EiGENCOMM
More than connection

TOOL MANUAL

FotaToolkit User Guide_V2.0

EigenCOMM Wireless Microcontroller

Document Description

This document mainly introduces the features of FotaToolkit and also describes how to use it for the purpose of firmware upgrade.

Contents

1. General Description

FOTA (Firmware Over-The-Air) is a kind of software remote upgrade technology, which can provide firmware update services for the terminal devices with networking capabilities. When FOTA is performed, firmware software package will be downloaded from cloud server via networking and be updated for system repair or optimization.

Our products of NB-IOT or CAT.1 series support the differential (diff for short) upgrade of firmware; and users can download a diff file(s) to the terminal device and take it control from a local PC/MCU or cloud server to complete system upgrade.

1.1 Main Procedure

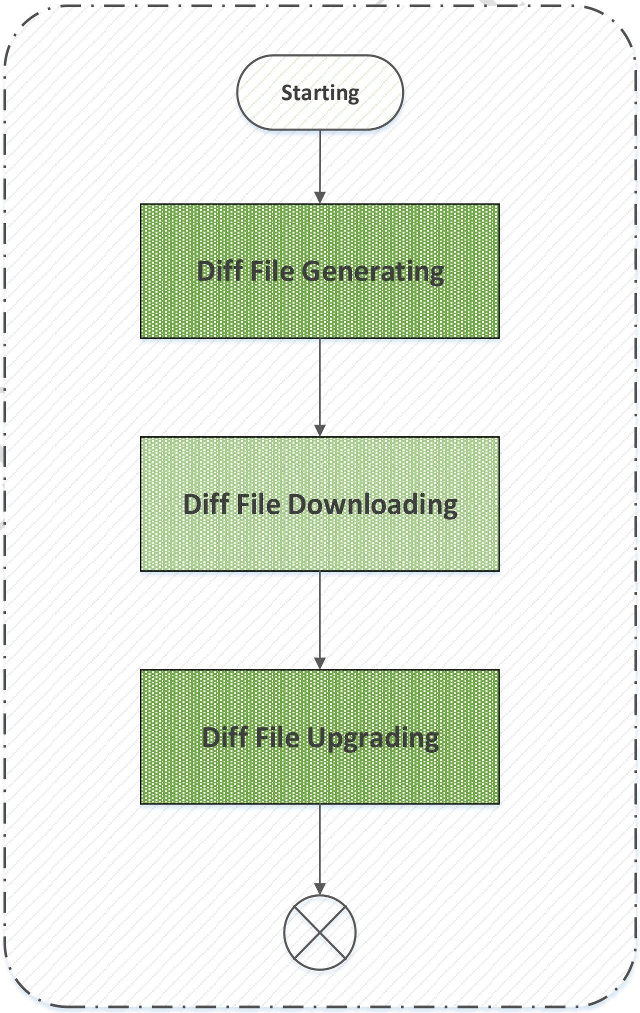
The FOTA service can be subdivided into three steps in terms of the process:

1) Diff file generating(DFG): to generate a FOTA diff file containing a single or multiple images.

2) Diff file downloading(DFD): to download a FOTA diff file to the device in a certain way.

3) Diff file upgrading(DFU): to restore target image(s) from a FOTA diff file and upgrade it.

Figure 1-1. FOTA main procedure diagram



Before performing the upgrade of device, firstly, users need to generate a diff file derived from the target and baseline images, using a kind of tool based on a certain differential algorithm, then the diff file will be downloaded to the device in a certain way. Finally, once upgrade process is triggered, the target image will be restored from the diff file and baseline image through the same or matched algorithm, and the baseline image will be updated and replaced!

The tool of FotaToolkit developed by EiGENCOMM mainly focuses on the first two steps of FOTA service mentioned above.

The following sections will introduce the features of this tool and how to use it.

1.2 Requirements

Table 1-1. Runtime Environment

Object	OS system/Platform
FotaToolkit(GUI)	Windows7/7+

2. How To Set JSON Format File

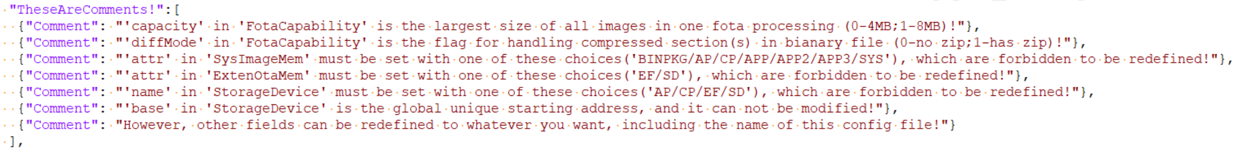
The configuration in JSON format file is essential, which is applied in all features of ‘FotaToolkit’. And it can be customized to meet the needs of their products by customers

Taking ‘config/ec618.json’ for example，all settings in the JSON format file will be introduced in this section.

2.1 Introduction of Comments

The purpose of these comments is to help customers to understand the meaning of some highlighted options and how to set them.

Figure 2-1. Comments in JSON File

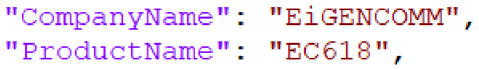


```
 "TheseAreComments!":[
  {"Comment": "'capacity' in 'FotaCapability' is the largest size of all images in one fota processing (0-4MB;1-8MB)!"},
  {"Comment": "'diffMode' in 'FotaCapability' is the flag for handling compressed section(s) in bianary file (0-no zip;1-has zip)!"},
  {"Comment": "'attr' in 'SysImageMem' must be set with one of these choices('BINPKG/AP/CP/APP/APP2/APP3/SYS'), which are forbidden to be redefined!"},
  {"Comment": "'attr' in 'ExtenOtaMem' must be set with one of these choices('EF/SD'), which are forbidden to be redefined!"},
  {"Comment": "'name' in 'StorageDevice' must be set with one of these choices('AP/CP/EF/SD'), which are forbidden to be redefined!"},
  {"Comment": "'base' in 'StorageDevice' is the global unique starting address, and it can not be modified!"},
  {"Comment": "However, other fields can be redefined to whatever you want, including the name of this config file!"}
 ],
```

2.2 Information of the Product

This section will introduce some basic information about the product, and it can be customized to whatever the customers want.

Figure 2-2. Product Info in JSON File



```
"CompanyName": "EiGENCOMM",
"ProductName": "EC618",
```

2.3 Configuration for FOTA Core Version

This section will introduce the version for FOTA core, and different features are included in different cores. The default value for version is matched with the ability of bootloader in the first official release. And it is recommended to double check the ability of bootloader when our customers want to change the version number.

Furthermore, the following features are included in FOTA core V2.x only, which are **not compatible** with core V1.x!

1) Support to handle the compressed sections in BIN file;

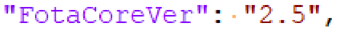
2) Support the maximum capacity (up to 16MB) in one FOTA upgrading procedure;

3) Support to set the size of differential block;

4) Support the compression for package header length to reduce bits;

5) Support the feature of HLS;

Figure 2-3. Core version in JSON File



```
"FotaCoreVer": "2.5",
```

2.4 Configuration for FOTA Basic Capability

This section will introduce some basic capabilities of FOTA.

