
RT-THREAD 云平台用户手册

RT-THREAD 文档中心

上海睿赛德电子科技有限公司版权 ©2019



WWW.RT-THREAD.ORG

Wednesday 17th October, 2018

目录

目录	i
1 RT-Thread 云平台介绍	1
1.1 云平台简介	1
1.2 云端功能	1
1.2.1. 设备接入	1
1.2.2. 安全管理	1
1.2.3. 设备管理	1
1.3 云平台优势	1
1.4 名词解释	2
2 RT-Thread 云平台使用	2
2.1 注册登录	2
2.2 创建产品	2
2.3 添加设备	4
3 设备接入及使用	5
3.1 设备接入	5
3.2 设备使用	5
3.2.1. Web Shell 功能	6
3.3 Web Log 功能	7
3.4 OTA 升级功能	8
3.4.1. OTA 工具介绍	8
3.4.2. OTA 功能介绍	9
4 注意事项	12

1 RT-Thread 云平台介绍

1.1 云平台简介

RT-Thread 云平台 是由上海睿赛德电子科技有限公司开发的一套物联网设备维护云平台。其目的是帮助开发者搭建安全有效的数据传输通道，方便设备终端和云端的双向通讯，在云端实现设备的升级、维护与管理功能。

该云平台旨在对接入产品和设备进行安全且高效的管理，可以实现对设备的远程操控、日志存储管理以及对固件的版本管理功能，帮助开发者快速搭建稳定可靠的物联网设备维护云平台。

1.2 云端功能

RT-Thread 云平台为用户提供快速搭建安全可靠的物联网开发应用的方法。其主要功能特点如下：

1.2.1. 设备接入

- 支持多种协议的接入云端，例如：MQTT、HTTP/HTTPS 等协议，满足用户对短实时连接以及长连接通讯的需求；
- 可适配不同平台多种型号的设备，满足不同设备接入需求；

1.2.2. 安全管理

- 每个产品、每个设备提供唯一的认证和加密机制，确保用户设备使用安全性；
- 设备支持 TLS 安全协议连接云端，设备数据传输为加密传输，确保数据安全性；
- 设备提供安全的设备密钥传输和存储功能，防止设备信息泄露；

1.2.3. 设备管理

- 提供产品和设备完整的生命管理周期，实现注册、修改、发布、删除整个过程；
- **Web Shell 功能：**在云端实现远程 Shell 控制功能，用户无需连接串口设备即可完成对设备的控制、管理，满足用户对设备远程管理的需求。
- **Web Log 功能：**在云端实现设备日志的实时显示和存储功能，实现设备数据的采集以及设备状态的查看功能，方便用户随时查看设备动态及历史记录。
- **OTA 升级功能：**实现设备远程升级功能，支持加密压缩升级、多固件升级、断点续传等功能，满足用户对多种设备的 OTA 升级需求。

1.3 云平台优势

- **易于管理：**云端侧重设备的管理维护，实现覆盖设备全生命周期的管理与维护方式，实时监控设备在线状态；
- **安全可靠：**提供设备动态激活认证过程，确保设备安全且唯一，传输数据双重加密方式防止被篡改，设备密钥安全存储防止被获取；

- 开放兼容：可适配多种设备的接入，连接协议支持 MQTT 和 HTTP/HTTPS。支持 QuickLZ、FastLZ 等压缩方式，减小升级固件体积；支持 AES256 加密方式，确保安全传输固件。更多连接协议和加密压缩方式将陆续支持。

1.4 名词解释

名词	解释
产品	一类设备的统称，用户可以通过产品划分不同的设备，并且对其设备进行统一管理
ProductID	云端颁布每个产品的唯一标识符，用户需要保存用于该产品的设备激活
ProductKey	云端颁布每个产品的密钥，用户需要保存用于该产品的设备激活
设备	产品的实例化，每个设备对应一个产品，每个产品对应多个设备
SN	设备的唯一标识符，用于区分不同的设备，需要在设备中获取，用于云端该设备的新建
DeviceID	云端颁布每个设备的唯一标识符，设备激活后获取，用于设备连接云端
DeviceKey	云端颁布每个设备的密钥，设备激活后获取，用于设备连接云端

