

# BG96\_MQTT\_Attach\_AWS\_IoTCore\_Application\_Note

## LPWA Module Series

### Contents

<b>BG96_MQTT_Attach_AWS_IoTCore_Application_Note .....</b>	<b>1</b>
<b>1. Module MQTTS attach AWS IoT Core.....</b>	<b>2</b>
1.1 Get the certificates .....	2
1.2 Connection address and port.....	4
1.3 Module load certificates and use the MQTTS connect AWS IoT Core: .....	5
1.4 Use the Device Shadow service .....	6
<b>2. Detailed AT command steps for module connection to AWS IoT core. ....</b>	<b>7</b>
2.1 Configure the network .....	7
2.2 Load certificates .....	8
2.3 Active PDP context .....	10
2.4 Configure SSL option .....	11
2.5 Configure MQTT option .....	12
2.6 MQTT connect and data interaction .....	12
<b>Support band list.....</b>	<b>13</b>

# 1. Module MQTTS attach AWS IoT Core.

AWS IoT Core supports TLS client and server certificates authentication, with the MQTT protocol as a message broker. Module import certificates, then it can be used to connect the AWS IoT Core by MQTTS.

Connection step as below:

## 1.1 Get the certificates

Sign in to the AWS IoT platform and register your device in the registry. Certificates will be created in the enrolled device. Note that these certificates should be downloaded to the local computer. These certificates will be imported into the module later. On the AWS website, have details about how to register the device, refer to the link: [https://docs.aws.amazon.com/iot/?id=docs\\_gateway](https://docs.aws.amazon.com/iot/?id=docs_gateway)

Please save the certificates which the red box note in the below, this will load into the module, the certificates need to be active, then attach policy and attach thing.

**Certificate created!**

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

**In order to connect a device, you need to download the following:**

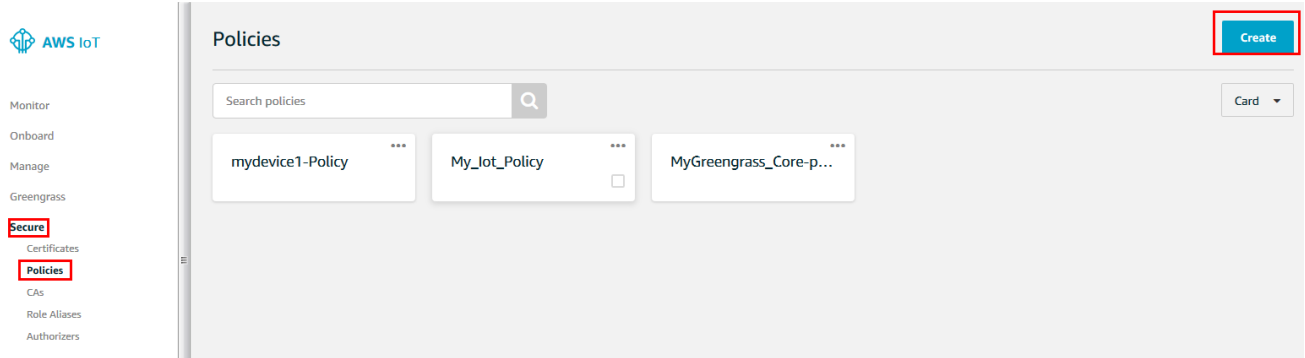
A certificate for this thing	facf1ae17f.cert.pem	<a href="#">Download</a>
A public key	facf1ae17f.public.key	<a href="#">Download</a>
A private key	facf1ae17f.private.key	<a href="#">Download</a>

