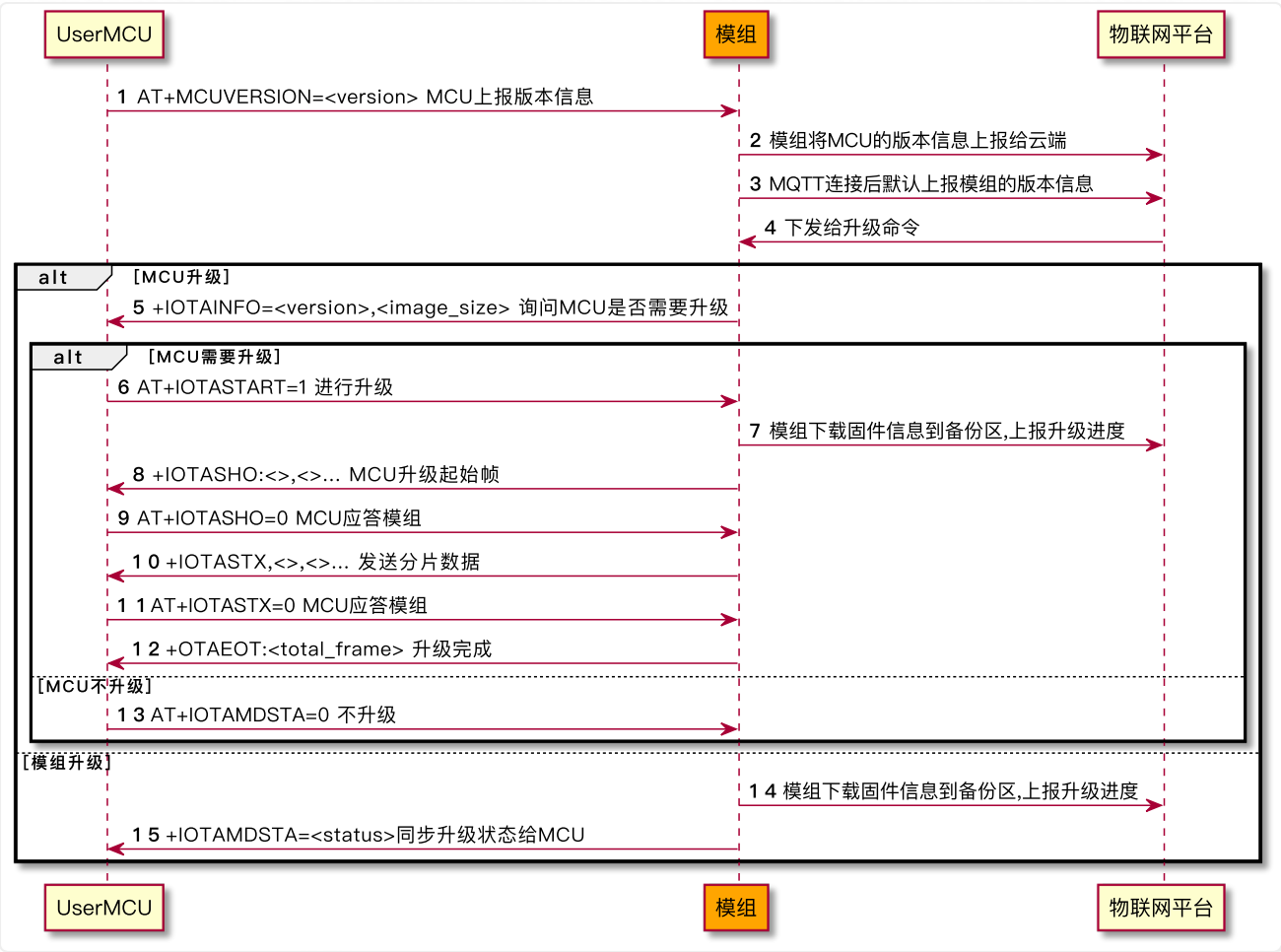


Haas210升级操作说明

1. 说明

Haas210模组可以支持在物联网平台上分别对模组和MCU进行独立的升级操作。

2. 升级交互流程



3. MCU升级操作

MCU通过模组实现OTA升级，有模组先将固件下载到模组内的备份区内，然后再分包发送给MCU。haas210进行升级时使用的是MQTT通道进行固件的下载，所以用户在在验证升级包和批量升级时需要选择MQTT方式升级如3.2和3.3说明。

