

---

# RT-THREAD Cloud Platform User Manual

---

**RT-THREAD** Document Center

Copyright ©2019 Shanghai Ruisaide Electronic Technology Co., Ltd.



**WWW.RT-THREAD.ORG**

**Wednesday 17th October, 2018**

Table of contents

Table of contents	i
1 Introduction to the RT-Thread Cloud Platform...	1
1.1 Introduction to the Cloud Platform.	1
1.2 Cloud Functionality ...	1
1.2.1. Device Access.	1
1.2.2. Security Management.	1
1.2.3. Device Management.	1
1.3 Advantages of cloud platforms.	1
1.4 Glossary ...	2
2 RT-Thread Cloud Platform Usage...	2
2.1 Registration and Login ...	2
2.2 Creating a Product ...	2
2.3 Adding a device ...	4
3. Device access and use.	5
3.1 Device Access ...	5
3.2 Equipment Use ...	5
3.2.1. Web Shell Functionality ...	6
3.3 Web Log Function.	7
3.4 OTA upgrade function.	8
3.4.1. OTA Tool Introduction ...	8
3.4.2. OTA Function Introduction ...	9
4 Notes ...	12

## 1 Introduction to the RT-Thread Cloud Platform

### 1.1 Introduction to the Cloud Platform

**RT-Thread Cloud Platform** It is an IoT device maintenance cloud platform developed by Shanghai Ruisaide Electronic Technology Co., Ltd. Its purpose is to help developers build a secure and effective data transmission channel, facilitate two-way communication between device terminals and the cloud, and realize device upgrades, maintenance and management functions in the cloud.

The cloud platform is designed to manage access products and devices securely and efficiently, enabling remote control of devices and log storage management. And the firmware version management function helps developers quickly build a stable and reliable IoT device maintenance cloud platform.

### 1.2 Cloud Functions

The RT-Thread cloud platform provides users with a way to quickly build secure and reliable IoT applications. Its main features are as follows:

#### 1.2.1. Device Access

- Supports multiple protocols for accessing the cloud, such as MQTT, HTTP/HTTPS, etc., to meet users' needs for short real-time connections and long connections.  
The need for communication;
- Adaptable to various models of devices on different platforms to meet different device access requirements;

#### 1.2.2. Security Management

- Each product and each device provides a unique authentication and encryption mechanism to ensure the security of user device use; • The device supports TLS security protocol to connect to the cloud, and device data transmission is encrypted to ensure data security; • The device provides secure device key transmission and storage functions to prevent device information leakage;

#### 1.2.3. Device Management

- Provide a complete life cycle management of products and equipment, and realize the whole process of registration, modification, release and deletion; • **Web Shell** function: realize remote Shell control function in the cloud, users can control the equipment without connecting to the serial port device, Management, meeting users' needs for remote device management.
- **Web Log** function: Real-time display and storage of device logs in the cloud, collection of device data and status of device  
The viewing function allows users to view device dynamics and historical records at any time.
- **OTA** upgrade function: realize the remote upgrade function of the device, support encryption and compression upgrade, multiple firmware upgrade, breakpoint resume and other functions, meet Users demand OTA upgrades for multiple devices.

### 1.3 Advantages of Cloud Platform

- Easy to manage: The cloud focuses on the management and maintenance of equipment, achieving a management and maintenance method covering the entire life cycle of the equipment, and real-time monitoring of the equipment  
Line status;
- Safe and reliable: Provides dynamic device activation and authentication process to ensure the device is safe and unique, and double encryption of transmitted data to prevent tampering.  
Device keys are securely stored to prevent them from being obtained;

- Open and compatible: It can adapt to the access of various devices, and the connection protocol supports MQTT and HTTP/HTTPS. Supports QuickLZ, FastLZ

Compression methods such as .txt and .bin can be used to reduce the size of the upgraded firmware; support AES256 encryption to ensure secure firmware transmission. More connection protocols and encryption

Compression methods will be supported in due course.

## 1.4 Glossary

noun	explain
product	A general term for a type of equipment, users can classify it by product Different devices and unified management of their devices
ProductID	The cloud issues a unique identifier for each product, and users need To save device activations for this product
ProductKey	The cloud issues a key for each product, which users need to save Device activation for this product
equipment	Product instantiation, each device corresponds to a product, Each product corresponds to multiple devices
SN	Unique identifier of the device, used to distinguish different devices Equipment, need to be obtained in the device, for the device in the cloud New construction
DeviceID	The cloud issues a unique identifier for each device, and the device activates Obtained after activation, used for device connection to the cloud
DeviceKey	The cloud issues a key to each device, and the device gets Get, used for device to connect to the cloud

## 2 RT-Thread Cloud Platform Usage

### 2.1 Registration and Login

RT-Thread cloud platform address: [iot.rt-thread.com](http://iot.rt-thread.com)

Log in with your user account and password to enter the RT-Thread cloud platform. New users need to register a new account using their mobile phone and then log in.