1) **“capacity”**: means the total size of all images in one single FOTA upgrading procedure (0-4MB，1-16MB)，which should be MATCHED with the capability of bootloader;

2) **“diffMode”**: means the mode of generating the differential file(0-treat the files as raw binaries, 1-consider the compressed sections in BIN files);

3) **“hasPkgHlc”**: means the flag to enable/disable the compression of package header size(0-disable, 1-enable);

4) **“metaPseg”**: means the meta size to increase/decrease the differential block, the value of which should be multiple of 4 and the max value is 32, and the real size is the value of ‘metaPseg’ * 1KiB;

5) **“bkupMemSize”**: means the differential block size in FOTA upgrading, which is set to 32KB that is NOT permitted to be changed in core V1.x and to the size of multiple 32KB in core V2.x;

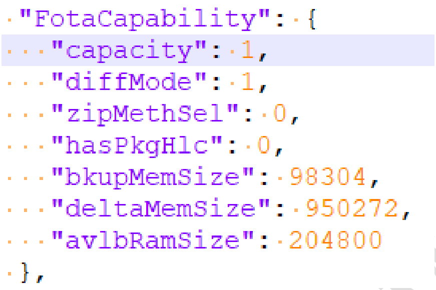
6) **“deltaMemSize”**: means the maximum size of delta file to be stored in FOTA download zone and it can be resized by customers;

7) **“avlbRamSize”**: means the maximum available size that can be applied in FOTA upgrading procedure;

8) **“zipAlgorithm”**: is used to set the compressed algorithm used in FOTA precedure;

- **“zipMeth”**: means the compressed algorithm used in differential file(0-mixed algorithm, 1-only bizp algorithm);
- **“bzipMode”**: means the buffer size mode during bzip running time(0-large size, 1-small size);

Figure 2-4. FOTA Basic Capability Info in JSON File



```
 "FotaCapability": {
   "capacity": 1,
   "diffMode": 1,
   "zipMethSel": 0,
   "hasPkgHlc": 0,
   "bkupMemSize": 98304,
   "deltaMemSize": 950272,
   "avlbRamSize": 204800
 },
```

2.5 Configuration for HLS feature(TBD)

This section will introduce the some settings for HLS (short for HuLu Suite) feature.

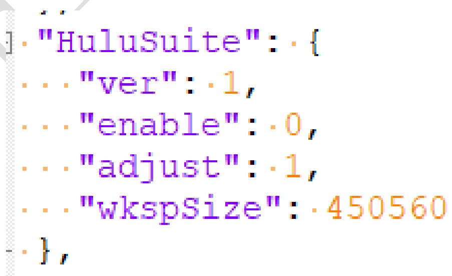
1) **“ver”**: means the internal version of HLS suite;

2) **“enable”**: means the flag to enable/disable HLS(0-disable, 1-enable);

3) **“adjust”**: means the internal flag for HLS;

4) **“wkspSize”**: means the maximum workspace(RAM) size for HLS;

Figure 2-5. Settings for AT Command in JSON File



```
 "HuluSuite": {
   "ver": 1,
   "enable": 0,
   "adjust": 1,
   "wkspSize": 450560
 },
```

2.6 Configuration for Downloading via AT Command

This section will introduce the meaning of some fields in download procedure via AT command and how to set it for customers' needs.

1) **“mode”**: means the data format in downloading procedure via AT command (0-Hex String, 1-Raw Binary);

2) **“ate”**: means AT echo is turned off or on(0-off, 1-on)

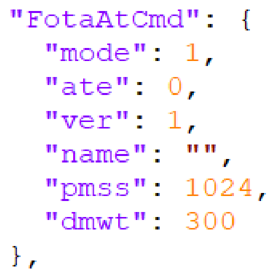
3) **“ver”**: means the version for AT command set(1-NFWUPD, 2-ECOTA);

4) **“name”**: means the customized AT command name, whose format should BE COMPLIED WITH that of ‘NFWUPD’ or ‘ECOTA’;

5) **“pmss”**: means the maximum segment size of the packet, which is LIMITED by that in UE side;

6) **“dmwt”**: means the maximum waiting time of the downloading procedure when it is set to raw binary mode;

Figure 2-6. Settings for AT Command in JSON File



```
"FotaAtCmd": {
  "mode": 1,
  "ate": 0,
  "ver": 1,
  "name": "",
  "pmss": 1024,
  "dmwt": 300
},
```

2.7 Configuration for System Images

This section will introduce the settings about system images and the meanings for the fields in the structure.

1) **“attr”**: means the type of system images and is set in string format, such as BINPKG/AP/CP/APP/SYS/…;

2) **“addr”**: means the starting address for a certain system image;

3) **“size”**: means the maximum size of a certain system image, that is the zone size for the image in the layout;

Now, we will introduce some key points to which we should pay attention. For example, when a file is BINPKG type, it means that all system related images, such as AP/CP/APP(if any)/SYS and etc., have been packed into an archive from whose control header the information of “attr”, ”addr” and “size” can be straightly got, wherefores, it means those fields of ‘SysImageMem’ in JSON file will not be used; however, when an image pair of specific kind will be made to a delta file or data in a specific memory zone will be read out, the fields of ‘SysImageMem’ are essential and customers should set them to proper values based on the features of product.

For the value of “**attr**” filed, it is necessary to explain the meanings of some particular value.

1) **“BINPKG”**: means an image archive containing all the needed images, which is introduced above;

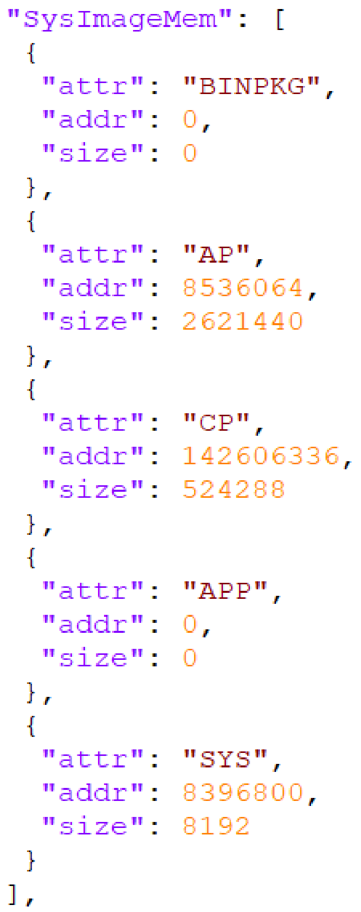
2) **“AP”**: means the AP image;

3) **“CP”**: means the CP image;

4) **“APP”**: means a particular customized image by customers if any;

5) **“SYS”**: short for SYstem Security, means a special control header for validating the images of AP, CP, APP (if any), which is mainly used for the feature of secure boot;

Figure 2-7. System Image Memory Zone in JSON File



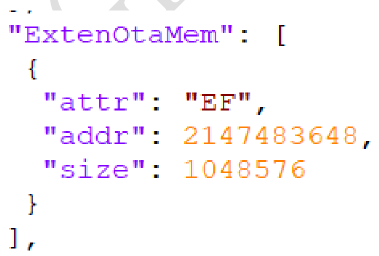
```
"SysImageMem": [
  {
   "attr": "BINPKG",
   "addr": 0,
   "size": 0
  },
  {
   "attr": "AP",
   "addr": 8536064,
   "size": 2621440
  },
  {
   "attr": "CP",
   "addr": 142606336,
   "size": 524288
  },
  {
   "attr": "APP",
   "addr": 0,
   "size": 0
  },
  {
   "attr": "SYS",
   "addr": 8396800,
   "size": 8192
  }
 ],
```

2.8 Configuration for Extensional OTA Storage Zone

This section will introduce the settings about extensional OTA storage zone and the meanings for the fields in the structure. And for now, it is supported to download data to external FLASH device only, which could be divided to several memory zones for storing different types of data by customers.

The meanings of “attr”, ”addr” and “size” are the same as those in ‘SysImageMem’, and for now, the only value for “attr” field is EF, which means External Flash device.