**You also need to download a root CA for AWS IoT:**

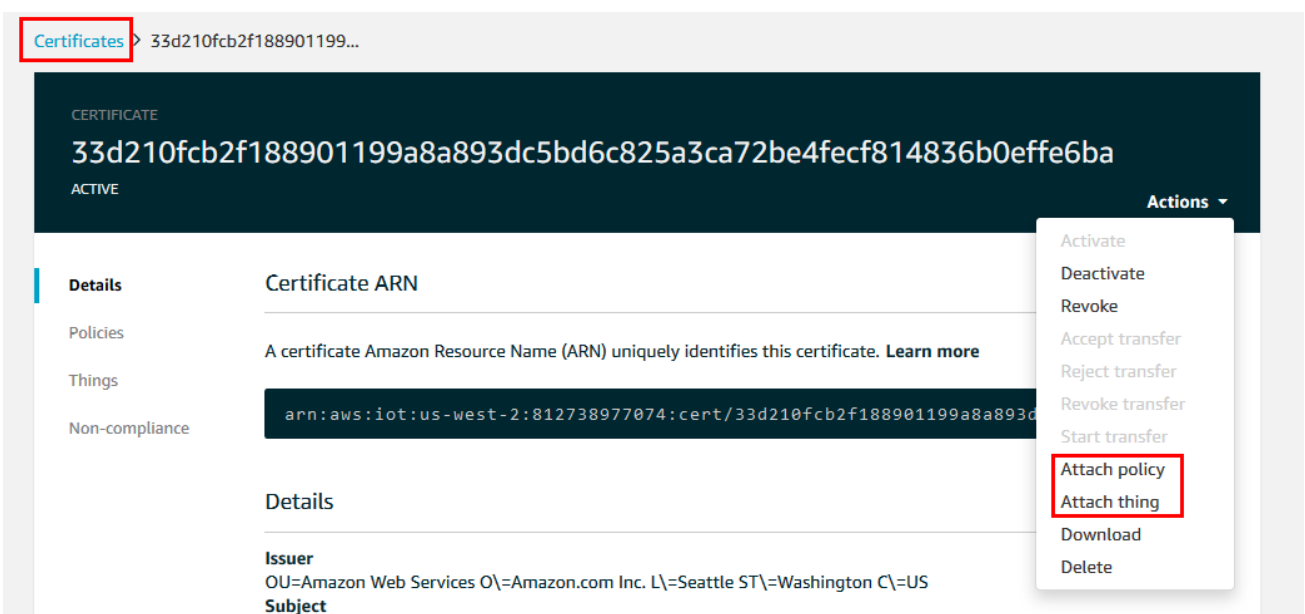
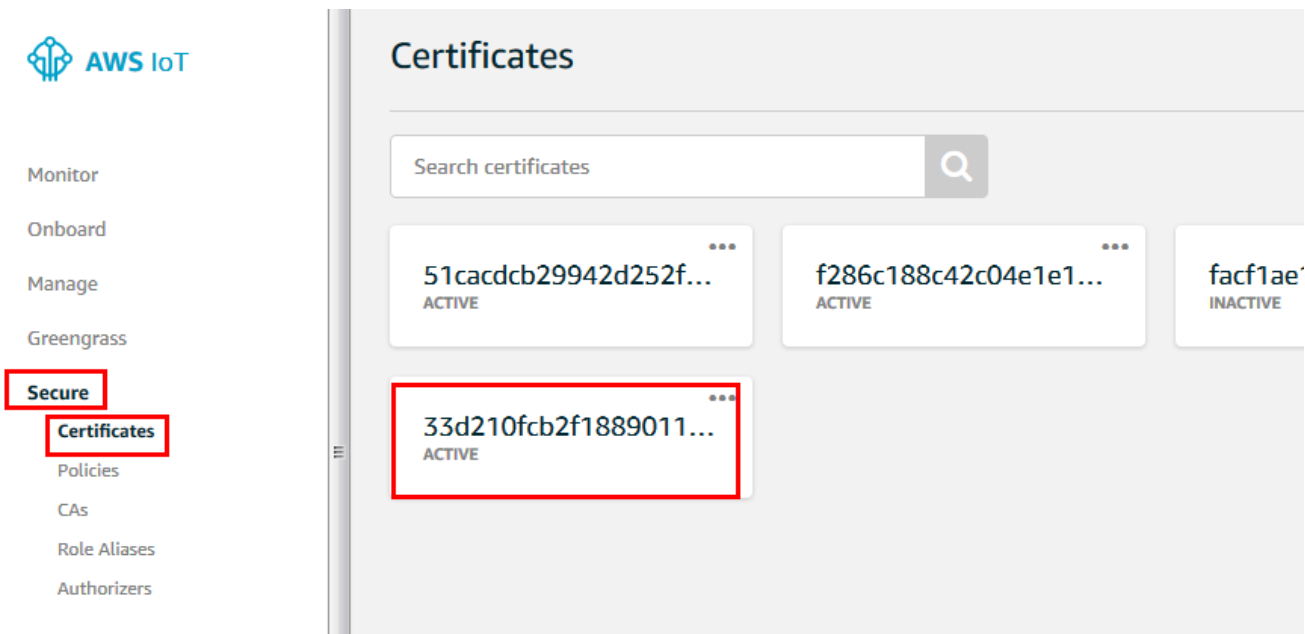
[A root CA for AWS IoT](#) [Download](#)