### 2.2 Create a product

When entering the RT-Thread cloud platform for the first time, you need to create a new product first: click New Product, enter the relevant product information, and create a new product.

Product.

The screenshot shows the 'RT-Thread IOT管理系统' (RT-Thread IOT Management System) interface. On the left is a sidebar with navigation links: '首页' (Home), '新建产品' (New Product), '产品管理' (Product Management), and '设置' (Settings). The '新建产品' link is highlighted with a red box and a red arrow points to the '新建产品' (New Product) form on the right. The form contains the following fields:

- \* 产品名称 (Product Name): TEXT
- \* 产品类型 (Product Type): 智能家居 (Smart Home)
- \* 产品型号 (Product Model): ART\_WIFI
- \* 设备连接类型 (Device Connection Type): Wi-Fi设备 (Wi-Fi Device)
- \* 设备数 (Number of Devices): 100
- 产品描述 (Product Description): (Empty text area)

At the bottom of the form are two buttons: '立即创建' (Create Immediately) and '取消' (Cancel).

Figure 1: Create a new product

- Product Name: Name the product and keep it unique within the account;
- Product type: Define according to the type of your product;
- Product Model: The specific model of the equipment used in the product;
- Device connection type: how the product connects to the cloud;
- Number of devices: The maximum number of devices supported by the product. The number of devices cannot be changed after the product is released.

After the product is created successfully, you can click Product Management to view the information of the product you just created in the product information interface.

Click the Modify button in the upper right corner to modify the information of unpublished products, and click the Delete button to delete unpublished products.

The screenshot shows the 'RT-Thread IOT管理系统' (RT-Thread IOT Management System) interface. On the left is a sidebar with navigation links: '首页' (Home), '新建产品' (New Product), '产品管理' (Product Management), 'TEST', '产品信息' (Product Information), '设备信息' (Device Information), '模块管理' (Module Management), '固件升级' (Firmware Upgrade), and '设置' (Settings). The '产品信息' link is highlighted with a red box. The main content area is titled '产品列表/产品信息' (Product List/Product Information). It displays the following product details:

- 产品名称: TEST
- 产品类型: 智能家居
- 产品型号: ART\_WIFI
- 设备连接类型: Wi-Fi设备
- 已使用设备数: 1
- 设备总数: 100
- 创建时间: 2018-03-19 16:54
- 更新时间: 2018-03-19 16:54
- Productid: 4056fb5de8f84925
- 产品密钥: f682b07a-8de7-4a4f-abca-358c48b5d19a
- 产品描述: 无
- 是否发布: 未发布

At the bottom of the form is a '发布' (Publish) button. In the top right corner of the main content area, there are two buttons: '修改' (Modify) and '删除' (Delete), both highlighted with a red box.

Figure 2: Product Information

- ProductID: The product is an identifier that the user needs to store for later device connection;

- Product key: ProductKey, which users need to store for subsequent device connections;

Each device has a unique **ProductID** and ProductKey. Users need to record this information for **CloudSDK** library migration.

Connect devices to the cloud.

## 2.3 Adding Devices

After the product is created, you can add a device to the product. Enter the product management interface, click Device Information, and enter the unique Identifier SN (can be customized by the user, and must be consistent with the CloudSDK library porting file), device name and device MAC Address, click New Device to create a new device.