Figure 2-8. Extensional OTA Memory Zone in JSON File



```
"ExtenOtaMem": [
  {
   "attr": "EF",
   "addr": 2147483648,
   "size": 1048576
  }
 ],
```

2.9 Configuration for Storage Devices

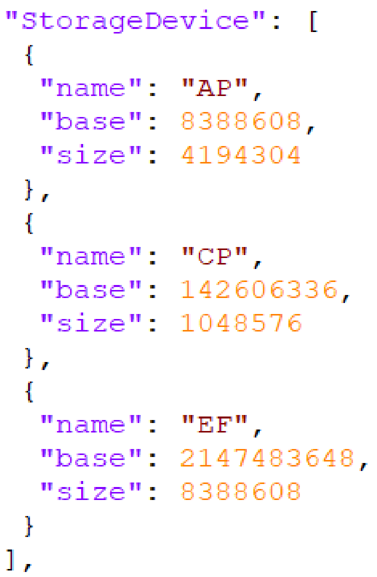
This section will introduce the basic settings for storage device, which is used to get the global starting address of the device for creating the delta file and downloading the data via AT command.

1) **“name”**: means the storage device type, which could be “AP”, “CP” or “EF”, meaning where the image is stored;

2) **“base”**: means the starting address of the device;

3) **“size”**: means the total size of the device;

Figure 2-9. Setting for Storage Device in JSON File



```
"StorageDevice": [
 {
  "name": "AP",
  "base": 8388608,
  "size": 4194304
 },
 {
  "name": "CP",
  "base": 142606336,
  "size": 1048576
 },
 {
  "name": "EF",
  "base": 2147483648,
  "size": 8388608
 }
],
```

2.10 Configuration for Customized Features

This section will introduce some settings for customized features which aims to support some special scenarios and all of which are **optional**. For now, resizing the loading zone is the only customized feature, and others will be developed later.

2.10.1 Resizing the Loading Zone

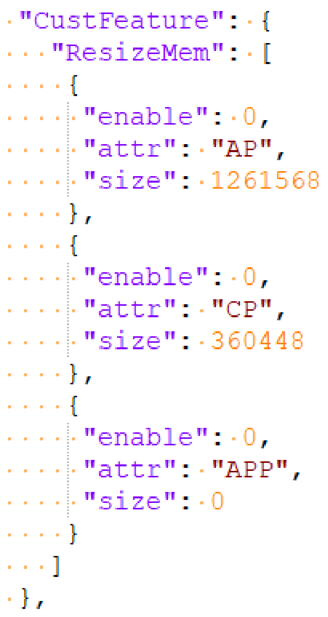
The settings aim to support to generate the differential file out of a file pair with a different size, and customers can customize and enable to resize whichever loading zone they want.

1) **“enable”**: means the flag to enable/disable the resizing a certain loading zone (0-disable, 1-enable);

2) **“attr”**: the type of system images and is set in string format, such as BINPKG/AP/CP/APP/SYS/…;

3) **“size”**: means the new adjusted size for a certain system image;

Figure 2-10. Setting for Resizing the Loading Zone in JSON File

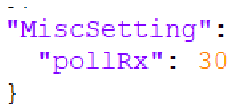


```
 "CustFeature": {
   "ResizeMem": [
    {
     "enable": 0,
     "attr": "AP",
     "size": 1261568
    },
    {
     "enable": 0,
     "attr": "CP",
     "size": 360448
    },
    {
     "enable": 0,
     "attr": "APP",
     "size": 0
    }
   ]
 },
```

2.11 Configuration for Miscellaneous Options

This section will introduce a few miscellaneous settings for the tool, and for now, the only field is ‘pollRx’, which means the polling time (ms) for serial port receiving.

Figure 2-11. Miscellaneous Settings in JSON File



```
"MiscSetting": {
   "pollRx": 30
}
```

3. How To Use The GUI Tool

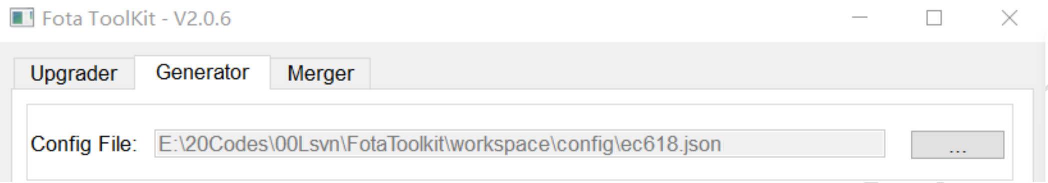
In this section, three important features will be introduced:

1) How to use it to generate a diff file;

2) How to merge diff files into one;

3) How to convert a diff file into AT command lines and upgrade via AT command.