2 RT-Thread 云平台使用

2.1 注册登录

RT-Thread 云平台地址：iot.rt-thread.com

用户账号密码[登录](#)，进入 RT_Thread 云平台，新用户需使用手机[注册](#)新账号，然后再登录账号。

2.2 创建产品

初次进入 RT-Thread 云平台，需要先创建新产品：点击[新建产品](#)，输入相关产品信息，创建一个新产品。

RT-Thread IOT管理系统

新建产品

* 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

* 设备数: 100

产品描述

立即创建 取消

图 1: 创建新产品

- 产品名称: 对产品命名, 在账号内保持唯一;
- 产品类型: 根据自己产品的类型定义;
- 产品型号: 产品使用设备的具体型号;
- 设备连接类型: 产品连接云端的方式;
- 设备数: 该产品最大支持的设备数量, 产品发布之后设备数不可变。

产品创建成功后, 可以点击产品管理, 在产品信息界面中查看刚才创建产品的信息。并且可以通过界面右上角修改按键, 修改未发布的产品的信息, 通过删除按键, 删除未发布的产品。

RT-Thread IOT管理系统

产品列表/产品信息

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

发布

图 2: 产品信息

- ProductID : 产品为一标识符, 用户需要存储用于后面设备连接;

- 产品密钥：即 ProductKey，用户需要存储用于后面设备连接；

每个设备对应唯一的 **ProductID** 和 **ProductKey**，用户需要记录信息用于 CloudSDK 库移植，实现设备连接云端。

2.3 添加设备

产品创建完成之后，可以为该产品添加设备。进入产品管理界面，点击设备信息，输入设备获取的唯一标识符 **SN**（可以由用户自定义设置，需要确保和 ClouSDK 库移植文件中一致）、设备名称和设备的 MAC 地址，点击新建设备创建新设备。