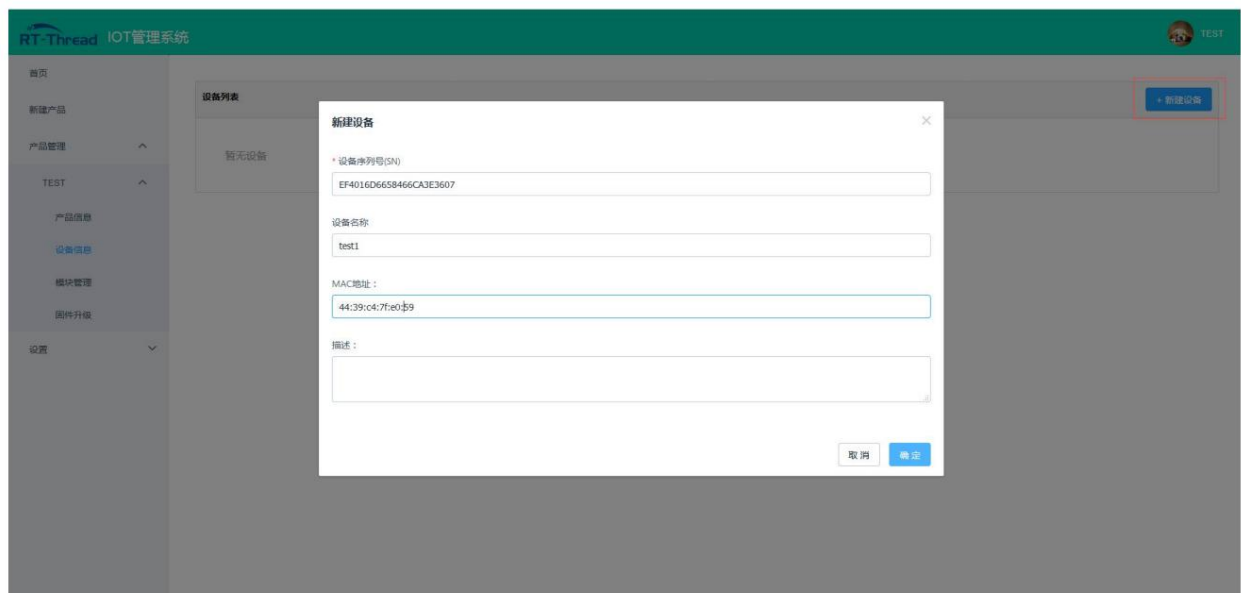


Figure 3: Add a device

The device is added successfully. You can view the current status of the device in Device Information->Details: Inactive, Offline.

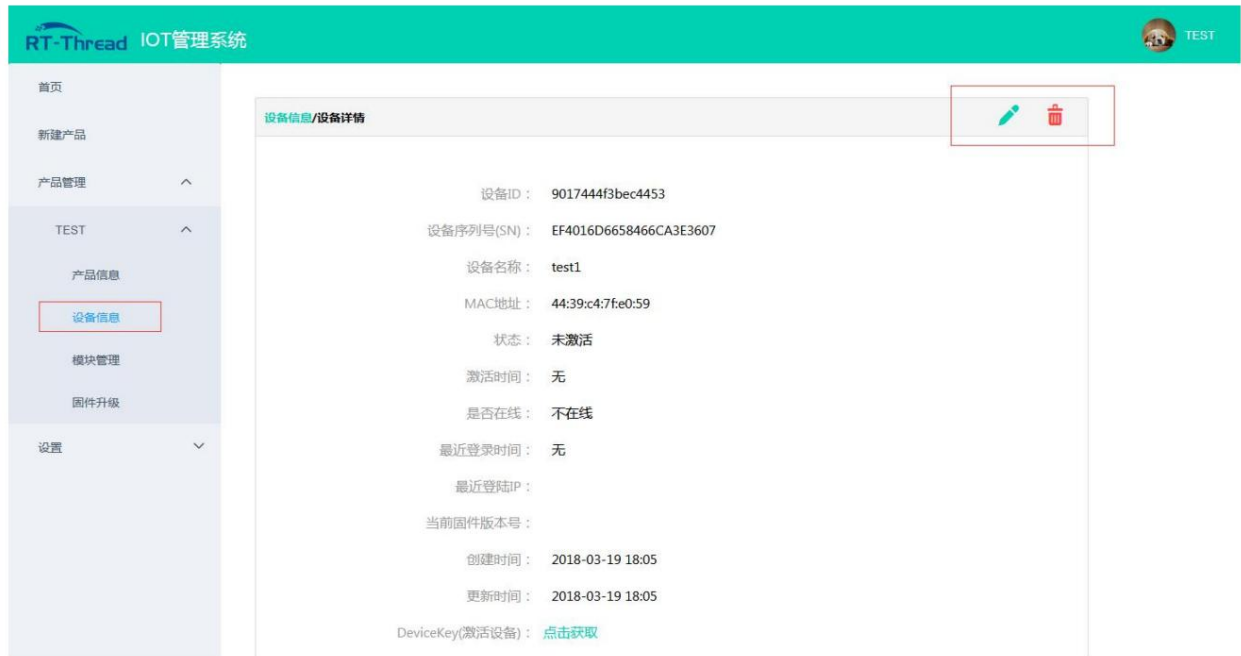


Figure 4: Device Information

After the device is created successfully, get the parameters such as ProductID and ProductKey. Then in the `rt_cld_port.c` file in the project Configure the connection parameters, download and compile the CloudSDK library file, connect the device to the RT-Thread cloud platform and perform data communication.