Figure 3-1. FotaToolkit Main GUI



Config File: E:\20Codes\00Lsvn\FotaToolkit\workspace\config\ec618.json

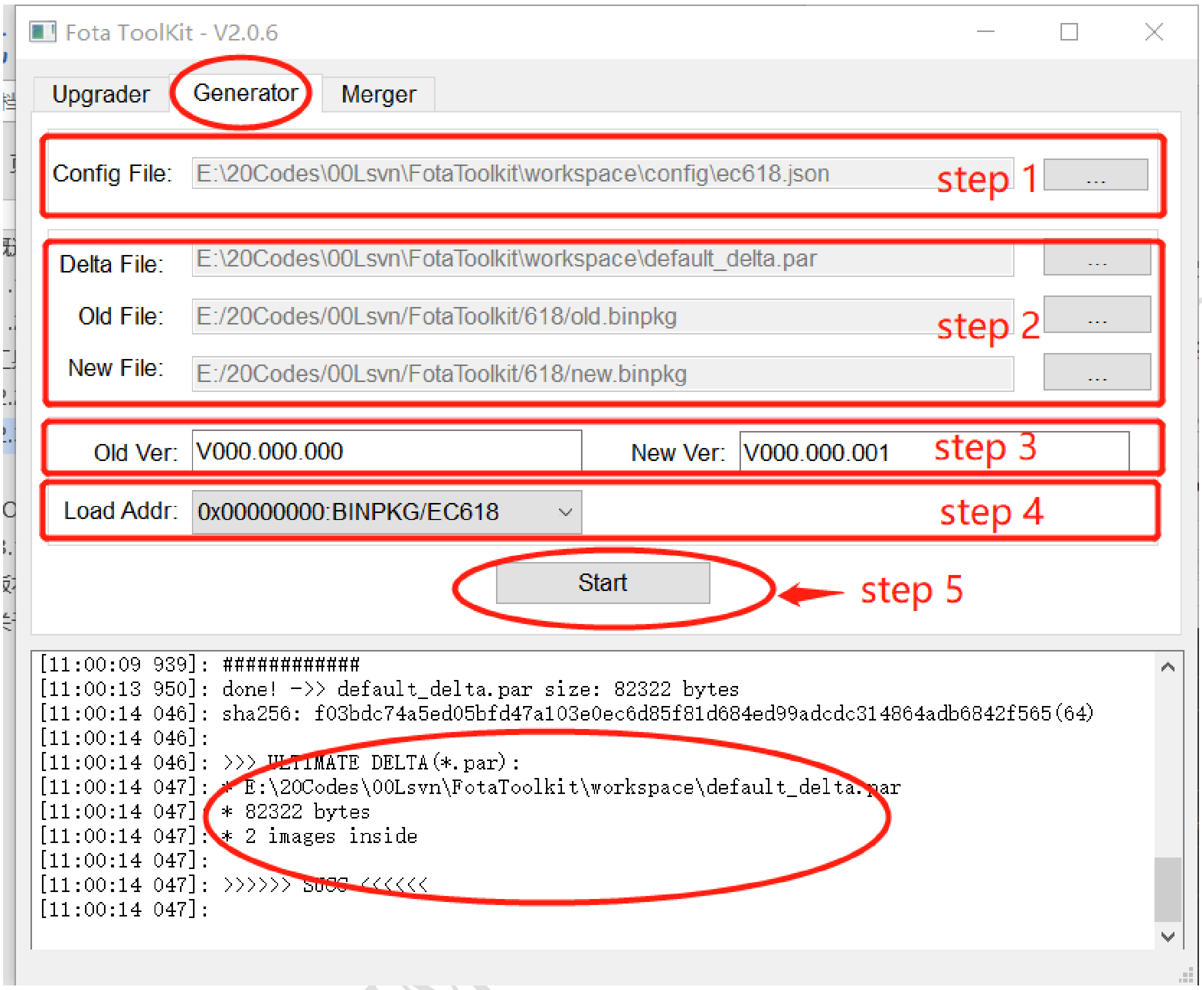
3.1 To Set the Config File

The configuration file in JSON format for this tool is essential, which will be used to generate, merge or download a diff/delta file. What we concern most about the file is the starting address and size of the FOTA and IMAGE areas, which varies per to different products; so do the flags for the features.

For more details, you can refer to Section 2 “How To set JSON Format File”!

3.2 To Generate A Diff File

Figure 3-2.　GUI for Generating Diff File



Config File: E:\20Codes\00Lsvn\FotaToolkit\workspace\config\ec618.json

Delta File: E:\20Codes\00Lsvn\FotaToolkit\workspace\default_delta.par

Old File: E:/20Codes/00Lsvn/FotaToolkit/618/old.binpkg

New File: E:/20Codes/00Lsvn/FotaToolkit/618/new.binpkg

Old Ver: V000.000.000　New Ver: V000.000.001

Load Addr: 0x00000000:BINPKG/EC618

```
[11:00:09 939]: ############
[11:00:13 950]: done! ->> default_delta.par size: 82322 bytes
[11:00:14 046]: sha256: f03bdc74a5ed05bfd47a103e0ec6d85f81d684ed99adcdc314864adb6842f565(64)
[11:00:14 046]:
[11:00:14 046]: >>> ULTIMATE DELTA(*.par):
[11:00:14 047]: * E:\20Codes\00Lsvn\FotaToolkit\workspace\default_delta.par
[11:00:14 047]: * 82322 bytes
[11:00:14 047]: * 2 images inside
[11:00:14 047]:
[11:00:14 047]: >>>>>> SUCC <<<<<<
[11:00:14 047]:
```

Open the FotaToolkitGUI.exe, and select the ‘Generator’ tab to generate a diff file in following five steps.

1) Select the configuration file, which would be used as the default one once it is selected and the GUI is never closed, of a specific product project, such as config/ec618.json;

2) Set the name and path for the diff file (whose default path is the directory where FotaToolkitGUI.exe is located and default name is ‘default_delta.par’), and then select the files of baseline image and target image, from which the diff file derives;

3) **This step is optional.** Fill the version of the baseline and target images, whose format is Vmmm.nnn.ppp (m, n, and p are figure 0-9, representing major, minor and the patch version of the image respectively). It should be noted that the version number of the target & baseline image cannot be the same; additionally, **the version number is only for tracking and recording the information of the baseline and target image and it will not be checked and verified**;

4) Select the proper FLASH configuration information according to the attributes of the baseline and target images in step 2, which have been set in the JSON file;

5) Press "START" button to create a diff file straightly when the above steps are completed. And the display window will output all the LOG information of the running process. Once it is done, a pop-up will show the execution result: "success" or "failure", and also more detailed information of the ultimate file will also be revealed in the display window.

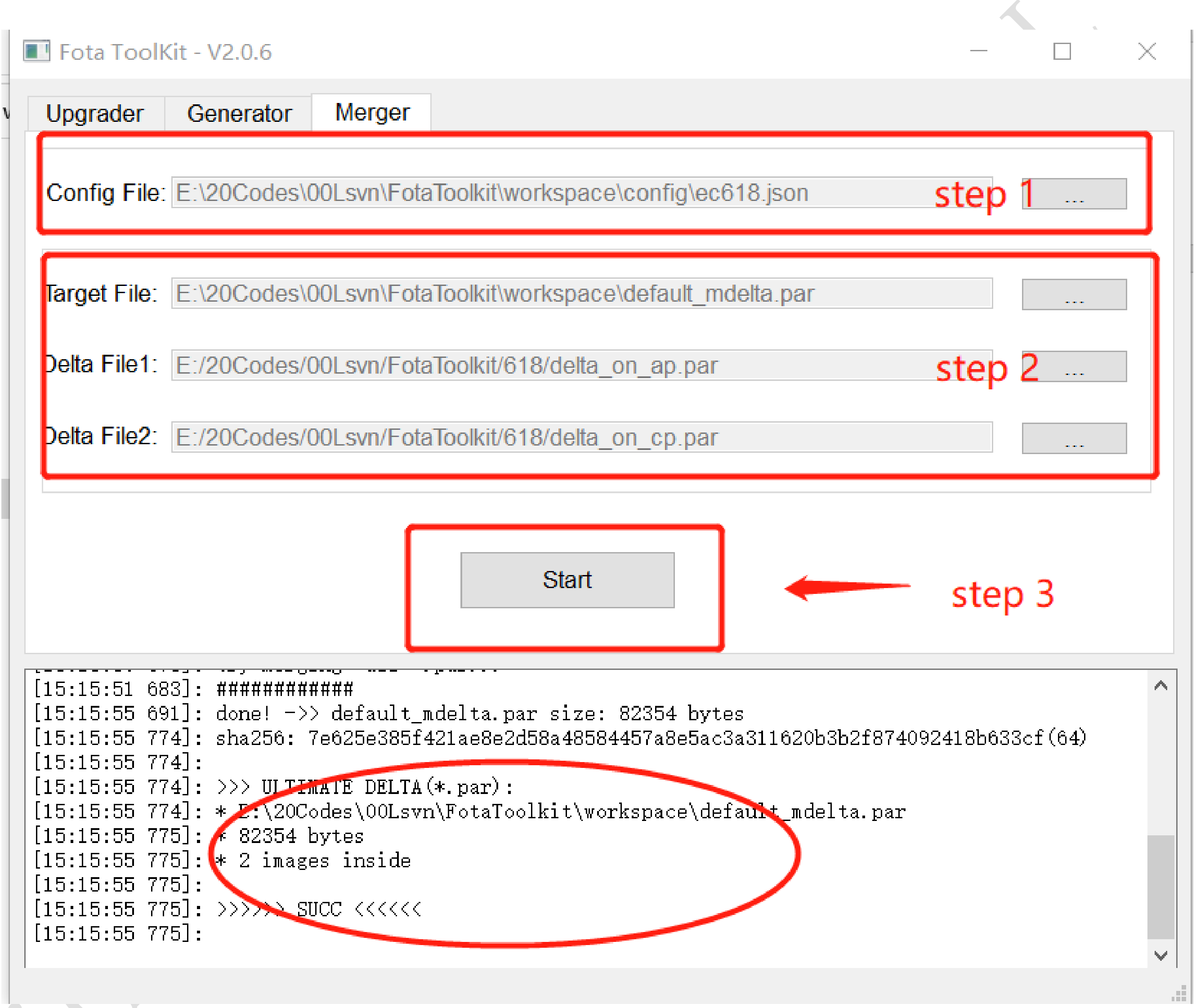
※※NOTE※※

The suffix name (.par) is the default file suffix of EiGENCOMM FOTA scheme. Once the diff file name is set, it will be

appened automatically by the FotaToolkitGUI.exe.

