" link is also visible.

Specify file subpub/sub as subscribe MQTT topic file subscribing topic IDG450/subtest

Subscribe to a topic
Publish to a topic

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

Q IDG450/subtest X

Message payload

```
{
  "message": "Hello from AWS IoT console"
}
```

▶ Additional configuration

Publish

Subscriptions IDG450/subtest Pause Clear Export Edit

IDG450/subtest ♥ X

Message payload

```
{
  "message": "Hello from AWS IoT console"
}
```

▶ Additional configuration

Publish

Publish a message to topic IDG450/subtest through MQTT test client publisher and can see the device received a message in the subscribe file.

```
2023-09-12T08:52:14.228Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on subscribe
topic, size: 45 bytes
root@sdxlemur:/tmp/subpub# cat sub
{
  "message": "Hello from AWS IoT console"
}root@sdxlemur:/tmp/subpub#
```

11 Troubleshooting

If you need troubleshooting tips for resolving common or potential problems with your device, please contact service@amit.com.tw or sales@amit.com.tw.

For more information, refer to the AWS online documentation on Troubleshooting AWS IoT.