Telink SIG Mesh

Android & iOS APP User Guide

AN-23031700-E1

Ver1.0.0

2023.03.17

Keyword

SIG Mesh, Android/iOS

Brief

This document is SIG Mesh Android/iOS APP User Guide.

Published by
Telink Semiconductor

Bldg 3, 1500 Zuchongzhi Rd,
Zhangjiang Hi-Tech Park, Shanghai, China

Legal Disclaimer

This document is provided as-is. Telink Semiconductor reserves the right to make improvements without further notice to this document or any products herein. This document may contain technical inaccuracies or typographical errors. Telink Semiconductor disclaims any and all liability for any errors, inaccuracies or incompleteness contained herein.

Information

For further information on the technology, product and business term, please contact Telink Semiconductor Company www.telink-semi.com

For sales or technical support, please send email to the address of:

telinksales@telink-semi.com

telinksupport@telink-semi.com

Revision History

Version	Change Description
V1.0.0	Initial release

Contents

List of Figures

1 Device Network

Networking is divided into manual networking mode and automatic networking mode.

1.1 Manual Provision Networking

1.1.1 Add Device in Manual Mode

Android APP

The Android version of the APP enables manual provision mode by default. After launching the app, click the upper right corner of the "+" button of the main interface to enter the add interface, the APP will automatically search for peripheral devices. You can click the ADD button on the right side of the corresponding device, or you can click × button on the top right corner of the corresponding device to delete the corresponding device, or click the ADD ALL button to network all the devices in the list.

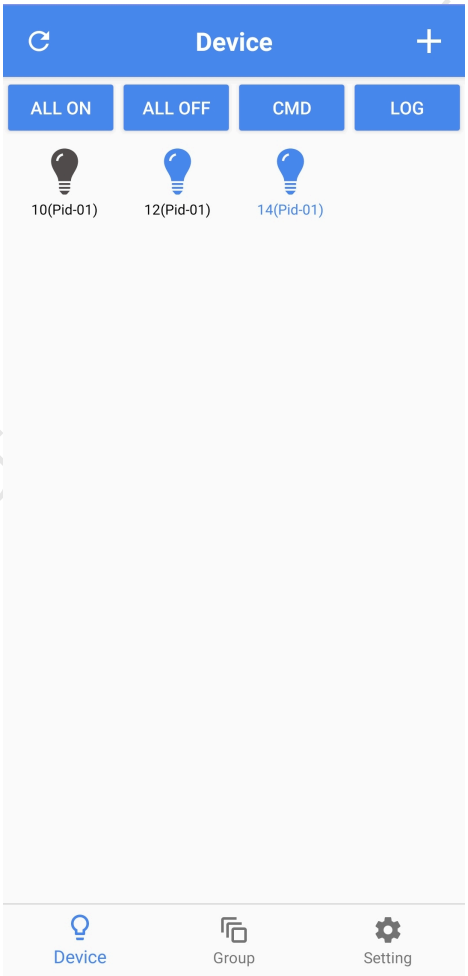


Figure 1.1: APP home page

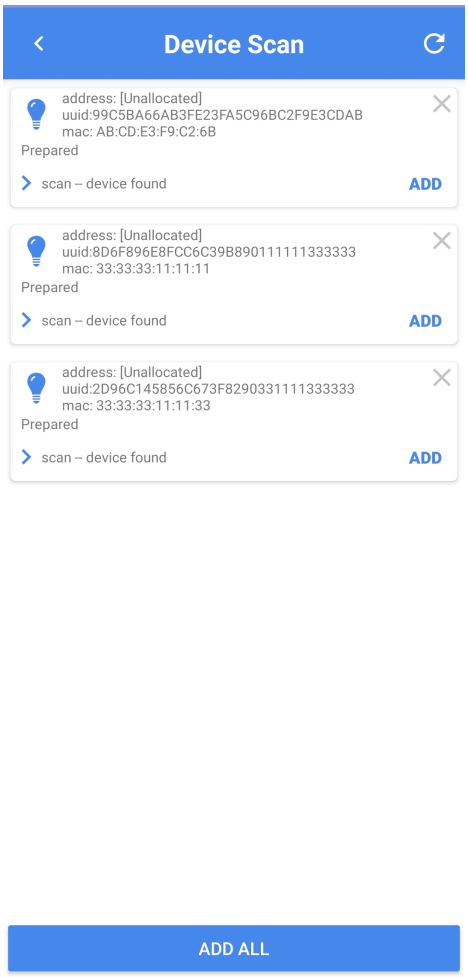


Figure 1.2: Device list in Android app

iOS APP

For the iOS version of the APP, implement manual networking through the following precedure: the APP home page - Setting - Choose Add Devices - Scan to search for the surrounding devices to be networked - Check the devices to be added, and then click Add Device button, the networked devices will prompt the keybound.

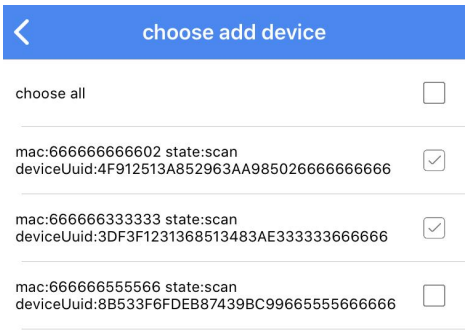


Figure 1.3: Device list in iOS app

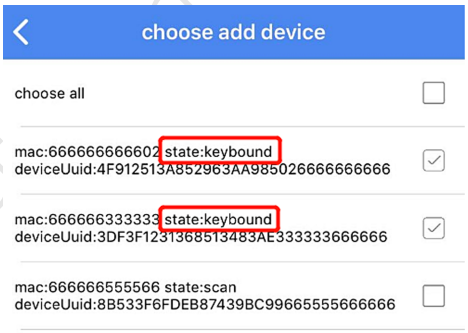


Figure 1.4: Device status in iOS app

1.1.2 Status During Manually Adding Devices

Android APP

The Scan-device found is the status that the device is scanned and found, and the left arrow expands the status of each state during the networking process (as shown below).

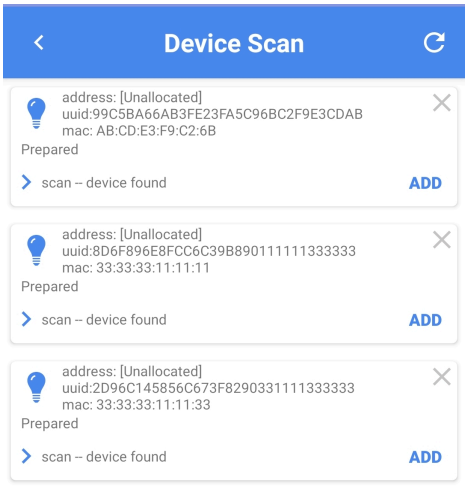


Figure 1.5: Device status in Android app

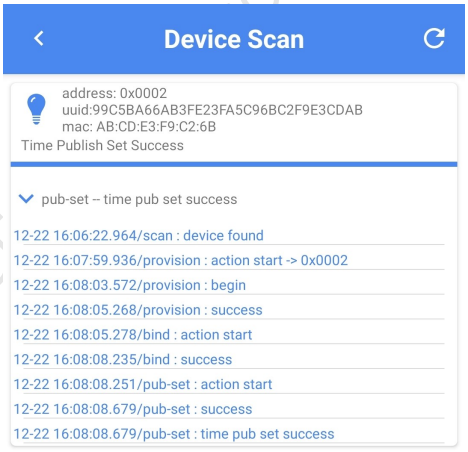


Figure 1.6: Device status list

iOS APP

The iOS version of the APP only displays the current status, more detailed information can be viewed in the APP's log record.

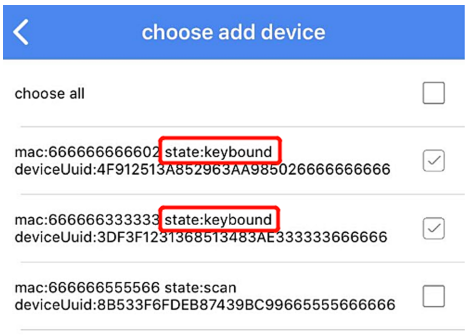


Figure 1.7: Device status in iOS app

1.2 Auto Provision Networking

Android APP

For Android app, switch to Auto provision mode through the following procedure: APP home page - setting - settings start Auto provision. At home page click the upper right corner of the + sign to enter the Device Scan interface, at this time the title will show Device Scan (Auto provision) and it automatically adds all the surrounding un-networked devices as the figure belwo.

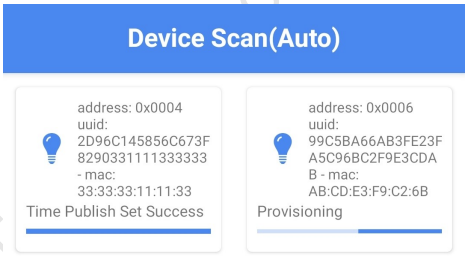


Figure 1.8: Auto Provision Networking in Android app

iOS APP

The iOS version of the APP enables automatic networking mode by default. Directly click the + sign on the upper right corner of the APP home page to enter the Device Scan interface to automatically scan and add devices.

Figure 1.9: Auto Provision Networking in iOS app

1.3 Rescan Peripheral Devices

The  at the upper right corner of Device Scan interface is the reload button. The manual provision networking mode can reload the device list as the figure below. The auto provision mode can re-scan the peripheral devices and automatically network as the figure below.

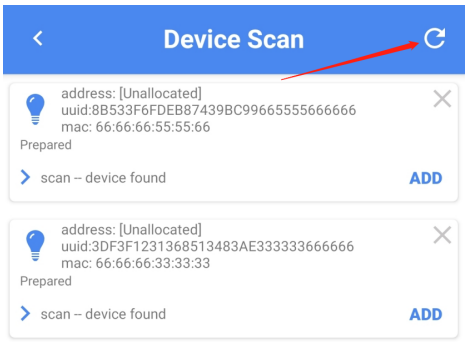


Figure 1.10: Reload device list

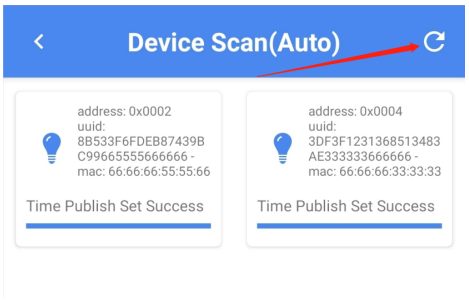


Figure 1.11: Rescan and network in Android app

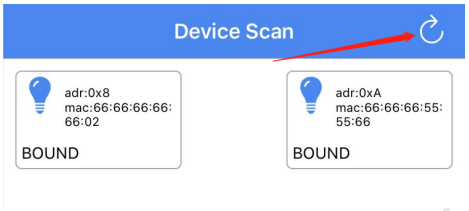


Figure 1.12: Rescan and network in iOS app

1.4 Certify Base

When networking, the certificate is verified to determine whether the device is allowed to join the mesh network. When the certificate passes, the networking will be successful, but if it does not pass, the networking will fail and prompt "certificate recordcheck error".

Preset conditions:

(1) Copy the files in draft_features\firmware\certify_base to vendor\common\certify_base.

(2) In the mesh_config.h file, enable CERTIFY_BASE_ENABLE macro, in certify_base_crypto.c file select a certificate for CERT_TYPE, set the specified certificates and then compile the firmware to burn into the device.

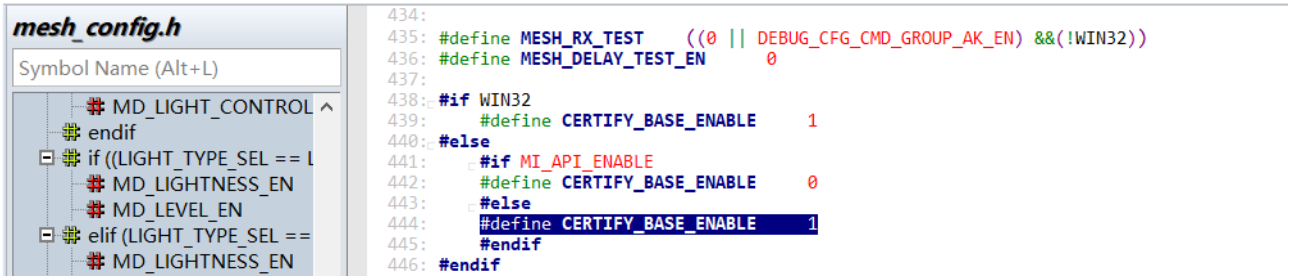


Figure 1.13: Enable CERTIFY_BASE_ENABLE

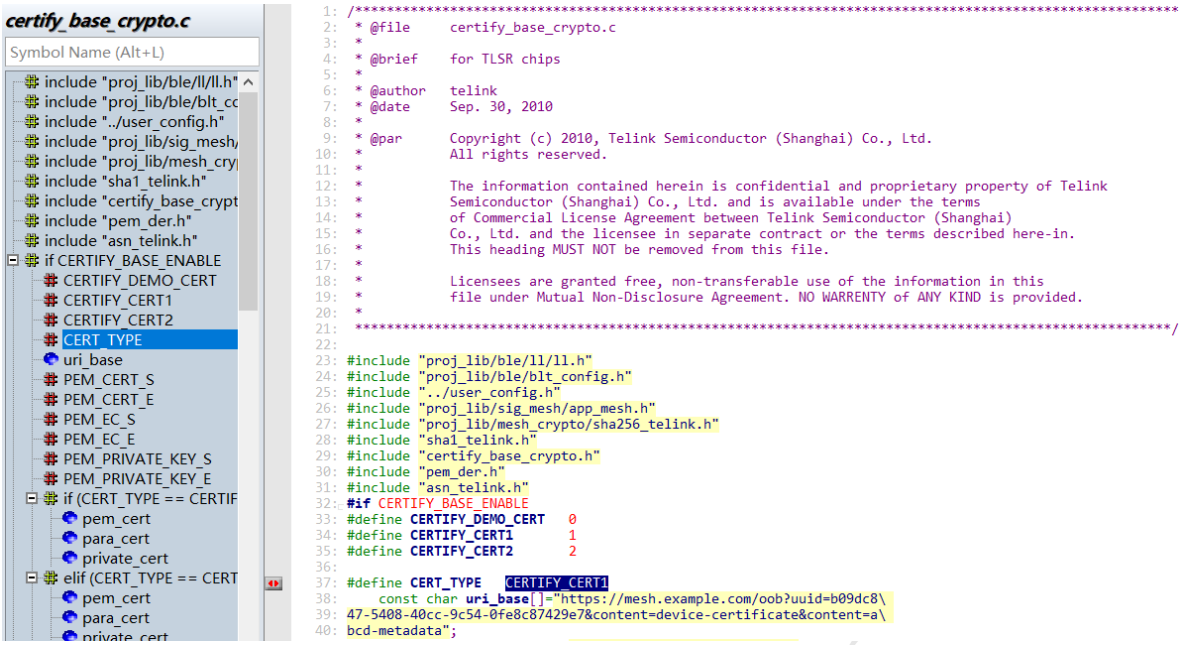


Figure 1.14: Select a cert for CERT_TYPE

Operating steps:

(1) Manual networking mode

The certificate icon will be displayed in the upper left corner when the device that needs to verify the certificate to nework, and the networking will be successful after it passes the certificate verification, while the verification will prompt certificate recordcheck error if the verification fails.

