Realtek Ameba-1 JoyLink user guide

The document describes how to porting JoyLink into Ameba SDK, to use JD APK to connect Ameba to JD cloud and to update new firmware over the air.

Table of Contents

1 Introduction

A smart device has to be bound together with JD APK or your own controller APK before connecting to JD cloud service at the first time. Before bind procedure, the smart device must first connect to a target AP which has connected to the network. After that, it can connect to JD cloud through the AP.

To accomplish the above procedures, you need a controller APK named “京东微联” to bind and control the smart device.

Notice : you may have to install the APK with a version num no less than **v3.3.0**.

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

Key word	Description
uuid	Static device ID. An uuid identifies an unique type of products.
feedid	Dynamic device ID. In bind procedure, device will be distributed a new dynamic feedid.
accesskey	An encryption key used in authentication procedure with JD cloud.
localkey	An encryption key used in LAN communication.
version	Device’s firmware version number.

Before using JD cloud, we should register an account as a developer. The website is http://devsmart.jd.com/dev/index. After approved by JD administrator, one type of your device will be assigned an unique uuid and a QR code which can identify the device. Then you can create device information and control parameters on developer platform.

The picture below is an example of Ameba's QR code (RTL8195AM, uuid : 7YGP6H) :

RTL8195A

To operate a smart device by the control interface on a controller APK, you need to offer an URL of HTML5 file on developer platform for controlling your device. As an example, a default generated RTL8195A controller interface is shown while no HTML5 file is offered.

To know more details about JD cloud service and to configure more function options, you can login JD smart cloud official website.

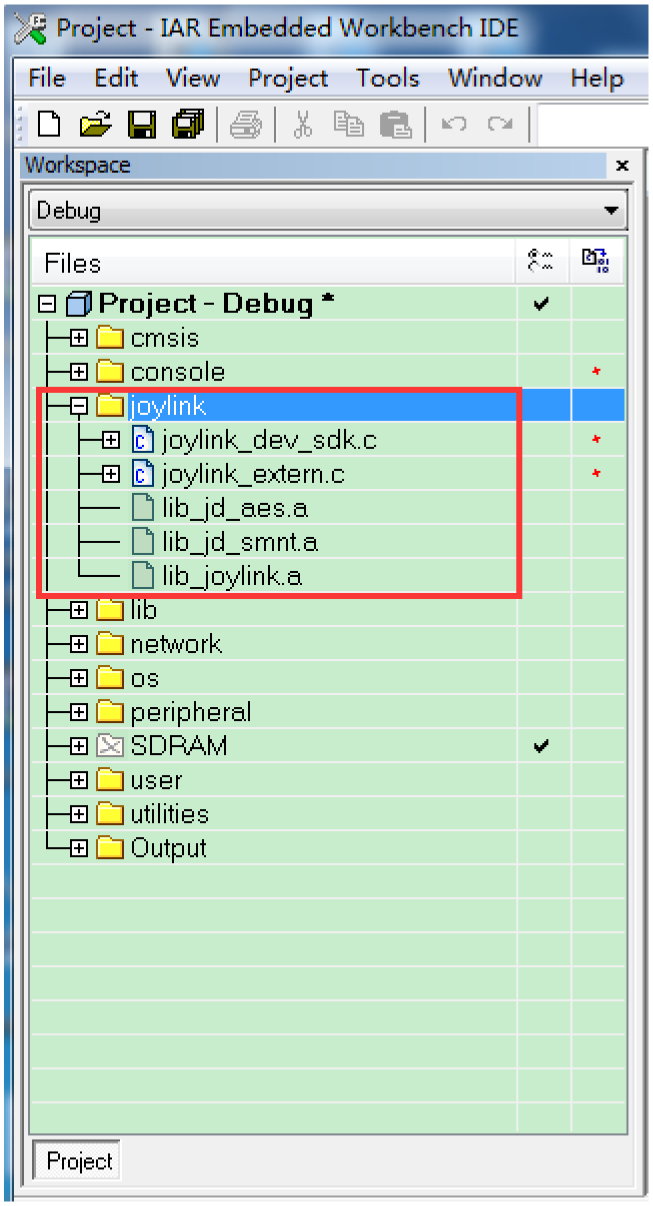
2 Porting JoyLink patch into Ameba's project

Step 1: **Extract Ameba released SDK**, as sdk-ameba1-v3.4b for example.

Step 2: **Extract all content from 3.4b_patch_joylink.zip** to the newly built folder "$sdk\component\common\application\joylink"

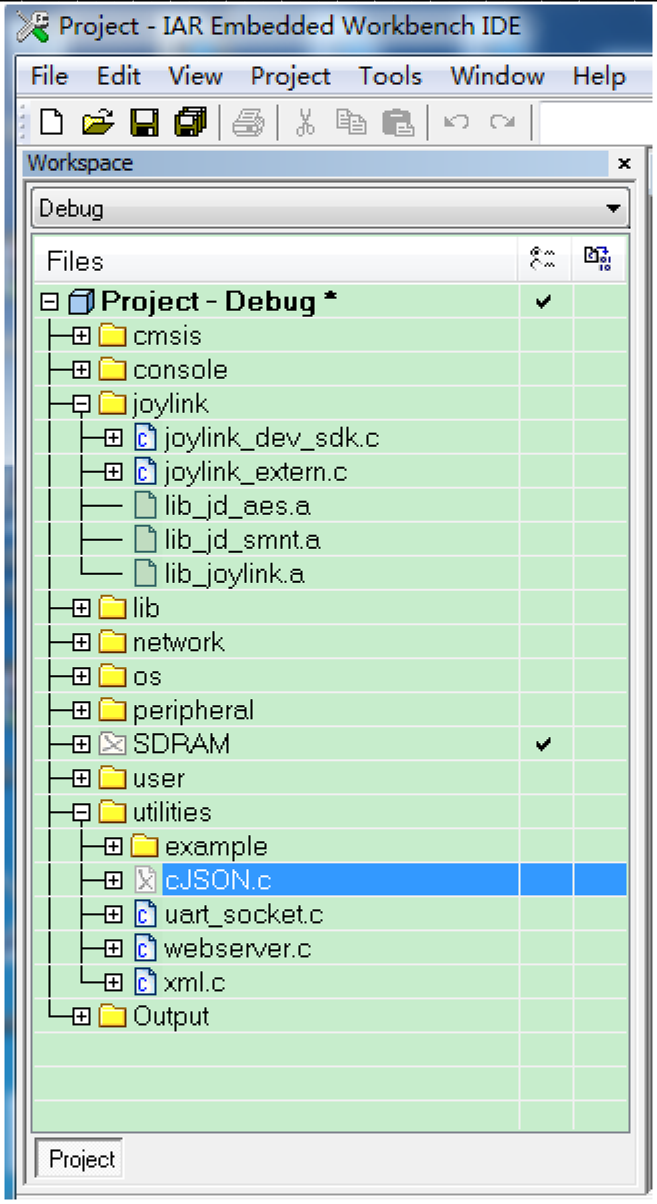
Step 3: Open IAR project and add a new group "**joylink**".

Step 4: Add "**joylink_extern.c**" and "**joylink_dev_sdk.c**" located at "$sdk\component\common\application\joylink\" into joylink group.



Step 5: Add "**lib_jd_aes.a**" "**lib_jd_smnt.a**" and "**lib_joylink.a**" located at "$sdk\component\common\application\joylink\lib\" into joylink group

Step 6: Exclude the below file from build.