[Activate](#)

Create Policies:



Attach policy and thing:



## 1.2 Connection address and port

Find the MQTT connection server as below:

The screenshot shows the AWS IoT console interface. On the left, the 'Manage' menu is expanded, and 'Things' is selected. The main area displays a list of things, with 'mydevice1' highlighted. Below this, the details for 'mydevice1' are shown. The 'Interact' tab is selected, displaying the MQTT connection information. The MQTT connection address is highlighted in a red box, and a red arrow points to it with the label 'connection address'.

**Things**

Search things

**mydevice1** NO TYPE

**MyGreengrass\_Core** NO TYPE

**MylotThing** NO TYPE

**THING**  
**mydevice1**  
NO TYPE

Details: This thing already appears to be connected. [Connect a device](#)

Security

Thing Groups

Billing Groups

Shadow

**Interact**

Activity

Jobs

Violations

Defender metrics new

**HTTPS**

Update your Thing Shadow using this Rest API Endpoint. [Learn more](#)

`af5c714003fcr-ats.iot.us-west-2.amazonaws.com`

**MQTT**

Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow)

[Learn more](#)

**Update to this thing shadow**

`$aws/things/mydevice1/shadow/update`

connection address

AWS-IOT support connection port as below:

## Protocols, Port Mappings, and Authentication

The following table shows each protocol supported by AWS IoT, the authentication method, and port used for each protocol.

Protocol, Authentication, and Port Mappings			
Protocol	Authentication	Port	ALPN ProtocolName
MQTT	X.509 client certificate	8883, 443 <sup>†</sup>	x-amzn-mqtt-ca
HTTPS	X.509 client certificate	8443, 443 <sup>†</sup>	x-amzn-http-ca
HTTPS	SigV4	443	N/A
MQTT over WebSocket	SigV4	443	N/A

## 1.3 Module load certificates and use the MQTTS connect AWS IoT Core:

6d1fa87feb-certificate.pem.crt	2020/1/20 14:48	安全证书	2 KB	
6d1fa87feb-private.pem.key	2020/1/20 14:49	KEY 文件	2 KB	client.pem
6d1fa87feb-public.pem.key	2020/1/20 14:48	KEY 文件	1 KB	user_key.pem
AmazonRootCA1.pem	2020/1/20 15:00	PEM 文件	2 KB	cacert.pem

Load the certificates in the module:

```

AT+QFUPL="cacert.pem",1206,100
CONNECT
+QFUPL: 1206,5a63

OK
AT+QFUPL="client.pem",1220,100
CONNECT
+QFUPL: 1220,293e

OK
AT+QFUPL="user_key.pem",1679,100
CONNECT
+QFUPL: 1679,7f66

OK