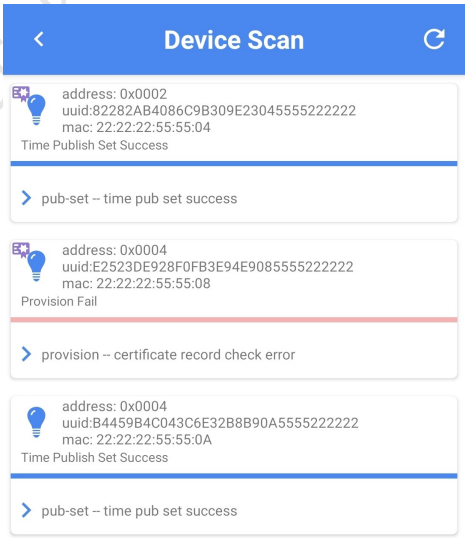


Figure 1.15: Manual network certify prompt in Android app

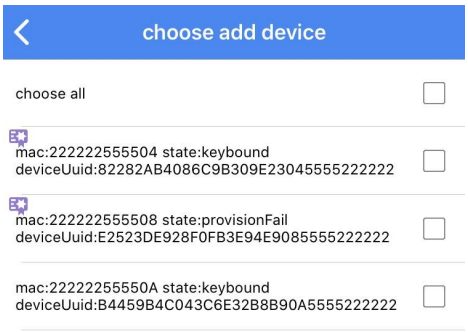


Figure 1.16: Manual network certify prompt in iOS app

(2) Auto networking mode

For the devices that need to verify the certificate, when networking in the Device Scan (Auto) interface, the certificate icon will be displayed in the upper left corner of the device, and the networking will be successful after the certificate verification is passed, while a certificate recordcheck error will be prompted if the verification fails.

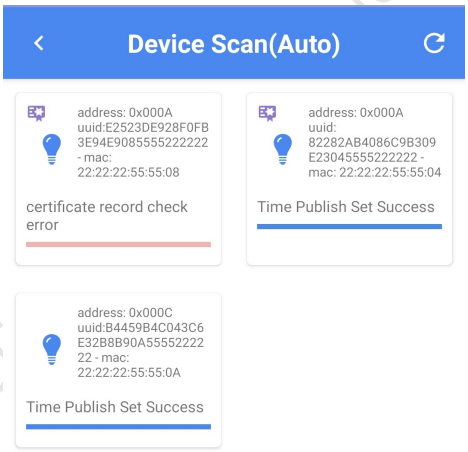


Figure 1.17: Auto network certify prompt in Android app

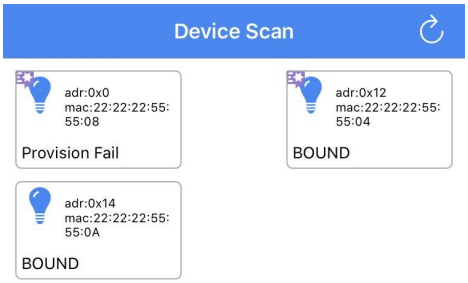


Figure 1.18: Auto Auto network certify prompt in iOS app

2 Device Interface

Figure 2.1: Android APP Device interface

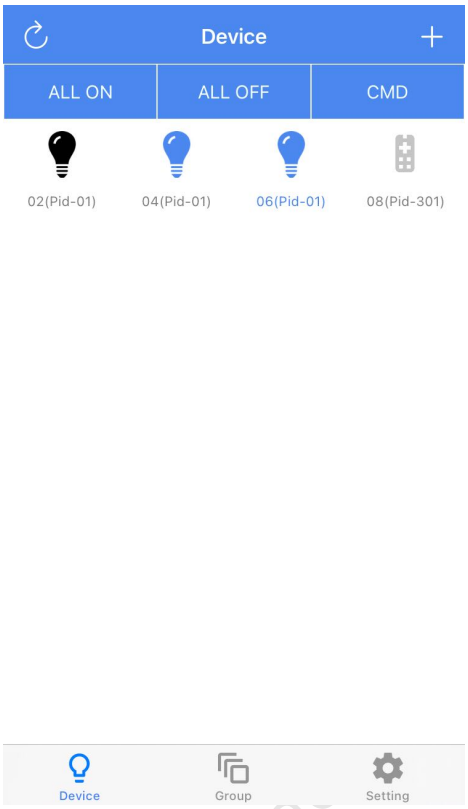


Figure 2.2: iOS APP Device interface

2.1 Refresh Device

The  icon on the top left corner of the Android/iOS APP homepage can refresh the current networked device status.

2.2 All on/off

The Android/iOS APPs control all networked devices to turn on/off the lights by sending "all on/off" commands. Blue is on, grey is off. The device with blue name is the directly connected device.

2.3 Single Device on/off

Click on the corresponding device icon to turn on/off the light. Blue is on, grey is off. The device with blue name is the directly connected device.

2.4 CMD Command

The CMD command has built-in Vendor on/off, Vendor on/off no-ACK, Vendor on/off get, Generic on/off commands (For iOS, it is not built-in yet). Users can also customize the commands through APP Custom for

Android (APP Vendor Data for iOS), which can customize Access Type, dst adr, opcode, params, rsp opcode, rsp max, retry count, ttl, tid position.

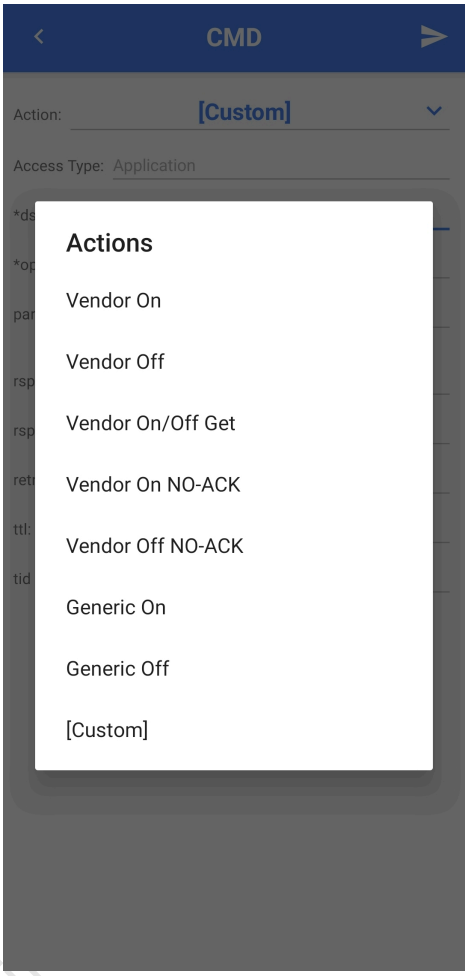


Figure 2.3: Android CMD -1

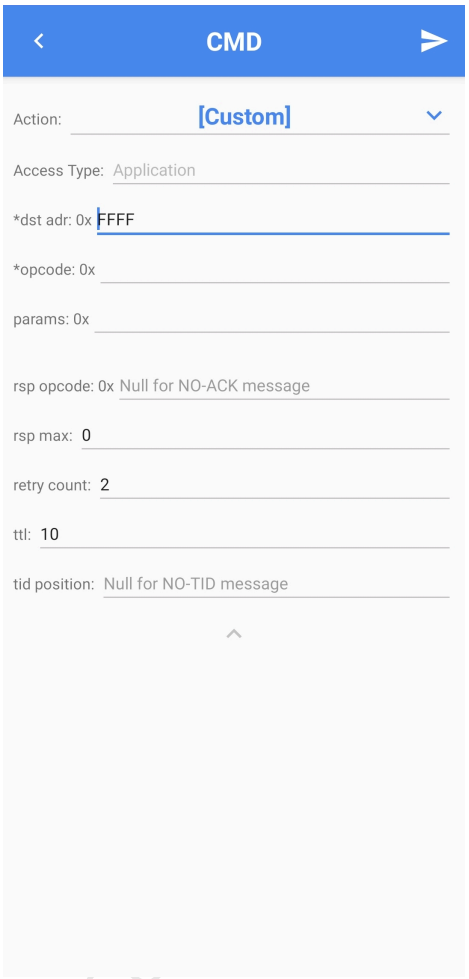


Figure 2.4: Android CMD -2

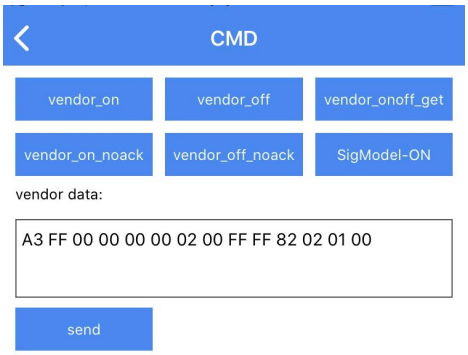


Figure 2.5: iOS APP CMD interface

2.5 Log

Android APP:

The Log interface will record the log information of the current operation (user need to open "Enable LOG" in the APP homepage – Setting – Settings in advance, which is turned off by default), "Save In File" is used to save log (save path: default storage/TelinkBleMesh), "Refresh" is used to refresh log, "Clear" is used to clear log information.

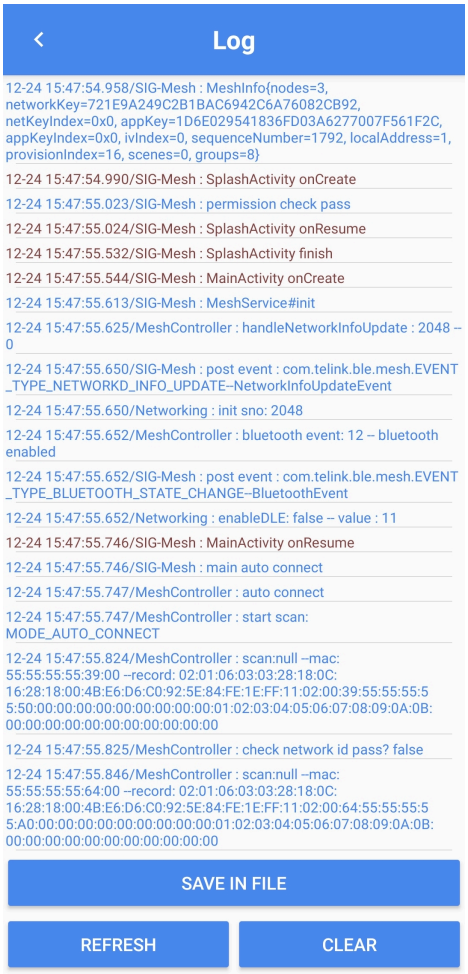


Figure 2.6: Android Log interface

iOS APP:

The iOS APP log view: APP homepage – Setting – Log. Click the "Clear" icon in the upper right corner to clear the log information. If users want to import log into the PC, users can get it to the computer through iTunes and other ways.

Figure 2.7: iOS Log interface

2.6 Device Setting (Light device)

Long press the icon of the networked Light device on the Android/iOS APP homepage to enter Device Setting interface.

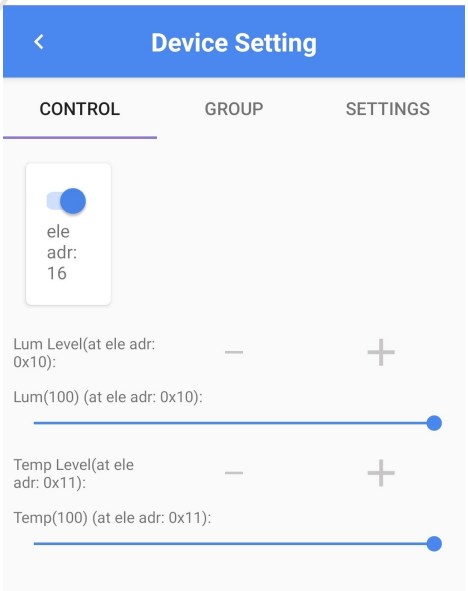


Figure 2.8: Light device Device Setting interface

2.6.1 Light Device Control

The "Ele Adr X" is used to switch the device on and off; "Lum Level" is used to adjust the brightness of the device; "Temp Level" is used to adjust the colour temperature of the device.

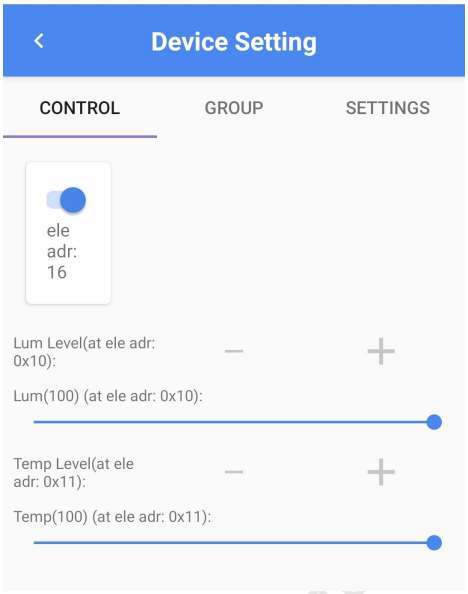


Figure 2.9: Android control interface

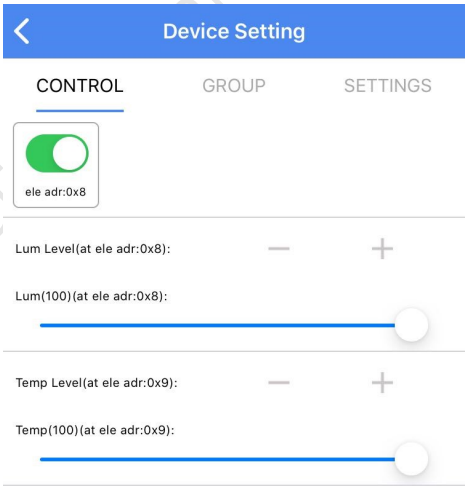


Figure 2.10: iOS control interface

2.6.2 Single Device Group

The "GROUP" is used to group the device (a single device supports a maximum of 8 groups).

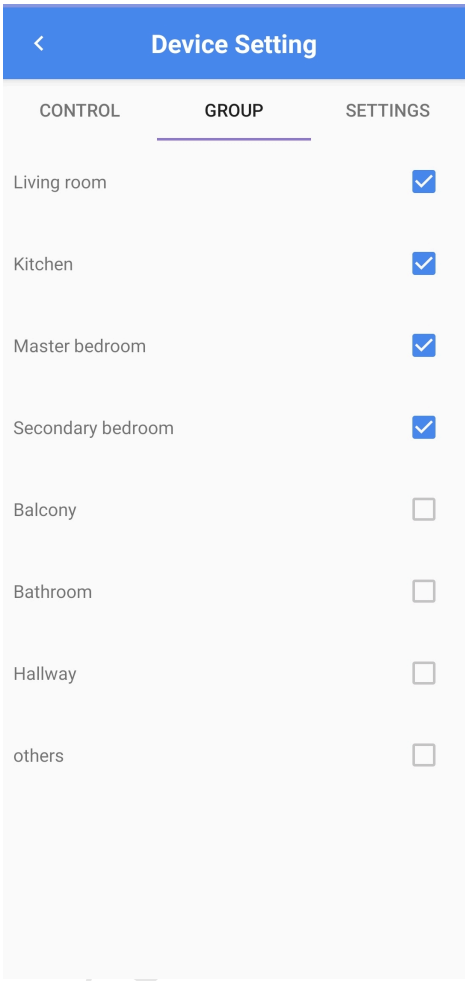


Figure 2.11: Android add group interface

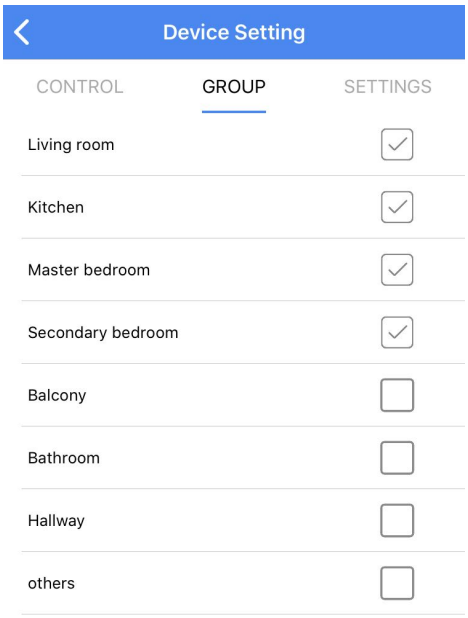


Figure 2.12: iOS add group interface

2.6.3 Light Device Settings

The Settings menu enables user to view the UUID, and execute composition data, schedulesetting, subscrIption models, device OTA, publication and kick out.