Step 7: Open “$sdk\project\realtek_ameba1_va0_example\inc\platform_opts.h”, add “#define CONFIG_JOYLINK 1” if the define does not exist or just change to 1 if the define has already existed.

```
#define CONFIG_JOYLINK                                   1
```

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

```
#define CONFIG_EXAMPLE_WLAN_FAST_CONNECT     0
```

Step 9: Open "$sdk\component\common\utilities\update.c", change the define to 1 if it is not.

```
#define WRITE_OTA_ADDR                              1
```

Step 10: 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_JOYLINK
    example_joylink();
#endif
    ... ...
}
```

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

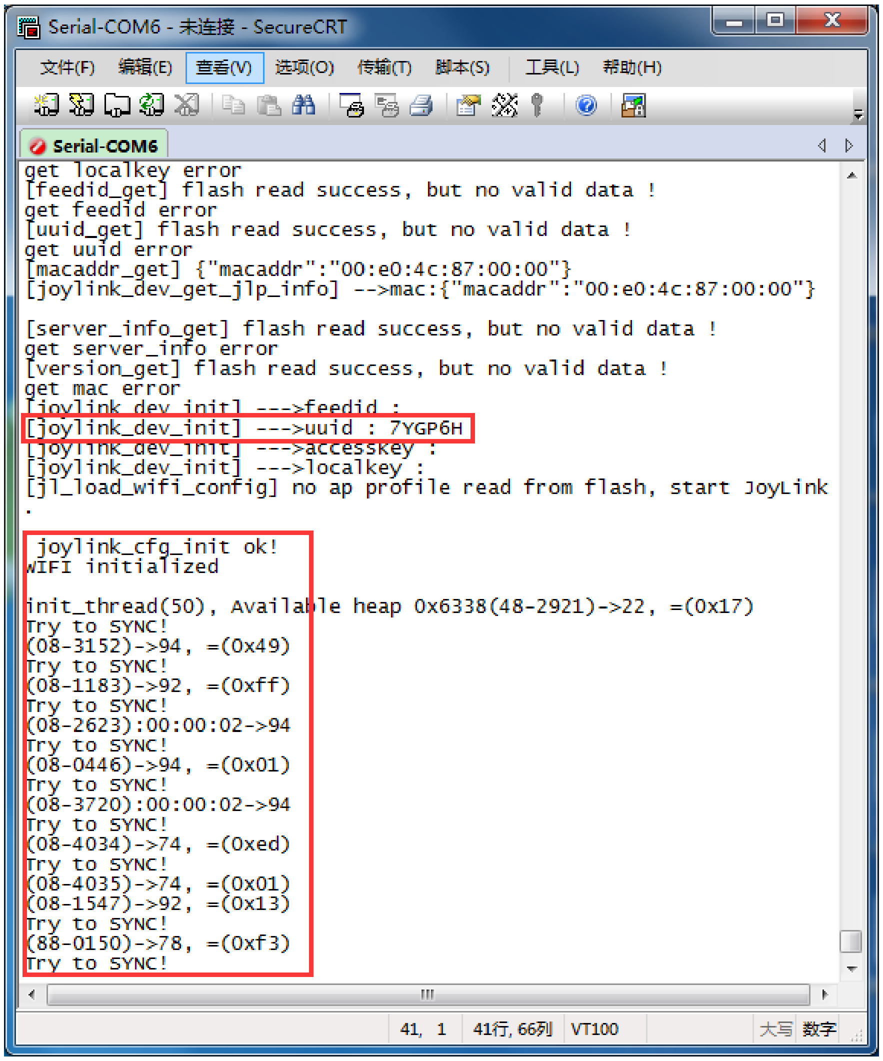
3 Example: connect Ameba to JD cloud

In this section, we use "京东微联" mentioned above to connect Ameba to JD cloud.

At present, use JoyLink protocol v1.3.3 to demonstrate.

PS : if ameba is reset at the first time, you can follow below procedures to operate. After that, ap profile and joylink related information is saved in flash, which means ameba will connect to ap and cloud automatically when reset again. If you wanna repeat below procedures rather than automatically operating, you need to use command "**ATCJ**" to erase above information.

Step 1: **Power on** your Ameba development board.


Serial-COM6 - 未连接 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
Serial-COM6
41, 1 41行, 66列 VT100 大写 数字

```
get localkey error
[feedid_get] flash read success, but no valid data !
get feedid error
[uuid_get] flash read success, but no valid data !
get uuid error
[macaddr_get] {"macaddr":"00:e0:4c:87:00:00"}
[joylink_dev_get_jlp_info] -->mac:{"macaddr":"00:e0:4c:87:00:00"}

[server_info_get] flash read success, but no valid data !
get server_info error
[version_get] flash read success, but no valid data !
get mac error
[joylink_dev_init] --->feedid :
[joylink_dev_init] --->uuid : 7YGP6H
[joylink_dev_init] --->accesskey :
[joylink_dev_init] --->localkey :
[jl_load_wifi_config] no ap profile read from flash, start JoyLink
.