3.3 To Merge Diff Files

Figure 3-3. GUI for Merging Diff Files



Config File: E:\20Codes\00Lsvn\FotaToolkit\workspace\config\ec618.json

Target File: E:\20Codes\00Lsvn\FotaToolkit\workspace\default_mdelta.par

Delta File1: E:/20Codes/00Lsvn/FotaToolkit/618/delta_on_ap.par

Delta File2: E:/20Codes/00Lsvn/FotaToolkit/618/delta_on_cp.par

```
[15:15:51 683]: ############
[15:15:55 691]: done! ->> default_mdelta.par size: 82354 bytes
[15:15:55 774]: sha256: 7e625e385f421ae8e2d58a48584457a8e5ac3a311620b3b2f874092418b633cf(64)
[15:15:55 774]:
[15:15:55 774]: >>> ULTIMATE DELTA(*.par):
[15:15:55 774]: * E:\20Codes\00Lsvn\FotaToolkit\workspace\default_mdelta.par
[15:15:55 775]: * 82354 bytes
[15:15:55 775]: * 2 images inside
[15:15:55 775]:
[15:15:55 775]: >>>>>> SUCC <<<<<<
[15:15:55 775]:
```

Open the FotaToolkitGUI.exe, and select the ‘Merger’ tab to merge diff files in following three steps.

1) Select the configuration file, which would be used as the default one once it is selected and the GUI is never closed, of a specific product project, such as config/ec618.json;

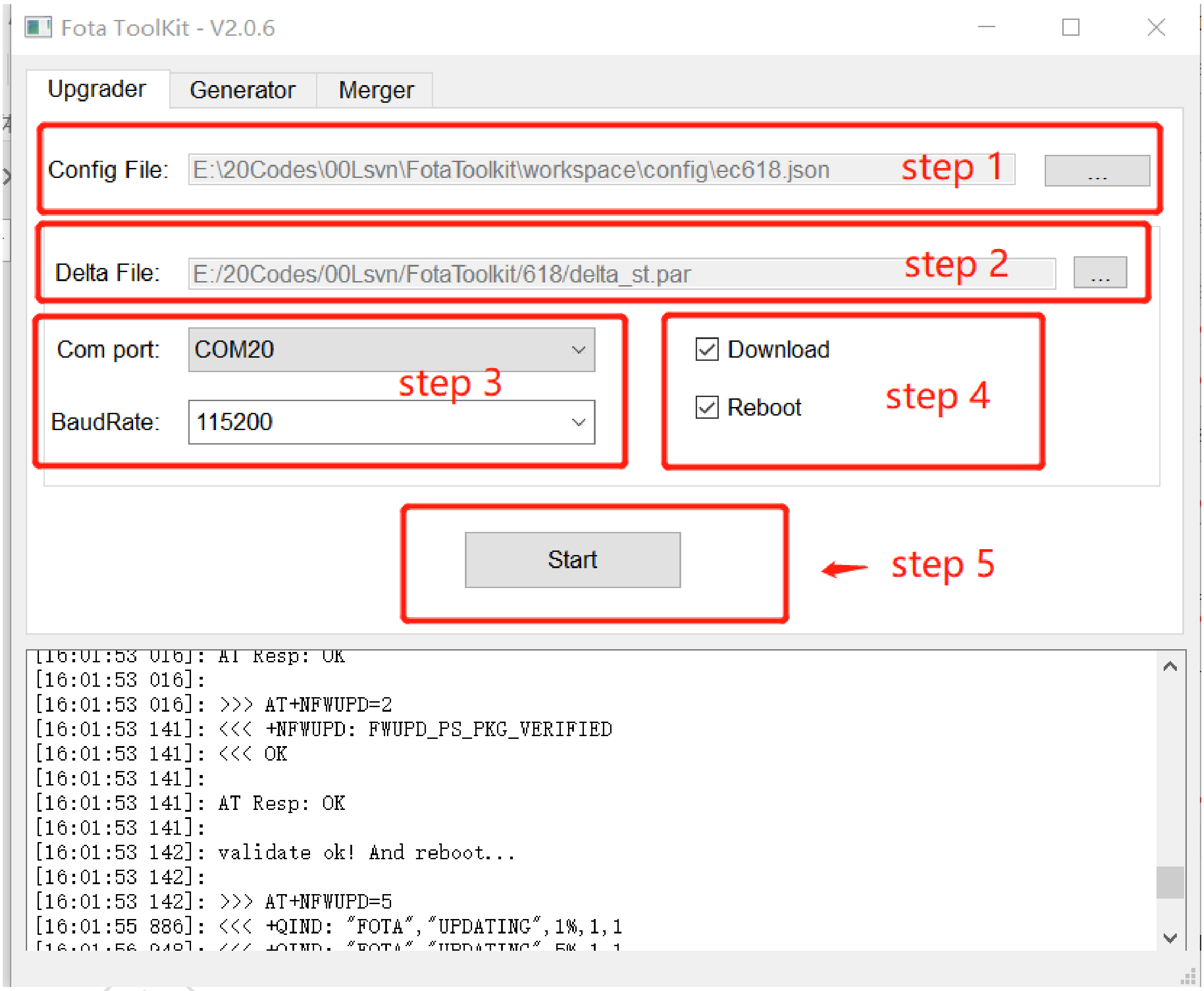
2) Set the name and path for the target merged file (whose default path is the directory where FotaToolkitGUI.exe is located and default name is ‘default_mdelta.par’), and then select the diff file 1 and file 2 as the input files, such as ‘delta_on_ap.par’ and ‘delta_on_cp.par’ in Figure2-4;

3) Press "START" button to merge the diff files straightly when the above steps are completed. Once it is done, a pop-up will show the execution result: "success" or "failure", and also more detailed information of the ultimate file will also be revealed

in the display window.

3.4 To Upgrade A Diff File

Figure 3-4. GUI for Upgrading A Diff File via AT Command



Config File: E:\20Codes\00Lsvn\FotaToolkit\workspace\config\ec618.json

Delta File: E:/20Codes/00Lsvn/FotaToolkit/618/delta_st.par

Com port: COM20

BaudRate: 115200

☑ Download

☑ Reboot

```
[16:01:53 016]: AT Resp: OK
[16:01:53 016]:
[16:01:53 016]: >>> AT+NFWUPD=2
[16:01:53 141]: <<< +NFWUPD: FWUPD_PS_PKG_VERIFIED
[16:01:53 141]: <<< OK
[16:01:53 141]:
[16:01:53 141]: AT Resp: OK
[16:01:53 141]:
[16:01:53 142]: validate ok! And reboot...
[16:01:53 142]:
[16:01:53 142]: >>> AT+NFWUPD=5
[16:01:55 886]: <<< +QIND: "FOTA","UPDATING",1%,1,1
[16:01:56 048]: <<< +QIND: "FOTA","UPDATING",5%,1,1
```

Open the FotaToolkitGUI.exe, and select the ‘Upgrader’ tab to upgrade the device in following five steps.

