Realtek Ameba-1 QQLink user guide

The document describes how to put QQLink into Ameba SDK and demonstrates how to use mobile QQ to connect Ameba to QQ cloud and how to update new image over the air.

Table of Contents

1 Introduction to QQLink

A smart device has to be bound together with your mobile QQ before connecting to QQ cloud service for the first time. Before bind procedure, the smart device must first connect to a target AP which has connected to the network. QQLink is one way help the smart device to connect to the target AP and then the cloud service.

A smart device using QQLink receives and analysis configuration packets sent from your mobile QQ. By doing this for a short moment, it can get enough information of the target AP. After that, the smart device will connect to the target AP automatically.

QQLink works on promiscuous mode and can only receive packets while can't transmit any. So it is not able to communicate with AP or mobile phone proactively and to ensure on which channel that mobile QQ transmits configuration packets. Based on above reasons, QQLink need to adjust channels between 1 and 13 circularly to lock the exact channel.

The table below shows some key words and their descriptions in this document.

Key word	Description
Mobile QQ	QQ application on your mobile phone. QQ version : Android > 5.7, iOS > 5.7
PID	Product id : one PID identifies an unique type of products.
SN	Serial number : one SN identifies an unique device of one PID

It can be seen that the combination of "PID && SN" can identify an unique smart device. Different devices of the same type must have different SN and the same PID. To bind a smart device, the mobile QQ need to scan a QR code consists of a PID and a SN that identify the smart device.

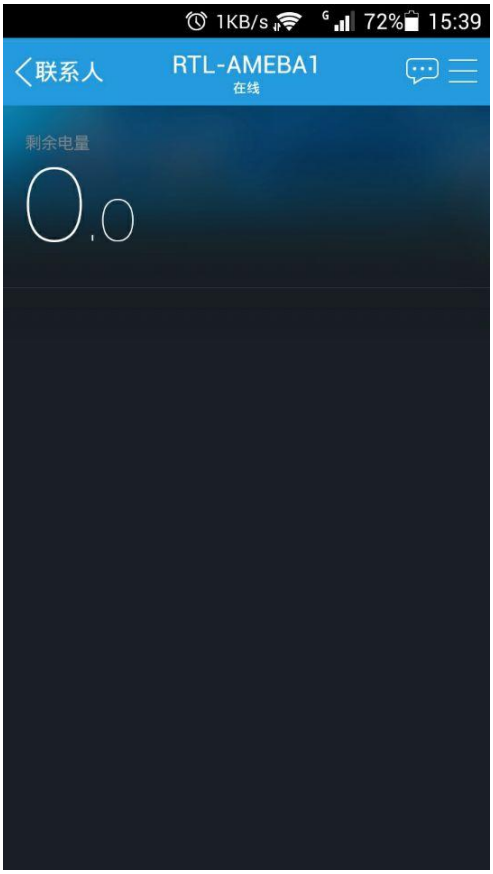
The picture below is an example of Ameba's QR code (RTL8711AF, SN : 04F94177669D4027):

On the developer platform, we default enabled RTL-AMEBA1 with two operable functions: “device sends message to mobile QQ(ID: 9999)” and “mobile QQ sends text message to device(ID: 10000)”, as shown below. Mobile QQ is able to send “text message” to smart device and vice versa.

The function “battery left(ID: 10101)” is also enabled as an example to show in controller interface, because it is visible to users which means users can see it in controller interface while the two functions mentioned above is invisible.

ID	功能名称	功能描述	修改日期	操作
9999	设备给手机QQ发消息	给手机QQ发消息，...	2016-01-19 16:52:41	
10000	手机QQ给设备发文本消息	请务必对是否支持进...	2016-01-19 16:52:41	
10001	手机QQ给设备发视频消息	请务必对是否支持进...	2016-01-19 16:52:41	
10002	手机QQ给设备发图片消息	请务必对是否支持进...	2016-01-19 16:52:41	
10003	手机QQ给设备发语音消息	请务必对是否支持进...	2016-01-19 16:52:41	
10015	手机QQ给设备发送文件	手机QQ 6.2及以上生...	2016-01-19 16:52:41	
10101	剩余电量	剩余电量	2016-01-22 14:47:56	
10102	退出控制界面后断开连接(BLE)	退出控制界面后断开...	2016-01-22 13:58:34	