 joylink_cfg_init ok!
WIFI initialized

init_thread(50), Available heap 0x6338(48-2921)->22, =(0x17)
Try to SYNC!
(08-3152)->94, =(0x49)
Try to SYNC!
(08-1183)->92, =(0xff)
Try to SYNC!
(08-2623):00:00:02->94
Try to SYNC!
(08-0446)->94, =(0x01)
Try to SYNC!
(08-3720):00:00:02->94
Try to SYNC!
(08-4034)->74, =(0xed)
Try to SYNC!
(08-4035)->74, =(0x01)
(08-1547)->92, =(0x13)
Try to SYNC!
(88-0150)->78, =(0xf3)
Try to SYNC!
```

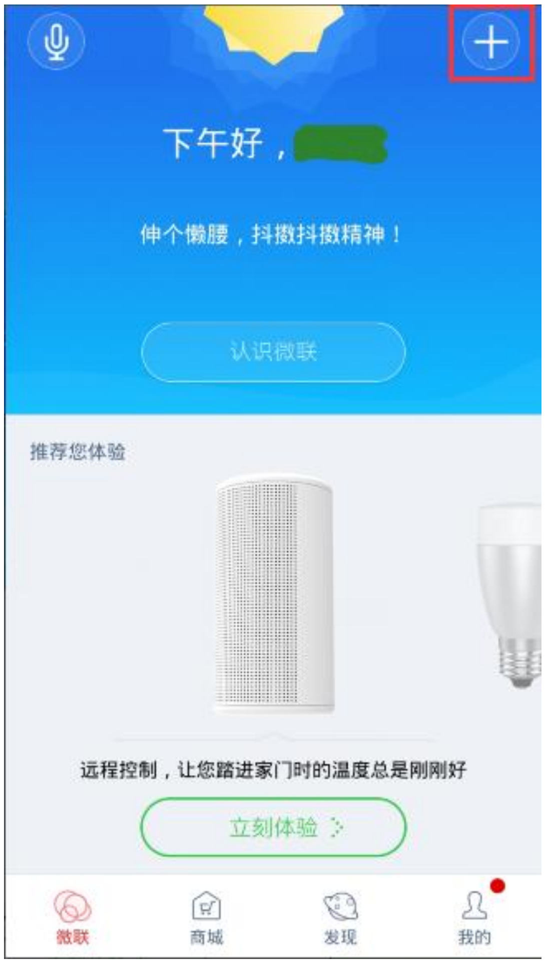
As above shows, Ameba's uuid is "7YGP6H".

After wifi is initialized, it works on promisc mode and scanning channels begins.

Step 2: Make your smart phone wifi connected to a target AP with internet.

Step 3: Install APK "京东微联" mentioned above on your phone.

Step 4: Run “京东微联” and login with a JD account, click on the top right corner.



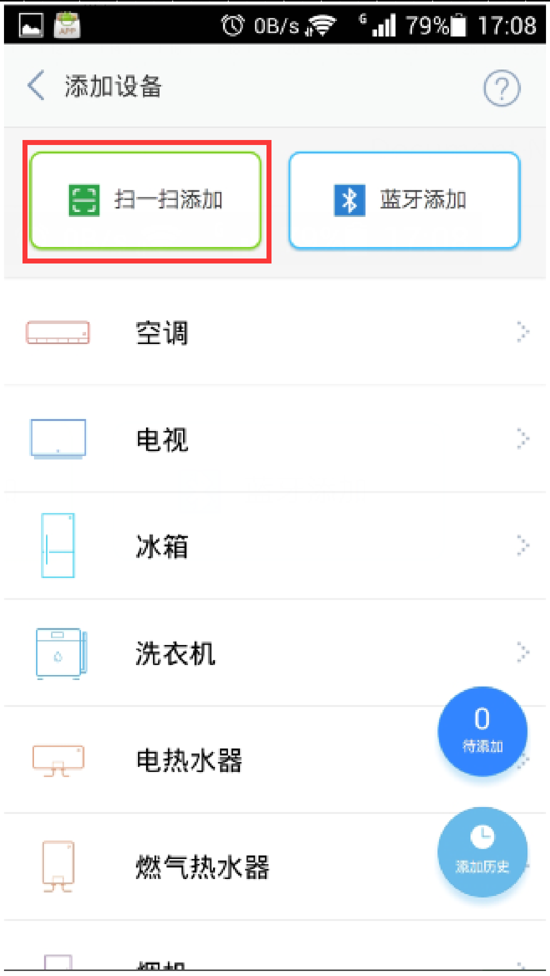
下午好，[redacted]

伸个懒腰，抖擞抖擞精神！

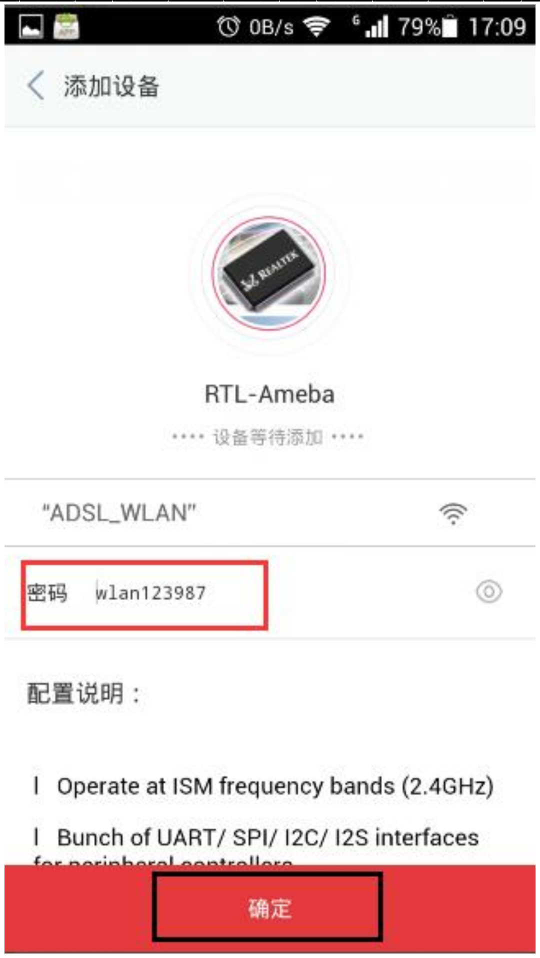
推荐您体验

远程控制，让您踏进家门时的温度总是刚刚好

Step 5: Click “**scan to add**” to scan QR-code mentioned above.



Step 6: After scanning, target AP’s SSID will be shown. Input its password and click “**confirm**”.

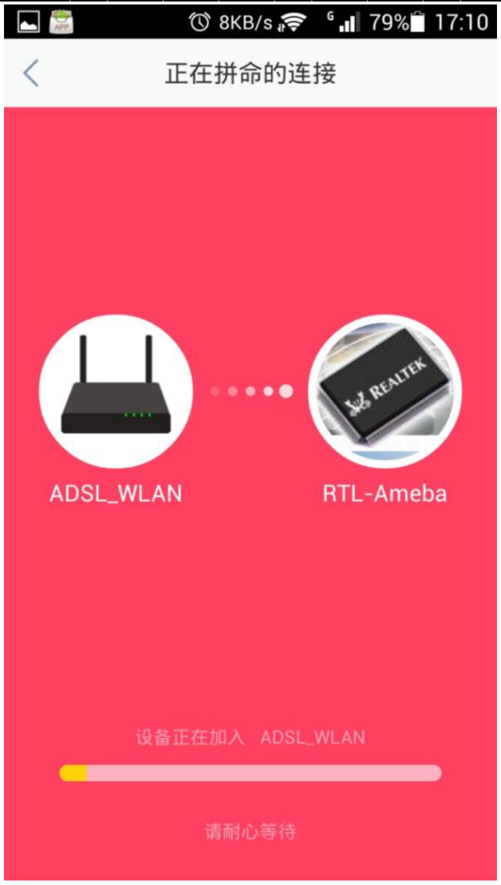

0B/s 79% 17:09
添加设备
RTL-Ameba
•••• 设备等待添加 ••••
"ADSL_WLAN"
密码 wlan123987
配置说明：
l Operate at ISM frequency bands (2.4GHz)
l Bunch of UART/ SPI/ I2C/ I2S interfaces for peripheral controllers
确定

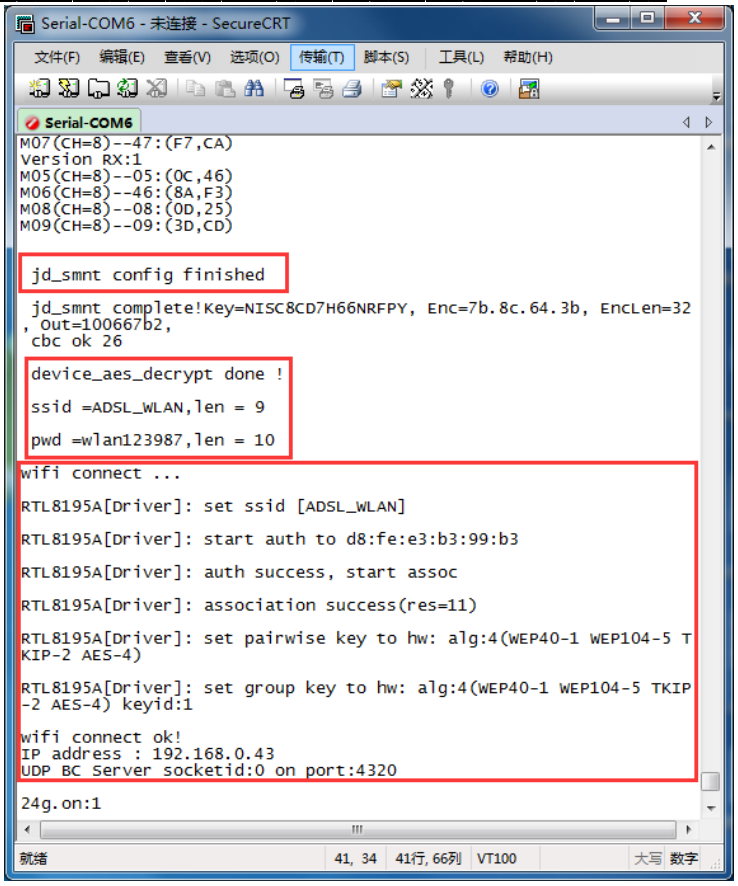
Step 7: As Ameba's log shows below, it first receives wifi configuration information.

After Ameba receives whole AP profile, prompts "**jd_smnt config finished**"

After Ameba decrypts AP profile, prompts "**device_aes_decrypt done !**" and shows **AP's ssid and password**.

Then, ameba starts to connect to AP automatically.



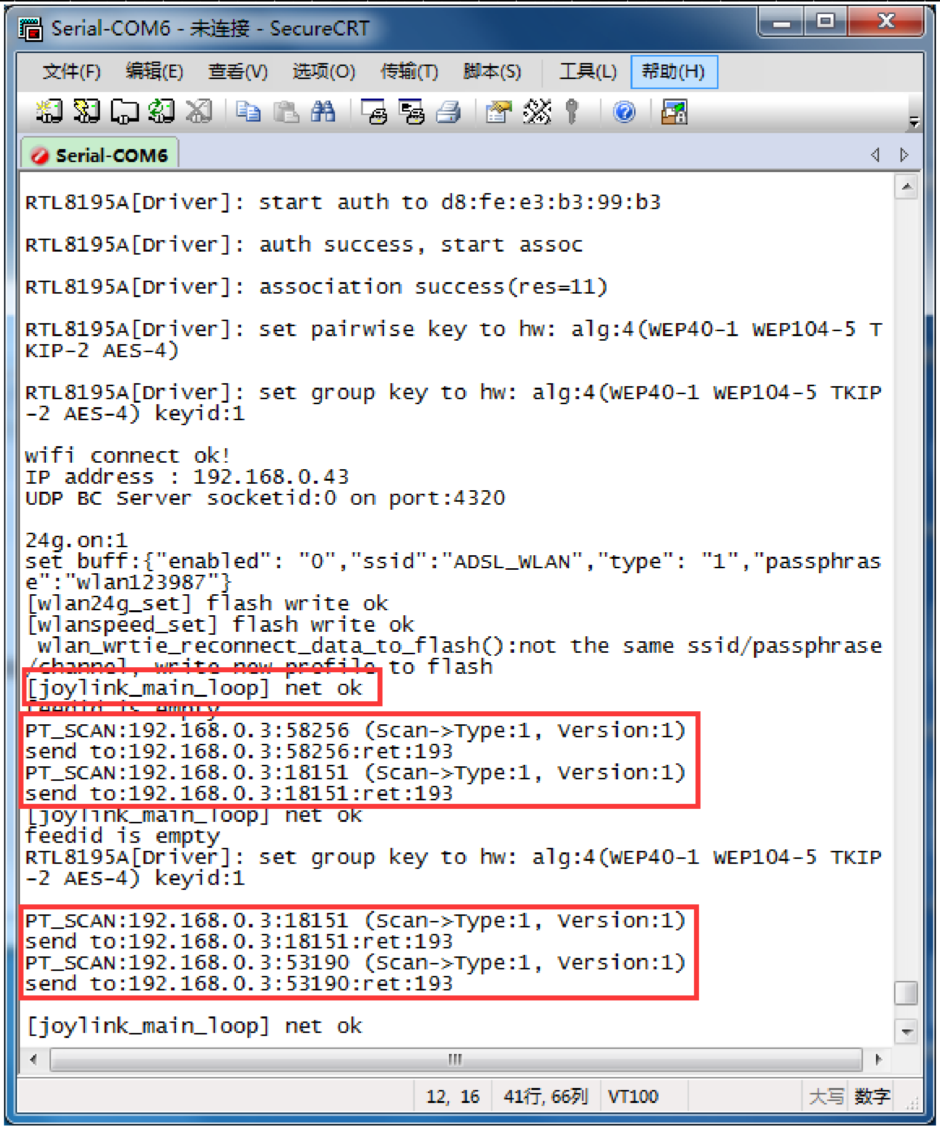

Serial-COM6 - 未连接 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
Serial-COM6
就绪 41, 34 41行, 66列 VT100 大写 数字