3.1 物联网平台上传固件包

在物联网平台上的升级服务中默认支持的是MCU的升级操作添加升价包时升级包模块选择default，默认升级的是MCU固件。如下图红框内选项必须是default：

添加升级包

升级包类型

整包

差分

升级包名称

请输入升级包名称

所属产品

路灯

升级包模块

default

+ 添加模块

升级包版本号

请输入升级包版本号

签名算法

MD5

+

▼

选择升级包

上传升级包

升级包是否需要平台验证

是

否

升级包描述

请输入升级包描述

0/1024

推送给设备的自定义信息

请输入推送给设备的自定义信息

0/4096

升级包安全检测服务

确认

取消

3.2 验证升级包：

如下图，haas210使用MQTT协议下载升级固件，升级包下载协议必须选择MQTT，如果HTTP模组将不执行升级操作。

验证升级包

×

待升级版本号

请选择版本号

▽

* 待验证设备

请选择设备

▽

* 云端主动推送升级

☒ 是

☐ 否

* APP 确认升级

☐ 是

☒ 否

* 升级包下载协议

MQTT

▽

i

使用 MQTT 协议升级时下发给设备的升级消息格式跟 HTTPS 协议有差异，设备端需要采用支持 MQTT 协议下载升级包的 OTA SDK 及 MQTT 协议下载文件的 SDK。

设备升级超时时间（分钟） ?

请输入超时时间（分钟）

批次标签 ?

+ 新增标签

确定

取消

3.3 批量升级：

当进行批量升级时用户在升级策略配置页面，选择升级包下载协议，必须要选择MQTT，如果选择HTTP模组将不执行升级操作。

1 升级范围配置

2 升级策略配置

3 完成

* 升级时间

立即升级

* 云端主动推送升级 ?

☒ 是 ☐ 否

* 升级包推送速率 ?

请输入每分钟推送的设备数

* 升级失败重试间隔

不重试

设备升级超时时间 (分钟) ?

请输入超时时间 (分钟)

< 根据历史升级记录, 推荐的超时时间为 10 分钟

* 是否覆盖设备之前的升级任务 ?

☐ 是 ☒ 否

* APP确认升级 ?

☐ 是 ☒ 否

* 升级包下载协议

MQTT

i

使用 MQTT 协议升级时下发给设备的升级消息格式跟 HTTPS 协议有差异, 设备端需要采用支持 MQTT 协议下载升级包的 OTA SDK 及 MQTT 协议下载文件的 SDK。

上一步

完成

取消


本次批量升级共 1 个设备

3.4 MCU版本信息上报

MCU通过模组实现OTA升级, 有模组先将固件下载到模组内的备份区内, 然后再分包发送给 MCU。

3.4.1 MCU版本信息上报


C++

 复制代码

```
1 // 测试命令
2 AT+IOTAVER=?
3     +IOTAVER:"version"
4
5 // 查询格式
6 AT+IOTAVER?
7     +IOTAVER:<version>
8
9 // 通过模组上报MCU版本信息
10 AT+IOTAVER=<version>
11     OK
12     或者
13     version: MCU当前的版本信息
```

3.4.2 MCU 主动查询云端是否有升级部署


C++

 复制代码

```
1 // 测试命令
2 AT+IOTAQUERY=?
3     +IOTAQUERY:
4
5 // 查询格式
6 AT+IOTAQUERY?
7
8 // 查询云端是否有升级任务
9 AT+IOTAQUERY
10     OK
11 收到命令后先应答OK
```

3.4.3 模组收到云端下发的升级信息

C++

 复制代码

```
1 // 模组收到云端升级信息后，将升级信息发送给MCU
2     +IOTAINFO:<version>,<image_size>
3
4 // MCU主动轮询命令
5 AT+IOTAINFO?
6     +IOTAINFO:<version>,<image_size>
7 说明：正在进行MCU的固件下载时，应答下载固件的信息，无升级任务时应答为空
8 version:固件的版本信息，无应答时空
9 image_size:固件的大小，无应答时空
```

3.4.4 MCU应答模组开始升级

C++ | 复制代码

```

1 // 格式查询
2 AT+IOTASTART?
3     +IOTASTART:<enable>
4
5 // MCU收到+IOTAINF0后需要应答MCU是否需要升级
6 AT+IOTASTART=<enable>
7 enable:
8 0 放弃本次升级
9 1 开始升级, 模组开始下载固件并存储模组的flash备份区

```

3.4.5 模组传输固件给MCU

固件传输起始:

C++ | 复制代码

```

1 // 固件传输起始包: 模组-->MCU 发送文件头信息, 等待10S无应答会重新发送升级任务
2     +IOTASH0:<total_frame>,<frame_size>,<version>,<image_size>,<digest_method>,<expect_digest>
3     说明:
4         total_frame: 固件的总包数
5         frame_size: 每次发送固件的大小
6         image_size: 固件的总大小
7         digest_method: 固件的校验方式, md5或者sha256
8         expect_digest: 固件对应的校验码
9
10 // 格式查询
11 AT+IOTASH0?
12     +IOTASH0:<state>
13 // MCU应答ACK, 模组收到后开始发送+IOTASTX, 不应答超时3s重发
14 AT+IOTASH0=<state>
15     state:0 模组重发当前包数据
16           1 模组开始固件信息报文