1) Select the configuration file, which would be used as the default one once it is selected and the GUI is never closed, of a specific product project, such as config/ec618.json;

2) Select the diff file to be upgraded, such as ‘delta_st.par’ in Figure2-5;

3) Select the serial COM port for the AT channel of the connected device, and then set the matched baud rate (9600 and 115200 by default). And the user can input any baud rate they want in the Dropdown List if there is no proper option;

4) Before pressing "START" button, the user can choose whether to "Download" the differential file or "Reboot" the device as their needs. if “Download” option is not set, the window will output the AT command lines converted from the diff file; if “Reboot” option is not set, the diff file will be downloaded to the device only and the upgrade process, which can be performed whenever it is needed by users later, will not be triggered.

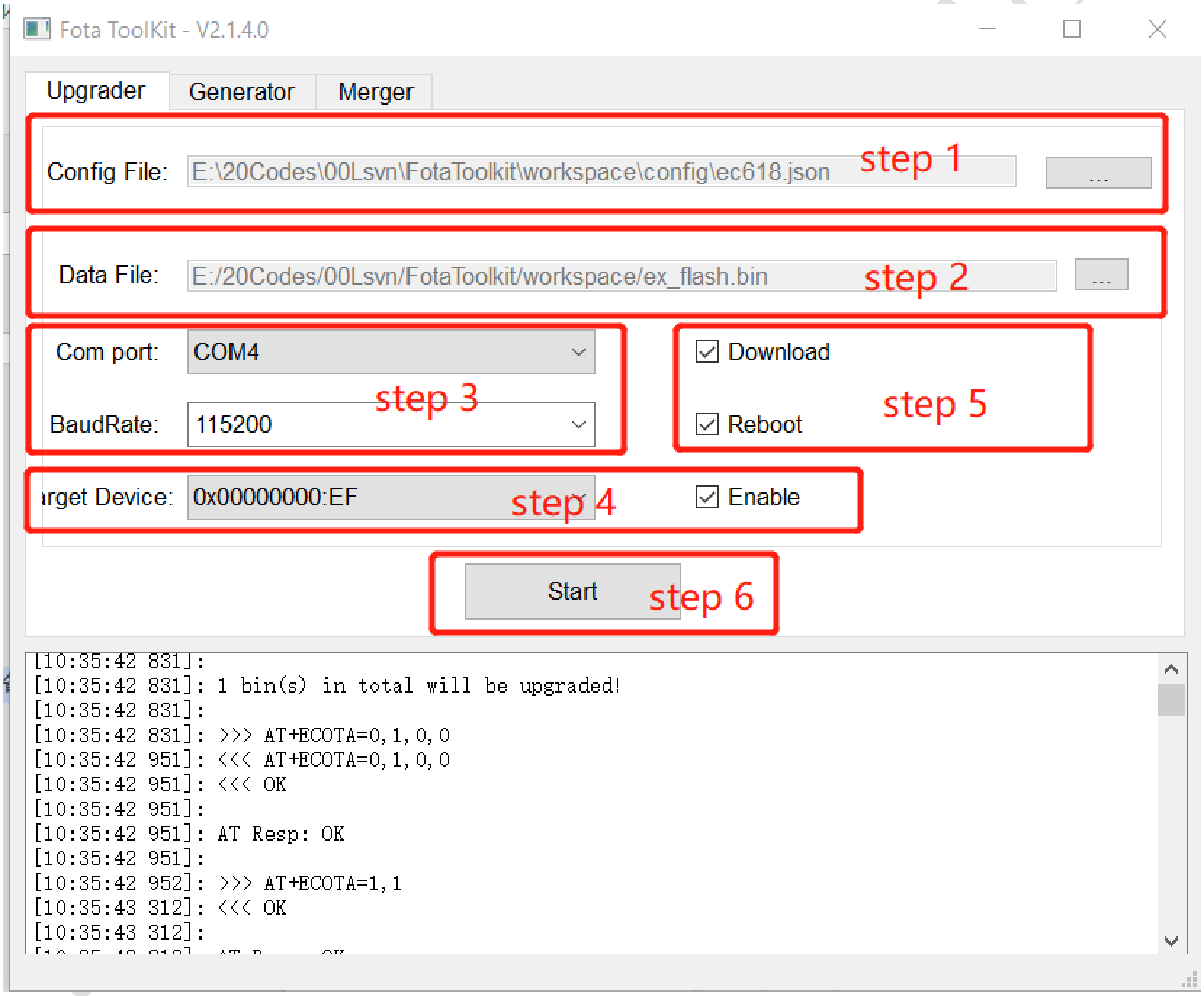
5) Press "START" button and the running log will be show in the display window.

※※NOTE※※

AT command is only a kind of way for downloading a diff file to the device from a PC or MCU, and also there is another networking way for downloading from a cloud server, which usually occurs through some application protocols, such as FTP, HTTP(s) and etc.

3.5 To Deliver Data To An Extensional Storage Device

Figure 3-5. GUI for Delivering Data To An Extensional Storage Device via AT command



Config File: E:\20Codes\00Lsvn\FotaToolkit\workspace\config\ec618.json

Data File: E:/20Codes/00Lsvn/FotaToolkit/workspace/ex_flash.bin

Com port: COM4

BaudRate: 115200

☑ Download

☑ Reboot

ırget Device: 0x00000000:EF

☑ Enable

```
[10:35:42 831]:
[10:35:42 831]: 1 bin(s) in total will be upgraded!
[10:35:42 831]:
[10:35:42 831]: >>> AT+ECOTA=0,1,0,0
[10:35:42 951]: <<< AT+ECOTA=0,1,0,0
[10:35:42 951]: <<< OK
[10:35:42 951]:
[10:35:42 951]: AT Resp: OK
[10:35:42 951]:
[10:35:42 952]: >>> AT+ECOTA=1,1
[10:35:43 312]: <<< OK
[10:35:43 312]:
```

Open the FotaToolkitGUI.exe, and select the ‘Upgrader’ tab to upgrade the device in following six steps.

1) Select the configuration file, which would be used as the default one once it is selected and the GUI is never closed, of a specific product project, such as config/ec618.json;

2) Select the binary file to be delivered, such as ‘ex_flash.bin’ in Figure2-6;

3) Select the serial COM port for the AT channel of the connected device, and then set the matched baud rate (9600 and 115200 by default). And the user can input any baud rate they want in the Dropdown List if there is no proper option;

4) “Enable” the feature of delivering to external storage device, then select the target zone whose starting address and total size

can be customized in *.json file; however, if the starting address and total size are not changed and still set with the default value, then the file will be delivered to the default external storage zone of the system.

