ICA AT 指令集测试规范

1. 适用范围

《ICA AT 指令集规范 v0.4》

《ICA AT 指令集规范 v0.6》

以下测试用例支持以上 ICA AT 指令集规范，适用于遵循并实现了 ICA AT 指令集的模组。

以下标注**可选**的测试项，若不支持该功能则无需测试，若支持则必测

2. MQTT AT 指令测试规

2.1 IMQTTMODE 指令测试（可选）

用例编号：WL-AT-MQTT-101

用例名称：IMQTTMODE 指令测试（可选）

测试目的：验证模组 IMQTTMODE 指令

测试步骤：

1.模组处于开放环境，模组上电

2.串口输入如下指令：

AT+IMQTTMODE=0

AT+IMQTTMODE?

3.串口输入如下指令：

AT+IMQTTMODE=2

AT+IMQTTMODE?

4.串口输入如下指令：

AT+IMQTTMODE=3

AT+IMQTTMODE?

5.串口输入如下指令：

AT+IMQTTMODE=1

AT+IMQTTMODE?

6.串口输入如下指令：

AT+IMQTTMODE=a

AT+IMQTTMODE?

预期结果：

1.步骤 2，设置成功，读取到为 mode0

2.步骤 3，设置失败，读取到为 mode0

3.步骤 4，设置失败，读取到为 mode0

4.步骤 5，设置成功，读取到为 mode1

5.步骤 6，设置失败，读取到为 mode1

备注：

2.2 IMQTTOPEN 指令（可选）

用例编号： WL-AT-CoAP-201

用例名称： IMQTTOPEN 指令

测试目的： 验证模组 IMQTTOPEN 指令

测试步骤：

1.模组处于开放环境，模组上电

2.分别通过串口输入如下内容：

AT+IMQTTOPEN="productxxx.aliyuncs.com",1883

AT+IMQTTOPEN="139.196.135.135",1883

3.串口输入如下：

AT+IMQTTOPEN="productxxx.aliyuncs.com",1883,1234

AT+IMQTTOPEN="139.196.135.135",1883,1234

4.串口输入如下：

AT+IMQTTOPEN="ABC",1883

AT+IMQTTOPEN="139.196.135.135",1883

5.串口输入如下：

AT+IMQTTOPEN="productxxx.aliyuncs.com",1883

AT+IMQTTOPEN="ABC",1883

预期结果：

1.步骤 2，设置成功

2.步骤 3，设置失败

3.步骤 4，设置失败

4.步骤 5，设置失败

备注：仅支持 ICA 联盟 AT 指令集规范 v0.6 及以上版本

2.3 IMQTTAUTH 指令测试

用例编号：WL-AT-MQTT-102

用例名称：IMQTTAUTH 指令测试

测试目的：验证模组 IMQTTAUTH 指令

测试步骤：

1.模组处于开放环境，模组上电

2.串口输入如下指令

AT+IMQTTAUTH="productkey","devicename","devicesecret"

AT+IMQTTAUTH?

3.串口输入如下指令：

AT+IMQTTAUTH="abc","def","xyz"

AT+IMQTTAUTH?

4.串口输入如下指令：

AT+IMQTTAUTH="productkey","devicename"

AT+IMQTTAUTH?

5.串口输入如下指令：

AT+IMQTTAUTH="productkey"

AT+IMQTTAUTH?

6.串口输入如下指令：

AT+IMQTTAUTH="productkey", "devicename", "devicesecret", "abc"

AT+IMQTTAUTH?