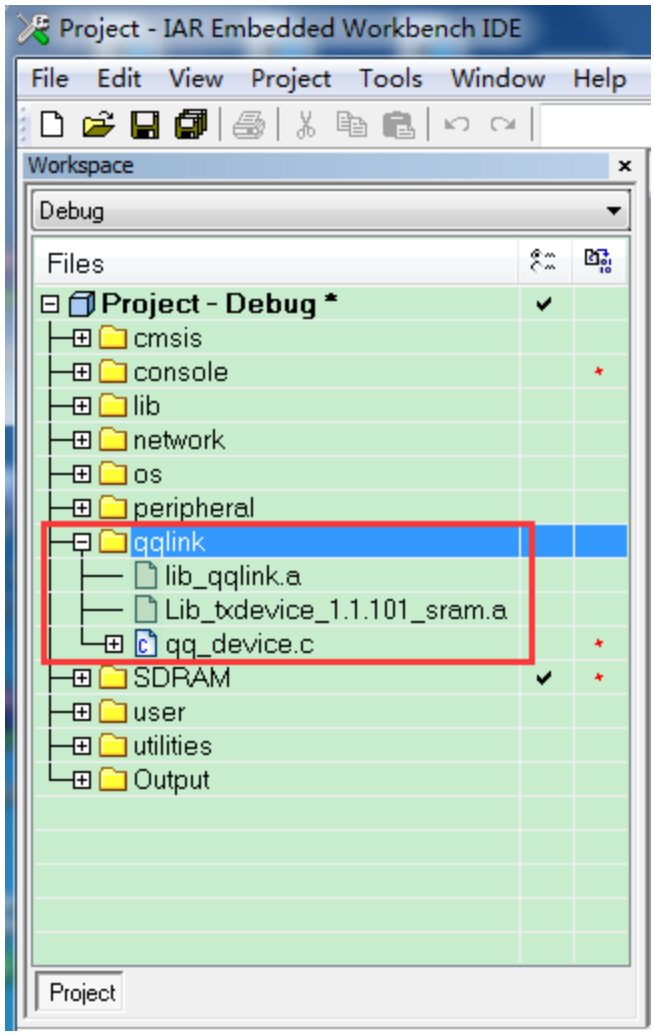
To operate a smart device by controller interface on mobile QQ, you need to offer a URL of HTML5 file for controlling your products. As an example, a default generated controller interface is shown below.



To know more details about QQ cloud service and to configure more function options, you can login "http://iot.open.qq.com/".

2 Porting QQLink patch into Ameba's project

Step 1: Extract Ameba released SDK, as sdk-ameba1-v3.4b for example.
Step 2: Extract all content from 3.4b_patch_qqlink.zip to the newly built folder "$sdk\component\common\application\qqlink."
Step 3: Open IAR project and add a new group **"qqlink"**.
Step 4: Add **"qq_device.c"** located at "$sdk\component\common\application\qqlink\" into qqlink group.



Step 5: Add into qqlink group with "lib_qqlink.a" located at "$sdk\component\common\application\qqlink\lib\" and "Lib_txdevice_1.1.101_sram.a" if RTL8711AF chip is used or "Lib_txdevice_1.1.101_sdram.a" if RTL8711AM / RTL8195AM chip is used. The picture above is an example of using RTL8711AF chip.

Step 6: Open "$sdk\project\realtek_ameba1_va0_example\inc\platform_opts.h", add "#define CONFIG_QQ_LINK 1" if the define does not exist or just change to 1 if the define has already existed.

```
#define CONFIG_QQ_LINK          1
```

It is better to close the following define since fast connect function has been included in qqlink.c.

```
#define CONFIG_EXAMPLE_WLAN_FAST_CONNECT   0
```

Step 7: Open $sdk\component\common\example\example_entry.c, in function example_entry(), add define as below if the define does not exist.

```
void example_entry(void)
{
    ... ...
#if CONFIG_QQ_LINK
    example_qq_link();
#endif
    ... ...
}
```

Step 8: Open $sdk\component\common\utilities\update.c, change to 1 if the define below is 0.

```
#define WRITE_OTA_ADDR      1
```

Now you can build and download image into Ameba and start QQLink journey.

3 Example: connect Ameba to QQ cloud

Use QQLink version 1.1.101 to demonstrate.

Step 1: Get a SN and its corresponding LICENSE.