## 3 Device access and use

### 3.1 Device Access

Before using cloud functions, you need to prepare the operating environment and port and configure the CloudSDK library. The preparation work is as follows:

- RT-Thread system operating environment;
- Obtain the SN, ProductID and ProductKey parameters corresponding to the current device from the cloud;
- Use the parameters obtained from the cloud to complete the porting of the CloudSDK library;
- Burn the bootloader for OTA upgrade;

### 3.2 Equipment Use

After the device is transplanted, the first time it is powered on, there will be a dynamic activation process for the device. After successful activation, the device obtains and stores the DeviceID and DeviceKey for the device to connect to the cloud. Activation is successful, the device is online successfully, and the cloud displays the device information: Activated, Online, as shown below:



Figure 5: Device Status

After successful activation and online, the device can complete the following functions:

- Web Shell functionality
- Web Log function
- OTA upgrade function

#### Web Shell Functionality

The implementation of Web Shell is based on TCP/IP protocol and MQTT protocol. Its main function is to realize the remote Shell control function. Users do not need to connect to a serial device, you can manage and debug the device in the cloud, and display the device printing information in real time.

The device is online successfully. In the cloud, click Device Information -> Device Details -> Shell: Connect to implement the FinSH terminal function in the cloud:



Figure 6: Web Shell

Location



After clicking Connect, the device console will switch to the cloud display. Similar to the FinSH console, you can enter commands in the cloud to get

The corresponding response is as shown below:

```

【shell】 命令行控制台 --当前用户: $username
用法提示:
  1.输入"命令 参数1 参数2 ...."回车
  2.鼠标选中文本即可复制

开始连接服务器
连接服务器成功

\ | /
- RT -   Thread Operating System
/ | \   3.0.3 build Mar 22 2018
2006 - 2018 Copyright by rt-thread team

ash /?ps
thread pri status sp stack size max used left tick error
-----
web_shell 9 suspend 0x0000017c 0x00000800 49% 0x00000003 000
MQTT 10 suspend 0x000001a4 0x000001000 31% 0x00000001 000
web_iap 19 suspend 0x00000108 0x000001800 24% 0x0000000e 000
wifi_dog 29 suspend 0x000000a4 0x000000400 70% 0x00000008 000
WICED 10 suspend 0x000000cc 0x00000800 21% 0x00000008 000
tshell 20 ready 0x000002bc 0x000001000 29% 0x00000003 000
elog_async 31 suspend 0x000000a4 0x000000400 29% 0x00000007 000
ntp_sync 26 suspend 0x000000a0 0x00000800 10% 0x00000002 000
  
```

Annotations in the image: "输入命令" (Enter command) points to the command input area, and "获取结果" (Get result) points to the output table.

Figure 7: Web Shell

Function

### 3.3 Web Log Function

Web Log is similar to Web Shell, and its main function is to store and query Shell input and output logs.

The main difference between Web Shell and Shell is that Web Shell can display and manage the input and output of Shell in real time, while Web Log can log the output of Shell in real time.

Input and output records are stored for easy viewing later.

The device is online successfully. Click Device Information->Device Details->Enable Log Function in the cloud: Enable, and the cloud will enable the device Web Log record.

Function, the input and output logs of the device console will be sent to the cloud for record keeping. Click to turn on the log function again: Turn off to turn off the log function.

The Web Log function has a timeout mechanism. If there is no data transmission within 5 minutes after the Web Log function is turned on, the server will automatically shut down.

Web Log function.



Figure 8: Web Log

Location

After turning on the Web Log function, you can enter the `ps` command in the local FinSH to view the current thread status, and the displayed log will be sent to the cloud.

And store it in the log list. Later, click View device log: Log list in the cloud to view historical log information.