预期结果：

1.步骤 2，执行成功，认证成功；读取到的 key 为 productkey

2.步骤 3，执行成功，认证失败；读取到的 key 为 abc

3.步骤 4，设置失败，读取到的 key 为 abc

4.步骤 5，设置失败，读取到的 key 为 abc

5.步骤 6，设置失败，读取到的 key 为 abc

备注：

2.4 IMQTTPARA 指令测试

用例编号：WL-AT-MQTT-103

用例名称：IMQTTPARA 指令测试

测试目的：验证模组 IMQTTPARA 指令

测试步骤：

1.模组处于开放环境，模组上电

2.分别通过串口输入 x 等于 0，0.5，1，1.5，10，a，空

AT+IMQTTPARA="TIMEOUT",x

2.分别通过串口输入 x 等于 0，0.5，1，1.5，10，a，空

AT+IMQTTPARA="CLEAN",x

3.分别通过串口输入 x 等于 0，1，60，100，180，200，a，空

AT+IMQTTPARA="KEEPALIVE",x

4.分别通过串口输入 x 等于 3.1，3.1.1，1.0，a，空

AT+IMQTTPARA="VERSION", x

预期结果：

1.步骤 2，0.1、0.5、1、1.5、10 执行返回成功，0、a、空 执行失败

2.步骤 3，0、1 执行返回成功，0.5、1.5、10、a、空 执行失败

3.步骤 4，60、100、180 执行返回成功，0、1、200、a、空执行失败

4.步骤 5，3.1、3.1.1 执行成功，1.0、a、空 执行失败

备注：timeout 取值范围由厂商合理定义，keepalive 取值范围 60~180 秒

2.5 IMQTTCONN 指令测试

用例编号：WL-AT-MQTT-104

用例名称：IMQTTCONN 指令测试

测试目的：验证模组 IMQTTCONN 指令

测试步骤：

1.模组处于开放环境，模组上电

2.模组已通过 AT 输入 IMQTTAUTH 相应参数，并配置了 IMQTTPARA 参数

3.通过串口输入：AT+IMQTTCONN

4.模组清空参数，不做 IMQTTAUTH，直接：AT+IMQTTCONN

5.模组清空参数，做了 AT+IMQTTAUTH，但是不设置 IMQTTPARA，进行

AT+IMQTTCONN

6.模组内置参数，做了 AT+IMQTTAUTH，但是不设置 IMQTTPARA，进行

AT+IMQTTCONN

预期结果：

1.步骤 3，返回执行成功

2.步骤 4，返回执行失败

3.步骤 5，返回执行失败

4.步骤 6，返回执行成功

备注：模组内置参数取值范围符合 ICA 标准

2.6 IMQTTPUB 指令测试

用例编号：WL-AT-MQTT-105

用例名称：IMQTTPUB 指令测试

测试目的：验证模组 IMQTTPUB 指令

测试步骤：

1.模组处于开放环境，模组上电

2.模组已通过 AT 输入 IMQTTAUTH 相应参数，并配置了 IMQTTPARA 参数，执行了 AT+IMQTTCONN 连接成功

3.通过串口输入如下指令：

AT+IMQTTPUB="/productkey/devicename/update",0,"hello world"

4.串口输入如下指令：

AT+IMQTTPUB="abc",1,"hello world"

5.串口输入如下指令：

AT+IMQTTPUB="abc",4,"hello world"

6.串口分别输入参数个数 1，2，4 个：

如 1 个参数，则串口输入: AT+IMQTTPUB="abc"

预期结果：

1.步骤 3，返回成功

2.步骤 4，返回执行成功

3.步骤 5，返回执行失败

4.步骤 6，参数非 3 个的，都返回执行失败

备注：

2.7 IMQTTSUB 指令测试

用例编号：WL-AT-MQTT-106

用例名称：IMQTTSUB 指令测试

测试目的：验证模组 IMQTTSUB 指令

测试步骤：

1.模组处于开放环境，模组上电

2.模组已通过 AT 输入 IMQTTAUTH 相应参数，并配置了 IMQTTPARA 参数，执行了 AT+IMQTTCONN 连接成功

3.串口输入如下指令：

AT+IMQTTSUB="/productkey/devicename/get",0

4.串口输入如下指令：

AT+IMQTTSUB="abc",1

5.串口输入如下指令：

AT+IMQTTPSUB="abc"