In folder “$sdk\component\common\application\qqlink\sn\”, there is an example SN 04F94177669D4027. Its corresponding LICENSE is in “licence.sign.file[04F94177669D4027].txt”.

Attention : if this example SN cannot be used, please contact Realsil for another one.

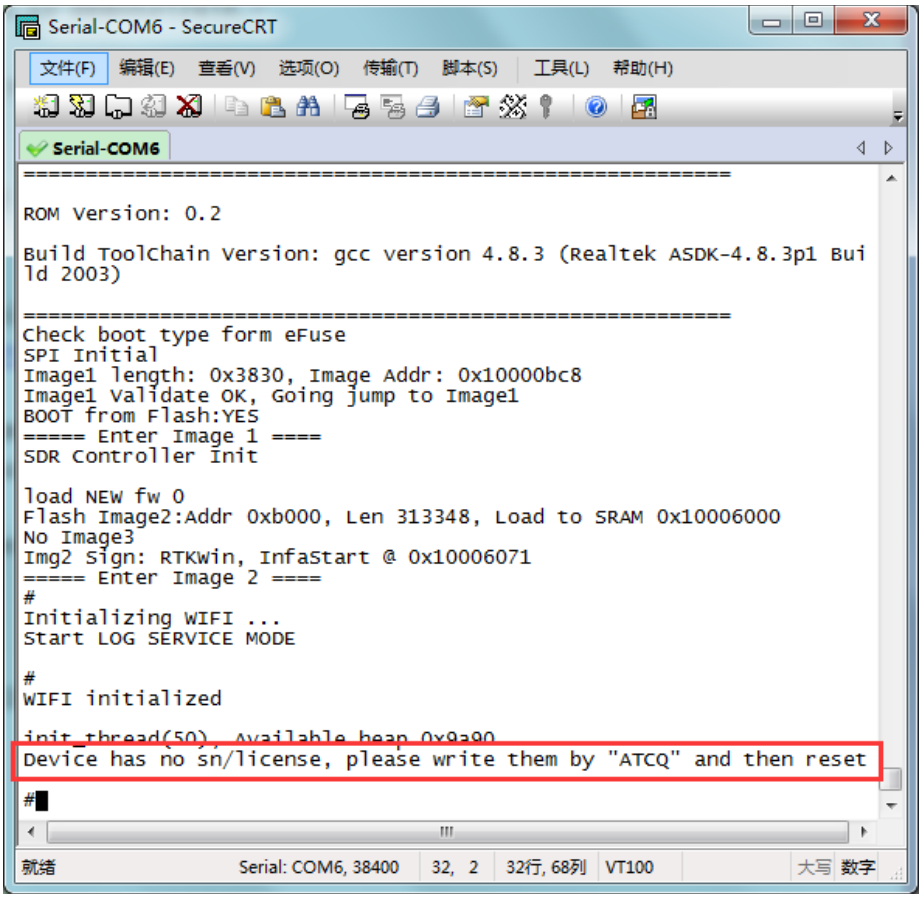
Step 2: Generate a QR code using the content template as below.

http://iot.qq.com/add?pid=1700002458&sn=xxxxxxxx

Replace “xxxxxxxx” by SN got in step 1.

Step 3: Power on your development board.

If the device is used at the first time and has no “SN/LICENSE” yet, you need to write them into flash first.