```

固件传输分片数据:

C++

复制代码

```

1  // 固件传输:
2      +IOTASTX:<frame_num>,<data_size>,<crc16>,<data>
3  // 格式查询
4  AT+IOTASTX?
5      +IOTASTX:<state>,<frame_num>
6  // MCU应答0, 模组收到后开发送下一帧数据, 不应答超时3s重发
7  AT+IOTASTX=<state>,<frame_num>
8      state:0 模组重发当前包数据
9              1 模组开始发送下一包数据
10     frame_num: 收到当前的帧编号
11
12     说明:
13     frame_num: 发送固件包的序号, 从0~total_frame-1
14     data_size: data数据的长度
15     crc16: data数据的CRC校验
16     data: 数据包

```

固件传输完成:

C++

复制代码

```

1  // 固件传输结束包:
2      +IOTAEOT:<total_frame>
3
4  // MCU不需要应答
5      total_frame: 发送完成的总包数

```

4. 模组升级操作

4.1 模块列表添加模块

在升级模块选项中, 点击"添加模块", 选择对应的产品, 添加模块名称"**ali_hass210_module**" (模块名称必选否则无法升级)。

升级包列表					
模块列表					
数据分析					
添加模块					
全部产品					
请输入模块名称					
模块名称	模块别名	所属产品	添加时间	模块描述	操作

添加模块

模块可用于区分设备不同类型的升级包，详细使用说明请查看 [文档](#)

* 所属产品

tewt123

* 模块名称

ali_hass210_module

模块别名

请输入模块别名

模块描述

请输入模块描述

0/100

确认

取消

4.2 添加升级包

添加升级包是先选择所属产品，升级包模块选择 "ali_hass210_module"

签名算法：MD5

https://yuque.antfin.com/docs/share/58cb1bff-a9f5-46d6-9f3a-ad53d4d6a562?#

8/24

添加升级包

✕

* 升级包类型 ?

整包

差分

* 升级包名称 ?

请输入升级包名称

* 所属产品

tewt123

▼

* 升级包模块

ali_hass210_module

▼

+ 添加模块

* 升级包版本号 ?

请输入升级包版本号

* 签名算法

MD5

▼

* 选择升级包 ?

上传升级包

* 升级包是否需要平台验证 ?

☒ 是 ☐ 否

升级包描述

请输入升级包描述

0/1024

推送给设备的自定义信息 ?

请输入推送给设备的自定义信息

4.3 模组验证升级包：

如下图，haas210使用MQTT协议下载升级固件，升级包下载协议必须选择MQTT，如果HTTP模组将不执行升级操作。

验证升级包

×

待升级版本号

请选择版本号

▽

* 待验证设备

请选择设备

▽

* 云端主动推送升级

☒ 是

☐ 否

* APP 确认升级

☐ 是

☒ 否

* 升级包下载协议

MQTT

▽

i

使用 MQTT 协议升级时下发给设备的升级消息格式跟 HTTPS 协议有差异，设备端需要采用支持 MQTT 协议下载升级包的 OTA SDK 及 MQTT 协议下载文件的 SDK 。

设备升级超时时间（分钟）

?

请输入超时时间（分钟）

批次标签

?

+ 新增标签

确定

取消

4.4 模组批量升级：

当进行批量升级时用户在升级策略配置页面，选择升级包下载协议，必须要选择MQTT，如果选择HTTP模组将不执行升级操作。

1 升级范围配置

2 升级策略配置

3 完成

* 升级时间

立即升级

* 云端主动推送升级 ?

☒ 是

☐ 否

* 升级包推送速率 ?

请输入每分钟推送的设备数

* 升级失败重试间隔

不重试

设备升级超时时间 (分钟) ?

请输入超时时间 (分钟)

< 根据历史升级记录, 推荐的超时时间为 10 分钟

* 是否覆盖设备之前的升级任务 ?

☐ 是

☒ 否

* APP确认升级 ?

☐ 是

☒ 否

* 升级包下载协议

MQTT

使用 MQTT 协议升级时下发给设备的升级消息格式跟 HTTPS 协议有差异, 设备端需要采用支持 MQTT 协议下载升级包的 OTA SDK 及 MQTT 协议下载文件的 SDK。

上一步

完成

取消

本次批量升级共 1 个设备

5 MCU升级Demo

5.1 初始化模组和串口

C++

复制代码

```
1     int32_t res = 0;
2     aiot_sysdep_set_portfile(&g_aiot_sysdep_portfile);
3     aiot_state_set_logcb(demo_state_logcb);
4     /* 初始化模组参数配置 */
5     if ((res = aiot_module_init(AIOT_MODULE_ALI_HASS210)) !=
STATE_SUCCESS) {
6         goto exit;
7     }
8
9     /* 初始化串口收发 */
10    if ((res = at_hal_init()) != STATE_SUCCESS) {
11
12        printf("Open usart failed res = %d.\r\n", res);
13        goto exit;
14    }
```