预期结果：

1.步骤 3，返回成功

2.步骤 4，返回执行失败

3.步骤 5，返回执行失败

备注：

2.8 IMQTTUNSUB 指令测试

用例编号：WL-AT-MQTT-107

用例名称：IMQTTUNSUB 指令测试

测试目的：验证模组 IMQTTUNSUB 指令

测试步骤：

1.模组处于开放环境，模组上电

2.模组已通过 AT 输入 IMQTTAUTH 相应参数，并配置了 IMQTTPARA 参数，执行了 AT+IMQTTCONN 连接成功

3.串口输入如下指令：

AT+IMQTTSUB="/productkey/devicename/get",0

4.串口输入如下指令：

AT+IMQTTUNSUB="/productkey/devicename/get"

5.串口输入如下指令：

AT+IMQTTUNSUB="/productkey/devicename/get",0

6.串口输入如下指令：

AT+IMQTTPUNSUB="abc"

预期结果：

1.步骤 3，返回成功

2.步骤 4，返回成功

3.步骤 5，返回执行失败

4.步骤 6，返回执行失败

备注：

2.9 IMQTTRCVPUB 指令测试

用例编号： WL-AT-MQTT-108

用例名称： IMQTTRCVPUB 指令测试

测试目的： 验证模组 IMQTTRCVPUB 指令

测试步骤：

1.模组处于开放环境，模组上电

2.模组已通过 AT 输入 IMQTTAUTH 相应参数，并配置了 IMQTTPARA 参数，执行了 AT+IMQTTCONN 连接成功

3.串口输入如下指令：

AT+IMQTTSUB="/productkey/devicename/get",0

4.从云端控制台，向/productkey/devicename/get publish 消息（qos=1/0）

预期结果：

1.步骤 3，返回成功

2.步骤 4，云端下发消息后，设备串口打印收到此消息内容

备注：

2.10 IMQTTSTATE/IMQTTDISCONN 指令测试

用例编号： WL-AT-MQTT-109

用例名称： IMQTTSTATE/IMQTTDISCONN 指令测试

测试目的： 验证模组 IMQTTSTATE/IMQTTDISCONN 指令

测试步骤：

1.模组处于开放环境，模组上电

2.模组已通过 AT 输入 IMQTTAUTH 相应参数，并配置了 IMQTTPARA 参数，执行了 AT+IMQTTCONN 连接成功

3.串口输入如下指令：

AT+IMQTTSTATE?

4.串口输入如下指令

AT+IMQTTDISCONN

5.串口输入如下指令：

AT+IMQTTSTATE?

预期结果：

1.步骤 3，返回状态 1 或 2

2.步骤 4，返回成功

3.步骤 5，返回状态 0

备注：

2.11 支持 TLS/非 TLS 通信(可选)

用例编号： WL-AT-MQTT-110

用例名称： 支持 TLS/非 TLS 通信(可选)