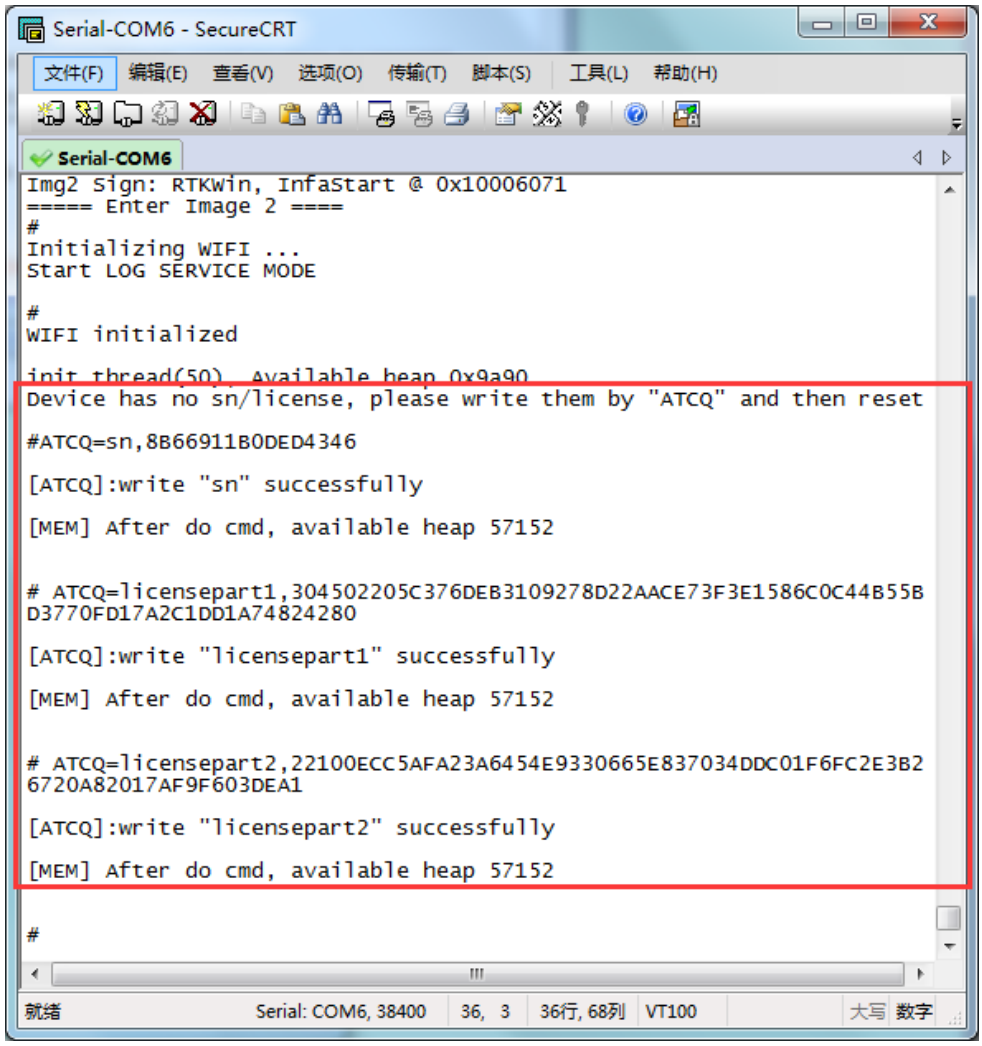


Step 4: Write SN and its corresponding LICENSE into flash.

Write SN first and then LICENSE by using "ATCQ". Input "ATCQ" to see how to use it.

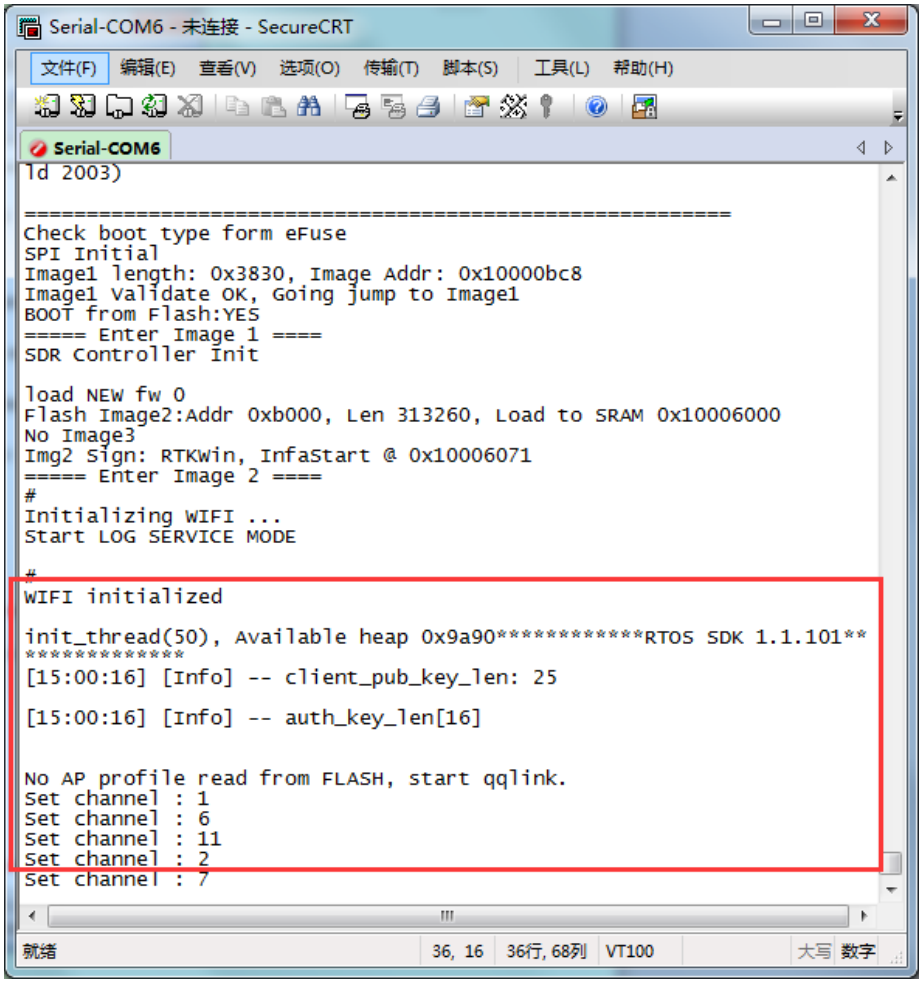
Because the LICENSE is too long, you need to divide it into two parts and write them respectively and sequentially.