5.2 MCU与模组进行AT命令握手

C++

复制代码

```
1     /* 等待与模组建立AT命令握手 */
2     if (aiot_module_handshake(10000) != STATE_SUCCESS) {
3         printf("With module handshake failed.\r\n");
4         goto exit;
5     }
```

5.3 配置WiFi和MQTT连接参数

C++ | 复制代码

```
1  /* 配置模组连接的WiFi信息 */
2  if (aiot_module_connect_wifi(AIOT_MODULE_ALI_HASS210,
    DEMO_WIFI_DEFAULT_SSID, DEMO_WIFI_DEFAULT_PSK) != STATE_SUCCESS) {
3      printf("aiot_module_connect_wifi failed.\r\n");
4      goto exit;
5  }
6  /* MQTT 连接参数配置和连接 */
7  aiot_mqtt_auth_t mqtt_auth = {
8      .product_key = product_key,
9      .device_name = device_name,
10     .device_secret = device_secret
11 };
12
13     aiot_mqtt_init(demo_default_recv_handler, demo_event_handler,
    NULL);
14     aiot_mqtt_auth(&mqtt_auth);
15     aiot_mqtt_connect(mqtt_host, 1883);
16     /* 订阅topic, 如果不指定 demo_topic_recv_handler 的回调, 接收到的消息都会会
    调到 demo_topic_recv_handler */
17     aiot_mqtt_sub("/a1P81YUbDih/at_module_test001/user/get",
    AIOT_MQTT_QOS0, demo_topic_recv_handler, NULL);
```

5.4 使能OTA升级, 上报MCU版本信息

C++ | 复制代码

```
1
2  /* 注册OTA功能回调函数 */
3  aiot_ota_init(demo_ota_recv_handle);
4  /* 上报MCU版本信息 */
5  aiot_ota_report_mcuverson(DEMO_CLIENT_VERSION);
```

5.5 循环下载固件

建立缓冲区从模组开始循环接收固件信息, 接收到分包的固件信息后用户需要自行写入到flash的备份区。

在升级过程中aiot_ota_request_firmware_package内会进行单包的CRC16和所有升级包的MD5校验, CRC16校验失败, 会在单包数据时通知用户。MD5校验失败会在最后一包通知用户。

```
1  /* 升级缓冲区 */
2      uint8_t buff[2048];
3      aiot_ota_packet_t package = {
4          .frame_num = 0,
5          .data_size = 0,
6          .data = buff,
7      };
8
9      while (1)
10     {
11         switch (demo_g_upgrade.state)
12         {
13             case AIOT_OTA_STA_WAIT_UPGRADE: {
14                 if (aiot_ota_is_start_upgrade(AIOT_OTA_ASSENT_VALUE) !=
STATE_SUCCESS) {
15                     continue;
16                 }
17                 demo_g_upgrade.frame_num = 0;
18                 demo_g_upgrade.frame_state = FRAME_HEAD;
19                 demo_g_upgrade.state = AIOT_OTA_STA_NONE;
20                 break;
21             }
22             case AIOT_OTA_STA_UPGRADE_MCU: {
23                 switch (demo_g_upgrade.frame_state)
24                 {
25                     case FRAME_HEAD:
26                         if
(aiot_ota_request_firmware_info(&demo_g_upgrade.info) !=
STATE_SUCCESS) {
27                             continue;
28                         }
29                         demo_g_upgrade.frame_state = FRAME_DATA;
30                         break;
31                     case FRAME_DATA:
32                         demo_g_upgrade.frame_state = FRAME_DATA;
33                         if
(aiot_ota_request_firmware_package(demo_g_upgrade.frame_num,
&package) != STATE_SUCCESS) {
34                             // 请求失败, 重新请求
35                             continue;
36                         }
37                         // TODO: 增加存储到flash
38
39                         // 请求下一包升级数据
40                         demo_g_upgrade.frame_num++;
41                         /* 升级完成 */
42                         if (demo_g_upgrade.frame_num ==
demo_g_upgrade.info.total_frame) {
43                             demo_g_upgrade.frame_state = FRAME_END;
44                         }
45                         break;
```

```
46         case FRAME_END:
47             // TODO: 升级结束
48             demo_g_upgrade.frame_state = FRAME_HEAD;
49             demo_g_upgrade.state = AIOT_OTA_STA_SUCCESS;
50             sleep(1);
51             break;
52         default:
53             break;
54     }
55 }
56 break;
57 case AIOT_OTA_STA_SUCCESS:
58     printf("Upgrade Success\r\n");
59     sleep(1);
60     // 上报新的Version
61     //
62     aiot_ota_report_mcuversion(demo_g_upgrade.info.version);
63     demo_g_upgrade.state = AIOT_OTA_STA_NONE;
64     break;
65 default:
66     break;
67 }
```