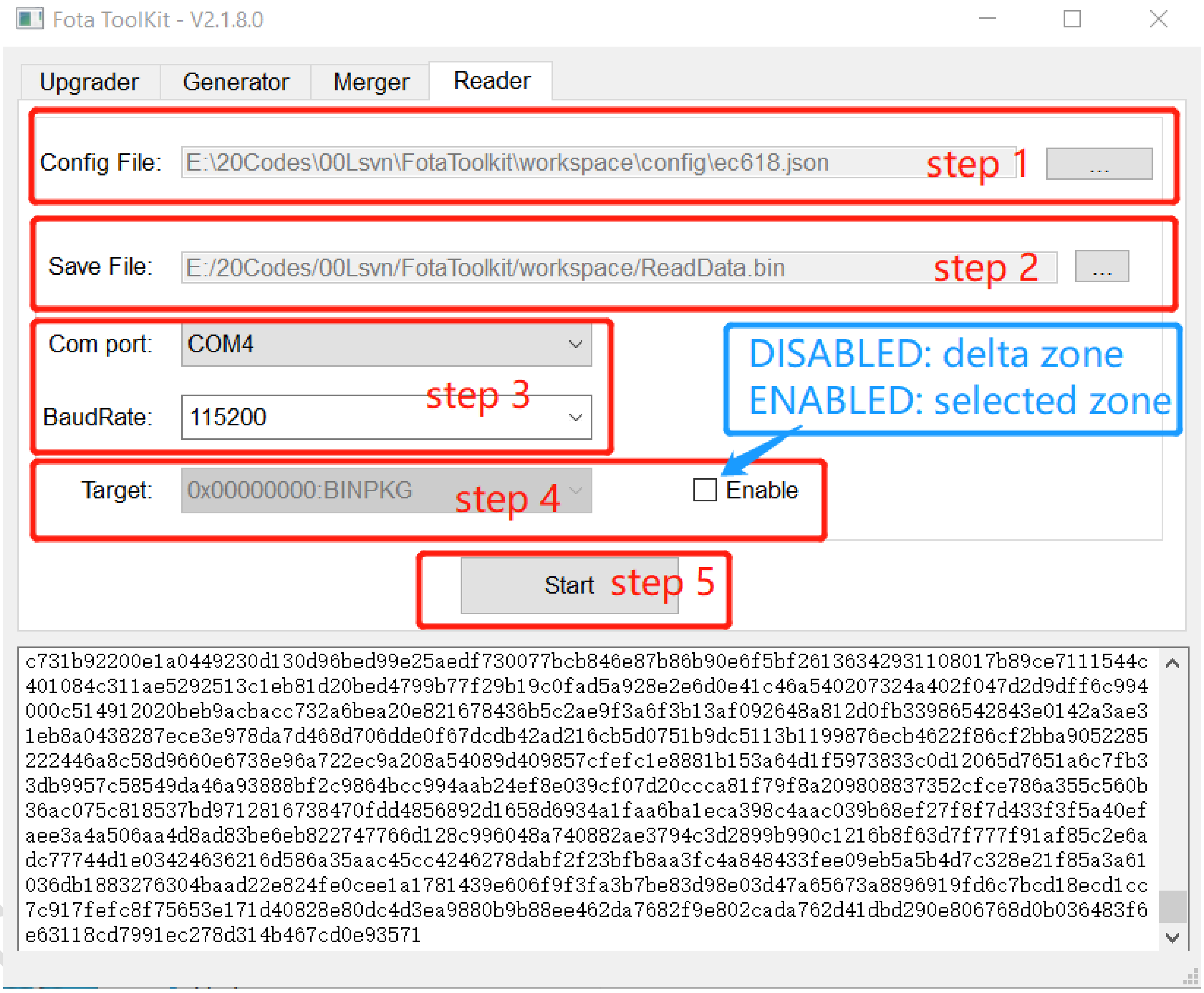
[NOTE] If the "Enable" button is NOT selected; the file will be delivered/downloaded to the FOTA zone of the system.

5) Before pressing "START" button, the user can choose whether to "Download" the file or "Reboot" the device as their needs. if "Download" option is not set, the window will output the AT command lines converted from the file; if "Reboot" option is not set, the file will be downloaded to the device only.

6) Press "START" button and the running log will be show in the display window.

3.6 To Read Data Out Of A Storage Zone

Figure 3-6. GUI for Reading Data Out Of A Storage Zone via AT Command



Config File: E:\20Codes\00Lsvn\FotaToolkit\workspace\config\ec618.json

Save File: E:/20Codes/00Lsvn/FotaToolkit/workspace/ReadData.bin

Com port: COM4

BaudRate: 115200

Target: 0x00000000:BINPKG

□ Enable

```
c731b92200e1a0449230d130d96bed99e25aedf730077bcb846e87b86b90e6f5bf26136342931108017b89ce7111544c
401084c311ae5292513c1eb81d20bed4799b77f29b19c0fad5a928e2e6d0e41c46a540207324a402f047d2d9dff6c994
000c514912020beb9acbacc732a6bea20e821678436b5c2ae9f3a6f3b13af092648a812d0fb33986542843e0142a3ae3
1eb8a0438287ece3e978da7d468d706dde0f67dcdb42ad216cb5d0751b9dc5113b1199876ecb4622f86cf2bba9052285
222446a8c58d9660e6738e96a722ec9a208a54089d409857cfefc1e8881b153a64d1f5973833c0d12065d7651a6c7fb3
3db9957c58549da46a93888bf2c9864bcc994aab24ef8e039cf07d20ccca81f79f8a209808837352cfce786a355c560b
36ac075c818537bd9712816738470fdd4856892d1658d6934a1faa6ba1eca398c4aac039b68ef27f8f7d433f3f5a40ef
aee3a4a506aa4d8ad83be6eb822747766d128c996048a740882ae3794c3d2899b990c1216b8f63d7f777f91af85c2e6a
dc77744d1e03424636216d586a35aac45cc4246278dabf2f23bfb8aa3fc4a848433fee09eb5a5b4d7c328e21f85a3a61
036db1883276304baad22e824fe0cee1a1781439e606f9f3fa3b7be83d98e03d47a65673a8896919fd6c7bcd18ecd1cc
7c917fefc8f75653e171d40828e80dc4d3ea9880b9b88ee462da7682f9e802cada762d41dbd290e806768d0b036483f6
e63118cd7991ec278d314b467cd0e93571
```

Open the FotaToolkitGUI.exe, and select the 'Reader' tab to read data out of a specific storage zone in following five steps.

1) Select the configuration file, which would be used as the default one once it is selected and the GUI is never closed, of a specific product project, such as config/ec618.json;

2) Set the binary file for saving, such as 'ReadData.bin' in Figure2-7;

3) Select the serial COM port for the AT channel of the connected device, and then set the matched baud rate (9600 and 115200

by default). And the user can input any baud rate they want in the Dropdown List if there is no proper option;

4) If the data in FOTA download zone is to be read out, do not “Enable” the target storage zone; however, if the data in other storage zone (such as AP/CP/EF zone), do “Enable” the target zone and select the zone which you want.

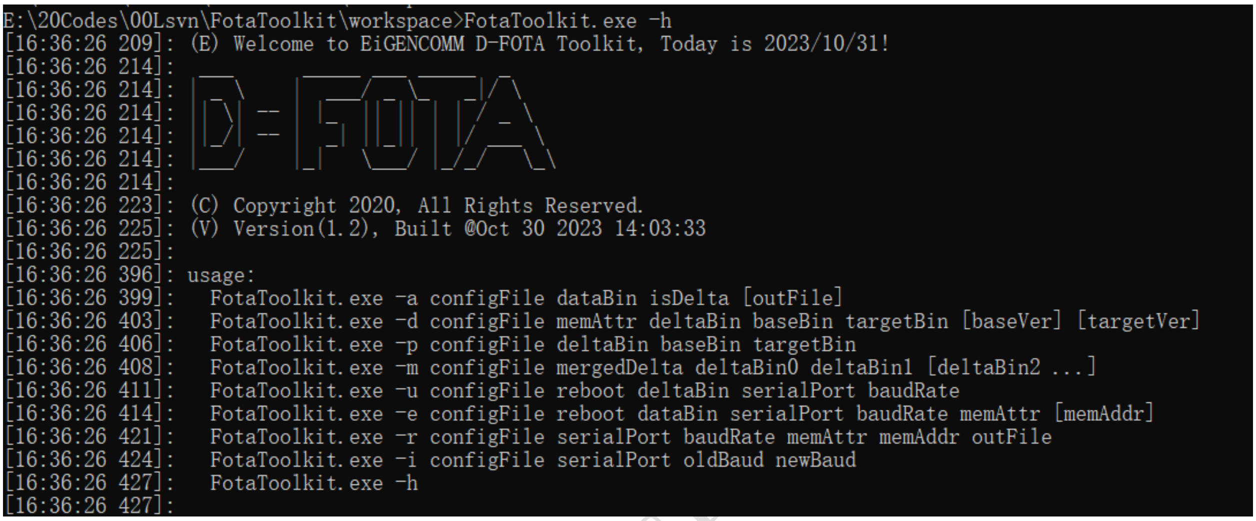
   [NOTE] If the “Enable” button is NOT selected; data in the FOTA download zone of the system will be read out.