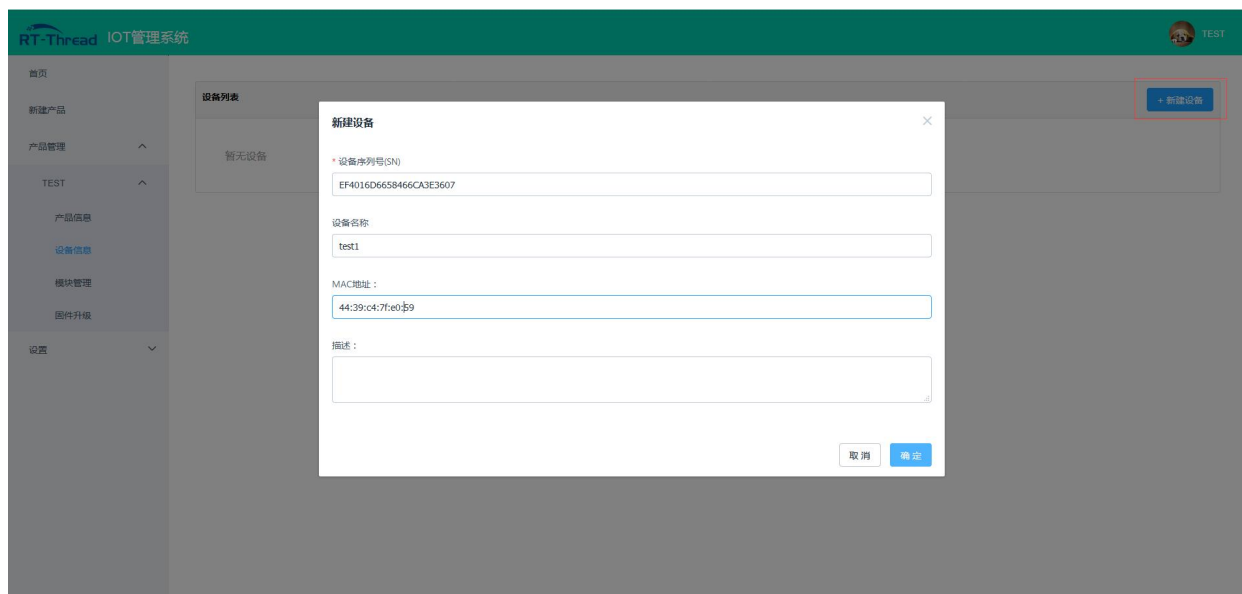


图 3：添加设备

设备添加成功，可以在设备信息->详情中查看设备当前状态：未激活、不在线状态。

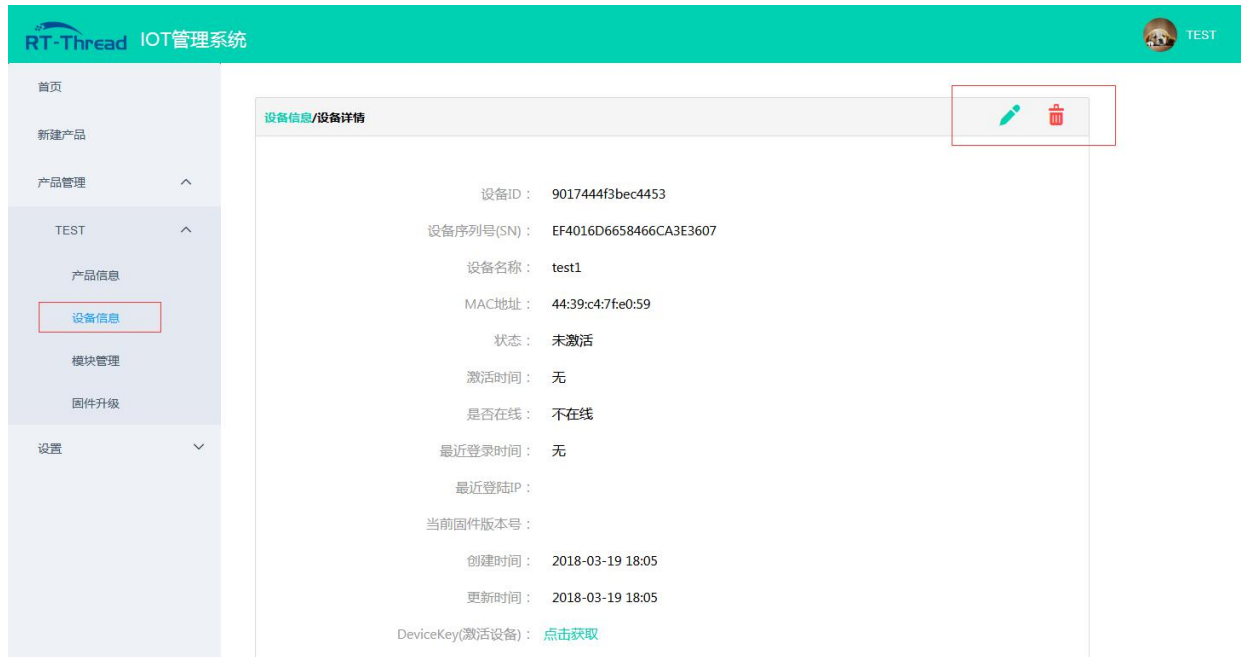


图 4: 设备信息

设备创建成功之后，获得 ProductID、ProductKey 这些参数，接下来在工程中 `rt_cld_port.c` 文件中配置连接参数，下载编译 CloudSDK 库文件，将设备连接上 RT-Thread 云平台并进行数据通信。

3 设备接入及使用

3.1 设备接入

云端功能使用之前，需要有运行环境准备和对 CloudSDK 库移植配置，准备工作如下：

- RT_Thread 系统运行环境；
- 云端获取当前设备对应 SN、ProductID 和 ProductKey 参数；
- 使用云端获取参数完成 CloudSDK 库的移植；
- 烧录 bootloader 用于 OTA 升级；