```

Configure TLS option:

```
AT+QSSLCFG="cacert",2,"cacert.pem"
OK
AT+QSSLCFG="clientcert",2,"client.pem"
OK
AT+QSSLCFG="clientkey",2,"user_key.pem"
OK
AT+QSSLCFG="ciphersuite",2,0xFFFF
OK
AT+QSSLCFG="sslversion",2,4
OK
AT+QSSLCFG="seclvl",2,2
OK
AT+QSSLCFG="ignorelocaltime",2,1
OK
```

Configure MQTTS and MQTT protocol version:

```
AT+QMTCFG="ssl",0,1,2
OK
AT+QMTCFG="version",0,4
OK
```

Connect server:

```
AT+QMTOPEN=0,"af5c714003fcr-ats.iot.us-west-2.amazonaws.com",8883
OK
```

```
+QMTOPEN: 0,0 [REDACTED] connect server success
```

```
AT+QMTCONN=0,"mydevice1"
```

```
OK
```

```
+QMTCONN: 0,0,0 [REDACTED] MQTT connect success
```

## 1.4 Use the Device Shadow service


In the Things->mydevice1->Interact, we can find the MQTT shadow topic. Use topics to enable applications and things to get, update, or delete the state information for a Thing.

The screenshot shows the AWS IoT console interface. On the left, a sidebar contains the following menu items: **Interact** (highlighted with a red box), Activity, Jobs, Violations, and Defender metrics. The main content area is titled **MQTT** (also highlighted with a red box). Below the title, there is a description: "Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow)". A link labeled "Learn more" is provided. Below this, there are four sections, each with a title and a corresponding MQTT topic in a dark blue box:

- Update to this thing shadow**: `$aws/things/mydevice1/shadow/update`
- Update to this thing shadow was accepted**: `$aws/things/mydevice1/shadow/update/accepted`
- Update this thing shadow documents**: `$aws/things/mydevice1/shadow/update/documents`
- Update to this thing shadow was rejected**: `$aws/things/mydevice1/shadow/update/rejected`


The shadow data report use json format.

```
AT+QMTSUB=0,1,"$aws/things/mydevice1/shadow/update/rejected",1
OK

+QMTSUB: 0,1,0,1
AT+QMTPUB=0,1,1,0,"$aws/things/mydevice1/shadow/update"
> {"state":{"reported":{"color":"red"}}}
OK  Ctrl+Z send data

+QMTPUB: 0,1,0
AT+QMTPUB=0,1,1,0,"$aws/things/mydevice1/shadow/update"
> 1234567890
OK

+QMTPUB: 0,1,0

+QMTRECV: 0,1,"$aws/things/mydevice1/shadow/update/rejected","{"code":400,"message":"Missing
required node: state"}"  receive reject
```

## 2. Detailed AT command steps for module connection to AWS IoT core.

### 2.1 Configure the network

```
// AT+QCFG="nwscanmode", [<scanmode>]
```

```
//<scanmode>: 0 Automatic
```

```
// 1 GSM only
```

```
// 3 LTE only
```

//E.g configure the scan mode is LTE only.

**AT+QCFG="nwscanmode",3**

**OK**

```
//AT+QCFG="iotopmode"[,<mode>]
```

//<mode> Number format. Network category to be searched under LTE RAT.

```
// 0 eMTC
```

```
// 1 NB-IoT
```

```
// 2 eMTC and NB-IoT
```

//E.g configure the network is NB-IoT.

**AT+QCFG="iotopmode",1**

**OK**

```
//AT+QCFG="band"[,<gsmbandval>,<emtcbandval>,<nbiotbandval>]
```

//<gsmbandval>: A hexadecimal value that specifies the GSM frequency band. If it is set to 0, it //means

not to change GSM frequency band.

`<emtcbandval>`: A hexadecimal value that specifies the eMTC frequency band. If it is set to 0 or 0x40000000, it means not to change the frequency band

`<nbbandval>`: A hexadecimal value that specifies the NB-IoT frequency band. If it is set to 0 or 0x40000000, it means not to change the frequency band

//Which band need configure please confirm for **chapter 7 <Support band list>**.