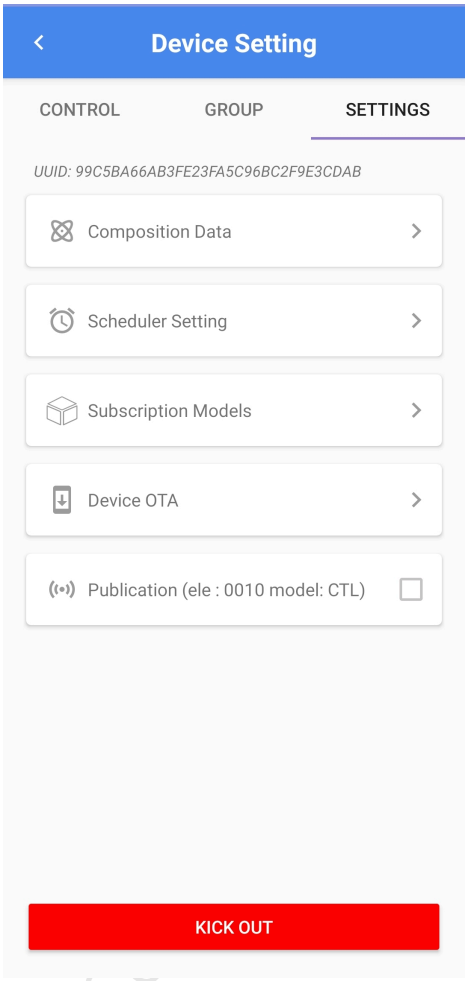


Figure 2.13: Android Settings interface

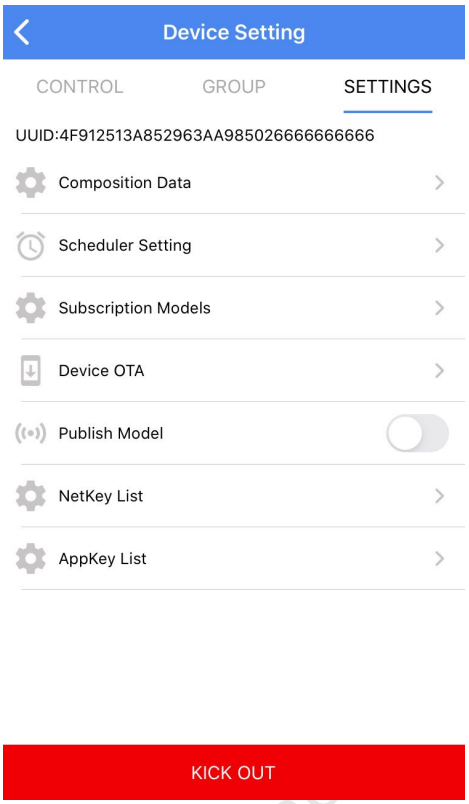


Figure 2.14: iOS Settings interface

2.6.3.1 Composition Data

The "Composition Data" is used to view the data of the device (including: cid/pid/vid/crpl/features/relay support/proxy support/freind support/low power support/position type of each sig model and vendor model). Clicking the icon on the top right corner can refresh the data.

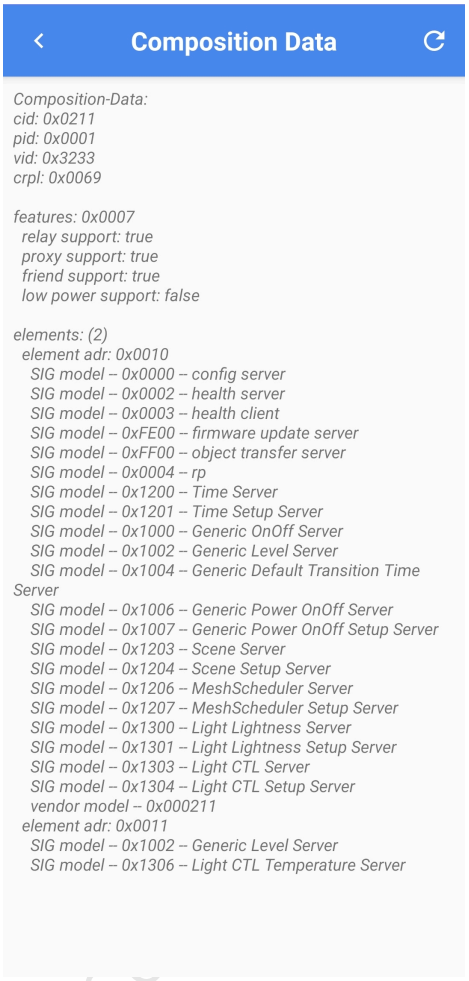


Figure 2.15: Android Composition Data

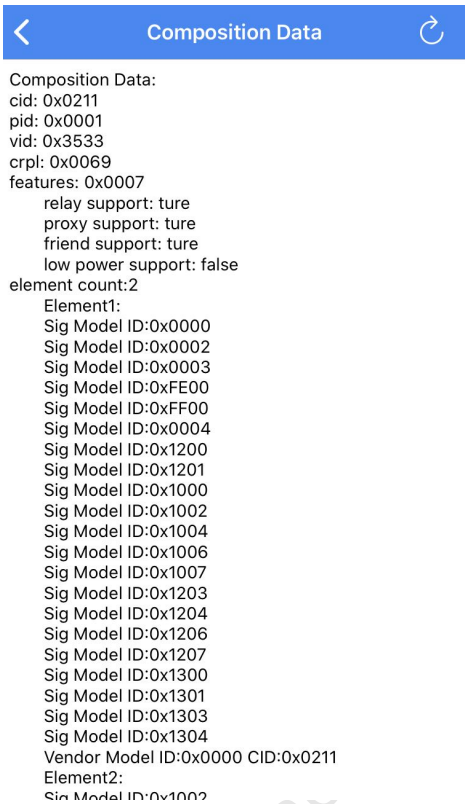


Figure 2.16: iOS Composition Data

2.6.3.2 Networ Keys (iOS：NetKey List / AppKey List)

The Network Keys (iOS: NetKey List / AppKey List) enables user to view the keys bound to the current device, as well as add a new Networ Key to the specified node so that different keys can be used to connect to different devices, or share the device by sharing the key. As follows:

Preset conditions: prepare two mobile phones A and B; add more than 2 devices to mobile phone A.

Steps for Android: (Android as mobile phone A)

(1) Mobile phone A creates a new Net Key for the specified device. The detailed operation:

Long press a device that needs to creat a new Net Key on the APP homepage – Settings – Network Keys – Click "+" on the upper right corner to select a Net key (Currently, there are two built-in Net key and APP key, which can be viewed in APP Home – Setting – Mesh info).

(2) Mobile phone A shares device to mobile phone B by sharing Net Key:

Mobile phone A APP home page – Setting – Share – Export – select the newly created Net Key – export by file/QR code;

Mobile phone B APP home page – Setting – Share – Import – import by file/QR code.

At this time, mobile phone B can only get the status of the device corresponding to the Net key, and the other device shows offline status due to different Net key.

Steps for iOS: (iOS as mobile phone A)

(1) Mobile phone A creates a new Net Key for the specified device. The detailed operation:

a. Click Setting in the lower right corner of APP homepage – Mesh info – Netkey List – create a new Net Key;

b. Return to Mesh info – App Key List – create a new App Key (Note: the key is the same as the currently existing App Key; index, BoundNetkey bind to the newly created Net Key);

c. Long press a device that needs to creat a new Net Key/App Key on the APP homepage – Settings – NetKeys List – Click "+" on the upper right corner to select a Net key – return to Device Setting – select AppKey List – Click "+" on the upper right corner to select a App key.

(2) Mobile phone A shares device to mobile phone B by sharing Net Key:

Mobile phone A APP home page – Setting – Share – Export – select the newly created Net Key – export by file/QR code;

Mobile phone B APP home page – Setting – Share – Import – import by file/QR code.

At this time, mobile phone B can only get the status of the device corresponding to the Net key, and the other device shows offline status due to different Net key.

2.6.3.3 Subnet Bridge Setting

The Subnet Bridge feature allows the configuration of bridge tables to nodes with multiple subnets, allowing messages to be forwarded to specific subnets. For example, node 1 is on networks A and B, and node 2 is on network A: if an APP with data on network B wants to control a node on network A, it needs to configure a bridge table from network B to A on node 1.

Mobile phone A operation (initial added device):

a. Long press a public device on the APP homepage – Setting – Subnet Bridge Setting;

b. Turn on "Enable Subnet Bridge" switch;

c. Click "Add Subnet Bridge" button;

d. In the "Add Bridging Table" interface, "Net key 1" fill in the shared Net key; "Net Key 2" fill in the Net key that needs to be converted (Note: this the Net key of the device that will be added);

e. "Address 1" fill in the Local Address of the shared Net key (view steps: mobile phone B importing Net key by sharing: APP homepage – Setting – Mesh info);

f. ""Address 2"" fill in the short address of the device to be controlled;

g. Click "Add Bridge Table" to save.

Mobile phone B operation (get netkey by sharing):

Long press the node specified by mobile phone A to enter Device Setting interface to switch the control node.

2.6.3.4 Scheduler

Click "+" on the top right corner of the Scheduler list interface to add scheduler. After setting the conditions in the scheduler setting interface, click  to get and set the time (click setTime for iOS). Then click  to save scheduler (Note: Scheduler is turned off by default, the device needs to enable MD_TIME_EN macro).

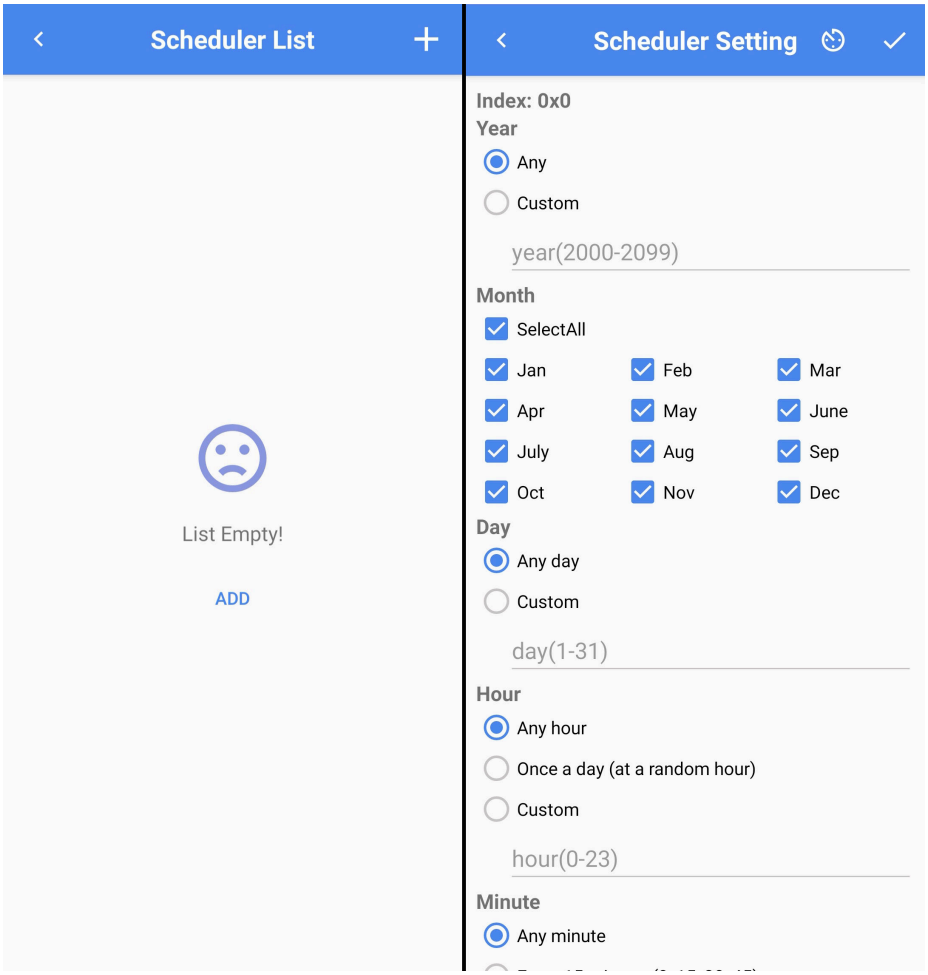


Figure 2.17: Android Scheduler

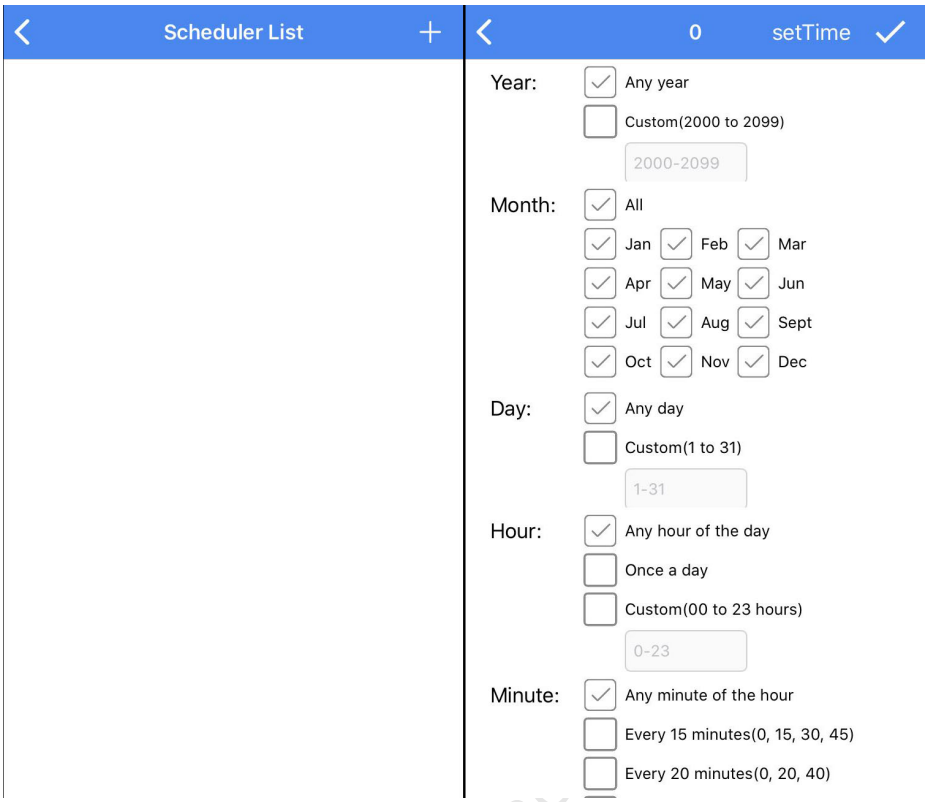


Figure 2.18: iOS Scheduler

The Schedulers added in the Scheduler list interface can also be edited by clicking . After setting the conditions in the scheduler setting interface, click to get and set the time (click setTime for iOS). Then click to save scheduler.