测试目的： 验证模组可以通过 TLS 认证连接至阿里云

测试步骤：

1.模组处于开放环境，模组上电

2.通过串口工具发送指令 AT+IMQTTMODE=1（如不支持 TLS，则此处为默认值）

3.配置认证、连接参数，建立连接，pub 消息

4.设备订阅 /productkey/devicename/get

5.服务端 web 控制台向/productkey/devicename/get 发布消息

6.串口发送 AT+IMQTTMODE=0（如不支持 TLS，则此处为默认值）

7.重复步骤 3~5

预期结果：

1.步骤 2、3，每一步都设置成功，且模组 pub 消息，服务端收到此消息

2.步骤 5，服务端发布消息后，设备端收到

3.步骤 7，mode 为 0 后，设备发布订阅可正常通信

备注：

2.12 支持 QoS0/QoS1

用例编号： WL-AT-MQTT-111

用例名称： 支持 QoS0/QoS1

测试目的： 验证模组可以支持发布 QoS0 和 QoS1 消息

测试步骤：

1.模组处于开放环境，模组上电

2.通过串口工具发送指令 AT+IMQTTMODE=1（如不支持 TLS，则此处为默认值）

3.配置认证、连接参数，建立连接，pub QoS 为 1 的消息

4.再 pub QoS 为 0 的消息

预期结果：

1.步骤 3，设备端可以发布 qos=1 的消息

2.步骤 4，设备端可以发布 qos=0 的消息

备注：

2.13 支持心跳

用例编号： WL-AT-MQTT-112

用例名称： 支持心跳

测试目的： 验证模组心跳机制生效

测试步骤：

1.模组处于开放环境，模组上电

2.通过串口工具发送指令 AT+IMQTTMODE=1（如不支持 TLS，则此处为默认值）

3.配置认证、连接参数，其中，keepalive 字段，设置为 90

4.建立连接

5.keepalive 字段再设置为 180，建立连接

预期结果：

1.步骤 4，模组每隔 90s 发送 ping request

2.步骤 5，模组每隔 180s 发送 ping request

备注：

2.14 支持 session 配置

用例编号： WL-AT-MQTT-113

用例名称： 支持 session 配置

测试目的： 验证模组 session 配置生效

测试步骤：

1.模组处于开放环境，模组上电

2.通过串口工具发送指令 AT+IMQTTMODE=1（如不支持 TLS，则此处为默认值）

3.配置认证、连接参数，其中，clean session 字段，设置为 0，

4.建立连接，设备订阅/productkey/devicename/get

5.断开连接（使用 AT+IMQTTDISCONN 断开连接），服务端 web 控制台向 /productkey/devicename/get 发布消息

6.建立连接，设备端可以收到服务端下发的缓存信息

7.clean session 字段再设置为 1，重复步骤 4、5、6

预期结果：

1.步骤 4、6，可成功建立连接

2.步骤 6，clean=0 时，设备端可以收到云端的缓存信息

3.步骤 7，clean=1 时，设备端不可以收到云端的缓存信息

备注：

2.15 取消订阅

用例编号：WL-AT-MQTT-114

用例名称：取消订阅

测试目的：验证模组取消订阅生效

测试步骤：

1.模组处于开放环境，模组上电

2.通过串口工具发送指令 AT+IMQTTMODE=1（如不支持 TLS，则此处为默认值）

3.配置认证、连接参数，建立连接

4.模组 sub /productkey/devicename/get

5.服务端通过控制台向/productkey/devicename/get 发布消息

6.模组 AT+IMQTTUNSUB="/productkey/devicename/get"

