EC2x&EG9x&EG25-G Series QuecOpen User Application and Configuration Parameter Backup and Restoration Reference Guide

LTE Standard Module Series

Version: 1.0

Date: 2021-01-07

Status: Released

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to support@quectel.com.

General Notes

Quectel offers the information as a service to its customers. The information provided is based upon customers' requirements. Quectel makes every effort to ensure the quality of the information it makes available. Quectel does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information. All information supplied herein is subject to change without prior notice.

Disclaimer

While Quectel has made efforts to ensure that the functions and features under development are free from errors, it is possible that these functions and features could contain errors, inaccuracies and omissions. Unless otherwise provided by valid agreement, Quectel makes no warranties of any kind, implied or express, with respect to the use of features and functions under development. To the maximum extent permitted by law, Quectel excludes all liability for any loss or damage suffered in connection with the use of the functions and features under development, regardless of whether such loss or damage may have been foreseeable.

Duty of Confidentiality

The Receiving Party shall keep confidential all documentation and information provided by Quectel, except when the specific permission has been granted by Quectel. The Receiving Party shall not access or use Quectel's documentation and information for any purpose except as expressly provided herein. Furthermore, the Receiving Party shall not disclose any of the Quectel's documentation and information to any third party without the prior written consent by Quectel. For any noncompliance to the above requirements, unauthorized use, or other illegal or malicious use of the documentation and information, Quectel will reserve the right to take legal action.

Copyright

About the Document

Revision History

Version	Date	Author	Description
-	2020-06-13	Tinker SUN	Creation of the document
1.0	2021-01-07	Tinker SUN	First official release

Contents

Table Index

Figure Index

1 Introduction

Quectel LTE Standard EC2x series, EG9x series and EG25-G modules support QuecOpen® solution. QuecOpen is an open-source embedded development platform based on Linux system. It is intended to simplify the design and development of IoT applications. For more information on QuecOpen®, see ***document [1]***.

This document mainly introduces the reference guide for backing up and restoring user applications and configuration parameters in QuecOpen® solution. The Linux operating system partitions are used to store, read and write these applications and configuration parameters.

1.1. Applicable Modules

Table 1: Applicable Modules

Module Series	Module
EC2x series	EC25 series
	EC21 series
	EC20 R2.1
EG9x series	EG95 series
	EG91 series
EG25-G	EG25-G

2 Related Partitions

The following table describes the relevant partitions in the Linux operating system used to store, read and write user applications and configuration parameter data in QuecOpen® solution. Partitions that can be used for user application storage, backup and restoration are *system* partition and *usr_data* partition.

Table 2: Overview of Partitions That Can be Used for User Application Storage, Backup and Restoration

User Partition	Partition Format	Mount Point	Use
system	UBI file system	/ (Root directory)	● *rootfs* volume. ● Store, read and write user applications and configuration parameters.
usr_data	UBI file system	*/usrdata*	● Store, read and write user applications and configuration parameters.

3 Backup and Restoration Reference Guide

3.1. Reference Guide

User applications and configuration parameters are stored in *system* partition and *usr_data* partition respectively. When the module is turned on, after running the *find_partitions.sh* script in the */etc/init.d* path, the module runs the detection script *app_auto_backup_restore.sh* for automatic backup and restoration (as shown in ***Figure 1*** below, which has been attached to the sidebar of the document, and you can choose the storage path) to detect whether user applications exist in the two partitions:

- If user applications do not exist in one of the partitions, copy the user applications and configuration parameters from the other partition to itself.
- If user applications exist in both partitions, check whether the application versions in the two partitions are consistent.
  - If the version numbers are inconsistent, copy the user applications and configuration parameters with the higher version number from the other partition to the partition and overwrite the original ones.

The specific process is shown in ***Figure 2*** below.

Since the *system* partition and *usr_data* partition will not be damaged at the same time, even if the user applications have been upgraded, the backup and restoration mechanism can ensure that the user applications and configuration parameters are not lost; if the file system is upgraded, it can also ensure that the user applications are of the latest version.