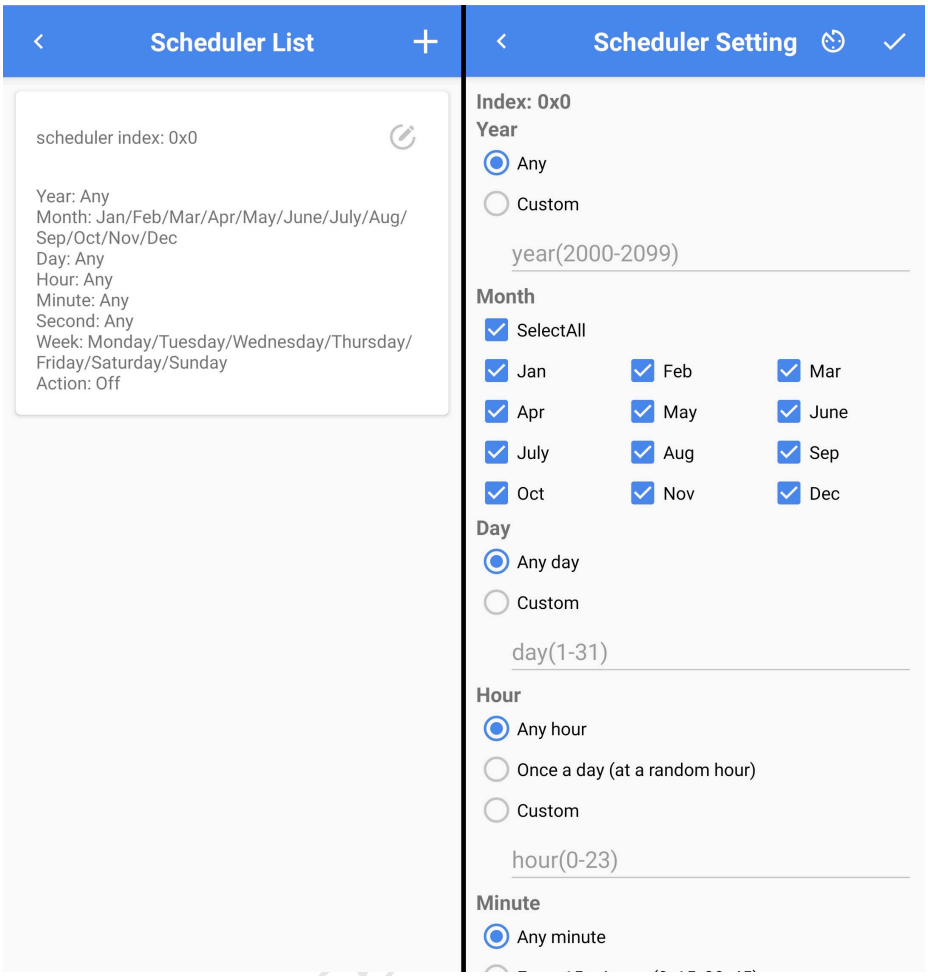


Figure 2.19: Android edit Scheduler

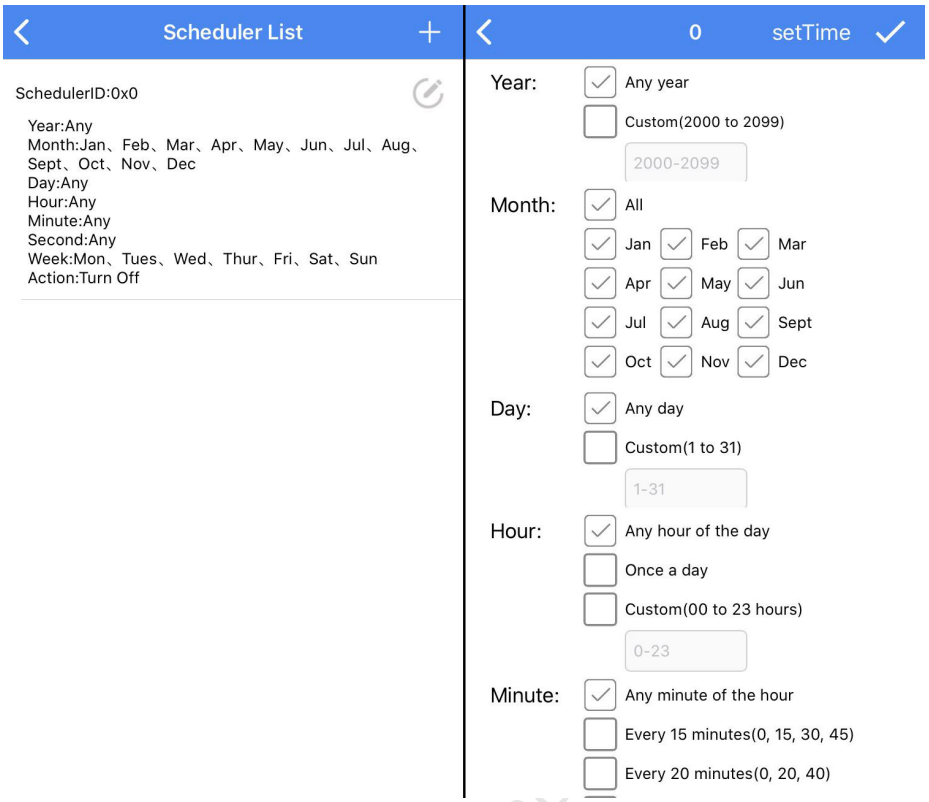


Figure 2.20: iOS edit Scheduler

2.6.3.5 Subscription Models

Subscription models can be viewed:

ID: 0x1000 (model name:Generic onoff server)

ID: 0x1300 (model name:Light Lightness server)

ID: 0x1303 (model name:Light CTL server)

ID: 0x1306 (model name:Light CTL Temperature server)

ID: 0x1307 (model name:Light HSL server)

2.6.3.6 Device OTA

Android APP:

Device OTA can perform GATT OTA upgrade on the device. The OTA interface can display the current device information, the different pid upgrade options in devices (unticked by default, users can tick the item as needed), the target version information (click select file to select firmware). Click "START" to start the upgrade, it will prompt start OTA and display the upgrade progress. When the upgrade is completed, the progress is displayed 100%, prompt OTA_SUCCESS, and the device flashes slowly. To check whether the device is upgraded to the target version, users can refresh and view the vid data by long pressing the device on the APP homepage – settings – Composition Data (refer to section 2.6.3.1 Composition Data).

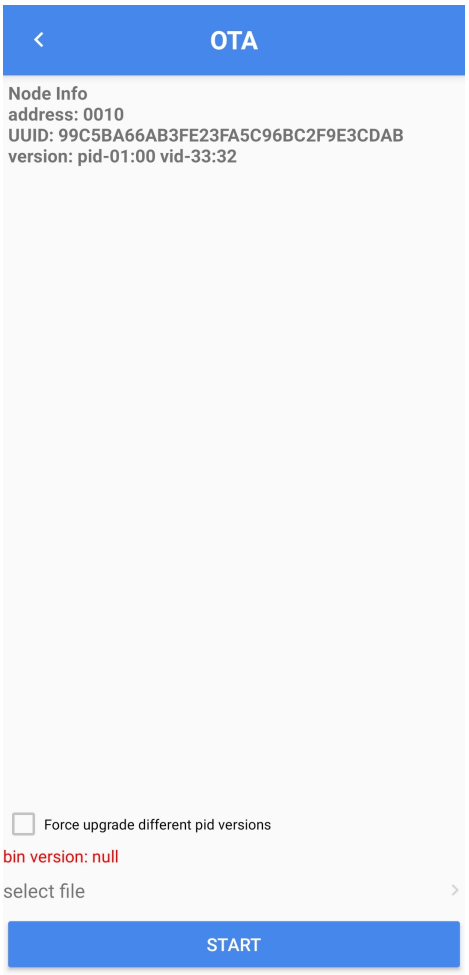


Figure 2.21: Android OTA interface-1

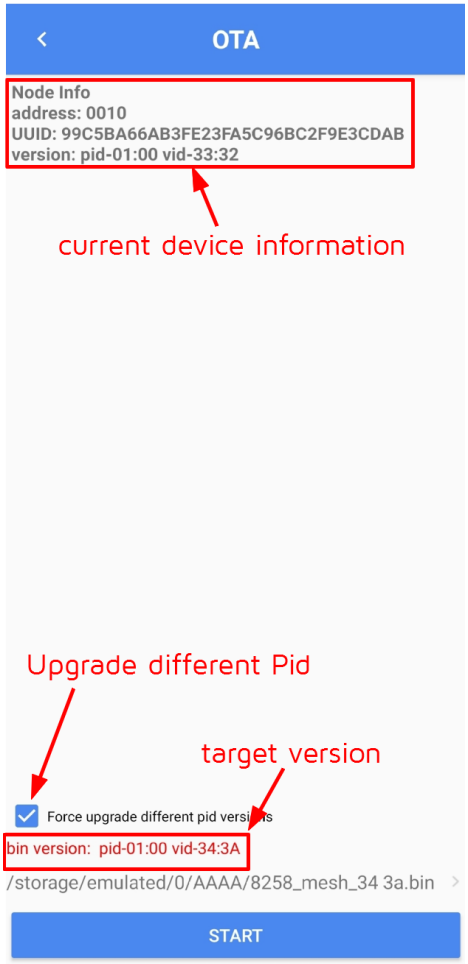


Figure 2.22: Android OTA interface-2

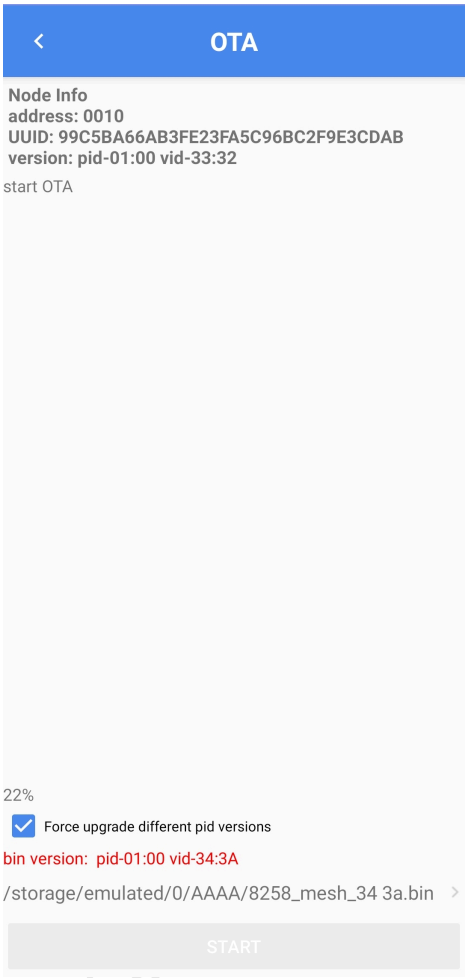


Figure 2.23: Android OTA interface-3

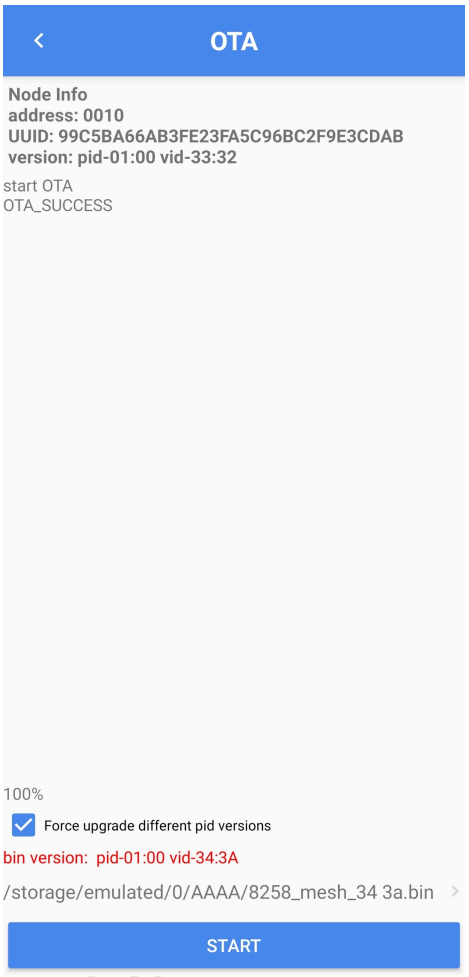
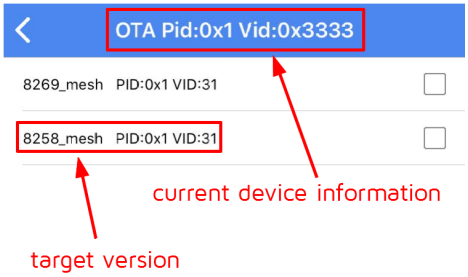


Figure 2.24: Android OTA interface-4

iOS APP:

The OTA interface can display the current device information and the target version Pid and Vid. Tick the corresponding version and click "START" to start the upgrade. When the upgrade is completed, the device flashes slowly. To check whether the device is upgraded to the target version, users can refresh and view the vid data by long pressing the device on the APP homepage – settings – Composition Data (refer to section 2.6.3.1 Composition Data).



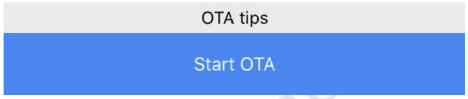
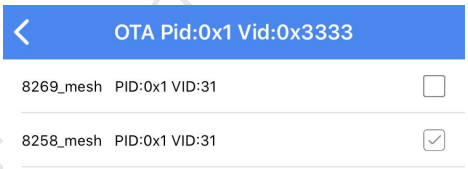


Figure 2.25: iOS OTA interface -1



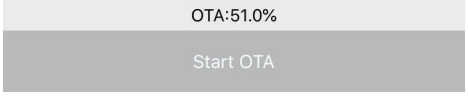


Figure 2.26: iOS OTA interface -2

2.6.3.7 Publication (ele：xxxx model：CTL)

The Android/iOS APP will send status every 20 seconds after opening the corresponding device publication (it can be viewed in the log interface, CT light reports Ctlstatusmessage, HSL light reports Hslstatusmessage).

2.6.3.8 KICK OUT

The "KICK OUT" is used to kick out the current device. After kicking out, the device flashes slowly, and the device will be in the state of pending network.