//E.g configure NB-IoT network BAND 5, the hexadecimal value is 0x10, only need input 10.

**AT+QCFG="band",0,0,10**

**OK**

```
//Query the network status.
```

**AT+CEREG?;+QNWINFO;+QCSQ**

```
+CEREG: 0,1 //Registered NB-IoT network
```

+QNWINFO: "CAT-NB1","46011","LTE BAND 5",2506

**+QCSQ: "CAT-NB1",-80,-94,103,-15**

**OK**

## 2.2 Load certificates

```
//If module already have the certificates, can use AT+QFDEL delete it first.
```

**AT+QFDEL="cacert.pem"**

**OK**

**AT+QFDEL="client.pem"**

**OK**

AT+QFDEL="user\_key.pem"

**OK**

//The server certificate size is 1188 bytes, the timeout is 5000s, after echoing "CONNECT" load it.

**AT+QFUPL="cacert.pem",1188,5000**

## CONNECT

-----BEGIN CERTIFICATE-----

MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQsF  
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6  
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTEL  
MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvcjEZMBcGA1UEAxMQQW1hem9uIFJv  
b3QgQ0EgMTCCASlwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj  
ca9HgFB0fW7Y14h29Jlo91ghYPI0hAEvrAlhtOgQ3pOsQTNroBvo3bSMgHFzZM  
9O6lI8c+6zf1tRn4SWiw3te5djdYZ6k/ol2peVKVuRF4fn9tBb6dNqcmzU5L/qw  
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWA6  
VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L  
93FcXmn/6pUCyziKrlA4b9v7LWlBxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm



```
jgSubJrlqg0CAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBBYEFIQYzIU07LwMIJQuCFmcx7lQTgoIMA0GCSqGSib3DQEBCwUA
A4IBAQCY8jdaQZChGsV2USggNiMOruYou6r4lK5lpDB/G/wkjUu0yKGX9rbxenDI
U5PMCCjmmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbl1Bjtt/msv0tadQ1wUs
N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKWlJbYK8U90vv
o/ufQJVtMVT8QtPHRh8jrdkPSHCa2XV4cdFyQzR1bldZwgJcJmApzyMZFo6lQ6XU
5Msl+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
rqXRfboQnoZsG4q5WTP468SQvvG5
-----END CERTIFICATE-----
+QFUPL: 1188,2d13
```

OK

//The client certificate size is 1224 bytes, the timeout is 5000s, after echoing "CONNECT" load it.

**AT+QFUPL="client.pem",1224,5000**

**CONNECT**

-----BEGIN CERTIFICATE-----

```
MIIDWjCCAkkKAwIBAgIWAObjU0wHkgefuhH3vJLBlUJD0IMA0GCSqGSib3DQEB
CwUAME0xSzBjBgNVBAsMQkFtYXpvbiBXZWlU2VydmljZXMGZTz1BbWF6b24uY29t
IEluYy4gTD1TZWF0dGxIFNUPVdhc2hpbmd0b24gQz1VUzAeFw0xOTA1MjAyMDM3
NThaFw00OTEyMzEyMzU5NTIaMB4xHDAaBgNVBAMME0FXUyBjb1QgQ2VydGlmZWVh
dGUwggEiMA0GCSqGSib3DQEBAQUAA4IBDwAwggEKAoIBAQDF99Sr+7QjYsScC4uU
xpNj2Z4+InchoC8afsuCXAS14FJDJhs5iuG+rrV4fo3oBUfWxU0YeMueVOE4wcG7
gJco3RX13COWrQqmKAdEMd7JCFk874MLHx4LILqHWJiF7lwdCMX+fGx9cTL/EKj4
iX3x0le+KrtRr0ax2Jjig6A9lAdmSYRPPpqQaiZJA+ytSwf6fi2iafscpA8r8d5R
c+alZbE5ocChkd5SOu99JZzEsgB1gCBdlsY5POEENc9iGvl8x1GNtzzr/NG83Woq
6ZmA6DjJSLUqyYX0vHOWMBWdNOrkLKu2dG85XJJn1CxZWPhcP88RSvQGO6xGJdOR
UrT9AgMBAAGjYDBeMB8GA1UdIwQYMBaAFP09lWYvUvKRzDCIcBpc+AmFEI4MB0G
A1UdDgQWBBTSt4AT/FfXa/JDjKMLpKvEcneeXDAMBgNVHRMBAf8EAjAAMA4GA1Ud
DwEB/wQEAwIHgDANBgkqhkiG9w0BAQsFAAOCQAQEA+3wJL4MJ78JZIRqfLxtCD6h
hEK0PhBw9Oiyfrzl4zsrRe0Tme/PZ6xOcxixS2mDnE3uPfbGO5LP4MQ7ACFuV+9
Pu00W3gAnKU2w7MaqBkuUmWYd1yMVbD0osx6BH0hMUzOqqX6oNFwZw5E0gnSAubj
ztqfCKXGSs6R2oeUfX+gCAhMPbV7atmi6wMZG6vaeP3HWF5p0YJOLQysHYsRjgIN
0zr+P1mHxkpKeiwE3S9xxMOF2C+TXjKvKkarm2YIRie42Xz26rb0IL/ifz5KkgQU
2dxq8ohryewCXBeqwyzzQILbGkhz5Ot+dYXpbWuxCow7Eocobmdx7+KSI5+n4A==
-----END CERTIFICATE-----
+QFUPL: 1224,7a6c
```