```
M07(CH=8)--47:(F7,CA)
Version RX:1
M05(CH=8)--05:(0C,46)
M06(CH=8)--46:(8A,F3)
M08(CH=8)--08:(0D,25)
M09(CH=8)--09:(3D,CD)

 jd_smnt config finished

 jd_smnt complete!Key=NISC8CD7H66NRFPY, Enc=7b.8c.64.3b, EncLen=32
, Out=100667b2,
 cbc ok 26

 device_aes_decrypt done !

 ssid =ADSL_WLAN,len = 9

 pwd =wlan123987,len = 10

wifi connect ...

RTL8195A[Driver]: set ssid [ADSL_WLAN]

RTL8195A[Driver]: start auth to d8:fe:e3:b3:99:b3

RTL8195A[Driver]: auth success, start assoc

RTL8195A[Driver]: association success(res=11)

RTL8195A[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 T
KIP-2 AES-4)

RTL8195A[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP
-2 AES-4) keyid:1

wifi connect ok!
IP address : 192.168.0.43
UDP BC Server socketid:0 on port:4320

24g.on:1
```

Step 8: After Ameba connects to AP, it can receive **LAN “scan”** from phone through the AP and then response back.

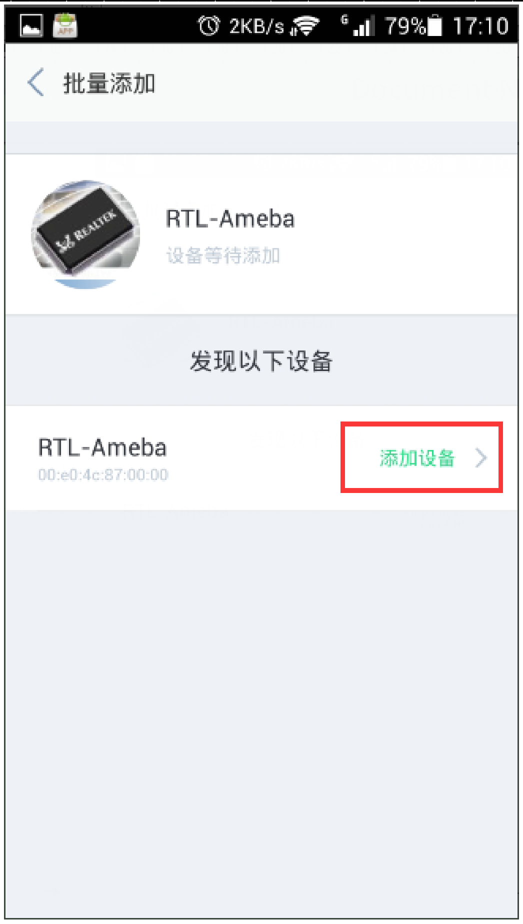
“[joylink_main_loop] net ok” means wifi connected ok.

“192.168.0.3” is my testing phone’s IP address.


Serial-COM6 - 未连接 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
Serial-COM6
12, 16 41行, 66列 VT100 大写 数字

```
RTL8195A[Driver]: start auth to d8:fe:e3:b3:99:b3

RTL8195A[Driver]: auth success, start assoc

RTL8195A[Driver]: association success(res=11)

RTL8195A[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 T
KIP-2 AES-4)

RTL8195A[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP
-2 AES-4) keyid:1

wifi connect ok!
IP address : 192.168.0.43
UDP BC Server socketid:0 on port:4320

24g.on:1
set buff:{"enabled": "0","ssid":"ADSL_WLAN","type": "1","passphras
e":"wlan123987"}
[wlan24g_set] flash write ok
[wlanspeed_set] flash write ok
 wlan_wrtie_reconnect_data_to_flash():not the same ssid/passphrase
/channel], write new profile to flash
[joylink_main_loop] net ok
feedid is empty
PT_SCAN:192.168.0.3:58256 (Scan->Type:1, Version:1)
send to:192.168.0.3:58256:ret:193
PT_SCAN:192.168.0.3:18151 (Scan->Type:1, Version:1)
send to:192.168.0.3:18151:ret:193
[joylink_main_loop] net ok
feedid is empty
RTL8195A[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP
-2 AES-4) keyid:1