3.2 设备使用

设备移植完成，第一次上电会有设备动态激活的过程，激活成功之后设备获取并存储 DeviceID 和 DeviceKey，用于设备连接云端。激活成功，设备上线成功，云端显示设备信息：已激活、在线状态，如下图所示：



图 5: 设备状态

激活成功，上线成功之后，设备可完成如下功能：

- Web Shell 功能
- Web Log 功能
- OTA 升级功能

3.2.1. Web Shell 功能

Web Shell 的实现基于 TCP/IP 协议和 MQTT 协议，主要作用是实现远程 Shell 控制功能，用户无需连接串口设备即可在云端完成设备的管理和调试，并且实时显示设备打印信息。

设备上线成功，云端点击设备信息->设备详情->shell：连接，在云端实现 FinSH 终端功能：



图 6: Web Shell 位置

点击连接之后，设备端控制台将切换到云端显示。类似于 FinSH 控制台，此时在云端输入命令可以得到相应响应，如下图所示：

```

【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 0x00001000 31% 0x00000001 000
web_iap 19 suspend 0x00000108 0x00001800 24% 0x0000000e 000
wifi_dog 29 suspend 0x000000a4 0x00000400 70% 0x00000008 000
WICED 10 suspend 0x000000cc 0x00000800 21% 0x00000008 000
tshell 20 ready 0x000002bc 0x00001000 29% 0x00000003 000
elog_async 31 suspend 0x000000a4 0x00000400 29% 0x00000007 000
ntp_sync 26 suspend 0x000000a0 0x00000800 10% 0x00000002 000
    
```

图 7: Web Shell 功能

3.3 Web Log 功能

Web Log 与 Web Shell 类似，主要作用是实现对 Shell 输入输出日志的存储和查询功能。它与 Web Shell 主要的区别是 Web Shell 功能是对 Shell 输入输出的实时显示与管理控制，Web Log 功能是对 Shell 输入输出的记录存储，方便后期查看。

设备上线成功，云端点击设备信息->设备详情->开启日志功能：开启，云端开启设备 Web Log 日志记录功能，设备控制台的输入输出日志会发送到云端记录保存，再次点击开启日志功能：关闭可关闭日志功能。

Web Log 功能自带超时处理机制，开启 Web Log 功能后 5 分钟内无数据传输，服务器会主动关闭 Web Log 功能。



图 8: Web Log 位置

开启 Web Log 功能后，可在本地 FinSH 中输入 `ps` 命令查看当前线程状态，显示的日志会发送到云端并存储在日志列表中，之后在云端点击查看设备日志：日志列表，可以查看历史日志信息。