2.7 Device Setting (Switch Device)

Long press the SW10 + SW13 keys of the unnetworked Switch remote control to trigger the broadcast (Note: the flicker frequency of the unnetworked state light is 200ms/s, and the networking state is 500ms/s) for networking. After the networking is successful, the broadcast also needs to be triggered. Long presses the Switch icon on the APP homepage to enter the device setting interface to connect the remote control. At this time, the bottom of the interface will show "device connected" which represents the connected. If the connection fails, the position will pop up a button and user can click to reconnect.

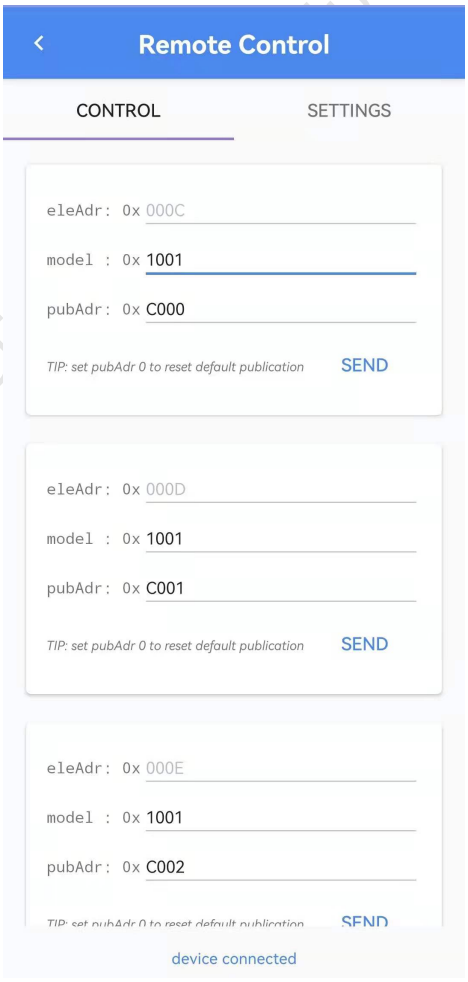


Figure 2.27: Android Switch device, Device Setting interface

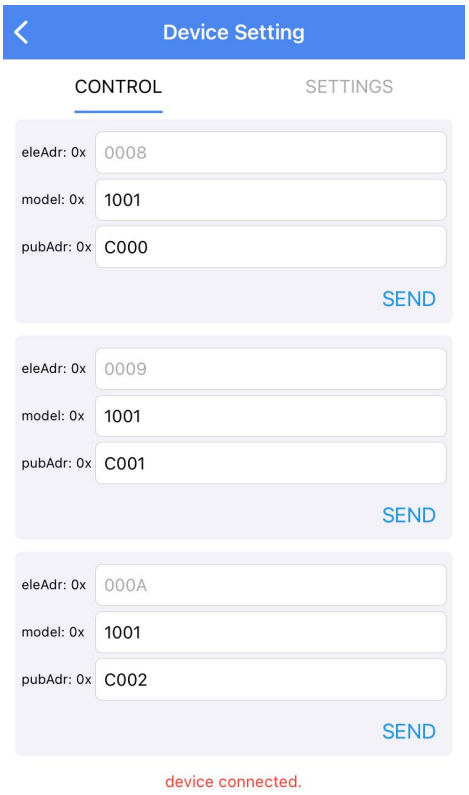


Figure 2.28: iOS Switch device Device, Setting interface

2.7.1 Switch Device Control

The 0x0008 in Ele adr corresponds to Switch remote control SW7/SW10 keys, 0x0009 corresponds to SW8/SW11, 0x000A corresponds to SW9/SW12, 0x000B corresponds to SW3/SW6, and Model can execute Switch-supported models (for details, please refer to Switch Device Composition Data). 0xC000 in Pubadr corresponds to Living room in Group, 0xC001 corresponds to Kichen, 0xC002 corresponds to Masterbedroom, and 0xC003 corresponds to Secondary bedroom. The Group to be controlled can be set at the specified key.

2.7.2 Switch Device Setting

Please refer to 2.6.3 Light Device Settings.

3 Group

There are 8 groups in the Group interface (refer to section 2.6.2 Single Device Group), and assign devices to groups before operation.

Grouping method：long press a device in Device interface to group.

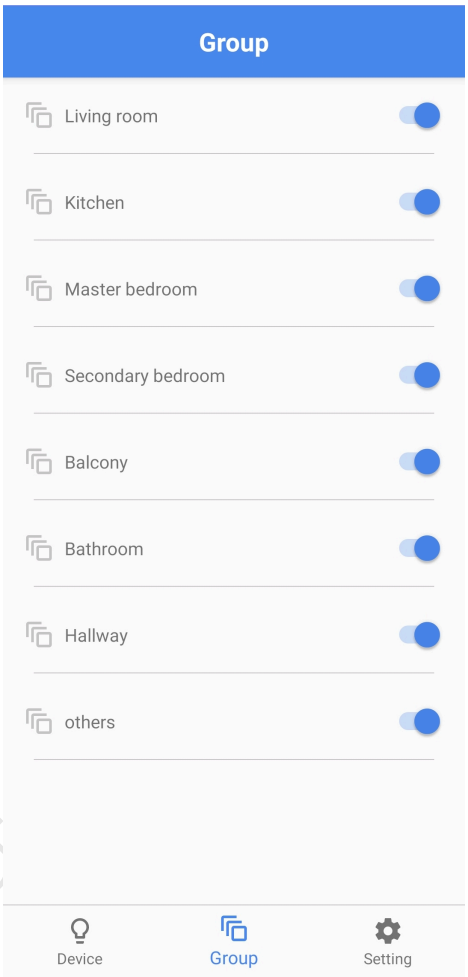


Figure 3.1: Group for Android

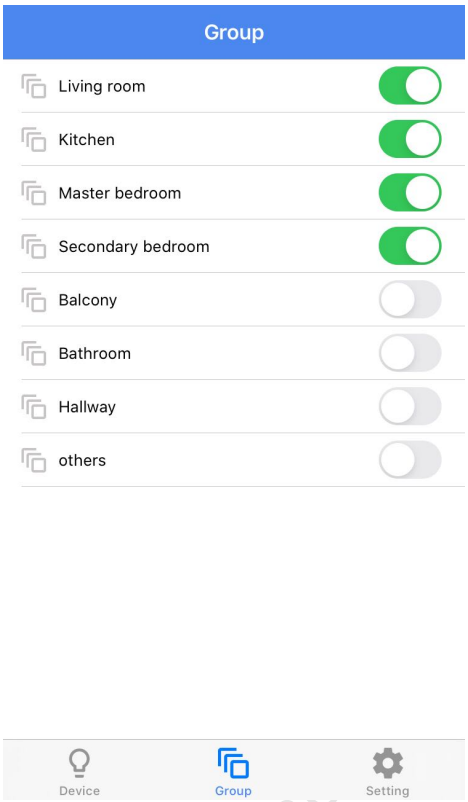


Figure 3.2: Group for iOS

3.1 On/Off Group

The Group interface enables user to On/Off the devices belonging to the corresponding group.

3.2 Group Setting

Long press the corresponding Group to enter the Group Setting interface.

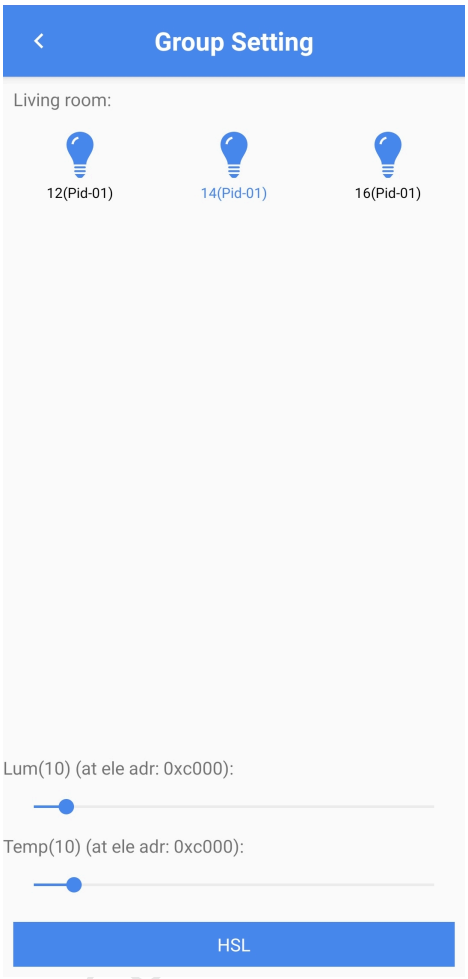


Figure 3.3: Group Setting for Android

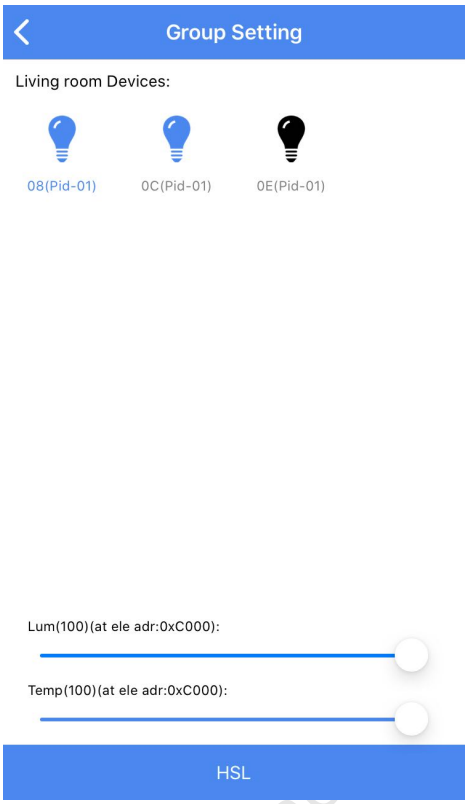


Figure 3.4: Group Setting for iOS

3.2.1 On/Off Group Devices Individually

Click the device icon in the Group setting interface to on/off the device (the blue icon is On status, the gray is Off status), the blue device name is the direct connection device.

3.2.2 Lum & Temp

Lum adjusts the brightness of the devices belonging to the group, and Temp adjusts the color temperature.

3.2.3 HSL

The color palette enables user to adjust the color of the GRB, or user can adjust the color by adjusting R, G, B or H, S, L individually, RGB corresponds to HSL color, and V below the palette can adjust the brightness.

Note:

> The device is required to enable the LIGHT_TYPE_HSL macro.

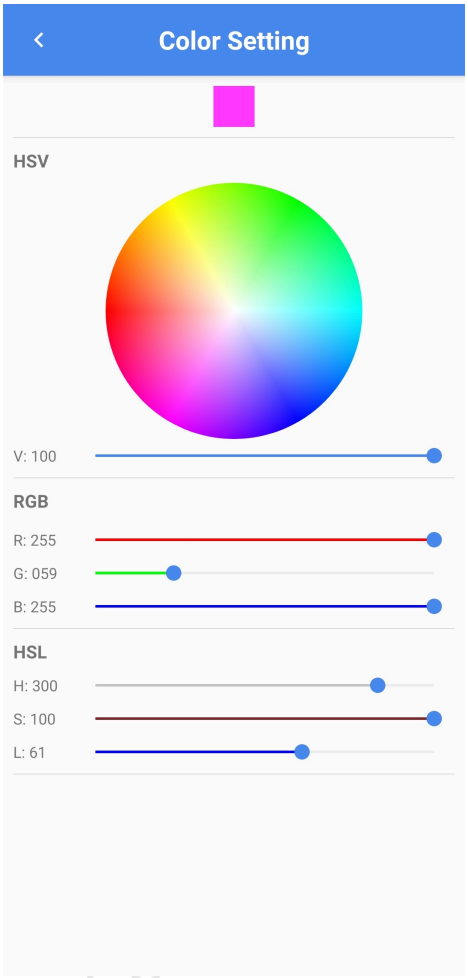


Figure 3.5: HSL for Android

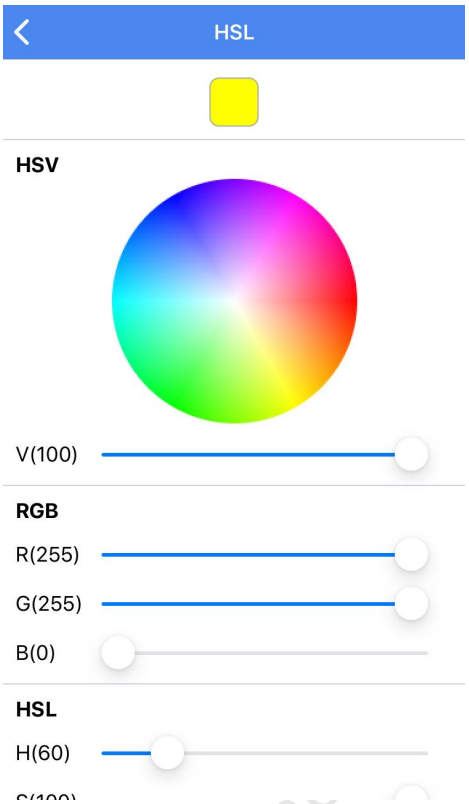


Figure 3.6: HSL for iOS

4 Setting Interface

The Setting interface enables user to execute: Scenes, Share, Mesh OTA, settings/Mesh Info (Enable LOG, Auto provision, Private Mode, Enable DLE Mode Extend Bearer, Fast Provision, OOB database, Use no-oob Automatically, Net Key/App Key, Online Status, Reset Mesh).

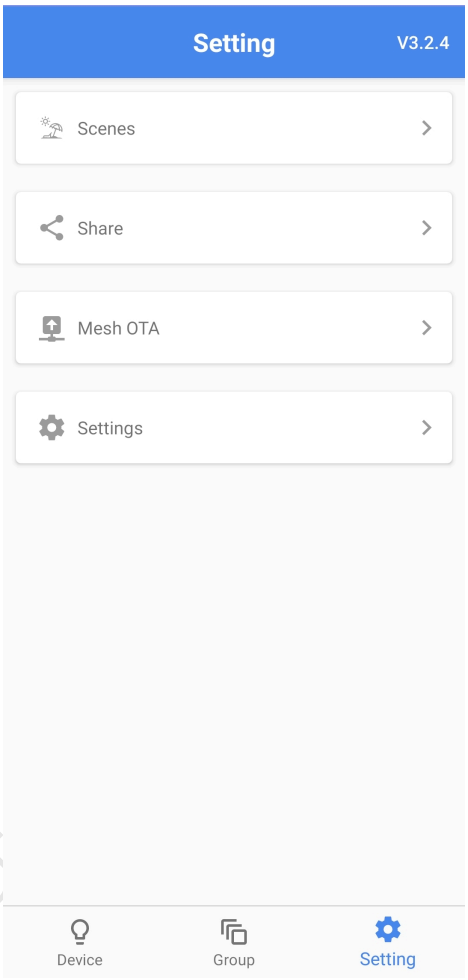


Figure 4.1: Setting for Android

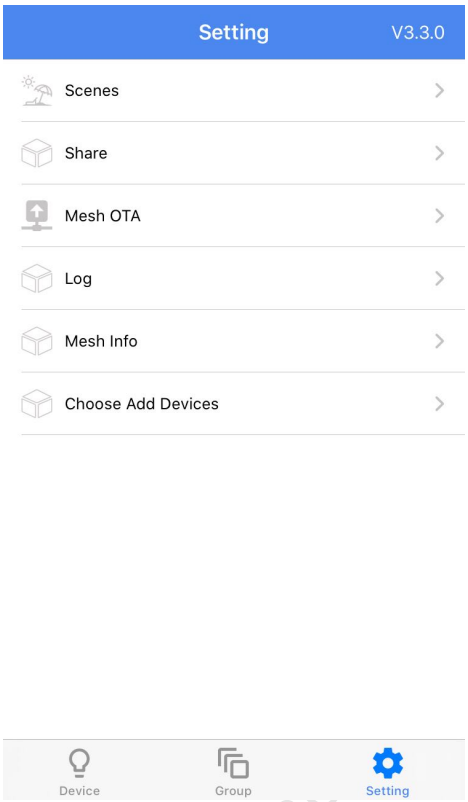


Figure 4.2: Setting for iOS

4.1 Scenes

User can use the symbol "+" in the upper right corner of the Android / iOS APP Scene interface to save the current state of the specified device as Scenes. After saving, user can start Scenes in Scene List by , edit Scenes by , and delete Scenes by long-pressing Scene ID.