PT_SCAN:192.168.0.3:18151 (Scan->Type:1, Version:1)
send to:192.168.0.3:18151:ret:193
PT_SCAN:192.168.0.3:53190 (Scan->Type:1, Version:1)
send to:192.168.0.3:53190:ret:193

[joylink_main_loop] net ok
```

Step 9: After phone receives **LAN response**, APK prompts "**find below device**" with the device's MAC address.

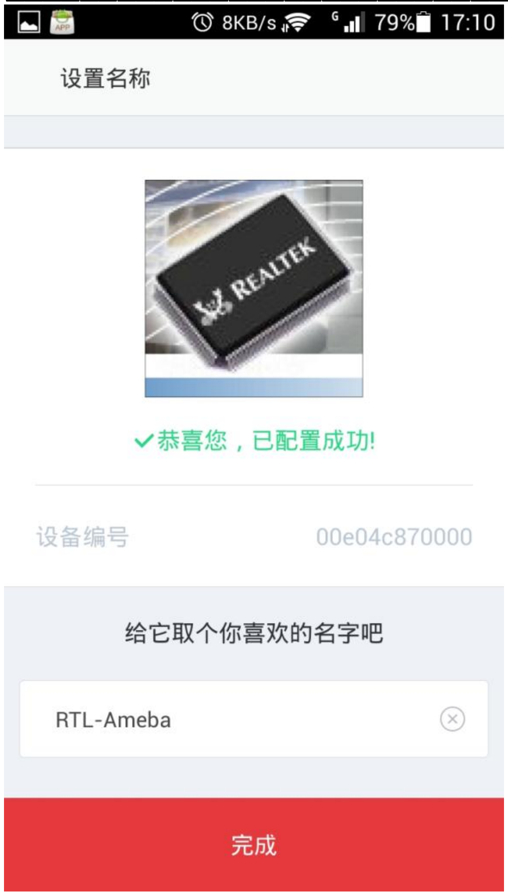


Step 10: Click “**add device**”. As Ameba’s log shows, parameters will be written into device and then APK prompts “**configure successfully**”.

After **being bound**, Ameba will connect to JD cloud server automatically :

“**server st : 0**” means establishing a long tcp connection with server.

“**AUTH OK**” means authentication ok with server and “server st” now is 1. After this step, Ameba is able to communicate with JD cloud server.


8KB/s 79% 17:10

设置名称

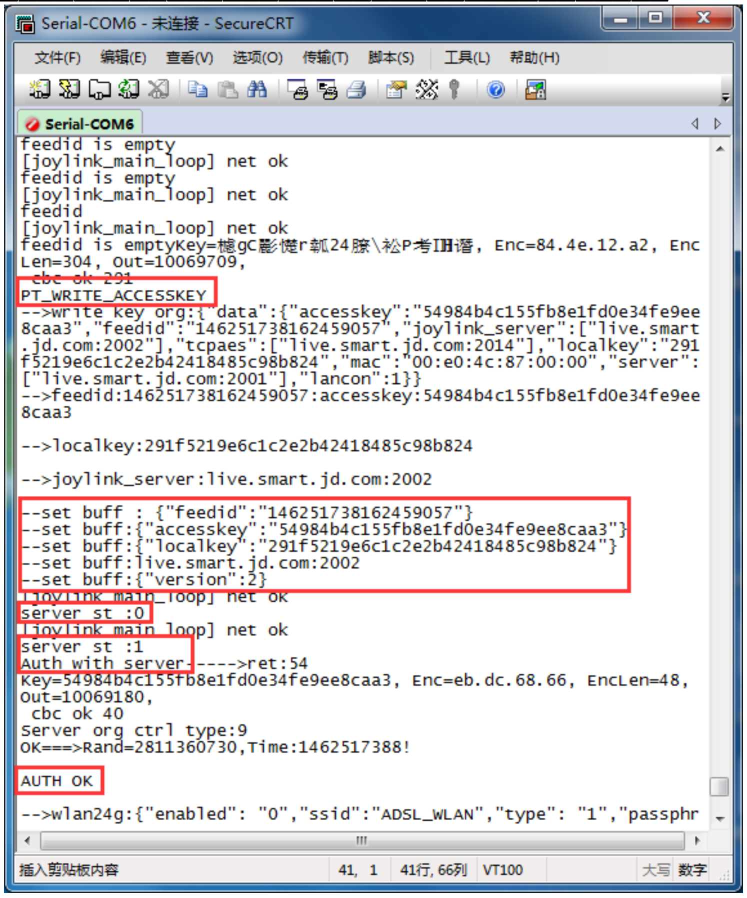
✓恭喜您，已配置成功!

设备编号 00e04c870000

给它取个你喜欢的名字吧

RTL-Ameba

完成


Serial-COM6 - 未连接 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
Serial-COM6

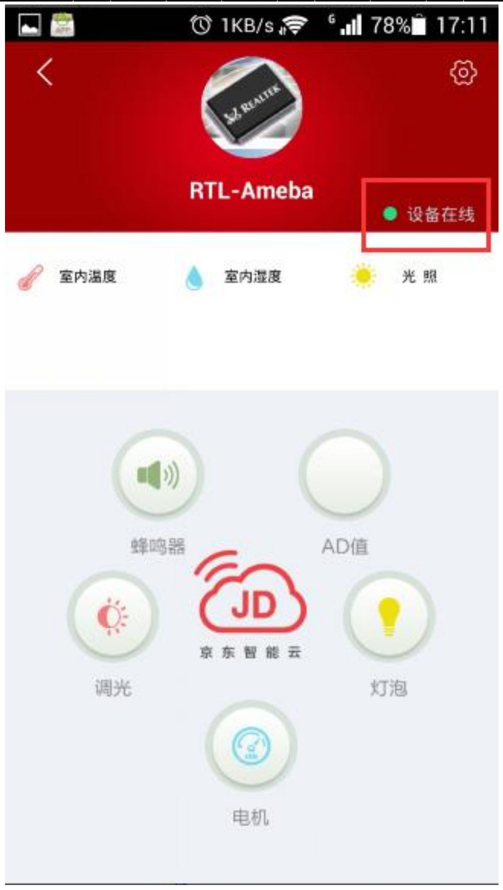
```
feedid is empty
[joylink_main_loop] net ok
feedid is empty
[joylink_main_loop] net ok
feedid
[joylink_main_loop] net ok
feedid is emptyKey=櫶gC影憔r韰24膝\祕P考IH谮, Enc=84.4e.12.a2, Enc
Len=304, out=10069709,
 cbc ok 291
PT_WRITE_ACCESSKEY
-->write key org:{"data":{"accesskey":"54984b4c155fb8e1fd0e34fe9ee
8caa3","feedid":"146251738162459057","joylink_server":["live.smart
.jd.com:2002"],"tcpaes":["live.smart.jd.com:2014"],"localkey":"291
f5219e6c1c2e2b42418485c98b824","mac":"00:e0:4c:87:00:00","server":
["live.smart.jd.com:2001"],"lancon":1}}
-->feedid:146251738162459057:accesskey:54984b4c155fb8e1fd0e34fe9ee
8caa3

-->localkey:291f5219e6c1c2e2b42418485c98b824

-->joylink_server:live.smart.jd.com:2002

--set buff : {"feedid":"146251738162459057"}
--set buff:{"accesskey":"54984b4c155fb8e1fd0e34fe9ee8caa3"}
--set buff:{"localkey":"291f5219e6c1c2e2b42418485c98b824"}
--set buff:live.smart.jd.com:2002
--set buff:{"version":2}
[joylink_main_loop] net ok
server st :0
[joylink_main_loop] net ok
server st :1
Auth with server----->ret:54
Key=54984b4c155fb8e1fd0e34fe9ee8caa3, Enc=eb.dc.68.66, EncLen=48,
Out=10069180,
 cbc ok 40
Server org ctrl type:9
OK===>Rand=2811360730,Time:1462517388!