OK

//The client key size is 1675 bytes, the timeout is 5000s, after echoing "CONNECT" load it.

**AT+QFUPL="user\_key.pem",1675,5000**

**CONNECT**

-----BEGIN RSA PRIVATE KEY-----

MIIEowIBAAKCAQEAxfUq/u0I2LEnAuLIMaTSdmePiJ3laAvGn7LglwEteBSQyYb  
OYrhvq61eH6N6AVH1sVNGHjLnIthOMHBu4CXKN0V9dwjsK0KpigHRDHeyQhZPO+D  
Cx8eCyC6h1iYheyMHQjF/nxsfXEy/xCo+Il98dJXviq7Ua9GsdiSYoOgPZQHZkmE  
T6aakGomSQPsUsH+n4tomn7HKQPK/HeUXPmpWWxOaHAoZHeUjrvfSWcxLIAdYAg  
XSLGOTzhBDXPYhr5fMdRjbc86/zRvN1qKumZgOgyY0i1KsmF9LxzljAVnTTq5Cyr  
tnRvOVySZ9QsWVj4XD/PEUr0BjusRiXTkVK0/QIDAQABAoIBACzg7wSCDIdLd7jM  
YTs7/R3ryJTbaVhllOiHwAjwZVz52oxAckGMO3rnzInfjrn/oHaOydh3Yeml4lzk  
h8/C6gGohRb0bd7ai6R5RJRwHcpDKz/GKFX4Gu543/BY5FLP+j+GwqPbfxi89cue  
8j9NpBCTF7MTqt1st8EPkLfmkQgERTzv3979VSALMSg0F5aPTKd8baJk5RQfyNqV  
TYuPc1YbwQGC7MfVgP/e0aXRjxq4FtRnLhsbaEhpp2kenuZZnpwAr5c03+Q2NIz9  
4l3tZp3BzA9+CczSE3DSSkqow+h0JWmMsc4h2JKn169nyini3/7d2MvYHhc9V9kZD  
8cSZQIECgYEA+d8NDKu5QpnDOCRXEeje5Wq0qMSiWw94pfWWvPMP9vwSI9FVj8vN  
gkZhAFFQnVtCFe5yHDP6/gTCcY5E2Q9WfxaWkKbAIWRMZul6FWdCRXnEgVxJZBUv  
ofERMyn5GdlEH4SmN4FFimxDRSeIALGwYRE7AJHpQuwHpodCOdauekCgYEAytLg  
yOvG0DHlhf4+aGlb3YVg1skGTZ5KdiV78/wa7jRTp16vF18yMUFkFT2X8yfnk1Se  
OHC9uAukXH5MN4aSbEa9KH8W0/fuBaAD3ZTFsm73Rfdvo+Gm6ar3/wymqiOAn5PN  
UZOY9lqnS8lchlmK6WzK+OqMuGBR34YF8PFU6fUCgYAbiP7xJINX2YXvy+FkcdsO  
QQIDjxetHTIYYIXp8lcT2pdNjyZcoELXVIYeKNfPX10a+ZnZA7lnALNEvD7OBKit  
fJm++wE1LM5WkOgJL8XXziFfeAlGa2dCW0Rf9QIkXRLqvj1q5G37YQ7QshiEqxAa  
wkFUQMAJxeTZ7xHXOdcM8QKBgHvREHiXt8SBRs4y1BiQTqSWfPE9JwdrcdCStE/  
bhFM0GS0YJ8b9kArczWkTEacbbp7Rbue17KrKMNUokPFgQTixZVv1HXYO7+KF30  
nFu5ciq8SuL4tjSfspV426Wx1quURxstx4pBuq+BVYKO10Z4q9AZH57O7ixOXQlH  
0XyAoGBAOeVZ7SmY2RVkB0ArfHb0lIYnnovG1rremCoxq+y5lgaYl1ArAkP0p0t  
GQadZ5yAShVEDuuL/x9OEZmMGkt+vWrvwO/5pg4x+BbnTmZJ+Ud5zWf7Dletx+Ce  
Fs0yJkWTJCpHA5SFGTeFdWyYcstv+nvVlg7wOPj/tLHWmTAuxTcd