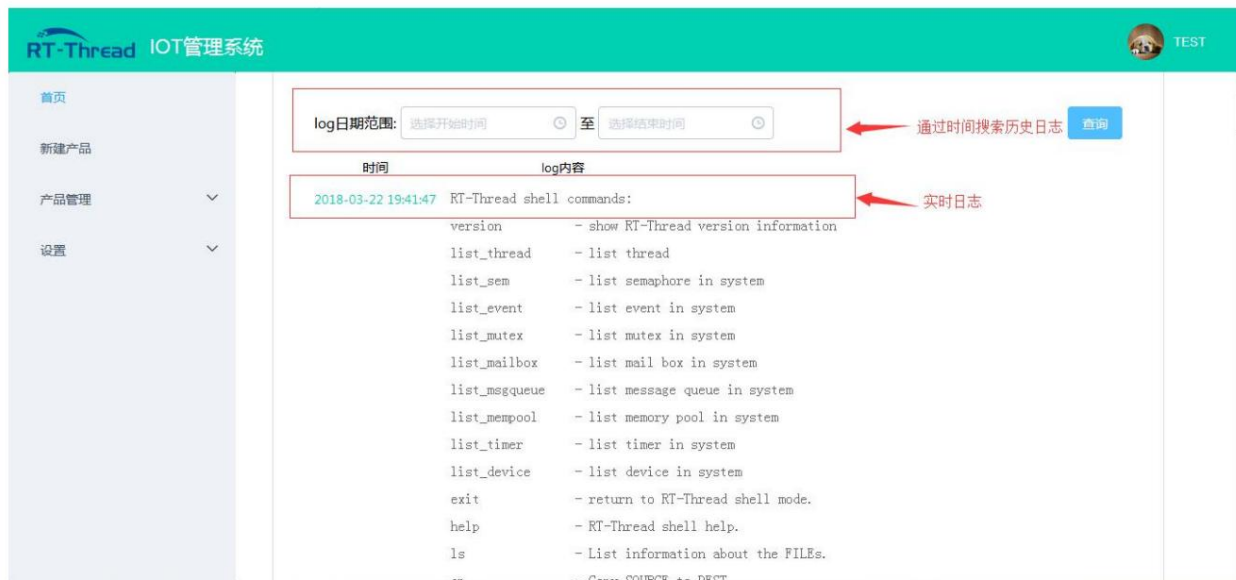


Figure 9: Historical log information

## 3.4 OTA upgrade function

RT-Thread cloud platform OTA upgrade function can realize remote device upgrade. Compared with other device upgrade methods, RT-Thread

The cloud platform has the following characteristics:

- Adaptable to different types of flash or file systems • Support encrypted data transmission • Support firmware encryption and compression
- Support breakpoint resume function • Support multiple firmware upgrade functions

### 3.4.1. OTA Tool Introduction

The firmware files required for cloud-based OTA upgrades require a specific firmware format. To address this, we provide the [RT-Thread OTA firmware packager](#), located at `/tools/ota_packager/rt_ota_packaging_tool.exe`. This firmware packaging tool encrypts and compresses the original firmware files, generating upgrade files in a specific format (.rbl extension) for later upload to the cloud and for creating new upgrade tasks in the cloud. The tool interface is shown below:

**RT-Thread** RT-Thread OTA 固件打包器

选择固件: C:\Users\Adminster\Desktop\test\rtthread.bin

保存路径: C:\Users\Adminster\Desktop\test\rtthread.rbl

压缩算法: 不压缩

加密算法: 不加密

加密密钥:

加密 IV:

固件名称: app 固件版本: 0.1

结果 :

HASH\_CODE : RAW\_SIZE :

HDR\_CRC32 : PKG\_SIZE :

BODY\_CRC32 : TIMESTAMP :

**开始打包**

COPYRIGHT (C) 2012-2018, Shanghai Real-Thread Technology Co., Ltd Ver: 1.0.0.0

Figure 10: OTA Packaging Tools

How to use the tool Users can choose whether to encrypt and compress the firmware according to their needs. The tool provides support for multiple compression and encryption algorithms. The specific steps are as follows: - Select the firmware to be packaged - Select the location to generate the firmware - Select the compression algorithm (support QuickLZ, FastLZ compression) - Select encryption algorithm (supports AES256 encryption) - Configure encryption key and IV (leave blank if not encrypted) - Fill in firmware name (firmware partition name) - Firmware version number (similar to xx) - Start packaging

### 3.4.2. OTA Function Introduction

Prepare the firmware:

This function introduces and demonstrates the multi-firmware upgrade and encryption compression upgrade functions. In the preparation stage, the latest firmware `rtthread.bin` and the latest firmware `filesystem.bin` of the device file system are compiled and generated. Then, the RT-Thread OTA firmware packager is used to generate `rtthread.rbl` and `filesystem.rbl`.