Note:

- Scenes are disabled by default and enabling the MD_SCENE_EN macro is necessary.
- Adjust the device scene before setting Scenes.

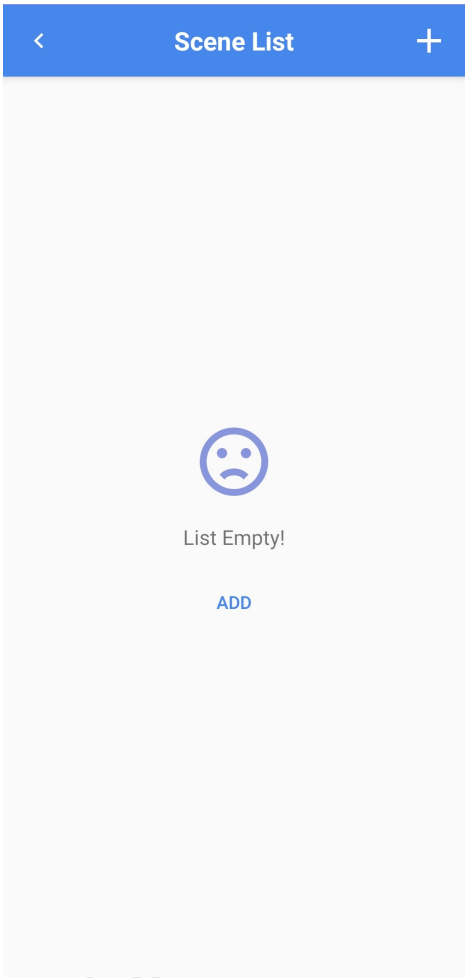


Figure 4.3: Scene-1 for Android

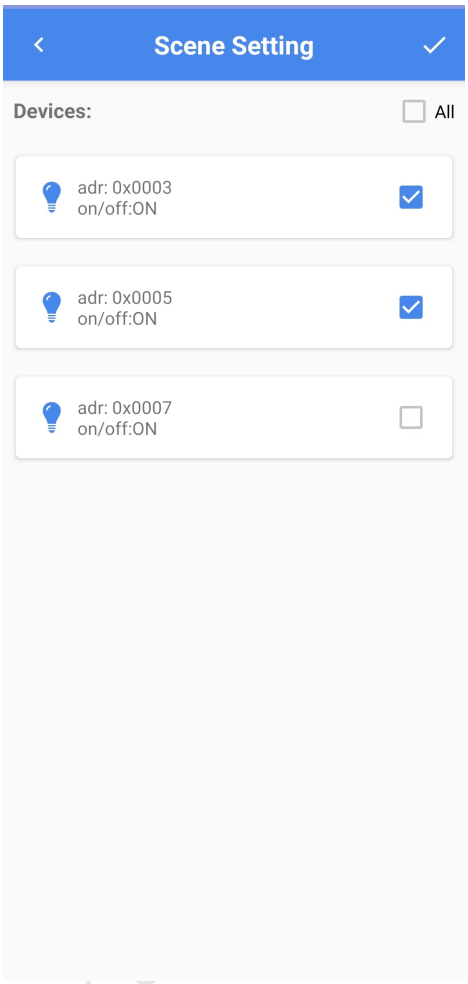


Figure 4.4: Scene-2 for Android

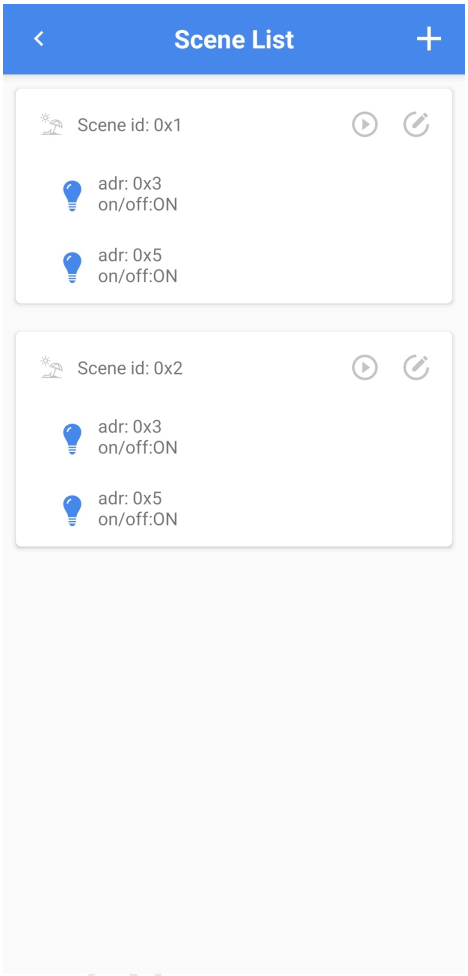


Figure 4.5: Scene-3 for Android

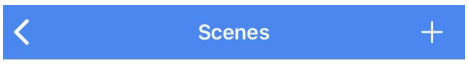


Figure 4.6: Scene-1 for iOS

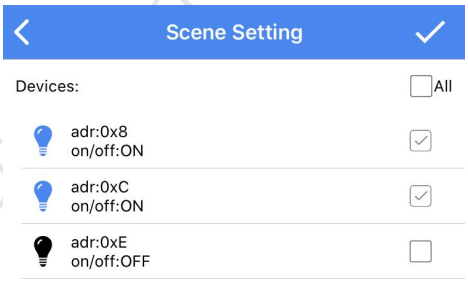


Figure 4.7: Scene-2 for iOS

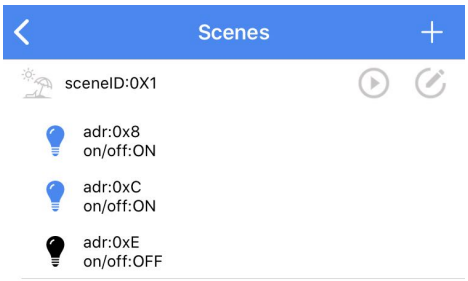


Figure 4.8: Scene-3 for iOS

4.2 Share

4.2.1 Export mesh

4.2.1.1 Export by Json File

Select the "Net Key" that needs to be shared on the Android/iOS APP Share interface "EXPORT" tab, select the "Json File" as the sharing method. Then click the "EXPORT" button to export the json file. The json file exported by the Android version appp will be saved at storage/emulated/0/TelinkBleMesh, and the json file exported by the iOS version app will be shared through the iTunes file.

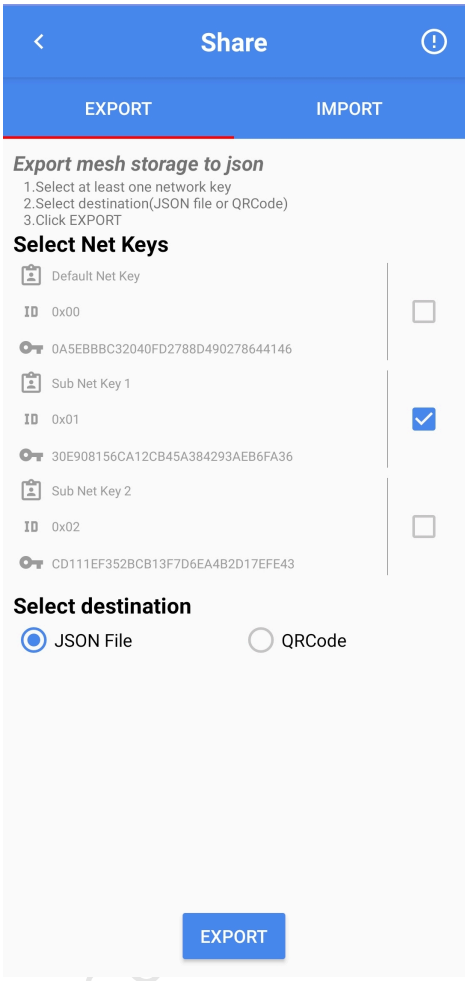


Figure 4.9: Android json file export

Figure 4.10: iOS json file export

4.2.1.2 Export by QR Code

Select the "Net Key" that needs to be shared on the Android/iOS APP Share interface "EXPORT" tab, select the "QRCode" as the sharing method. Clicking the "EXPORT" button will display the QRCode (the QR code has a time limit and will expire after 300 seconds).

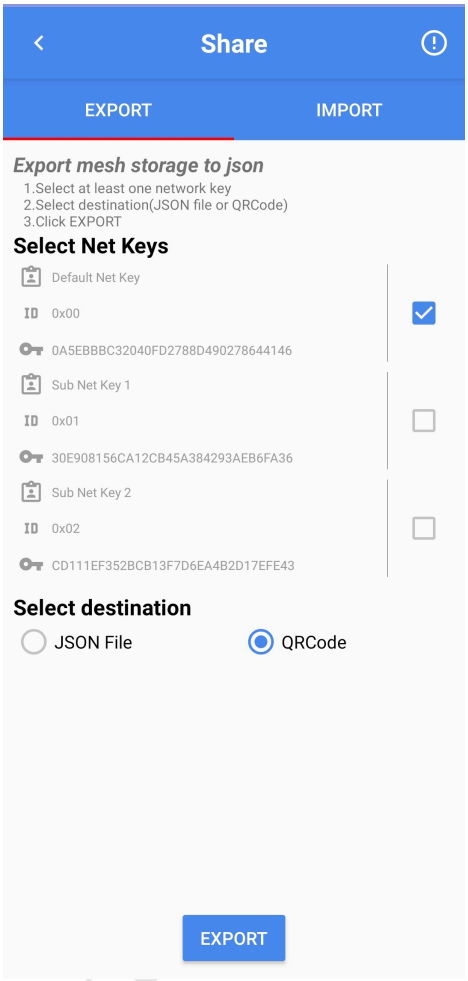


Figure 4.11: Android QRcode export -1

Figure 4.12: Android QRcode export -2

Figure 4.13: iOS QRcode export -1

Figure 4.14: iOS QRcode export -2

4.2.2 Import mesh

4.2.2.1 Import by Json File

Android APP:

On the Android APP Share interface "IMPORT" tab, select the "Json File" as the importing method. Select the json file that needs to be imported in the "Select File", then click the "IMPORT" button to import the mesh.

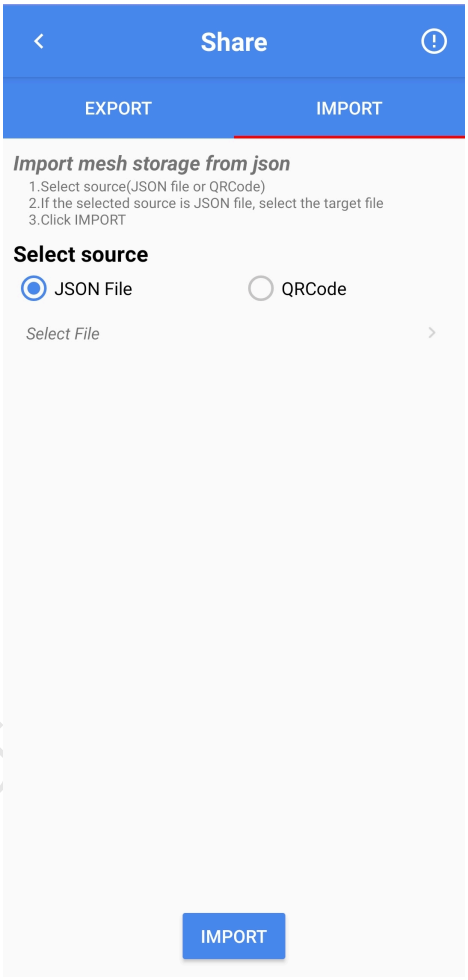


Figure 4.15: Android json file import

iOS APP:

Put the json file into TelinkSigMesh APP through iTunes, select the "Json File" as the importing method on the iOS APP Share Mesh interface "IMPORT" tab. Then click the "IMPORT" button, select the json file and click the "IMPORT" button again.

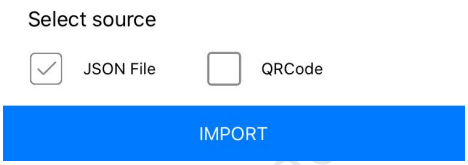


Figure 4.16: iOS json file import -1

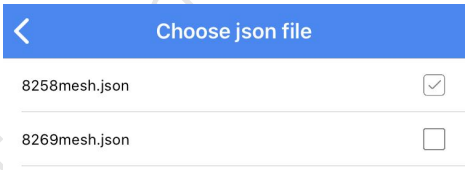


Figure 4.17: iOS json file import -2

4.2.2.2 Import by QRCode

Android/iOS APP:

On the Android/iOS APP Share interface "IMPORT" tab, select the "QRCode" as the importing method. Click the "IMPORT" button and then scan the QR Code to import mesh network.

4.2.3 Preview

After exporting a json file in Share interface export tab, user can view the content of the last exported json file via the preview button, and after selecting a json file in Share interface export tab, user can view the content of the selected json file via the preview button (currently supported only on the Android version).

4.2.4 Tip

Clicking the "!" button in the upper right corner of the Share interface of Android version can enter the Tips interface to view the operation guide of importing and exporting. Clicking the "Tips" button in the upper right corner of the Share interface of iOS version can enter the Share Tips interface to view the operation guide of importing and exporting.

4.3 Settings

The Settings interface enables user to execute Enable LOG, Auto provision, Private Mode, Enable DLE Mode Extend Bearer, Fast Provision, OOB database, Use no-oob Automatically, Net Key/App Key, Online Status, and Reset Mesh.