Here is an example.



Step 5: Reset your development board.

"**********RTOS SDK 1.1.101**********" is normally shown, and scanning channels begins.



Step 6: Make your **phone wifi connected** to some AP with internet.

Step 7: Install Mobile QQ on your smart phone. **The version of mobile QQ should be no less than 5.7.**

Step 8: **Run your mobile QQ.**

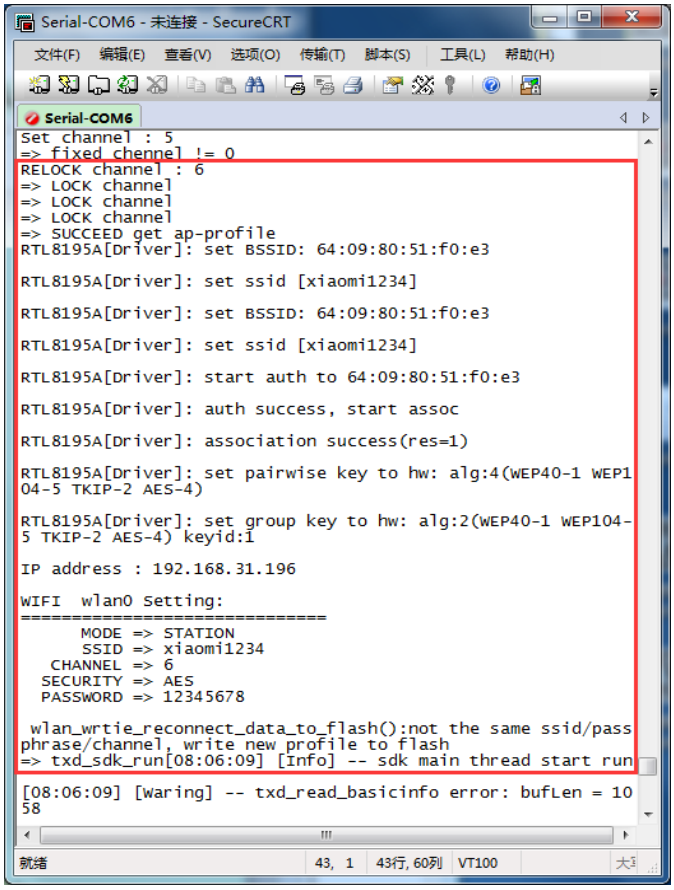
As shown below, login QQ and click **"find new devices"** and then **"scan"** to scan QR code generated in step 2.

Step 9: After scanning, target AP's SSID will be shown. Input its passphrase.

Step 10: Click "join" button to start QQLink.