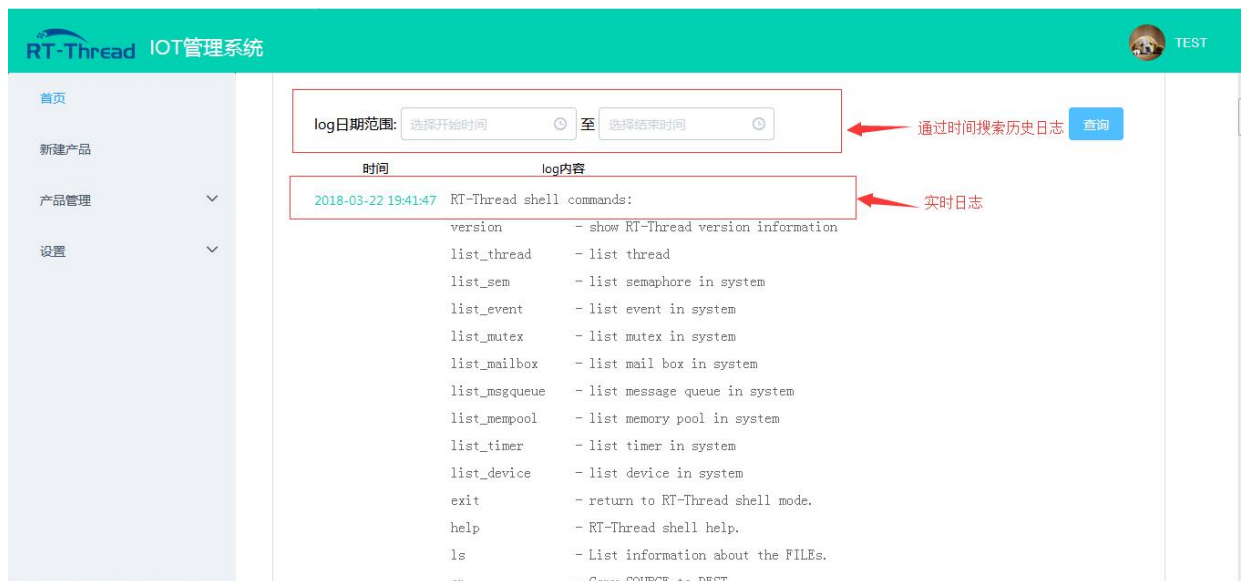


图 9：历史日志信息

3.4 OTA 升级功能

RT-Thread 云平台 OTA 升级功能可以实现设备远程升级。相比于其他的设备升级方式，RT-Thread 云平台具有以下特点：

- 可适配不同型号的 flash 或文件系统
- 支持加密数据传输
- 支持固件加密和压缩功能
- 支持断点续传功能
- 支持多固件升级功能

3.4.1. OTA 工具介绍

云端 OTA 升级时所需的固件文件需要特定的格式固件支持，为此我们提供 **RT-Thread OTA 固件打包器**，位于 `/tools/ota_packager/rt_ota_packaging_tool.exe`。固件打包工具可以将原格式固件文件做加密、压缩处理，生成特定格式（.rb1 后缀）的升级固件文件，用于后期上传至云端以及在云端新建升级任务。工具的界面如下图所示：



图 10: OTA 打包工具

工具使用方式用户可以根据需求，选择是否对固件进行加密和压缩，工具提供多种压缩和加密算法支持。具体操作步骤如下：- 选择待打包的固件 - 选择生成固件的位置 - 选择压缩算法（支持 QuickLZ、FastLZ 压缩）- 选择加密算法（支持 AES256 加密）- 配置加密密钥和 IV（不加密则留空）- 填写固件名称（固件分区名称）- 固件版本号（类似 x.x）- 开始打包

3.4.2. OTA 功能介绍

准备固件：

本功能介绍演示多固件升级和加密压缩升级功能，准备阶段编译生成设备最新固件 `rtthread.bin` 及设备文件系统最新固件 `filesystem.bin`，再利用 RT-Thread OTA 固件打包器生成 `rtthread.rbl` 和 `filesystem.rbl`。