5.6 运行日志

```

1  [INF][aiot_module_init][6]:Start moduel init param.[INF]
   [at_client_para
2  nit][1030]:### Start OS init.
3  init mutex: 0x7fa61a405a20
4  init client->lock: 0x7fa61a405a20
5  init sem: 0x3
6  client->resp_sem : 0x3
7  Entry client_parser
8  fcntl=0
9  isatty success!
10 fd->open=4
11 [INF][_recv_readline][899]:<<<<
12 recvline > 0D 0A | ..

13
14 [INF][_recv_readline][899]:<<<<
15 recvline > 4F 4B 0D 0A | OK..

16
17 [INF][at_clie_obj_wait_connect][654]:Connect OK.
18 [INF][_vprintfn][65]:>>>>
19 sendline > 41 54 2B 49 52 45 42 4F 4F 54 0D 0A |
   AT+IREBOOT..
20
21 [INF][_recv_readline][899]:<<<<
22 recvline > 0D 0A | ..

23
24 [INF][_recv_readline][899]:<<<<
25 recvline > 4F 4B 0D 0A | OK..

26
27 [INF][_recv_readline][899]:<<<<
28 recvline > 00 0D 0A | ...

29
30 [INF][_recv_readline][899]:<<<<
31 recvline > 4F 4B 0D 0A | OK..

32
33 [INF][at_clie
34 _obj_wait_connect][654]:Connect OK.
35 [INF][
36 _vprintfn][65]:>>>>
37 sendline > 41 54 2B 43 57 4D 4F 44 45 3D 31 0D 0A |
   AT+CWMODE=1..
38
39 [INF][
40 _recv_readline][899]:<<<<
41 recvline > 0D 0A | ..