Note: Android and iOS Settings are slightly different, please see the picture below for details:

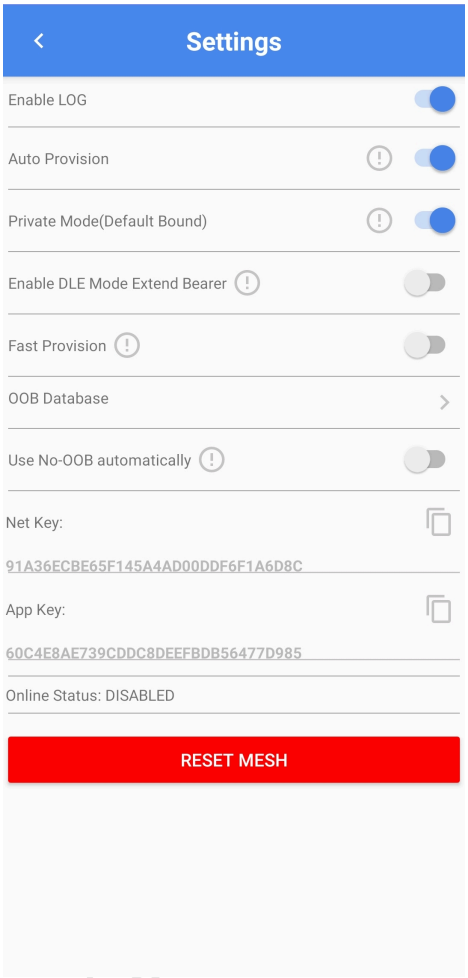


Figure 4.18: Android Setting/Settings

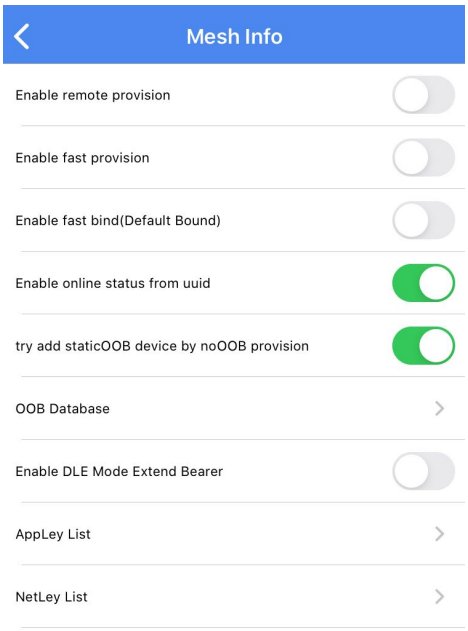


Figure 4.19: iOS Setting/Settings

4.3.1 Enable Log

Android APP:

Turning on Enable Log can record the log information when controlling mesh. This item is turned off by default and can be turned on as needed (please refer to the introduction of section 2.5 log).

iOS APP:

There is no Log switch, and the App turns on the Log function by default.

4.3.2 Auto Provision

When Auto Provison is enabled in Android APP, it will automatically add peripheral devices to be networked when adding devices, this item is off by default and can be turned on as needed.

Note:

> Both the Android and iOS APPs support both manual and automatic networking modes. However, iOS APP does not set the Auto Provison switch (refer to the introduction of section 1.1 Manual Provision networking and section 1.2 Auto Provision networking).

4.3.3 Private Mode（Default Bound）

Default Bound is the default binding mode, which requires device support. In this mode, the app key binding process can be completed only if the app key add is successfully executed, and the device will automatically bind the app key to all models that need to be bound.

Note:

> In iOS, this item shows Enable fast bind (Default Bound).

4.3.4 Remote Provision

The remote provision networking is to add multi-hop range of devices one by one, able to add a longer distance devices, while with the function of relay. The detailed operations:

(1) Network a device that supports Remote Provision in ordinary mode (with Remote Provision turned off), i.e. enable the MD_REMOTE_PROV macro;

(2) Click on "Setting" on the APP home page;

(3) Settings (Mesh info);

(4) Open Remote Provision switch in the APP;

(5) Click "+" for Remote Provision on the APP home page.

Note:

> The remote provision is turned off by default and requires the device to enable the MD_REMOTE_PROV macro.

4.3.5 Enable DLE Mode Extend Bearer

The Enable DLE Mode Extend Bearer is optional for sending long packets and requires device support. After enabling, the maximum length of the access layer short packet will be changed from 11 to 225.

4.3.6 Fast Provision

The Fast provision batch networking mode can simultaneously network multiple devices that are not networked in the multi-hop range. The device key used is generated based on the mac address according to certain rules, and does not need to be allocated separately. The operation steps are as follows:

(1) Click on "Setting" on the APP home page;

(2) In Settings (Mesh info), turn on the fast provision switch;

(3) Back to the APP home page, click on the upper right corner "+" to add the device.

Note:

> Fast provision is turned off by default and requires the device to enable the FAST_PROVISION_ENABLE macro.

4.3.7 OOB Database

The OOB Database is used for the App to look up the corresponding Auth Value of the device when the device supports static-oob method for provisioning.

When the App looks for Auth Value, it will use device UUID as the key to look up the table from the database. If the device writes OOB data, it is necessary to enter the corresponding UUID and OOB data in the APP to network normally, otherwise the network will fail.

4.3.7.1 Manually Add OOB Database

Click "+" in the upper right corner of the OOB List interface, select "Manual Input" to input UUID and OOB data.

The UUID and OOB queries are as follows:

OOB: burn 8258 mesh, 8269 mesh, 8278 mesh and other projects, write 16-bit OOB data at 77800 position.

UUID: View the UUID of the device with universal BLE APP in the state of device to be networked. The detailed operation:

(1) the APP scans the device that needs to obtain the UUID;

(2) View the Complete list of 16-bit Service UUID.

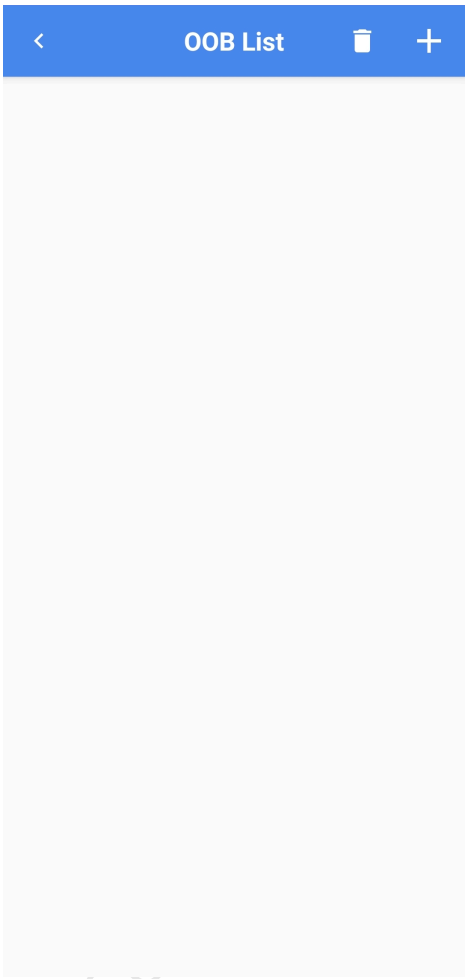


Figure 4.20: OOB List

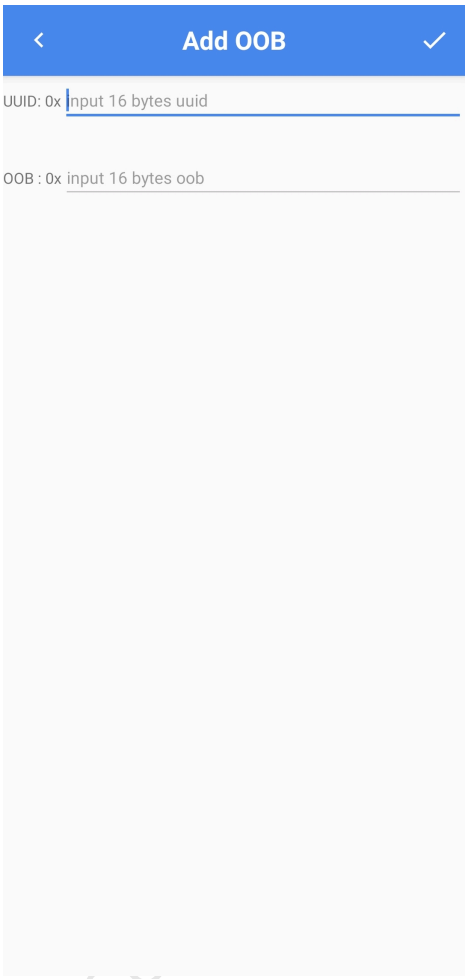


Figure 4.21: Add OOB

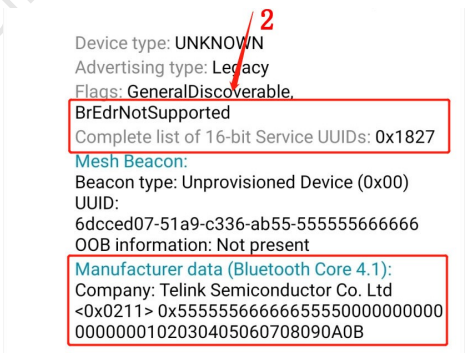


Figure 4.22: General APP

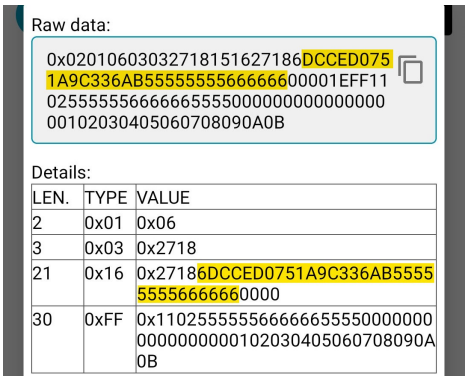


Figure 4.23: General APP RAW window

4.3.7.2 Import OOB Database by TXT File

Create a new txt document; enter a 16-byte UUID, enter OOB data in a blank space and save it; click on the upper right corner "+" in the OOB List interface to select "Import from file"; select the txt file saved.

4.3.7.3 Delete OOB Database

Long press on one of the OOB data to delete the OOB data individually, and click on the bin icon in the upper right corner to empty all OOB data.

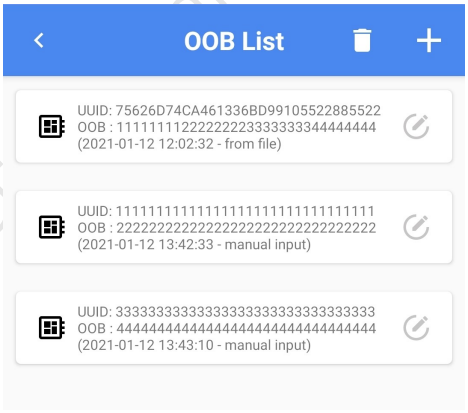


Figure 4.24: Delete OOB Database

4.3.8 Use No-OOB Automatically

Use No-OOB Automatically can add devices that have OOB data written at the 77800 location, but have not entered that OOB data on the APP (provided the ENABLE_NO_OOB_IN_STATIC_OOB macro is enabled).

4.3.9 Net Key / APP Key

Android APP:

The Net Key and APP Key of the current mesh can be viewed and copied.

iOS APP:

The NetKey and AppKey of the mesh can be viewed, added, and edited (in the condition of unbound devices).

4.3.10 Online Status

Online Status can be used to check whether the current connected device supports the Online Status function and report the status when the device status changes.

Note:

> This item shows Enable OnlineStatus From UUID on the iOS APP.

4.3.11 Reset Mesh

Reset Mesh can be used to reset the mesh by updating the Net Key and App Key.

Note:

> Currently only Android APP is supported.

4.4 Mesh OTA

Android APP:

The Mesh OTA can perform OTA upgrade on multiple devices specified by the mesh network at the same time. There are 3 ways to load Mesh OTA:

(1) No Extend (short packet loading for all nodes);

(2) Extend GATT Only (long packet loading for directly connected nodes, short packet loading for non-directly connected nodes);

(3) Extend GATT & ADV (long packet loading for all nodes, LPN node upgrading).

Path: Click "Setting" on the lower right corner of the APP home page – "Settings" – "Extend & GATT Mode".

```
Extend option
Extend Bearer Mode

0: short packets for all nodes                                                    No
 ↪  Extend
1: long packets for directly connected nodes, short packet for non-directly connected nodes
 ↪  Extend GATT Only
2: long packets for all nodes (LPN node upgrading)
 ↪  Extend GATT & ADV
```

(1) Mesh OTA is off by default, and the device needs to enable the Masterdongle and Gateway MD_MESH_OTA_EN macros respectively. If it is not enabled, Mesh OTA will not be supported, and the device cannot be ticked. Enable method: enable MD_MESH_OTA_EN in mesh_config.h file.

(2) After the Mesh OTA upgrade is completed, exit the Mesh OTA interface and re-enter to read the upgraded version.

(3) The iOS version needs to put the upgraded bin file into itunes – file sharing – TelinkSigMesh to be displayed on the mesh OTA upgrade interface.

(4) LPN node upgrades require the Extend GATT & ADV loading method.

4.4.1 Distributor: Phone Upgrade

Selecting "Phone" of "Distributor" to upgrade, it will directly transmit OTA data to the target device through the phone.

The detailed operation steps:

Android:

Click "Setting" – "Mesh OTA" on the bottom right of the APP home page, follow the steps in the figure below.

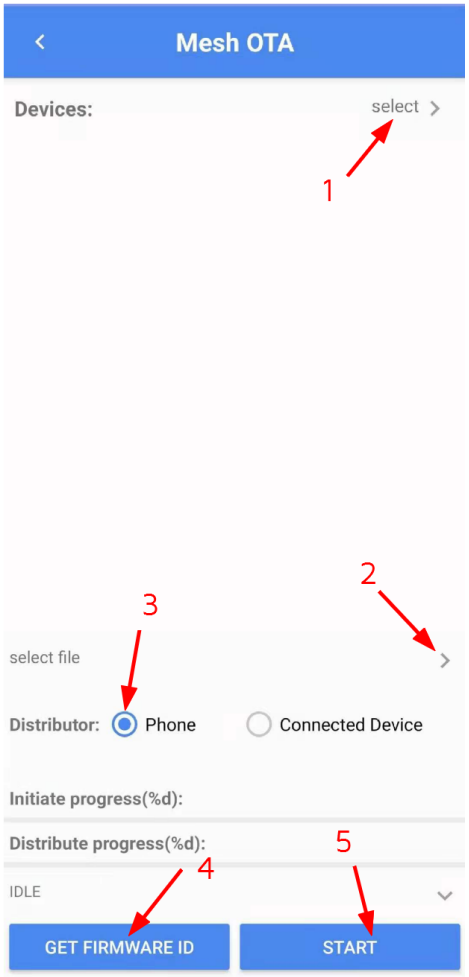


Figure 4.25: Android phone upgrade steps

(1) **Devices - select**: Select the device to be upgraded;

(2) **select file**: Select the file to be upgraded;

(3) **Distributor**: Select the phone loading method;

(4) **GET FIRMWARE ID**: Get the current device firmware ID;

(5) **START**: Start to upgrade.

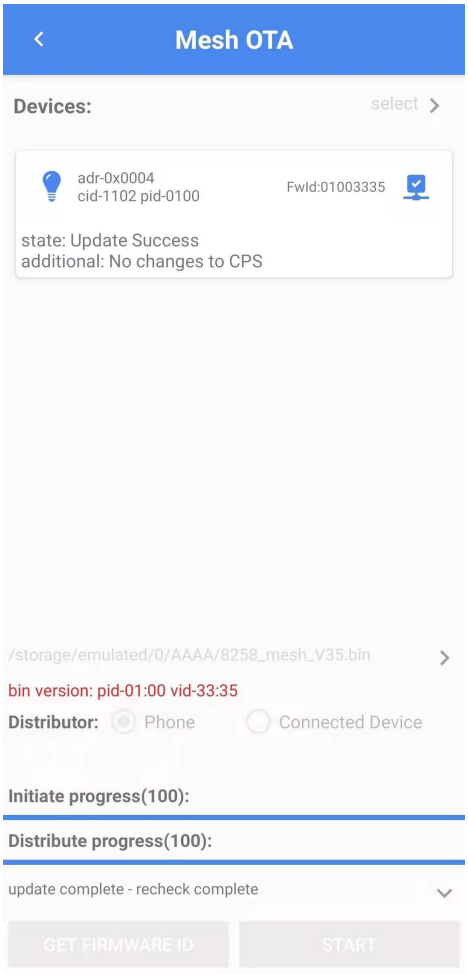


Figure 4.26: Android phone upgrade successful

iOS:

Click "Setting" – "Mesh OTA" on the bottom right of the APP home page, follow the steps in the figure below.