7.服务端通过控制台向/productkey/devicename/get 发布消息

预期结果：

1.步骤 5，模组收到服务端下发的消息

2.步骤 7，模组无法收到服务端下发的消息

备注：

2.16 断开连接

用例编号：WL-AT-MQTT-115

用例名称：断开连接

测试目的：验证模组断开连接生效

测试步骤：

1.模组处于开放环境，模组上电

2.通过串口工具发送指令 AT+IMQTTMODE=1（如不支持 TLS，则此处为默认值）

3.配置认证、连接参数，建立连接

4.模组 sub /productkey/devicename/get

5.服务端通过控制台向/productkey/devicename/get 发布消息

6.模组 AT+IMQTTDISCONN

7.服务端通过控制台向/productkey/devicename/get 发布消息

8.模组 pub 消息

预期结果：

1.步骤 5，模组收到服务端下发的消息

2.步骤 7，模组无法收到服务端下发的消息

3.步骤 8，模组无法 pub 出去消息

备注：

2.17 大数据包支持

用例编号：WL-AT-MQTT-116

用例名称：大数据包支持

测试目的：验证模组支持大数据包通信

测试步骤：

1.模组处于开放环境，模组上电

2.通过串口工具发送指令 AT+IMQTTMODE=1（如不支持 TLS，则此处为默认值）

3.配置认证、连接参数，建立连接

4.模组 sub /productkey/devicename/get

5.服务端通过控制台向/productkey/devicename/get 发布消息，payload 为 512B~2KB 大小

6.模组 pub /productkey/devicename/update，payload 为 512B 大小

预期结果：

1.步骤 5，模组收到服务端下发的消息，payload 与从服务端下发的消息完全一致

2.步骤 6，服务端收到模组上报的消息，payload 与从设备端上报的消息完全一致

备注：

2.18 反复建立连接

用例编号：WL-AT-MQTT-117

用例名称：反复建立连接

测试目的：验证模组反复建立连接不出现异常

测试步骤：

1.模组处于开放环境，模组上电

2.通过串口工具发送指令 AT+IMQTTMODE=1（如不支持 TLS，则此处为默认值）

3.配置认证、连接参数，其中 clean session 为 0

4.AT+IMQTTCONN

5.AT+IMQTTDISCONN

6.不断重复步骤 4、5，重复 100 次，每 N 秒循环一次

预期结果：

1.步骤 5，100 次不断建立连接时，设备不会出现死机、重启等异常，且无连接失败的情况

备注：N 为设备成功连接的平均时间

2.19 反复认证、连接

用例编号：WL-AT-MQTT-118

用例名称：反复认证、连接

测试目的：验证模组反复认证、连接不出现异常

测试步骤：

1.模组处于开放环境，模组上电

2.通过串口工具发送指令 AT+IMQTTMODE=1（如不支持 TLS，则此处为默认值）

3.配置认证、连接参数、建立连接

4.不断重复步骤 3，重复 100 次，每 N 秒循环一次

预期结果：

1.步骤 4，100 次不断建立连接时，设备不会出现死机、重启等异常，且无认证、连接失败的情况

备注：N 为设备成功 auth 的平均时间

3. CoAP AT 指令测试规范

3.1 ICOAPARA 指令

用例编号：WL-AT-CoAP-201

用例名称：ICOAPARA 指令

测试目的：验证模组 ICOAPARA 指令

测试步骤：

1.模组处于开放环境，模组上电

2.分别通过串口输入如下内容：

AT+ICOAPARA="TIMEOUT",1

3.串口输入如下：

AT+ICOAPARA="TIMEOUT",a

4.串口输入如下：

AT+ICOAPARA="TIMEOUT"

5.串口输入如下：

AT+ICOAPARA="ABC",1

6.串口输入如下：

AT+ICOAPARA="TIMEOUT",10

预期结果：

1.步骤 2，设置成功，读取到 timeout 为 1

2.步骤 3，设置失败，读取到 timeout 为 1

3.步骤 4，设置失败，读取到 timeout 为 1

4.步骤 5，设置失败，读取到 timeout 为 1

5.步骤 6，设置成功，读取到 timeout 为 10

备注：ICA 标准不对 timeout 做限制，由厂商合理定义取值范围

3.2 ICOAOPEN 指令（可选）

用例编号：WL-AT-CoAP-201

用例名称：ICOAPOPEN 指令

测试目的：验证模组 ICOAPOPEN 指令

测试步骤：

1.模组处于开放环境，模组上电

2.分别通过串口输入如下内容：

AT+ICOAPOPEN="productxxx.aliyuncs.com",5684

AT+ICOAPOPEN="139.196.135.101",5684

3.串口输入如下：

AT+ICOAPOPEN="productxxx.aliyuncs.com",5684,1234

AT+ICOAPOPEN="139.196.135.101",5684,1234

4.串口输入如下：

AT+ICOAPOPEN="ABC",5684

AT+ICOAPOPEN="139.196.135.101",5684

5.串口输入如下：

AT+ICOAPOPEN="productxxx.aliyuncs.com",5684

AT+ICOAPOPEN="ABC",5684

预期结果：

1.步骤 2，设置成功

2.步骤 3，设置失败

3.步骤 4，设置失败

4.步骤 5，设置失败

备注：仅支持 ICA 联盟 AT 指令集规范 v0.6 及以上版本

3.3 ICOAPAUTH 指令测试

用例编号：WL-AT-CoAP-202

用例名称：ICOAPAUTH 指令测试

测试目的：验证模组 ICOAPAUTH 指令

测试步骤：

1.模组处于开放环境，模组上电

2.分别通过串口输入如下内容

AT+ICOAPAUTH="productkey","devicename", "devicesecret"

AT+ICOAPAUTH?