```



```
42
43 [INF] [
44 _recv_readline][899]:<<<<
45 recvline > 4F 4B 0D 0A | OK..

46
47 [INF] [
48 _vprintfln][65]:>>>>
49 sendline > 41 54 2B 43 57 4A 41 50 3D 6C 69 6E 6B 6B 69 74 |
AT+CWJAP=linkkit
50 sendline > 2C 6C 6B 31 32 33 34 35 36 0D 0A |
, lk123456..

51
52 [INF] [
53 _recv_readline][899]:<<<<
54 recvline > 0D 0A | ..

55
56 [INF] [
57 _recv_readline][899]:<<<<
58 recvline > 4F 4B 0D 0A | OK..

59
60 [INF] [
61 _vprintfln][65]:>>>>
62 sendline > 41 54 2B 49 4E 45 54 53 54 41 54 0D 0A |
AT+INETSTAT..

63
64 [INF] [
65 _recv_readline][899]:<<<<
66 recvline > 0D 0A | ..

67
68 [INF] [
69 _recv_readline][899]:<<<<
70 recvline > 2B 49 4E 45 54 53 54 41 54 3A 32 2C 30 2E 30 2E |
+INETSTAT:2,0.0.
71 recvline > 30 2E 30 2C 42 34 3A 30 45 3A 43 46 3A 31 31 3A |
0.0,B4:0E:CF:11:
72 recvline > 38 43 3A 46 35 0D 0A |
8C:F5..

73
74 [INF] [
75 _recv_readline][899]:<<<<
76 recvline > 0D 0A | ..

77
78 [INF] [
79 _recv_readline][899]:<<<<
80 recvline > 4F 4B 0D 0A | OK..

81
82 [INF] [
```

```
83 ifi_status][111]:<<<<
84 [INF][_wifi_status][117]:WIFI St
85 e:2,ip:0.0.0.0,bssid:
86 [INF][_vprintfln][65]:>>>>
87 sendline > 41 54 2B 49 4E 45 54 53 54 41 54 0D 0A |
AT+INETSTAT..

88
89 [INF][_recv_readline][899]:<<<<
90 recvline > 0D 0A | ..

91
92 [INF][_recv_readline][899]:<<<<
93 recvline > 2B 49 4E 45 54 53 54 41 54 3A 33 2C 30 2E 30 2E |
+INETSTAT:3,0.0.
94 recvline > 30 2E 30 2C 42 34 3A 30 45 3A 43 46 3A 31 31 3A |
0.0,B4:0E:CF:11:
95 recvline > 38 43 3A 46 35 0D 0A |
8C:F5..

96
97 [INF][_recv_readline][899]:<<<<
98 recvline > 0D 0A | ..

99
100 [INF][_recv_readline][899]:<<<<
101 recvline > 4F 4B 0D 0A | OK..

102
103 [INF][ifi_status][111]:<<<<
104 [INF][_wifi_status][117]:WIFI St
105 e:3,ip:0.0.0.0,bssid:
106 [INF][_recv_readline][899]:<<<<
107 recvline > 0D 0A | ..

108
109 [INF][_recv_readline][899]:<<<<
110 recvline > 2B 49 4E 45 54 53 54 41 54 3A 34 2C 31 39 32 2E |
+INETSTAT:4,192.
111 recvline > 31 36 38 2E 31 2E 31 30 30 2C 42 34 3A 30 45 3A |
168.1.100,B4:0E:
112 recvline > 43 46 3A 31 31 3A 38 43 3A 46 35 0D 0A |
CF:11:8C:F5..

113
114 [INF][_vprintfln][65]:>>>>
115 sendline > 41 54 2B 49 4E 45 54 53 54 41 54 0D 0A |
AT+INETSTAT..

116
117 [INF][_recv_readline][899]:<<<<
118 recvline > 0D 0A | ..

119
120 [INF][_recv_readline][899]:<<<<
121 recvline > 2B 49 4E 45 54 53 54 41 54 3A 34 2C 31 39 32 2E |
+INETSTAT:4,192.
```

```
122  recvline > 31 36 38 2E 31 2E 31 30 30 2C 42 34 3A 30 45 3A |
123  168.1.100,B4:0E:
124  recvline > 43 46 3A 31 31 3A 38 43 3A 46 35 0D 0A |
125  CF:11:8C:F5..
126  [INF][_recv_readline][899]:<<<<
127  recvline > 0D 0A | ..