`rtthread.rbl`：只加密不压缩处理，加密算法为 AES256；`filesystem.rbl`：只压缩不加密处理，压缩算法为 QuickLZ；

固件上传：

生成的固件需要上传到云端模块管理，用于固件升级添加固件。如下图所示方式，上传 `rtthread.rbl` 和 `filesystem.rbl` 固件到云端：

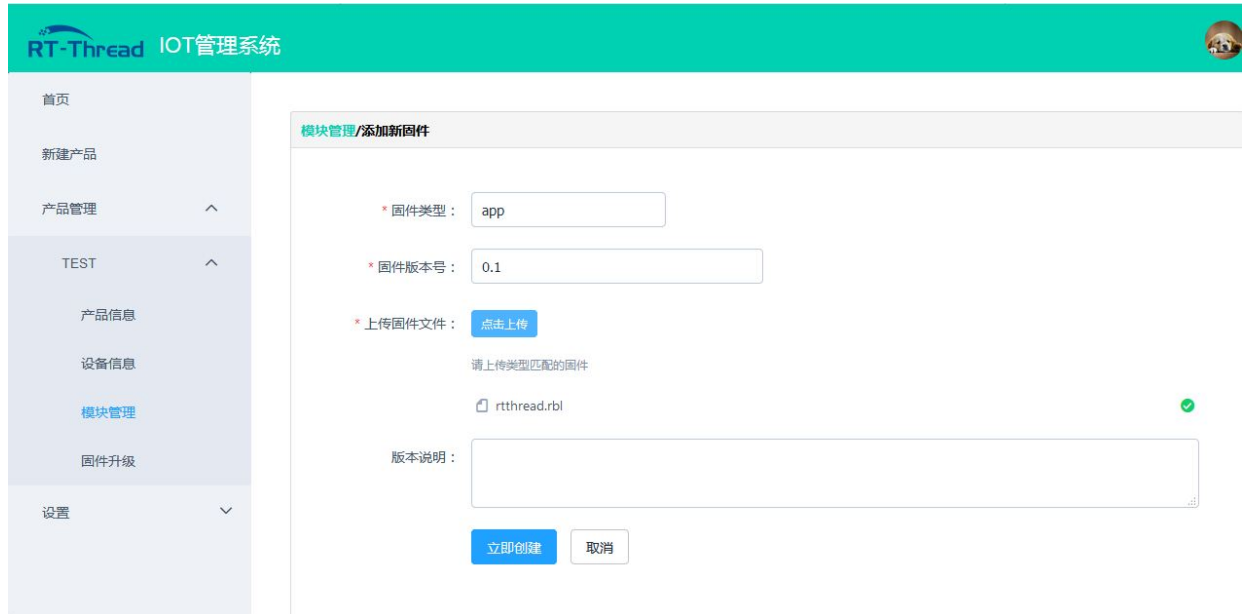


图 11：固件上传

- 固件类型：固件分区名称，与工具生成固件时填入固件名称一致；
- 固件版本号：需要与工具生成固件时填入固件版本一致，且高于云端最新版本号；

新建 OTA 升级任务：

固件上传成功后，可以通过云端新建 OTA 升级任务（支持多固件升级），云端会向设备推送升级请求，设备进行升级。如下图所示，是将两个固件打包生成一个 OTA 升级任务：