-----END RSA PRIVATE KEY-----

**+QFUPL: 1675,150**

**OK**

//Query the list file.

**AT+QFLST**

**+QFLST: "cacert.pem",1188**

**+QFLST: "client.pem",1224**

**+QFLST: "user\_key.pem",1675**

**OK**

## 2.3 Active PDP context

//AT+QICSGP=<contextID>[,<context\_type>,<APN>[,<username>,<password>[,<authentication>]]]

```
//<contextID>      Integer type. The context ID. The range is 1-16
//< context_type >: Integer type. The protocol type.
                  1 IPv4
//<APN> :          String type. The access point name.
//<username>:       String type. The username.
//<password> :      String type. The password
//<authentication> Integer type. The authentication methods.
//                0  None
//                1  PAP
//                2  CHAP
//                3  PAP or CHAP
// E.g configure Parameters of a TCP/IP Context, the <contextID> is 1,<context_type> is IPv4, the
//<APN> // is "m2m64.com.attz", the <username> and <password> is empty, the <authentication> //is 0.
AT+QICSGP=1,1,"m2m64.com.attz","", "",0
OK
```

```
//Active PDP contextID is 1.
AT+QIACT=1
OK
```

## 2.4 Configure SSL option

```
//Configure the path of server certificate for SSL context 2.
AT+QSSLCFG="cacert",2,"cacert.pem"
OK
```

```
//Configure the path of client certificate for SSL context 2.
AT+QSSLCFG="clientcert",2,"client.pem"
OK
```

```
//Configure the path of client private key for SSL context 2.
AT+QSSLCFG="clientkey",2,"user_key.pem"
OK
```

```
//Configure the authentication mode for SSL context 2, SSL authentication mode: server and client
authentication if requested by the remote server
AT+QSSLCFG="secllevel",2,2
OK
```

```
//SSL authentication version
AT+QSSLCFG="sslversion",2,4
OK
```

```
//Cipher suite
AT+QSSLCFG="ciphersuite",2,0xFFFF
OK
```

```
// Ignore the time of authentication
AT+QSSLCFG="ignorelocaltime",2,1
OK
```

## 2.5 Configure MQTT option

```
// Configure MQTT session into SSL mode for SSL context 2
AT+QMTCFG="SSL", 0, 1, 2
OK
```

```
//Configure MQTT protocol version MQTT v3.1.1
AT+QMTCFG="version",0,4
OK
```