128  [INF][
129  _recv_readline][899]:<<<<
130  recvline > 4F 4B 0D 0A | OK..
131  [INF][ifi_status][111]:<<<<
132  [INF][_wifi_status][117]:WIFI State:4,
133  :192.168.1.100,bssid:
134  init mutex: 0x7fa61a7040b0
135  [INF][_vprintfln][65]:>>>>
136  sendline > 41 54 2B 49 4D 51 54 54 41 55 54 48 3D 61 31 50 |
137  AT+IMQTTAUTH=a1P
138  sendline > 38 ** ** ** 62 44 69 68 2C 61 74 5F 6D 6F 64 75 |
139  ****bDih,at_modu
140  sendline > 6C 65 5F 74 65 73 74 30 30 31 2C 65 32 37 39 39 |
141  le_test001,e2799
142  sendline > 30 62 37 30 37 63 66 66 65 64 35 65 62 32 31 34 |
143  0b707cffed5eb214
144  sendline > 31 36 30 32 32 65 32 35 61 66 63 0D 0A |
145  16022e25afc..
146  [INF][_recv_readline][899]:<<<<
147  recvline > 0D 0A | ..
148  [INF][
149  _recv_readline][899]:<<<<
150  recvline > 4F 4B 0D 0A | OK..
151  [INF][_vprintfln][65]:>>>>
152  sendline > 41 54 2B 49 4D 51 54 54 43 4F 4E 4E 3D 61 31 50 |
153  AT+IMQTTCONN=a1P
154  sendline > 38 ** ** ** 62 44 69 68 2E 69 6F 74 2D 61 73 2D |
155  81***Dih.iot-as-
156  sendline > 6D 71 74 74 2E 63 6E 2D 73 68 61 6E 67 68 61 69 |
157  mqtt.cn-shanghai
158  sendline > 2E 61 6C 69 79 75 6E 63 73 2E 63 6F 6D 2C 31 38 |
159  .aliyuncs.com,18
160  sendline > 38 33 0D 0A | 83..
161  [INF][_recv_readline][899]:<<<<
```

```

158  recvline > 0D 0A | ..
159
160  [INF][_recv_readline][899]:<<<<
161  recvline > 2B 49 4D 51 54 54 53 54 41 54 45 3A 31 0D 0A |
+IMQTTSTATE:1..
162
163  [TRA][aiot_mqtt_event_handler][57
164  :aiot_mqtt_event_handler size:15
165  [INF][aiomqtt_event_handler][575]:res = 1
166  demo event handle event:1
167  AIOT_MQTTEVT_CONNECT
168  [INF][aiot_mqtt_event_ndler][581]:Receive mqtt state:1
169  [INF][_recv_readline][899]:<<<<
170  recvline > 0D 0A | ..
171
172  [INF][_recv_readline][899]:<<<<
173  recvline > 4F 4B 0D 0A | OK..
174
175  [INF][_vprintfln][65]:>>>>
176  sendline > 41 54 2B 49 4D 51 54 54 53 55 42 3D 2F 61 31 50 |
AT+IMQTTSUB=/a1P
177  sendline > 38 ** ** ** 62 44 69 68 2F 61 74 5F 6D 6F 64 75 |
****bDih/at_modu
178  sendline > 6C 65 5F 74 65 73 74 30 30 31 2F 75 73 65 72 2F |
le_test001/user/
179  sendline > 67 65 74 2C 30 0D 0A |
get,0..
180
181  [INF][_recv_readline][899]:<<<<
182  recvline > 0D 0A | ..
183
184  [INF][_recv_readline][899]:<<<<
185  recvline > 2B 49 4D 51 54 54 53 55 42 3A 31 0D 0A |
+IMQTTSUB:1..
186
187  [INF][_recv_readline][899]:<<<<
188  recvline > 0D 0A | ..
189
190  [INF][_recv_readline][899]:<<<<
191  recvline > 4F 4B 0D 0A | OK..
192
193  [INF][_vprintfln][65]:>>>>
194  sendline > 41 54 2B 49 4F 54 41 56 45 52 3D 31 2E 30 2E 30 |
AT+IOTAVER=1.0.0
195  sendline > 0D 0A | ..
196