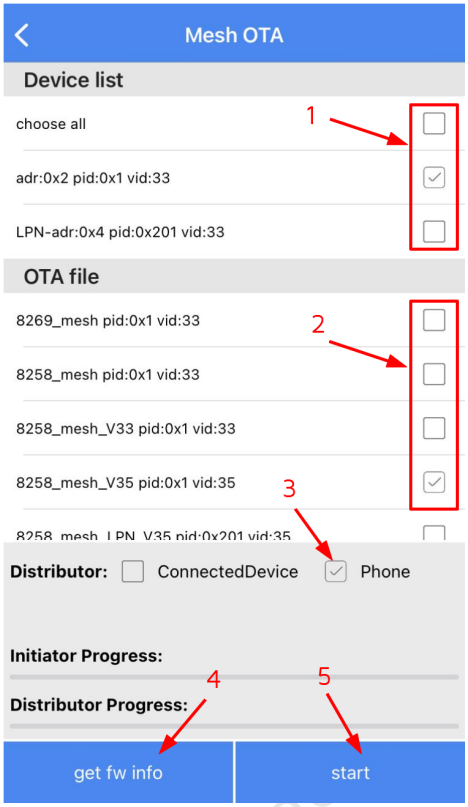


Figure 4.27: iOS phone upgrade steps

(1) **Device list**: Select the device to be upgraded;

(2) **OTA file**: Select the file to be upgraded;

(3) **Distributor**: Select the phone loading method;

(4) **get fw info**: Get the current device firmware ID;

(5) **start**: Start to upgrade.

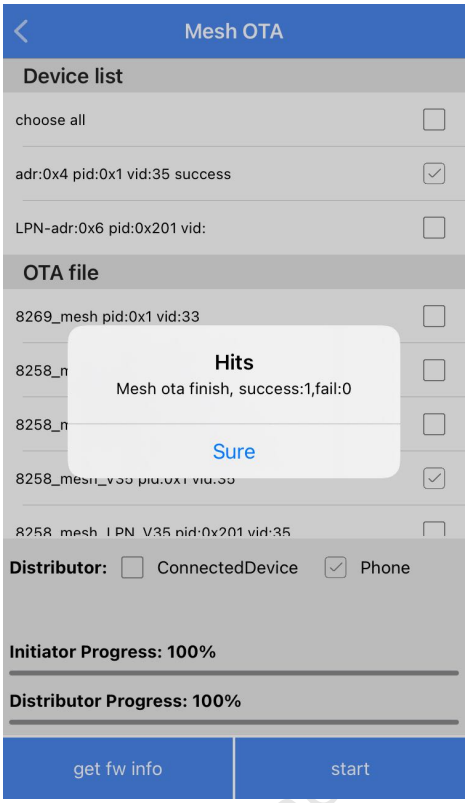


Figure 4.28: iOS phone upgrade successful

4.4.2 Distributor：Verify and Apply Upgrade

Selecting "Phone" of "Distributor" and "Verify and Apply" of "Apply Policy" to upgrade, it will upload the firmware to the direct connection node through the mobile phone, and then distributes it to the target node through the direct connection node, and automatically applies the new version after loading.

The detailed operation steps:

Android:

Click "Setting" – "Mesh OTA" on the bottom right of the APP home page, follow the steps in the figure below.

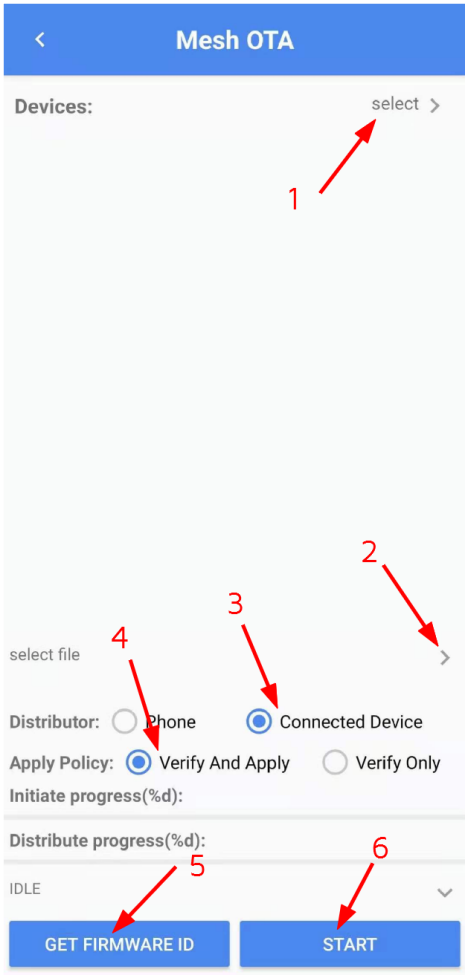


Figure 4.29: Android verify and apply upgrade steps

(1) **Devices - select**: Select the device to be upgraded;

(2) **select file**: Select the file to be upgraded;

(3) **Distributor**: Select Connected Device;

(4) **Apply Policy**: Select Verify And Apply;

(5) **GET FIRMWARE ID**: Get the current device firmware ID;

(6) **START**: Start to upgrade.

Figure 4.30: Android verify and apply upgrade successful

iOS:

Click "Setting" – "Mesh OTA" on the bottom right of the APP home page, follow the steps in the figure below.

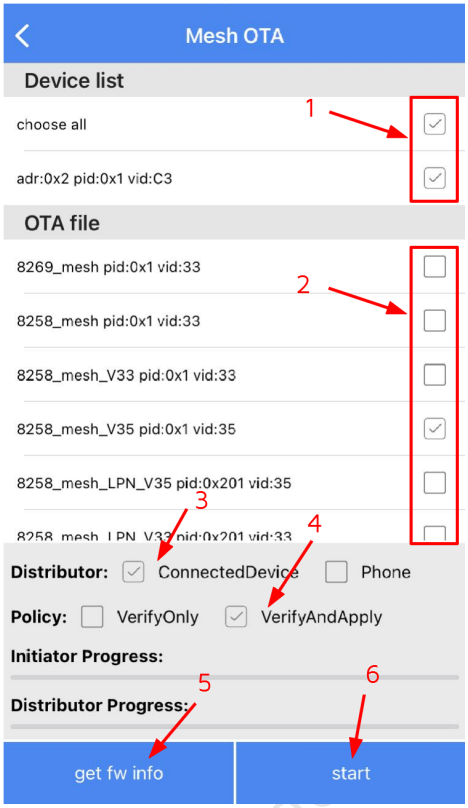


Figure 4.31: iOS verify and apply upgrade steps

(1) **Device list**: Select the device to be upgraded;;

(2) **OTA file**: Select the file to be upgraded;

(3) **Distributor**: Select ConnectedDevice;

(4) **Policy**: Select VerifyAndApply;

(5) **get fw info**: Get the current device firmware ID;

(6) **start**: Start to upgrade.

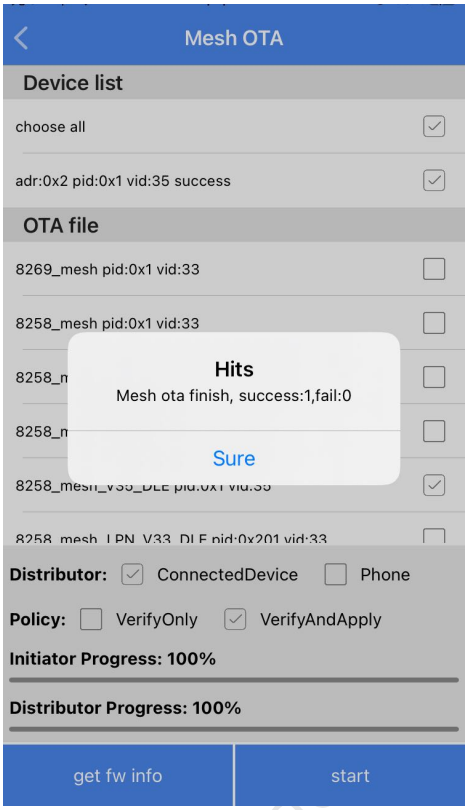


Figure 4.32: iOS verify and apply upgrade successful

4.4.3 Distributor：Verify Only Upgrade

Selecting "Verify only" of "Distributor" to upgrade, it will upload the firmware to the direct connection node through the mobile phone, and then distributes it to the target node through the direct connection node, and the APP needs to reconnect the node before applying the new version after loading.

The detailed operation steps:

Android:

Click "Setting" - "Mesh OTA" on the bottom right of the APP home page, follow the steps in the figure below.

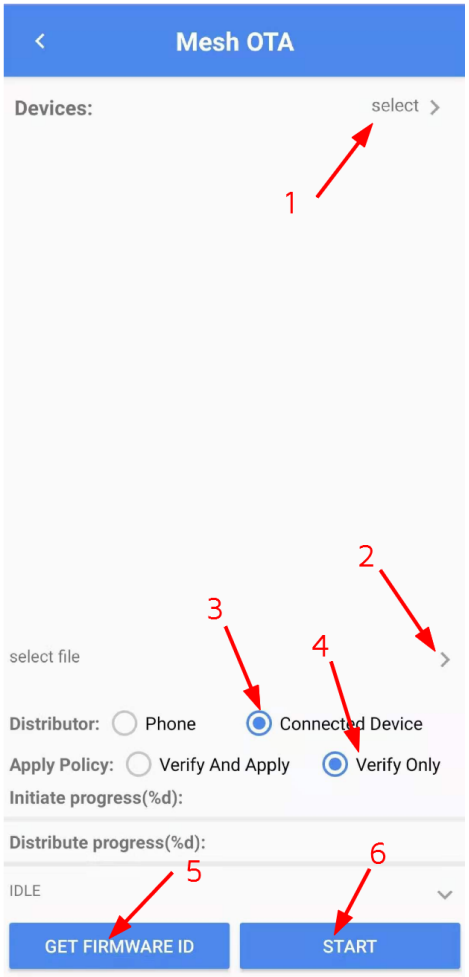


Figure 4.33: Android verify only upgrade steps

(1) **Devices - select**: Select the device to be upgraded;

(2) **select file**: Select the file to be upgraded;

(3) **Distributor**: Select Connected Device;

(4) **Apply Policy**: Select Verify Only;

(5) **GET FIRMWARE ID**: Get the current device firmware ID;

(6) **START**: Start to upgrade.

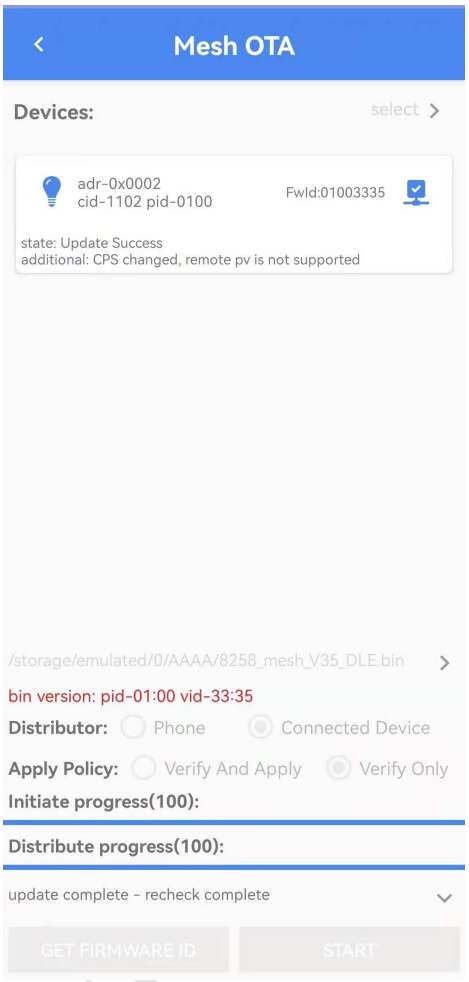


Figure 4.34: Android verify only upgrade successful

iOS:

Click "Setting" – "Mesh OTA" on the bottom right of the APP home page, follow the steps in the figure below.

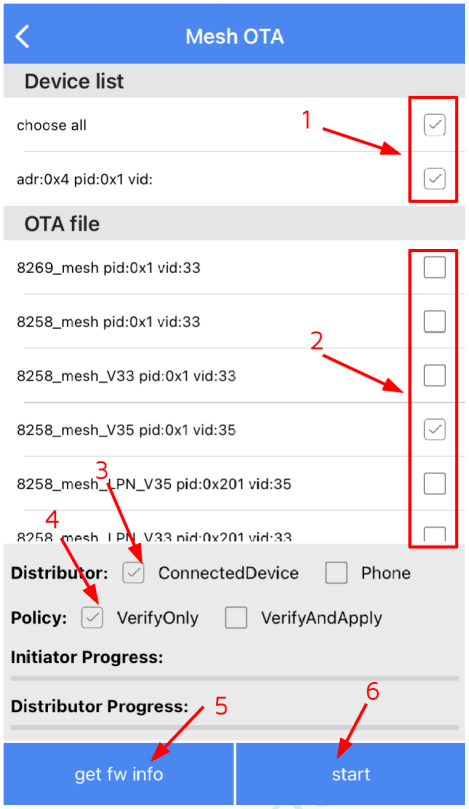


Figure 4.35: iOS verify only upgrade steps

(1) **Device list**: Select the device to be upgraded;;

(2) **OTA file**: Select the file to be upgraded;

(3) **Distributor**: Select ConnectedDevice;

(4) **Policy**: Select VerifyOnly;

(5) **get fw info**: Get the current device firmware ID;

(6) **start**: Start to upgrade.

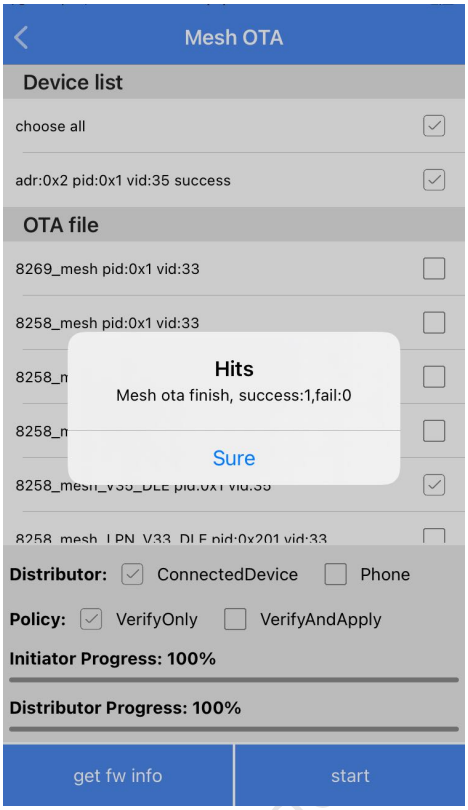


Figure 4.36: iOS verify only upgrade successful