Step 11: Then Ameba will get the target AP's profile and **connect to the target AP.**

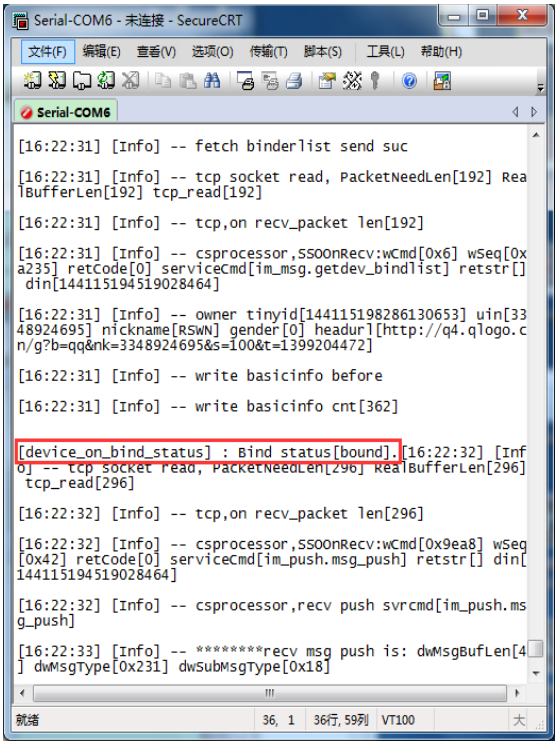
As log shows, after smart device locked the correct channel, it will connect to the target AP and gain an IP address.



Step 12: When “txd_sdk_run” is shown, QQ will change to “bind confirm” page. Please click “**agree to bind**”.

Step 13: Here, we **bind device successfully.**

Device's log shows "[device_on_bind_status] : Bind status[bound]."



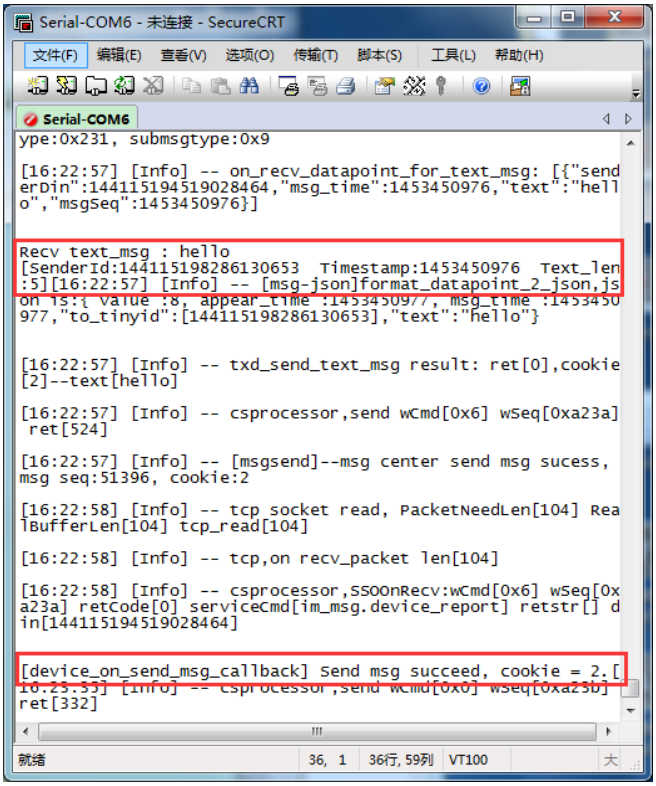
Step 14: Click “enter present device” to **enter controller interface.**

It is an example here, you can add more control functions in your own HTML5 file.

Step 15: Click “**message icon**” on the top right corner.

You can talk to the smart device now.

Step 16: Say “hello” to smart device, smart device receives it and then says the same content back.



4 OTA

OTA function is provided to upgrade the new firmware.

As an example for OTA test, a new firmware “ota.bin” with a higher assigned version num 102 is placed in the cloud. The current assigned version num in qq_device.c is 100.