```

```
197 [INF][_recv_readline][899]:<<<<
198 recvline > 0D 0A | ..

199
200 [INF][_recv_readline][899]:<<<<
201 recvline > 2B 49 4D 51 54 54 53 55 42 41 43 4B 3A 31 2C 30 |
+IMQTTSUBACK:1,0
202 recvline > 0D 0A | ..

203
204 [TRA][_urc_suback_handler][52
205 :_urc_suback_handler size:18
206 [INF][_c_suback_handler][531]:res=2
207 AIOT_MQTTRECV_SUB_ACK sub ack packet id:1 res:0
208 [INF][_recv_readline][899]:<<<<
209 recvline > 0D 0A | ..

210
211 [INF][_recv_readline][899]:<<<<
212 recvline > 4F 4B 0D 0A | OK..

213
214 [INF][_recv_readline][899]:<<<<
215 recvline > 2B 49 4F 54 41 53 54 41 54 45 3A 30 2C 31 2E 31 |
+IOTASTATE:0,1.1
216 recvline > 2E 31 2C 34 37 32 36 34 0D 0A |
.1,47264..

217
218 [INF][aiot_ota_recv_handler346]:Recv Fota message size:26
219 [INF][aiot_ota_rechandler][355]:Recv OTA state:0
220
221 [INF][_vprintfn][65]:>>>>
222 sendline > 41 54 2B 49 4F 54 41 53 54 41 52 54 3D 31 0D 0A |
AT+IOTASTART=1..

223
224 [INF][_recv_readline][899]:<<<<
225 recvline > 0D 0A | ..

226
227 [INF][_recv_readline][899]:<<<<
228 recvline > 4F 4B 0D 0A | OK..

229
230 [INF][_recv_readline][899]:<<<<
231 recvline > 2B 49 4F 54 41 53 54 41 54 45 3A 31 0D 0A |
+IOTASTATE:1..

232
233 [INF][aiot_ota_recv_handler346]:Recv Fota message size:14
234 [INF][aiot_ota_rechandler][355]:Recv OTA state:1
235
236 [INF][_recv_readline][899]:<<<<
237 recvline > 0D 0A | ..
```

```
238
239 [INF][_recv_readline][899]:<<<<
240 recvline > 2B 49 4D 51 54 54 53 55 42 41 43 4B 3A 32 2C 30 |
+IMQTTSUBACK:2,0
241 recvline > 0D 0A | ..

242
243 [TRA][_urc_suback_handler][52:_urc_suback_handler size:18
244 [INF][_c_suback_handler][531]:res=2
245
246 [INF][_recv_readline][899]:<<<<
247 recvline > 2B 49 4F 54 41 53 54 41 54 45 3A 32 0D 0A |
+IOTASTATE:2..

248
249 [INF][aiot_ota_recv_handler346]:Recv Fota message size:14
250 [INF][aiot_ota_rechandler][355]:Recv OTA state:2
251
252 [INF][_recv_readline][899]:<<<<
253 recvline > 2B 49 4F 54 41 53 54 41 54 45 3A 35 0D 0A |
+IOTASTATE:5..

254
255 [INF][aiot_ota_recv_handler346]:Recv Fota message size:14
256 [INF][aiot_ota_rechandler][355]:Recv OTA state:5
257
258 [INF][_vprintfln][65]:>>>>
259 sendline > 41 54 2B 49 4F 54 41 53 48 4F 0D 0A |
AT+IOTASH0..

260
261 [INF][_recv_readline][899]:<<<<
262 recvline > 2B 49 4F 54 41 53 48 4F 3A 31 38 35 2C 32 35 36 |
+IOTASH0:185,256
263 recvline > 2C 31 2E 31 2E 31 2C 34 37 32 36 34 2C 30 2C 64 |
,1.1.1,47264,0,d
264 recvline > 33 36 63 38 30 32 39 66 65 36 64 61 36 61 36 63 |
36c8029fe6da6a6c
265 recvline > 31 32 32 63 65 61 32 38 62 38 64 61 36 62 39 0D |
122cea28b8da6b9.
266 recvline > 0A | .

267
268 [INF][_recv_readline][899]:<<<<
269 recvline > 0D 0A | ..

270
271 [INF][_recv_readline][899]:<<<<
272 recvline > 4F 4B 0D 0A | OK..

273
274 [INF][aiot_ota_reest_firmware_info][148]:total_frame:185
275 [INF][aiot_ota_ruest_firmware_info][149]:frame_size:256
276 [INF][aiot_ota_quest_firmware_info][150]:version:1.1.1
277 [INF][aiot_ota_reqst_firmware_info][151]:image_size:47264
278 [INF][aiot_ota_reest_firmware_info][152]:digest_method:0
```

```
279 [INF][aiot_ota_request_firmware_info]
[153]:expecdigest:d36c8029fe6da6a6c122cea28b8da6b9
280 [INF][_vprintfln][65]:>>>>
281 sendline > 41 54 2B 49 4F 54 41 53 54 58 3D 30 0D 0A |
AT+IOTASTX=0..
282
283 [INF][_recv_readline][899]:<<<<
284 recvline > 2B 49 4F 54 41 53 54 58 3A 30 2C 32 35 36 2C 34 |
+IOTASTX:0,256,4
285 recvline > 32 31 36 34 2C 97 21 02 1F 93 81 01 80 73 70 04 |
2164,..!.....sp.
286 recvline > 30 97 32 00 00 93 82 42 1B 93 E2 32 00 73 90 52 |
0.2....B...2.s.R
287
288
289 .....
290 .....
291 .....
292 .....
293 .....
294 .....
295
296
297 sendline > 41 54 2B 49 4F 54 41 53 54 58 3D 31 38 34 0D 0A |
AT+IOTASTX=184..
298
299 [INF][_recv_readline][899]:<<<<
300 recvline > 2B 49 4F 54 41 53 54 58 3A 31 38 34 2C 31 36 30 |
+IOTASTX:184,160
301 recvline > 2C 35 36 39 33 34 2C 00 00 00 00 00 00 00 00 |
,56934,.....
302 recvline > 00 00 00 00 00 00 00 00 00 00 00 00 00 01 |
.....
303 recvline > 00 00 00 80 84 1E 00 03 01 00 40 00 00 00 00 |
.....@.....
304 recvline > 00 00 00 00 00 00 00 AC 10 02 42 00 00 00 00 |
.....B.....
305 recvline > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
306 recvline > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
307 recvline > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
308 recvline > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
309 recvline > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
310 recvline > 00 00 00 00 00 00 00 00 00 00 0C 11 02 42 0C |
.....B.
311 recvline > 11 02 42 FF 00 00 00 0D 0A |
..B.....
312
313 [INF][_recv_readline][899]:<<<<
```

```

314 recvline > 0D 0A | ..
315
316 [INF][_recv_readline][899]:<<<<
317 recvline > 4F 4B 0D 0A | OK..
318
319 data > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
320 data > 00 00 00 00 00 00 00 00 01 00 00 00 80 84 1E 00 |
.....
321 data > 03 01 00 40 00 00 00 00 00 00 00 00 00 00 00 00 |
...@.....
322 data > AC 10 02 42 00 00 00 00 00 00 00 00 00 00 00 00 |
...B.....
323 data > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
324 data > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
325 data > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
326 data > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
327 data > 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
.....
328 data > 00 00 00 00 0C 11 02 42 0C 11 02 42 FF 00 00 00 |
.....B...B....
329
330 [INF][aiot_ota_reqst_firmware_package][286]:ota package info
331 [INF][aiot_ota_quest_firmware_package][287]:frame_num:184
332 [INF][aiot_ota_quest_firmware_package][288]:data_size:160
333 [INF][aiot_otaquest_firmware_package][289]:crc16:0xDE66
334 [INF][aiot_ota_request_rmware_package][316]:Digest verify success
335 [INF][_recv_readline][899]:<<<<
336 recvline > 2B 49 4F 54 41 45 4F 54 3A 31 38 35 0D 0A |
+IOTAEOT:185..
337
338 [INF][aiot_ota_recv_handler346]:Recv Fota message size:14
339
340 [INF][_recv_readline][899]:<<<<
341 recvline > 0D 0A | ..
342
343 [INF][_recv_readline][899]:<<<<
344 recvline > 2B 49 4D 51 54 54 55 4E 53 55 42 41 43 4B 3A 33 |
+IMQTTUNSUBACK:3
345 recvline > 0D 0A | ..
346
347 Upgrade Success

```