AUTH OK

-->wlan24g:{"enabled": "0","ssid":"ADSL_WLAN","type": "1","passphr
```

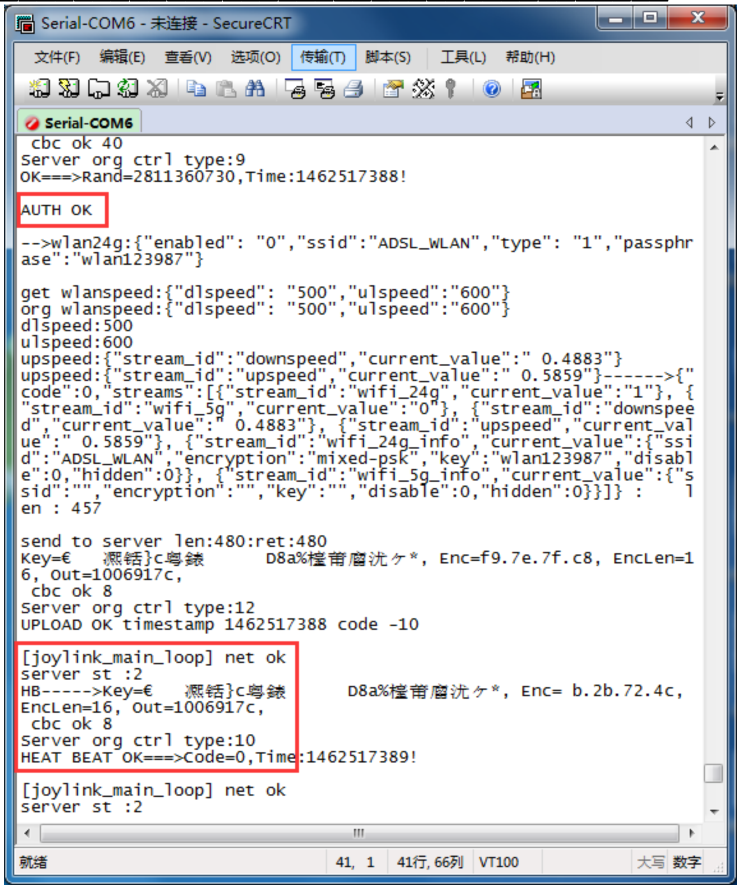
Step 11: Click "**accomplish**".

The left picture shows a default generated controller interface on JD APK. It shows "**device is online**". You can design your own device's controller interface.

The right picture shows, Ameba works on "**server st : 2**" which means cloud server is now working well to communicate with Ameba.

"**HEAT BEAT OK**" means the procedure of sending heartbeat to cloud server and getting a response is now working well.




Serial-COM6 - 未连接 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
Serial-COM6
就绪 41, 1 41行, 66列 VT100 大写 数字

```
 cbc ok 40
Server org ctrl type:9
OK===>Rand=2811360730,Time:1462517388!

AUTH OK

-->wlan24g:{"enabled": "0","ssid":"ADSL_WLAN","type": "1","passphr
ase":"wlan123987"}

get wlanspeed:{"dlspeed": "500","ulspeed":"600"}
org wlanspeed:{"dlspeed": "500","ulspeed":"600"}
dlspeed:500
ulspeed:600
upspeed:{"stream_id":"downspeed","current_value":" 0.4883"}
upspeed:{"stream_id":"upspeed","current_value":" 0.5859"}------>{"
code":0,"streams":[{"stream_id":"wifi_24g","current_value":"1"}, {
"stream_id":"wifi_5g","current_value":"0"}, {"stream_id":"downspee
d","current_value":" 0.4883"}, {"stream_id":"upspeed","current_val
ue":" 0.5859"}, {"stream_id":"wifi_24g_info","current_value":{"ssi
d":"ADSL_WLAN","encryption":"mixed-psk","key":"wlan123987","disabl
e":0,"hidden":0}}, {"stream_id":"wifi_5g_info","current_value":{"s
sid":"","encryption":"","key":"","disable":0,"hidden":0}}]} :    l
en : 457

send to server len:480:ret:480
Key=€   漑铦}c粤鍒      D8a%楃菁膤沋ケ*, Enc=f9.7e.7f.c8, EncLen=1
6, Out=1006917c,
 cbc ok 8
Server org ctrl type:12
UPLOAD OK timestamp 1462517388 code -10

[joylink_main_loop] net ok
server st :2
HB----->Key=€   漑铦}c粤鍒      D8a%楃菁膤沋ケ*, Enc= b.2b.72.4c,
EncLen=16, Out=1006917c,
 cbc ok 8
Server org ctrl type:10
HEAT BEAT OK===>Code=0,Time:1462517389!