Figure 1: Detection Script

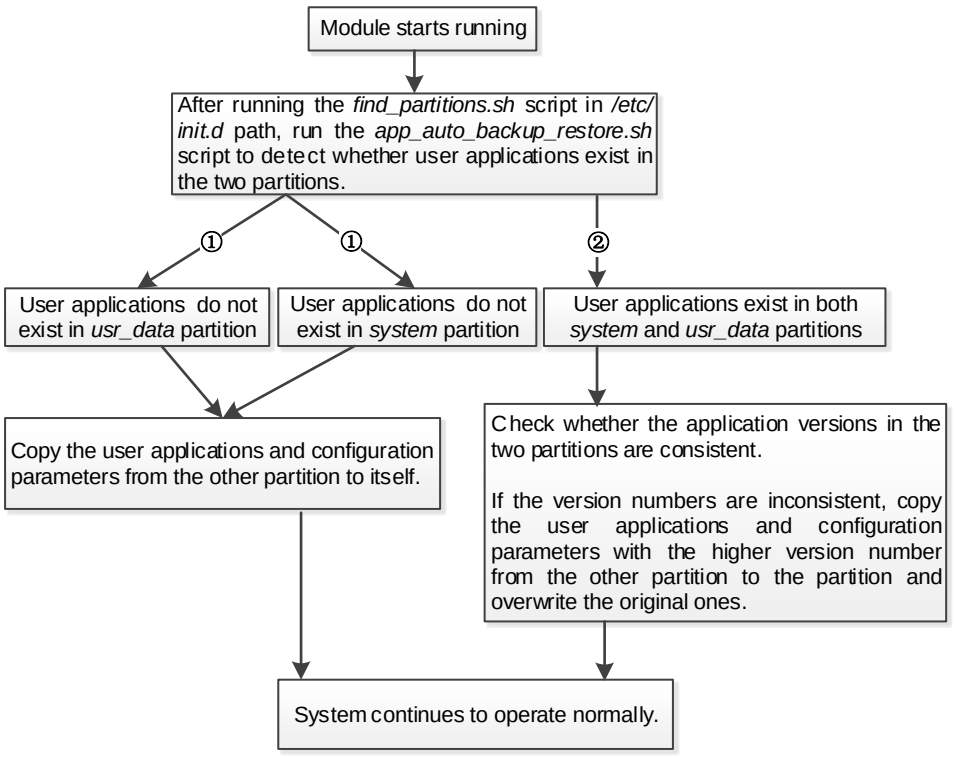

Figure 2: Flow Chart of Backup and Restoration Mechanism

NOTES

1. Due to the caching principle of the Linux file system, if a sudden power failure occurs during backup, the backup application may be incomplete, and cannot be detected by the code, thus the system cannot run during the next restoration. The solution is to copy user applications and configuration parameter files first, then perform the synchronization operation, and finally copy the user application version number tag file *app_ver.txt*.
2. If user applications and configuration parameters are stored in a partition other than *system* and *usr_data*, you can modify the script *app_auto_backup_restore.sh* by changing the name of the mount directory of the *system* or *usr_data* partition to the mount directory of the partition for backup and restoration.
3. After the backup, the two folders should be placed in the *system* and *usr_data* partitions, because the system kernel will format the entire partition after it detects the file system is damaged.

3.2. Possible Scenarios

This chapter mainly analyzes the scenarios and characteristics that may appear in the actual operation of user applications and configuration parameter backup and restoration. For example, if user applications and configuration parameters are stored in *system* partition (the storage path is */usrapp/apprun/apps*), the backup partition is *usr_data* (the backup path is */usrdata/appbackup/apps*), and the following four types may appear in actual operation scenarios.

- When module writes the firmware for the first time, the */usrdata/appbackup/app* backup area is empty, the user application version number tag file *app_ver.txt* under this path does not exist, then the *app_auto_backup_restore.sh* script will copy the folder under the */usrapp/apprun/apps* path and the tag file *app_ver.txt* to the */usrdata/appbackup/app* backup area to complete the backup.

- When the module runs later, if *usr_data* partition is damaged, it will be formatted when the *find_partitions.sh* script is loaded. At this time, the backup data and backup tag files are lost. The *app_auto_backup_restore.sh* script will copy the folder under the */usrapp/apprun/apps* path and the tag file *app_ver.txt* to the */usrdata/appbackup* path to complete the backup.

- After user applications are successfully upgraded, the version number in the *app_ver.txt* file under the */usrapp/apprun/apps* path will be incremented at the same time. Even if the backup area is not updated, when the *app_auto_backup_restore.sh* script detects the version number change at the next startup, it will copy the folder under the */usrapp/apprun/apps* path and the tag file *app_ver.txt* to the */usrdata/appbackup* path to complete the backup.

- User applications have been upgraded and the backup area has been updated. However, *system* partition of the module is damaged in the later use and the restoration occurs. At this time, the contents of the folder under the */usrapp/apprun/apps* path will be restored to factory settings, and the *app_auto_backup_restore.sh* script will determine that the application version number in the backup area is higher, and will copy the folder and tag file *app_ver.txt* in the */usrdata/appbackup/apps* path to the */usrapp/apprun/apps* path to complete the backup.