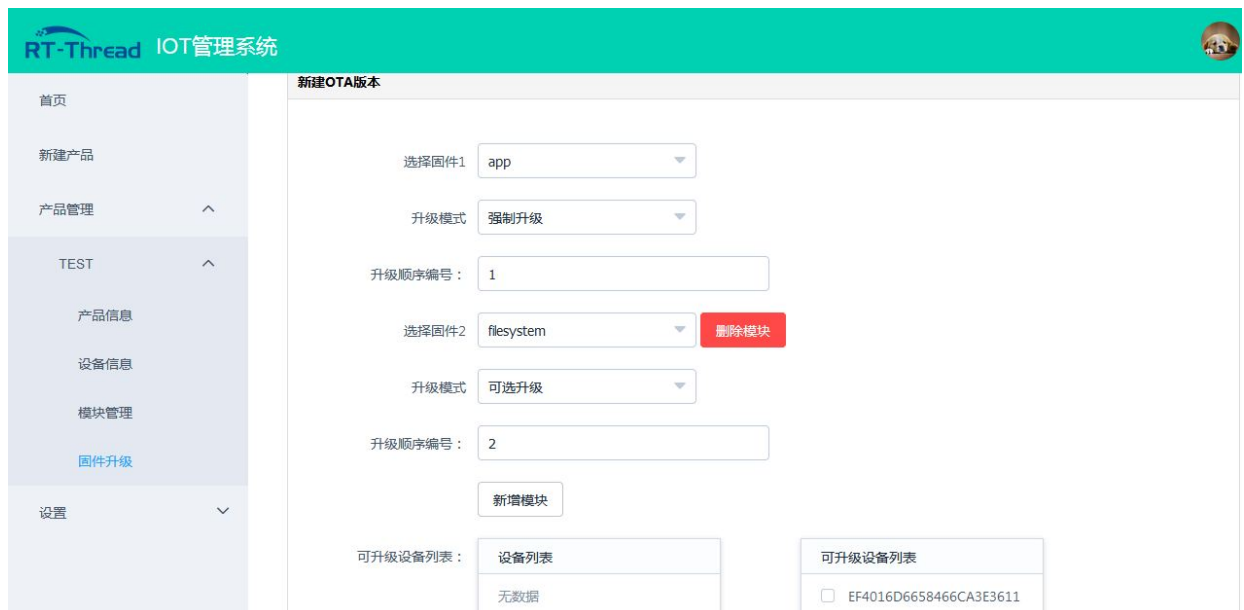


图 12：新建 OTA 升级任务

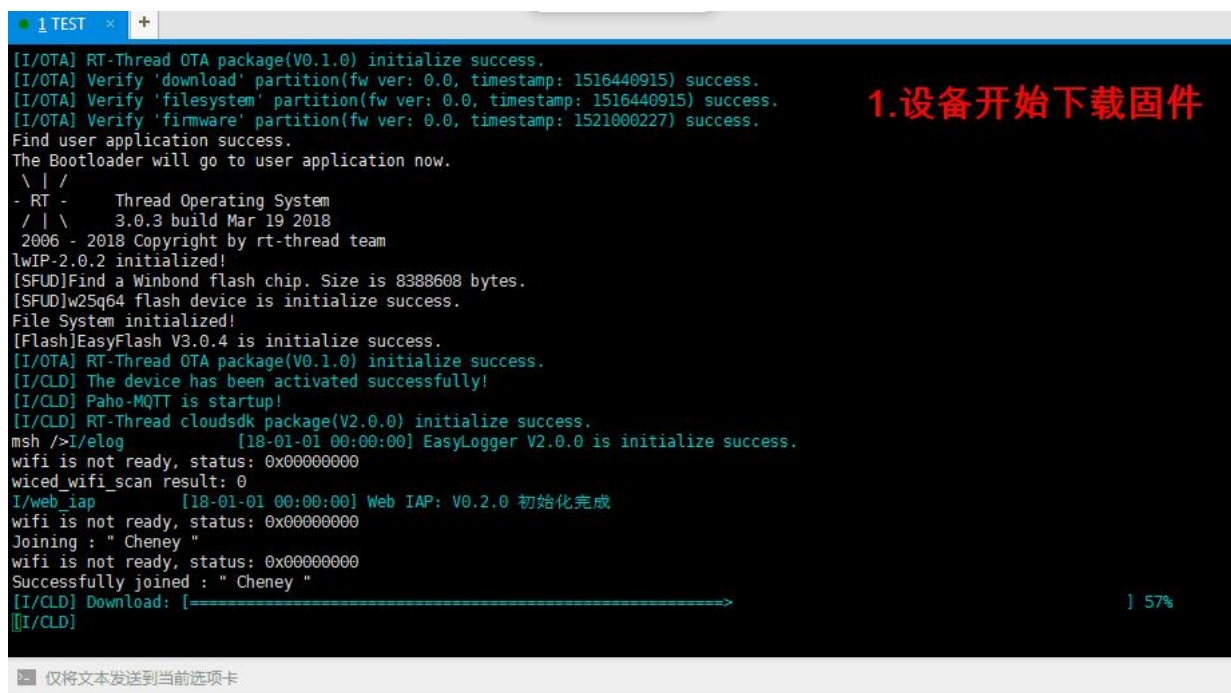
- 升级模式:

强制升级: 下发升级任务, 设备立刻执行下载升级
可选升级: 下发升级任务, 用户可自定义执行下载升级的条件
静默升级: 下发升级任务, 设备无提示执行下载升级

- 升级顺序: 多个固件升级时, 用户可自定义多个固件的升级顺序, 云端下发升级任务, 设备会根据固件升级顺序依次来升级固件
- 可升级列表: 用户可自定义添加需要升级的设备到设备列表中, 云端将下发本次升级任务到设备列表中的设备上;

设备 OTA 升级:

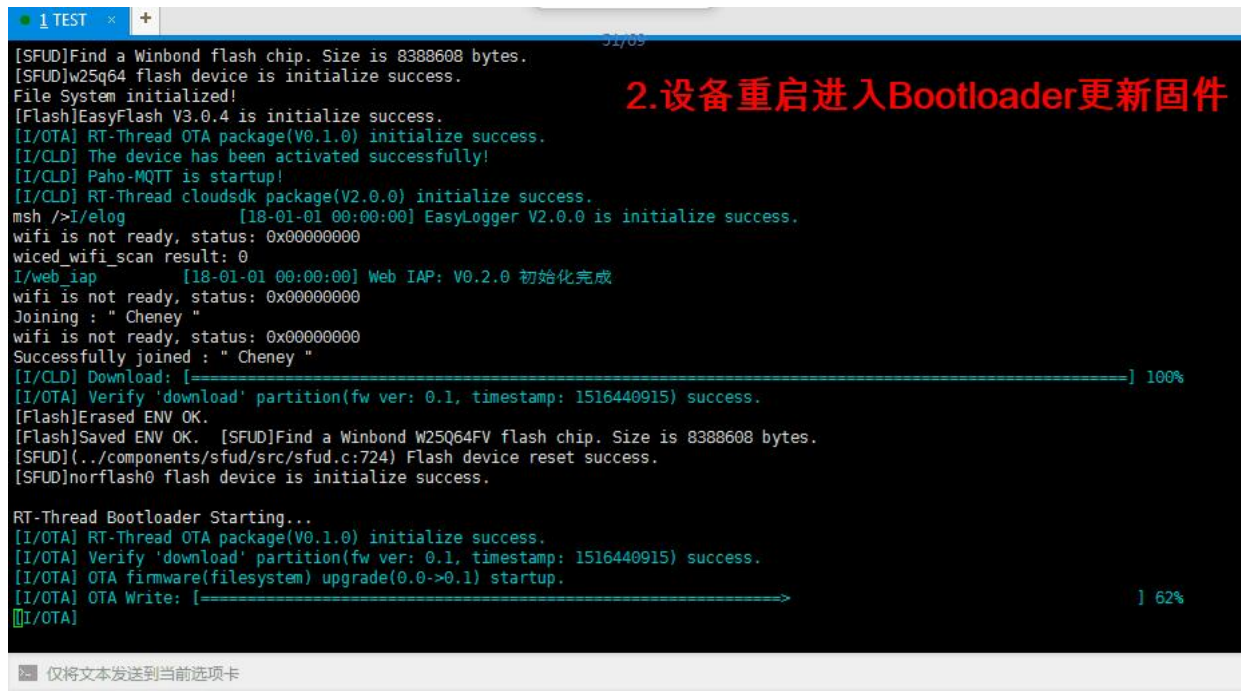
云端升级任务创建成功之后, 云端会通过 MQTT 协议下发固件升级信息, 设备获取升级信息后会下载新的固件。若为多固件升级, 设备每次升级完一个固件会重启一次, 直到最后一个固件升级成功。如果升级过程中若出现断电或者下载失败, 设备支持断点续传功能, 避免固件重复下载, 减少固件升级时间。下图为固件下载过程:



```
[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/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: [=====] 57%
[I/CLD]
```

图 13: 设备下载固件

设备固件下载完成之后, 程序会自动跳转到 Bootloader 中运行, Bootloader 会对固件进行解压解密, 并将固件拷贝到指定的应用分区 (这里是 **filesystem** 分区)。Bootloader 中解压解密和升级成功后设备状态如下图所示:



```
[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]
```

图 14: Bootloader 更新固件

Bootloader 中固件升级更新成功，设备自动运行新的固件，OTA 升级过程完成；

4 注意事项

- 云端新建设备时，输入的 SN 设备唯一标识符，需要和 CloudSDK 库移植文件中设置的 SN 一致；
- rt_cld_port.c 文件中配置的 ProductID 和 ProductKey 需要与云端新建对应设备的产品时获取的 ProductID 和 ProductKey 一致；
- 设备第一次上电启动 CloudSDK 组件需要有激活的过程，应确保设备联网才能激活成功。若出现激活失败返回400错误，检查用户自定义移植接口中获取的 SN 和云端新建设备时添加的 SN 是否一致；