[joylink_main_loop] net ok
server st :2
```

4 OTA

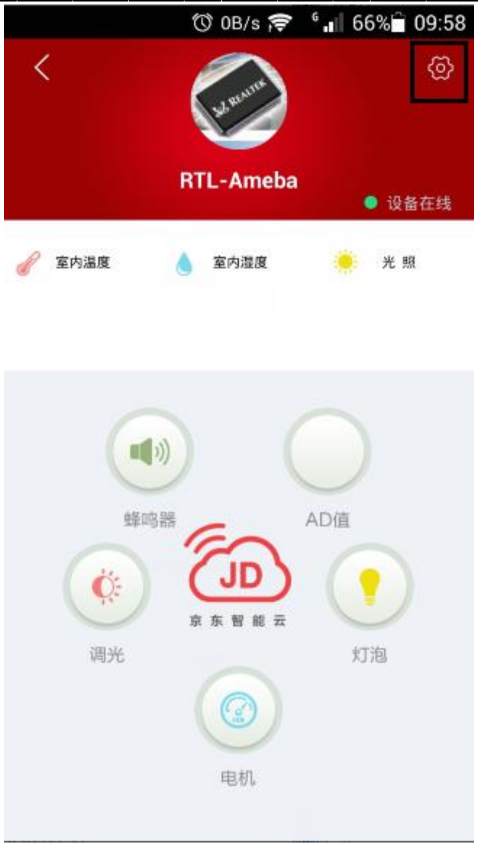
In this section, an example of OTA process is demonstrated.

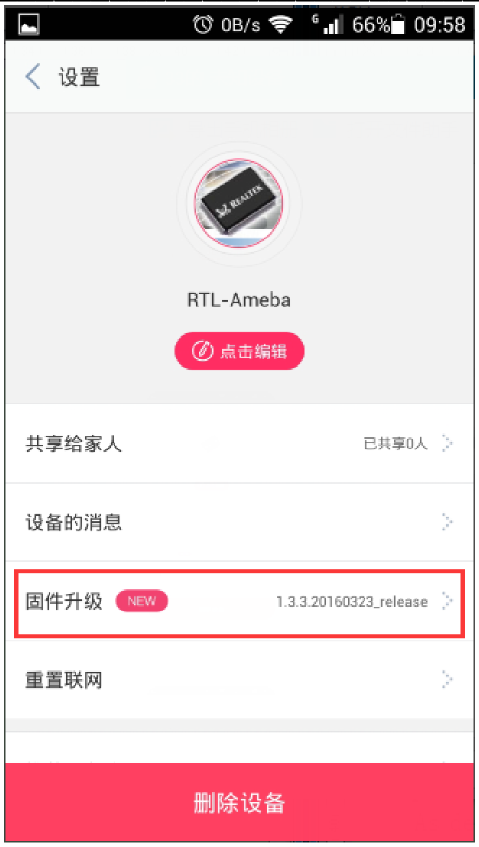
OTA function is provide to upgrade a new firmware over the air.

As an example of OTA test, a new firmware “ota.bin” with a higher assigned patch version num is placed in the cloud.

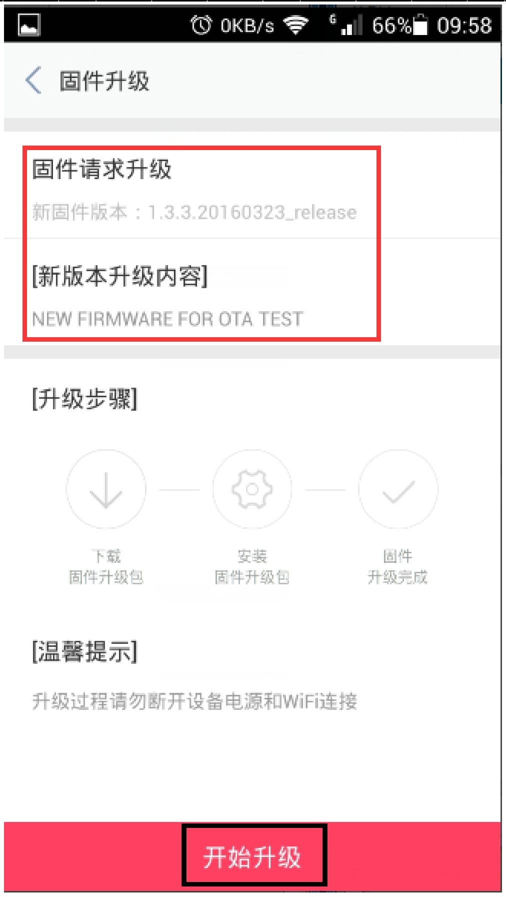
Step 1: On controller interface, click on the top right corner.

Step 2: When a higher version of firmware is detected by JD APK, it prompts “**firmware upgrade NEW**” with the version name.





Step 3: Click “firmware upgrade NEW” to get details about the new firmware.


0KB/s 66% 09:58

固件升级

固件请求升级

新固件版本：1.3.3.20160323_release

[新版本升级内容]

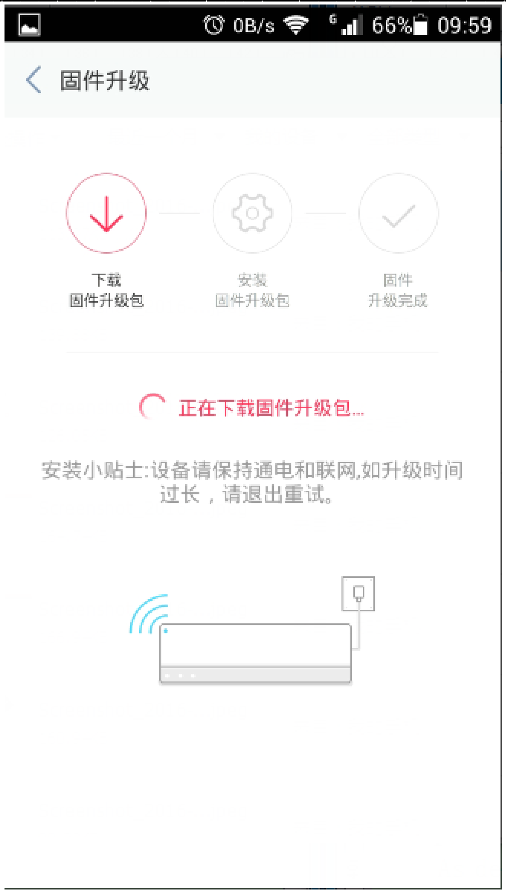
NEW FIRMWARE FOR OTA TEST

[升级步骤]

[温馨提示]

升级过程请勿断开设备电源和WiFi连接

Step 4: Click “**start to upgrade**” to start ota.



Step 5: Ameba starts to **download and upgrade** the new firmware.

As Ameba's log shown :

Total size of new firmware is 314632 bytes.

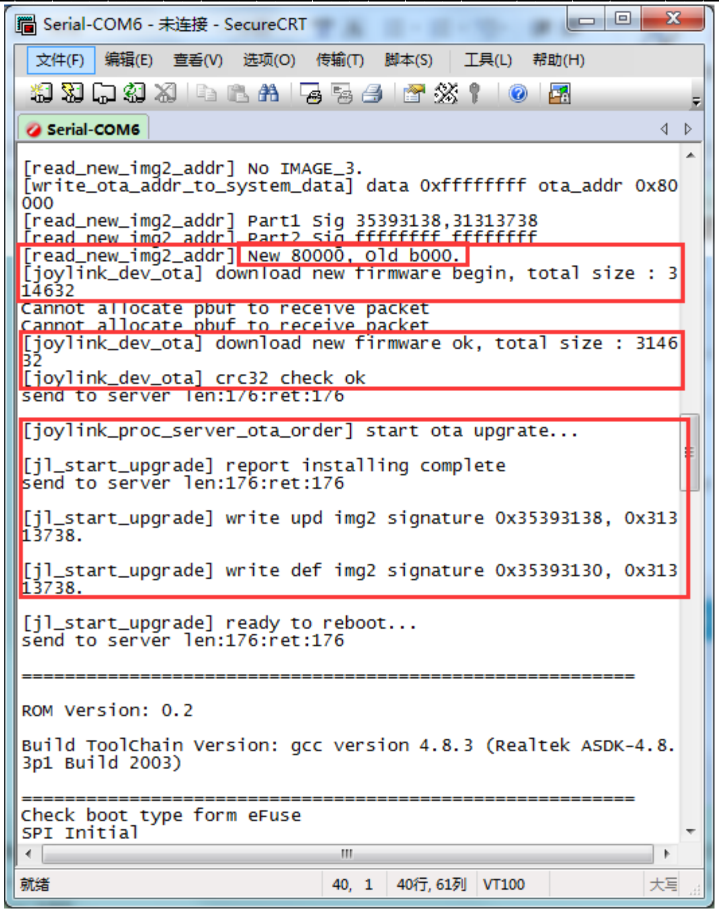
The old firmware is now located at **0xb000** in flash, the new firmware will be placed at **0x80000**.

After download 314632 bytes, check its crc32 value.

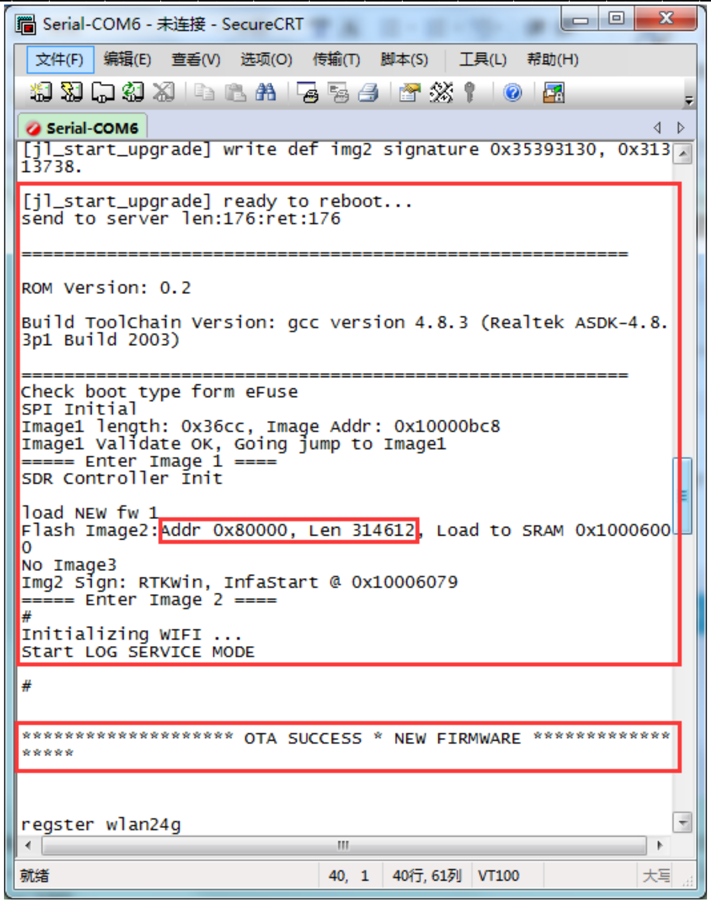
If the crc32 value check ok, start to upgrade and reboot !

After reboot, Ameba **runs the new firmware** (Addr 0x80000), it will show :

******************* OTA SUCCESS * NEW FIRMWARE *****************

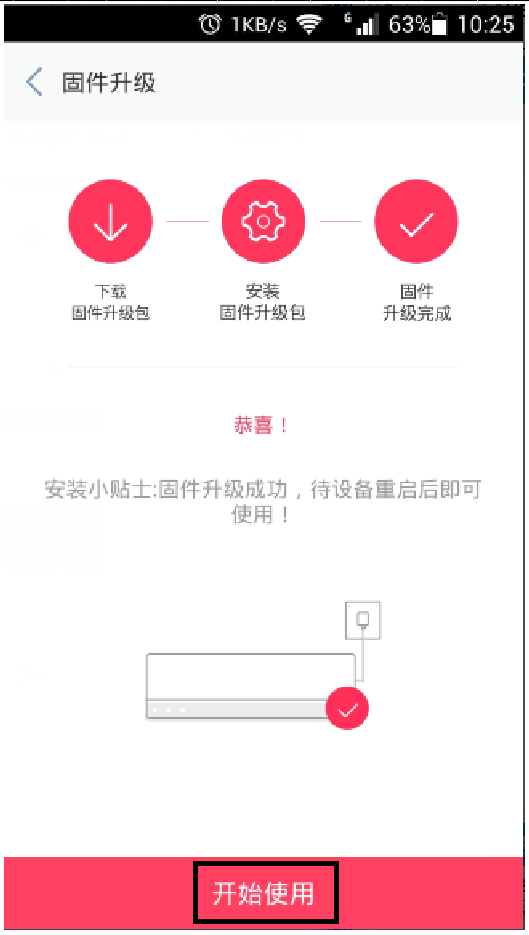

Serial-COM6 - 未连接 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
Serial-COM6
就绪 40, 1 40行, 61列 VT100 大写