3.串口输入如下：

AT+ICOAPAUTH="abc","def", "xyz"

AT+ICOAPAUTH?

4.串口输入如下：

AT+ICOAPAUTH="productkey","devicename"

AT+ICOAPAUTH?

5.串口输入如下：

AT+ICOAPAUTH="productkey"

AT+ICOAPAUTH?

6.串口输入如下：

AT+ICOAPAUTH="productkey", "devicename", "devicesecret", "abc"

AT+IMQTTAUTH?

预期结果：

1.步骤 2，执行成功，认证成功；读取到的 key 为 productkey

2.步骤 3，执行成功，认证失败；读取到的 key 为 abc

3.步骤 4，设置失败，读取到的 key 为 abc

4.步骤 5，设置失败，读取到的 key 为 abc

5.步骤 6，设置失败，读取到的 key 为 abc

备注：

3.4 收发字符串数据支持转义符

用例编号：WL-AT-CoAP-203

用例名称 ICOAPSENDREQ 指令测试

测试目的：验证模组 字符串数据支持转义符 指令

测试步骤：

1.模组处于开放环境，模组上电

2.模组已通过 AT 输入 ICOAPAUTH 相应参数

3.通过串口输入

AT+ICOAPSENDREQ=2,"topic/ProductKey/DeviceName/updata","hello world"

4.串口输入

AT+ICOAPSENDREQ=2,"topic/ProductKey/DeviceName/updata","{"action":"turnon"}"

5.串口输入

AT+ICOAPSENDREQ=2,"topic/ProductKey/DeviceName/updata","0001AC",1

预期结果：

1.步骤 3，返回成功，且云端收到的数据一致

2.步骤 4，返回成功，且云端收到的数据一致

3.步骤 5，返回成功，且云端收到的数据一致

备注：仅支持 ICA 联盟 AT 指令集规范 v0.6 及以上版本

3.5 ICOAPSENDREQ 指令测试

用例编号：WL-AT-CoAP-204

用例名称 ICOAPSENDREQ 指令测试

测试目的：验证模组 ICOAPSENDREQ 指令

测试步骤：

1.模组处于开放环境，模组上电

2.模组已通过 AT 输入 ICOAPAUTH 相应参数

3.通过串口输入

AT+ICOAPSENDREQ=2,"topic/ProductKey/DeviceName/updata","hello world"

4.串口输入

AT+ICOAPSENDREQ=2,"abc","hello worl"

5.串口输入

AT+ICOAPSENDREQ=2,"abc"

预期结果：

1.步骤 3，返回成功

2.步骤 4，返回执行失败

3.步骤 5，返回执行失败

备注：

3.6 大数据包支持

用例编号：WL-AT-CoAP-205

用例名称：大数据包支持

测试目的：验证模组支持大数据包

测试步骤：

1.模组处于开放环境，模组上电

2.模组已通过 AT 输入 ICOAPAUTH 相应参数

3.串口输入：

AT+ICOAPSENDREQ=2,"topic/ProductKey/DeviceName/updata","payload"

预期结果：

1.步骤 5，模组收到服务端下发的消息，payload 与从服务端下发的消息完全一致

2.步骤 6，服务端收到模组上报的消息，payload 与从设备端上报的消息完全一致

备注：其中 mqtt 通道 payload 最小支持 512B 大小；CoAP 通道 payload 最小支持 256B 大小

3.7 反复认证、数据通信

用例编号：WL-AT-CoAP-206

用例名称：反复认证、数据通信

测试目的：验证模组反复认证、数据通信不出现异常

测试步骤：

1.模组处于开放环境，模组上电

2.串口输入：

AT+ICOAPAUTH="productkey","devicename", "devicesecret"

3.串口输入：

AT+ICOAPSENDREQ=2,"topic/ProductKey/DeviceName/updata","hello world"

4.不断重复步骤 2，重复 100 次，每 X 秒循环一次

预期结果：

1.步骤 4，100 次不断建立连接时，设备不会出现死机、重启等异常，且无认证、连接失败的情况

备注：X 为成功 AUTH 一次的平均时间