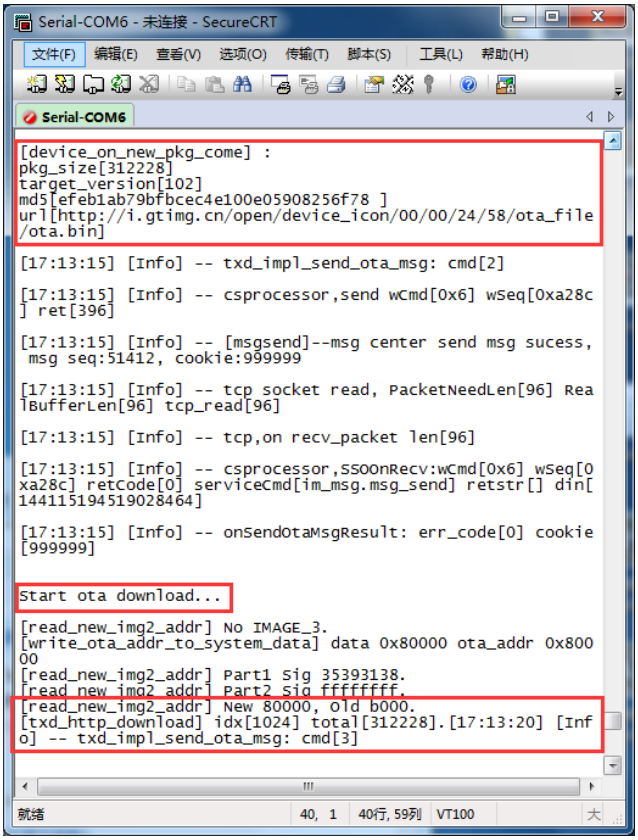
Step 1: When a higher version of firmware is detected by mobile QQ, it prompts “check device update NEW”. Click it to get details about the new firmware.

Step 2: Click “download update file” to start downloading the new firmware.

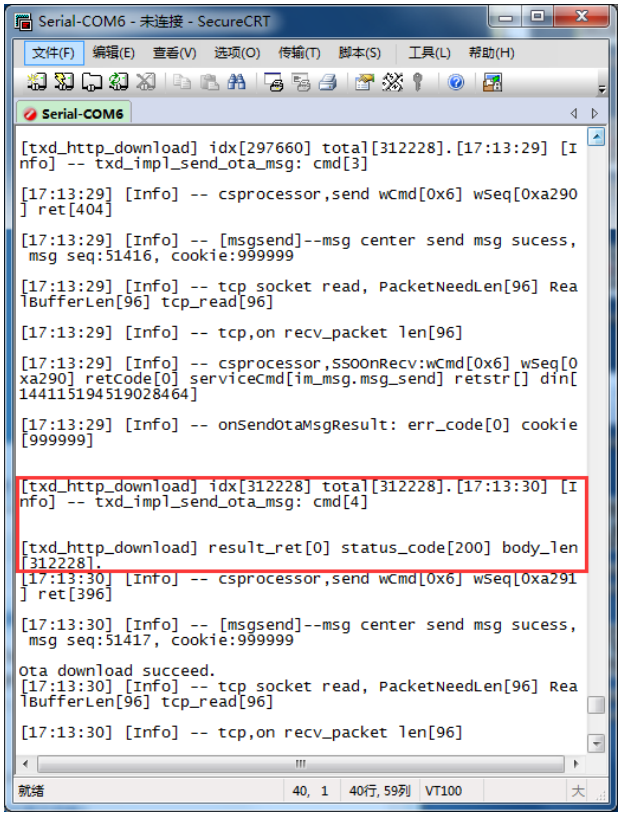
As device’s log shows, the target firmware is 312228 bytes large with version num 102.

“idx[1024] total[312228]” means 1024 bytes are downloaded and total size is 312228 bytes.

By the way, you can stop downloading the new firmware before download procedure is completed by clicking  .



Step 3: After completing download, “reset device to install update” appears. As device’s log shows, 312228 bytes are completly downloaded.