`rtthread.rbl`: Encryption only, no compression, using AES256 algorithm; `filesystem.rbl`: Compression only, no encryption, using QuickLZ algorithm;

Firmware upload:

The generated firmware needs to be uploaded to the cloud module management for firmware upgrade and firmware addition. As shown in the figure below, upload `rtthread.rbl`

And `filesystem.rbl` firmware to the cloud:

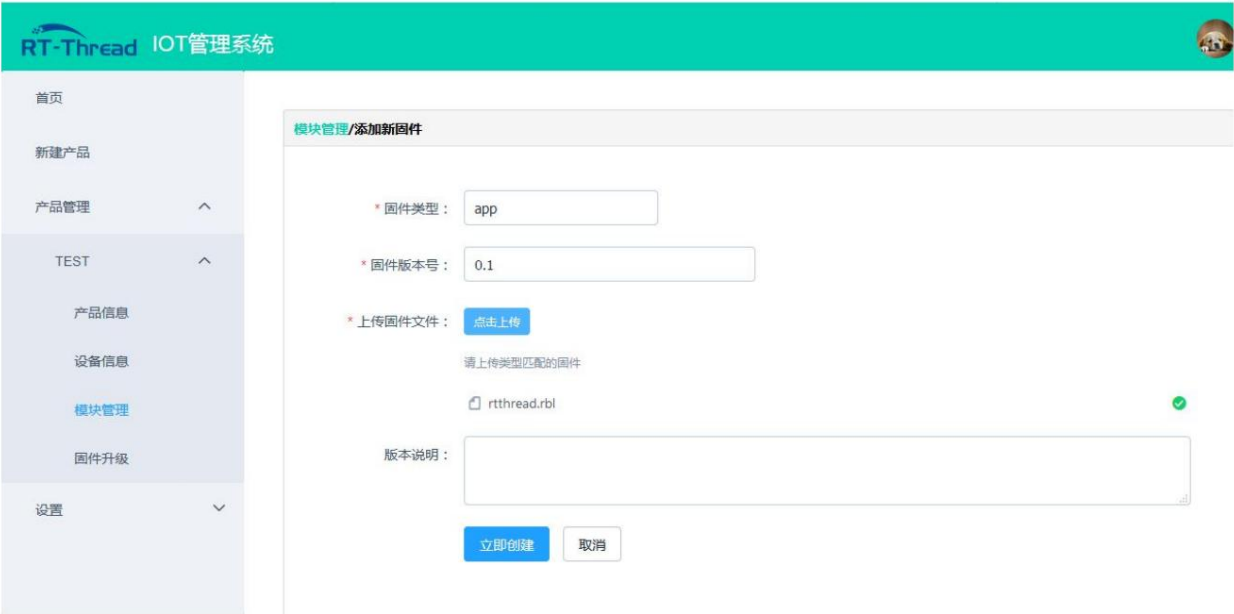


Figure 11: Firmware upload

- Firmware Type: The name of the firmware partition, which should be consistent with the firmware name entered when the tool generates the firmware;
- Firmware version number: This must be consistent with the firmware version entered when the tool generates the firmware, and must be higher than the latest version on the cloud;

Create a new OTA upgrade task:

After the firmware is uploaded successfully, you can create a new OTA upgrade task through the cloud (supports multiple firmware upgrades), and the cloud will push the upgrade request to the device.

The device is upgraded. As shown in the figure below, two firmwares are packaged to generate an OTA upgrade task:

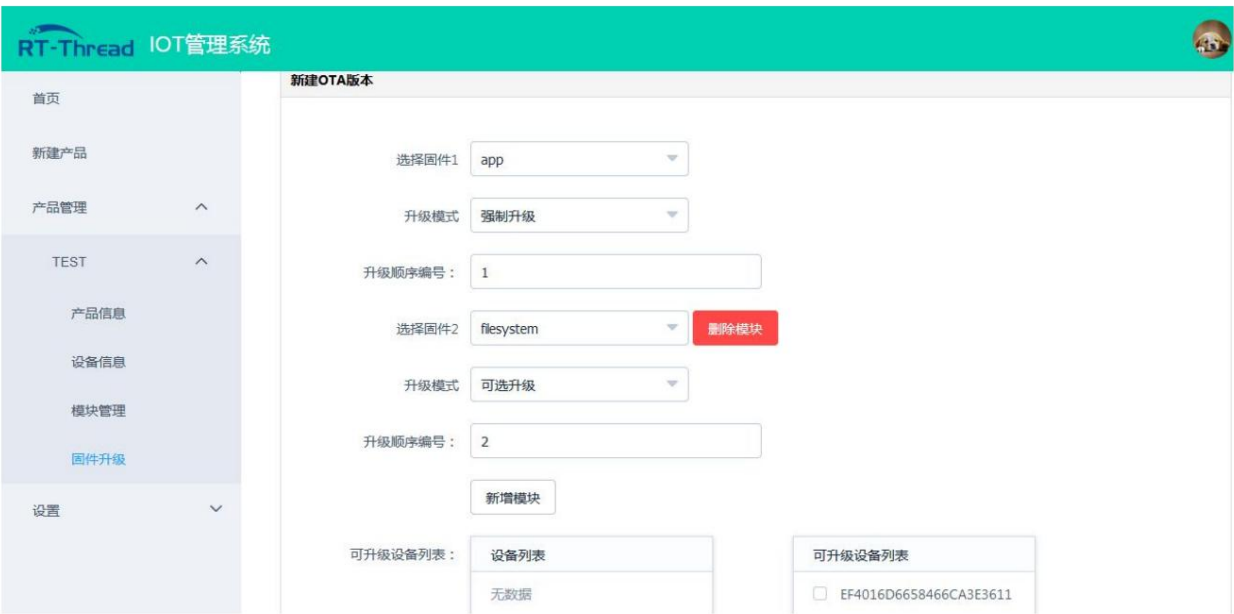


Figure 12: New ORDER Upgrade Tasks

- Upgrade Mode:

Forced upgrade: Send an upgrade task, and the device will immediately execute the download

upgrade. Optional upgrade: Send an upgrade task, and the user can customize the conditions for executing the download

upgrade. Silent upgrade: Send an upgrade task, and the device will execute the download upgrade without prompting.

- Upgrade order: When upgrading multiple firmware, users can customize the upgrade order of multiple firmware. The cloud sends the upgrade task, and the device will upgrade according to the firmware.

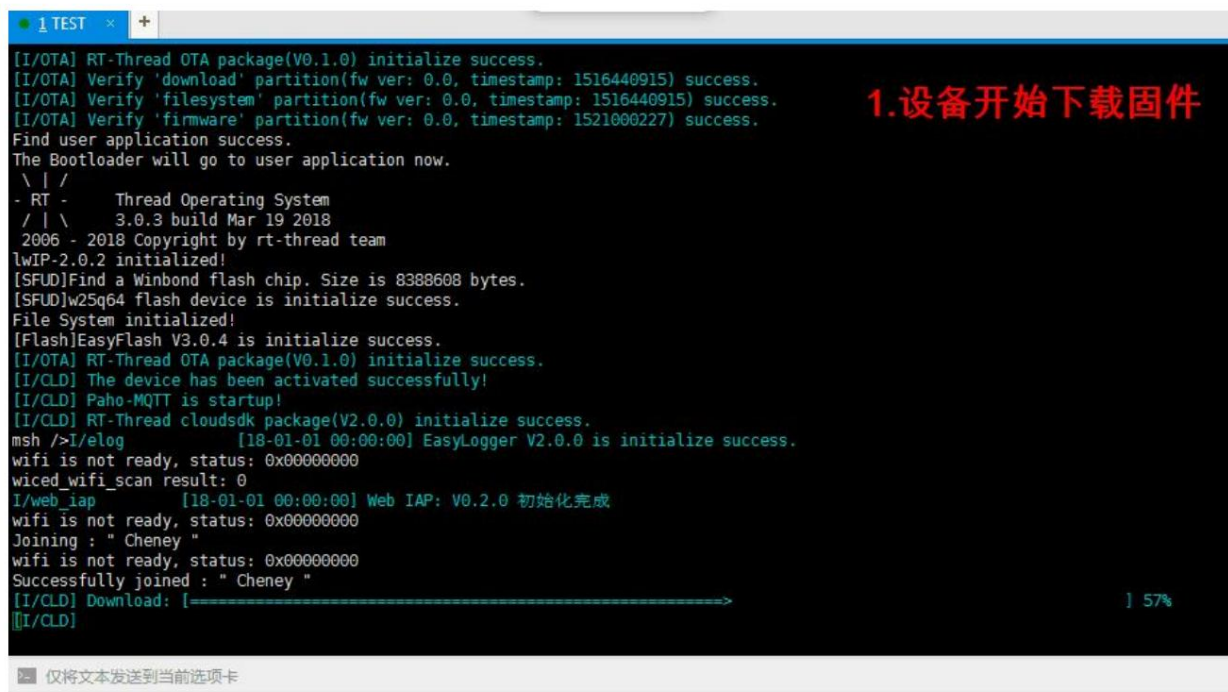
Upgrade the firmware in the order of the firmware upgrade

- Upgradeable list: Users can add devices that need to be upgraded to the device list, and the cloud will send the upgrade task to the device list.