```
[read_new_img2_addr] No IMAGE_3.
[write_ota_addr_to_system_data] data 0xffffffff ota_addr 0x80
000
[read_new_img2_addr] Part1 Sig 35393138,31313738
[read_new_img2_addr] Part2 Sig ffffffff ffffffff
[read_new_img2_addr] New 80000, old b000.
[joylink_dev_ota] download new firmware begin, total size : 3
14632
Cannot allocate pbuf to receive packet
Cannot allocate pbuf to receive packet
[joylink_dev_ota] download new firmware ok, total size : 3146
32
[joylink_dev_ota] crc32 check ok
send to server len:176:ret:176

[joylink_proc_server_ota_order] start ota upgrate...

[jl_start_upgrade] report installing complete
send to server len:176:ret:176

[jl_start_upgrade] write upd img2 signature 0x35393138, 0x313
13738.

[jl_start_upgrade] write def img2 signature 0x35393130, 0x313
13738.

[jl_start_upgrade] ready to reboot...
send to server len:176:ret:176

=========================================================

ROM Version: 0.2

Build ToolChain Version: gcc version 4.8.3 (Realtek ASDK-4.8.
3p1 Build 2003)

=========================================================
Check boot type form eFuse
SPI Initial
```


Serial-COM6 - 未连接 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
Serial-COM6
就绪 40, 1 40行, 61列 VT100 大写

```
[jl_start_upgrade] write def img2 signature 0x35393130, 0x313
13738.

[jl_start_upgrade] ready to reboot...
send to server len:176:ret:176

=========================================================

ROM Version: 0.2

Build ToolChain Version: gcc version 4.8.3 (Realtek ASDK-4.8.
3p1 Build 2003)

=========================================================
Check boot type form eFuse
SPI Initial
Image1 length: 0x36cc, Image Addr: 0x10000bc8
Image1 Validate OK, Going jump to Image1
===== Enter Image 1 ====
SDR Controller Init

load NEW fw 1
Flash Image2:Addr 0x80000, Len 314612, Load to SRAM 0x1000600
0
No Image3
Img2 Sign: RTKWin, InfaStart @ 0x10006079
===== Enter Image 2 ====
#
Initializing WIFI ...
Start LOG SERVICE MODE

#

******************** OTA SUCCESS * NEW FIRMWARE *************
*****


regster wlan24g
```

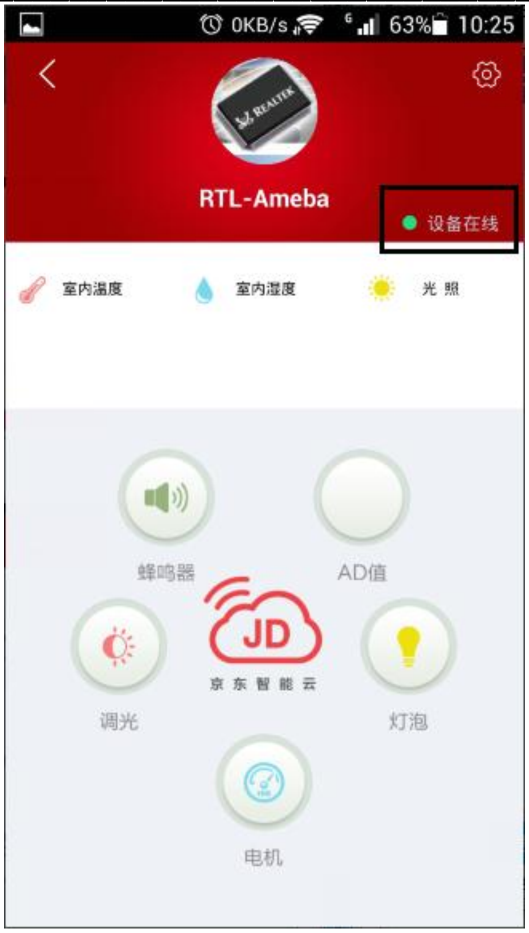
Step 6: After upgrading to the new firmware, APK prompts "**upgrade accomplish**".



恭喜！

安装小贴士:固件升级成功，待设备重启后即可使用！

Step 7: Click “**start to use**”.


0KB/s 63% 10:25
RTL-Ameba
设备在线
室内温度
室内湿度
光 照
蜂鸣器
AD值
调光
JD
京 东 智 能 云
灯泡
电机