## 2.6 MQTT connect and data interaction

```
//Start MQTT SSL connection
AT+QMTOPEN=0,"a2sgasbshsff52-ats.iot.us-west-2.amazonaws.com",8883
OK
```

```
+QMTOPEN: 0,0
```

```
//Connect to MQTT server
AT+QMTCONN=0,"Quectel"
OK
```

```
+QMTCONN: 0,0,0
```

```
//Subscribe to topics
AT+QMTSUB=0,1,"$aws/things/pabi/shadow/update/accepted",0
OK
```

```
+QMTSUB: 0,1,0,
```

```
//Publish messages, after echoing >, send the payload, and execution "ctrl+Z" send the data.
AT+QMTPUB=0,1,1,0,"$aws/things/pabi/shadow/update"
> { "state" : { "reported" : { "color" : "red" } } }
OK
```

+QMTPUB: 0,1,0

+QMTRECV:

0,0,"\$aws/things/pabi/shadow/update/accepted","{\"state\":{\"reported\":{\"color\":\"red\"}},\"metadata\":{\"reported\":{\"color\":{\"timestamp\":1578336900}}},\"version\":3,\"timestamp\":1578336900}"

//Close MQTTS connection

AT+QMTCLOSE=0

OK

+QMTCLOSE: 0,0

## Support band list

**<gsmbandval>**: (eg.: 0x0a=0x02(GSM1800)+0x08(GSM1900)). This parameter is valid on BG95-M3 only.

00000000 No change  
00000001 GSM 900MHz  
00000002 GSM 1800MHz  
00000004 GSM 850MHz  
00000008 GSM 1900MHz  
0000000F Any frequency band

**<emtcbandval>**: (eg.: 0x15=0x01(LTEB1)+0x04(LTE B3)+0x10(LTE B5))

0x1 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND1) LTE B1  
0x2 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND2) LTE B2  
0x4 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND3) LTE B3  
0x8 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND4) LTE B4  
0x10 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND5) LTE B5  
0x80 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND8) LTE B8  
0x800 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND12) LTE B12  
0x1000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND13) LTE B13  
0x2000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND14) LTE B14  
0x20000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND18) LTE B18  
0x40000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND19) LTE B19  
0x80000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND20) LTE B20  
0x1000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND25) LTE B25  
0x2000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND26) LTE B26  
0x4000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND27) LTE B27  
0x8000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND28) LTE B28  
0x40000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND31) LTE B31  
0x2000000000000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND66) LTE B66

0x8000000000000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND72) LTE B72  
0x1000000000000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND73) LTE B73  
0x1000000000000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND85) LTE B85  
0x4000000000000000 No change  
0x4001C200000000F0E389F (CM\_BAND\_PREF\_ANY) Any frequency band

**<nbiotbandval>:**

0x1 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND1) LTE B1  
0x2 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND2) LTE B2  
0x4 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND3) LTE B3  
0x8 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND4) LTE B4  
0x10 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND5) LTE B5  
0x80 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND8) LTE B8  
0x800 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND12) LTE B12  
0x1000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND13) LTE B13  
0x20000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND18) LTE B18  
0x40000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND19) LTE B19  
0x80000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND20) LTE B20  
0x1000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND25) LTE B25  
0x2000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND26) LTE B26  
0x8000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND28) LTE B28  
0x40000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND31) LTE B31  
0x2000000000000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND66) LTE B66  
0x4000000000000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND71) LTE B71  
0x8000000000000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND72) LTE B72  
0x1000000000000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND73) LTE B73  
0x1000000000000000 (CM\_BAND\_PREF\_LTE\_EUTRAN\_BAND85) LTE B85  
0x4000000000000000 No change  
0x4001C2000000004E0E189F (CM\_BAND\_PREF\_ANY) Any frequency band