3.3. Test Methods

You can use any of the following methods to test whether the user applications and configuration parameters are successfully backed up and restored.

1. Delete user applications and version number tag file in *system* or *usr_data* partition.

   - Delete user applications and version number tag file in *system* partition, restart the module (or run the *app_auto_backup_restore.sh* script directly), and check whether the applications and version number tag files in *usr_data* partition are synchronized to *system* partition.

- Delete user applications and version number tag file in *usr_data* partition, restart the module (or run the *app_auto_backup_restore.sh* script directly), and check whether the applications and version number tag file of *system* partition are synchronized to *usr_data* partition.

2. Update user applications and version number tag file in *system* or *usr_data* partition.

   - Add files to the user applications in *system* partition, and increase the version in the version number tag file by 1, restart the module (or run the *app_auto_backup_restore.sh* script directly), and check whether the applications and version number tag files in *system* partition are synchronized to *usr_data* partition.

   - Similarly, update user applications and version number tag file in *usr_data* partition, restart the module (or run the *app_auto_backup_restore.sh* script directly), and check whether the applications and version number tag files in *usr_data* partition are synchronized to *system* partition.

3. After the system is running normally, erase the *system* or *usr_data* partition in fastboot mode.

   - After the system is running normally, erase the *system* partition in fastboot mode, restart the module and wait for the system to run normally, and check whether the applications in *system* partition exists and is consistent with the ones in *usr_data* partition.

   - After the system is running normally, erase the *usr_data* partition in fastboot mode, restart the module and wait for the system to run normally, and check whether the applications in *usr_data* partition exists and is consistent with the ones in *system* partition.

4 Reference Script

The reference script of the user application and configuration parameter backup and restoration reference guide is as follows.

```
#!/bin/sh
# Copyright (c) 2014, The Linux Foundation. All rights reserved.
#
# example app_auto_backup_restore.sh
#
# if app.bin    stored in system partition (/usrapp/app.bin, /usrapp/app_Ver)
# app backup partition is usr_data in (/usrdata/appbackup/app.bin, /usrdata/appbackup/app_Ver)
#
#     /usrapp/apprun/apps
#     /usrapp/apprun/app_ver.txt
#     /usrdata/appbackup/apps
#     /usrdata/appbackup/app_ver.txt

Ver1=0
Ver2=0
app1="/usrapp/apprun/apps"
app1_Ver="/usrapp/apprun/app_ver.txt"
#just support 1,2,3,4,5,6,7 ......
app2="/usrdata/appbackup/apps"
app2_Ver="/usrdata/appbackup/app_ver.txt"
# you'd better make sure the usr_data partition is mount ok in here
if [ ! -f $app1_Ver ];then
      Ver1=-1
else
      Ver1=`cat $app1_Ver`
fi
if [ ! -f $app2_Ver ];then
      Ver2=-1
else
      Ver2=`cat $app2_Ver`
fi

if [ "$Ver1" -eq "$Ver2" ];then
      echo "app verion same , exit !!!"
```

```
        exit 0
fi

echo -n " app version update now!!!!"
if [ "$Ver1" -gt "$Ver2" ];then
        echo -n " app update to usr_data partition /usrdata/appbackup/"
        rm -rf $app2
        rm -rf $app2_Ver
        mkdir -p $app2
        cp -rf $app1    $app2
        sync
        cp -rf $app1_Ver    $app2_Ver
        sync
fi
if [ "$Ver1" -lt "$Ver2" ];then
        echo -n " app    update to system partition    /usrapp/apprun/"
        rm -rf $app1
        rm -rf $app1_Ver
        mkdir -p $app1
        cp -r $app2 $app1
        sync
        cp -rf $app2_Ver $app1_Ver
        sync
fi
```

5 Appendix References

Table 3: Related Documents

SN	Document Name	Description
[1]	Quectel_EC2x&EG9x&EG25-G_Series_QuecOpen_Quick_Start_Guide	Quick start guide applicable for EC2x series, EG9x series and EG25-G QuecOpen modules
[2]	Quectel_EC2x&EG9x&EG25-G_Series_QuecOpen_Linux_Partition_Adjustment_Guide_V1.0	Linux partition adjustment guide applicable for EC2x series, EG9x series and EG25-G QuecOpen modules

Table 4: Terms and Abbreviations

Abbreviation	Description
UBI	Unsorted Block Image