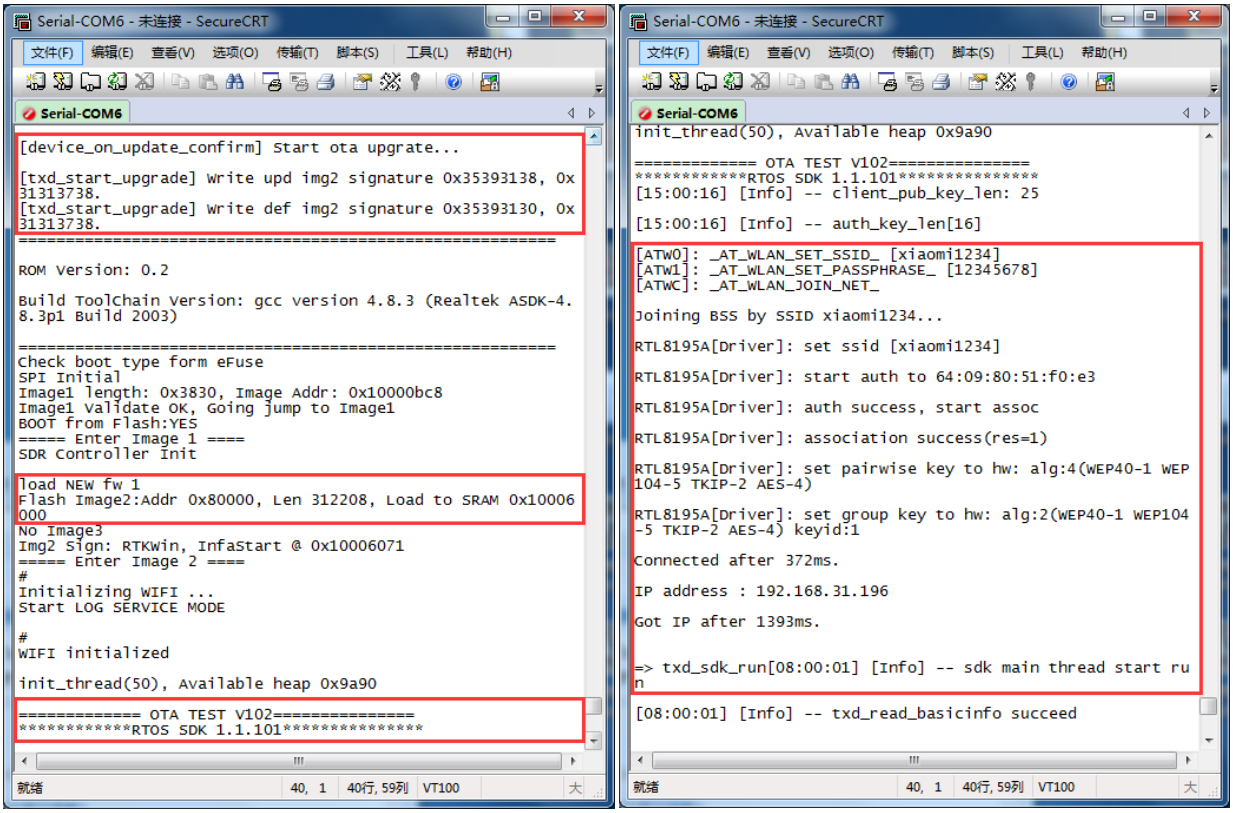


Step 4: Click “reset device to install update” and “confirm” to upgrade.

Step 5: After Clicking “confirm”, device will reset to **reload the new firmware.**

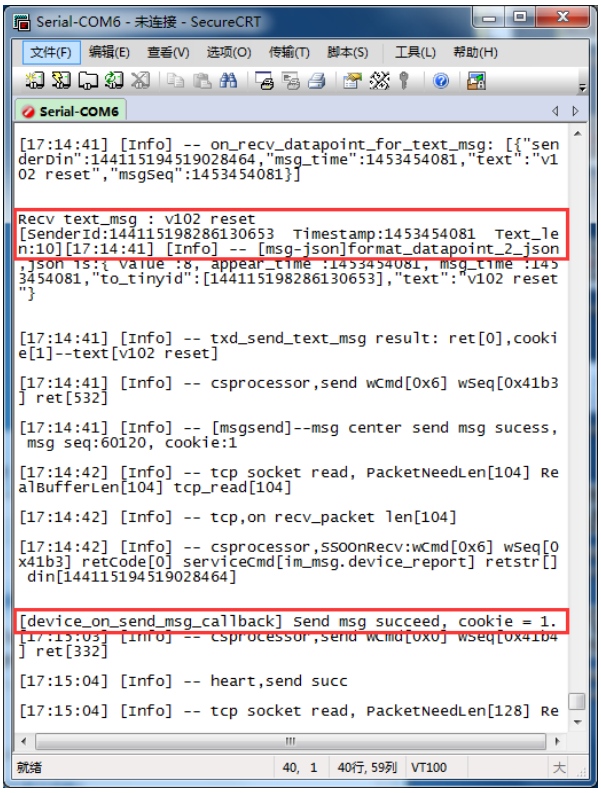
As left picture shows, the new firmware(Flash Image2) located at new address **ox80000** is entered and “============ OTA TEST V102==============” is printed.

As the right picture shows, after reset, device will connect to sap automatically by reading target ap’s profile from flash.



Step 6: After upgrading to the new firmware, mobile QQ prompts **“update accomplish”**.

Step 7: Say “v102 reset” to smart device, smart device receives it and then says the same content back.



Step 8: Click “check device update” again, it shows that “102” is already the newest version.