On the device in;

Device OTA upgrade:

After the cloud upgrade task is successfully created, the cloud will send the firmware upgrade information via the MQTT protocol. After the device receives the upgrade information, it will download the new firmware. If multiple firmware are being upgraded, the device will restart once after each firmware upgrade until the last firmware is successfully upgraded. If there is a power outage or download failure during the upgrade process, the device supports breakpoint resume function to avoid repeated firmware downloads and reduce firmware upgrade time. The following figure shows the firmware download process:



```

1 TEST
[I/OTA] RT-Thread OTA package(V0.1.0) initialize success.
[I/OTA] Verify 'download' partition(fw ver: 0.0, timestamp: 1516440915) success.
[I/OTA] Verify 'filesystem' partition(fw ver: 0.0, timestamp: 1516440915) success.
[I/OTA] Verify 'firmware' partition(fw ver: 0.0, timestamp: 1521000227) success.
Find user application success.
The Bootloader will go to user application now.
\ | /
- RT - Thread Operating System
/ | \ 3.0.3 build Mar 19 2018
2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
[SFUD]Find a Winbond flash chip. Size is 8388608 bytes.
[SFUD]w25q64 flash device is initialize success.
File System initialized!
[Flash]EasyFlash V3.0.4 is initialize success.
[I/OTA] RT-Thread OTA package(V0.1.0) initialize success.
[I/CLD] The device has been activated successfully!
[I/CLD] Paho-MQTT is startup!
[I/CLD] RT-Thread cloudsdk package(V2.0.0) initialize success.
msh />I/eLog [18-01-01 00:00:00] EasyLogger V2.0.0 is initialize success.
wifi is not ready, status: 0x00000000
wiced_wifi_scan result: 0
I/web_iap [18-01-01 00:00:00] Web IAP: V0.2.0 初始化完成
wifi is not ready, status: 0x00000000
Joining : " Cheney "
wifi is not ready, status: 0x00000000
Successfully joined : " Cheney "
[I/CLD] Download: [=====] 57%
[I/CLD]

```

Figure 13: Device firmware download

After the device firmware is downloaded, the program automatically jumps to the Bootloader to run. The Bootloader will decompress and decrypt the firmware and copy it to the designated application partition (here, the **filesystem** partition). After the Bootloader successfully decompresses, decrypts, and upgrades the firmware, the device status is as shown below:

```

1 TEST
[SFUD]Find a Winbond flash chip. Size is 8388608 bytes.
[SFUD]w25q64 flash device is initialize success.
File System initialized!
[Flash]EasyFlash V3.0.4 is initialize success.
[I/OTA] RT-Thread OTA package(V0.1.0) initialize success.
[I/CLD] The device has been activated successfully!
[I/CLD] Paho-MQTT is startup!
[I/CLD] RT-Thread cloudsdk package(V2.0.0) initialize success.
msh />I/eolog [18-01-01 00:00:00] EasyLogger V2.0.0 is initialize success.
wifi is not ready, status: 0x00000000
wiced_wifi_scan result: 0
I/web_iap [18-01-01 00:00:00] Web IAP: V0.2.0 初始化完成
wifi is not ready, status: 0x00000000
Joining : " Cheney "
wifi is not ready, status: 0x00000000
Successfully joined : " Cheney "
[I/CLD] Download: [=====] 100%
[I/OTA] Verify 'download' partition(fw ver: 0.1, timestamp: 1516440915) success.
[Flash]Erased ENV OK.
[Flash]Saved ENV OK. [SFUD]Find a Winbond W25Q64FV flash chip. Size is 8388608 bytes.
[SFUD](../components/sfud/src/sfud.c:724) Flash device reset success.
[SFUD]norflash0 flash device is initialize success.

RT-Thread Bootloader Starting...
[I/OTA] RT-Thread OTA package(V0.1.0) initialize success.
[I/OTA] Verify 'download' partition(fw ver: 0.1, timestamp: 1516440915) success.
[I/OTA] OTA firmware(filesystem) upgrade(0.0->0.1) startup.
[I/OTA] OTA Write: [=====>] 62%
[[I/OTA]

```

Figure 14: Bootloader

Update the firmware

The firmware upgrade in the bootloader is successful, the device automatically runs the new firmware, and the OTA upgrade process is completed;

## 4. Notes

- When creating a new device in the cloud, the SN device unique identifier entered must be consistent with the SN set in the CloudSDK library porting file ;
- The ProductID and ProductKey configured in the rt\_cld\_port.c file need to be the same as those obtained when creating a new product corresponding to the device on the cloud.  
ProductID and ProductKey are consistent;
- The first time the device is powered on and the CloudSDK component is started, an activation process is required. Ensure that the device is connected to the Internet for successful activation.  
If the migration fails, a 400 error is returned. Check whether the SN obtained in the user-defined migration interface is consistent with the SN added when creating a new device on the cloud .