5) Press "START" button and the running log will be show in the display window.

4. How To Use The Command Lines

The detail usage of command lines can be displayed by executing “FotaToolkit -h”, and the example of each option is listed as well.

Figure 4-1. Help Information for“FotaToolkit.exe –h”



```
E:\20Codes\00Lsvn\FotaToolkit\workspace>FotaToolkit.exe -h
[16:36:26 209]: (E) Welcome to EiGENCOMM D-FOTA Toolkit, Today is 2023/10/31!
[16:36:26 214]:
[16:36:26 214]:  ___      _____ ___ _____ _
[16:36:26 214]: |   \    |  ___/ _ \_   _|/ \
[16:36:26 214]: | |) | --| |_ | | | || | / _ \
[16:36:26 214]: | |_/| --|  _|| |_| || |/ ___ \
[16:36:26 214]: |___/    |_|   \___/ |_/_/   \_\
[16:36:26 214]:
[16:36:26 223]: (C) Copyright 2020, All Rights Reserved.
[16:36:26 225]: (V) Version(1.2), Built @Oct 30 2023 14:03:33
[16:36:26 225]:
[16:36:26 396]: usage:
[16:36:26 399]:   FotaToolkit.exe -a configFile dataBin isDelta [outFile]
[16:36:26 403]:   FotaToolkit.exe -d configFile memAttr deltaBin baseBin targetBin [baseVer] [targetVer]
[16:36:26 406]:   FotaToolkit.exe -p configFile deltaBin baseBin targetBin
[16:36:26 408]:   FotaToolkit.exe -m configFile mergedDelta deltaBin0 deltaBin1 [deltaBin2 ...]
[16:36:26 411]:   FotaToolkit.exe -u configFile reboot deltaBin serialPort baudRate
[16:36:26 414]:   FotaToolkit.exe -e configFile reboot dataBin serialPort baudRate memAttr [memAddr]
[16:36:26 421]:   FotaToolkit.exe -r configFile serialPort baudRate memAttr memAddr outFile
[16:36:26 424]:   FotaToolkit.exe -i configFile serialPort oldBaud newBaud
[16:36:26 427]:   FotaToolkit.exe -h
[16:36:26 427]:
```

4.1 To Generate A Diff File

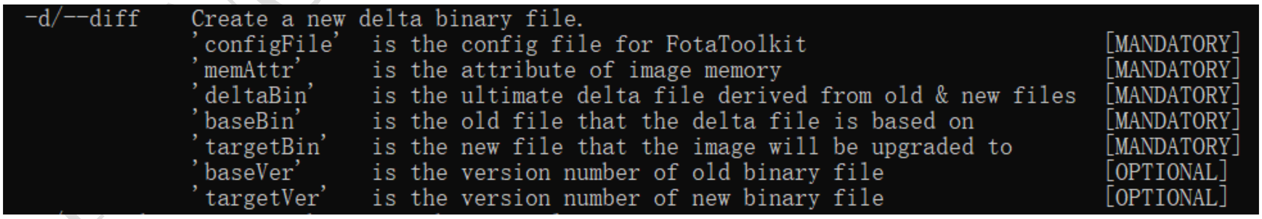
[FUNCTION]

To create a diff/delta file out of the image pair based on a certain of differential algorithm.

[FORMAT]

FotaToolkit.exe -d configFile memAttr deltaBin baseBin targetBin [baseVer] [targetVer]

[USAGE]



```
  -d/--diff     Create a new delta binary file.
                'configFile'  is the config file for FotaToolkit                 [MANDATORY]
                'memAttr'     is the attribute of image memory                   [MANDATORY]
                'deltaBin'    is the ultimate delta file derived from old & new files  [MANDATORY]
                'baseBin'     is the old file that the delta file is based on    [MANDATORY]
                'targetBin'   is the new file that the image will be upgraded to [MANDATORY]
                'baseVer'     is the version number of old binary file           [OPTIONAL]
                'targetVer'   is the version number of new binary file           [OPTIONAL]
```

[EXAMPLE]

FotaToolkit.exe -d config/ec618.json BINPKG delta.par old.binpkg new.binpkg

4.2 To Merge Diff Files

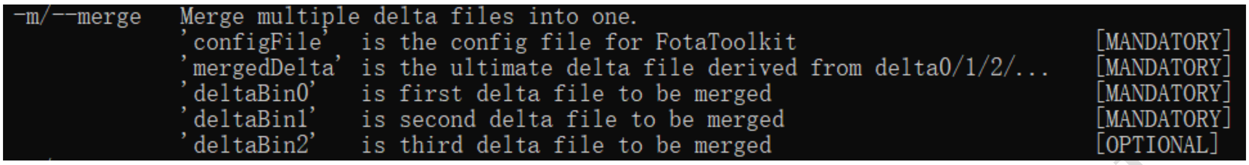
[FUNCTION]

To merge two or more diff/delta files (containing one or more images in each) into a diff/delta file (containing all the images).

[FORMAT]

FotaToolkit.exe -m configFile mergedDelta deltaBin0 deltaBin1 [deltaBin2 ...]

[USAGE]



```
-m/--merge   Merge multiple delta files into one.
             'configFile'  is the config file for FotaToolkit                [MANDATORY]
             'mergedDelta' is the ultimate delta file derived from delta0/1/2/... [MANDATORY]
             'deltaBin0'   is first delta file to be merged                  [MANDATORY]
             'deltaBin1'   is second delta file to be merged                 [MANDATORY]
             'deltaBin2'   is third delta file to be merged                  [OPTIONAL]
```

[EXAMPLE]

FotaToolkit.exe -m config/ec618.json mDelta.par delta1.par delta2.par

4.3 To Upgrade A Diff File

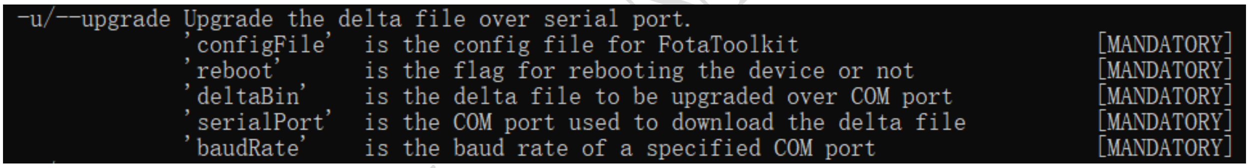
[FUNCTION]

To download a diff/delta file to UE side via AT commands for upgrading.

[FORMAT]

FotaToolkit.exe -u configFile reboot deltaBin serialPort baudRate

[USAGE]



```
-u/--upgrade Upgrade the delta file over serial port.
             'configFile'  is the config file for FotaToolkit                [MANDATORY]
             'reboot'      is the flag for rebooting the device or not       [MANDATORY]
             'deltaBin'    is the delta file to be upgraded over COM port    [MANDATORY]
             'serialPort'  is the COM port used to download the delta file   [MANDATORY]
             'baudRate'    is the baud rate of a specified COM port          [MANDATORY]
```

[EXAMPLE]

FotaToolkit.exe -u config/ec618.json 1 delta.par COM1 115200

4.4 To Convert To AT Commands

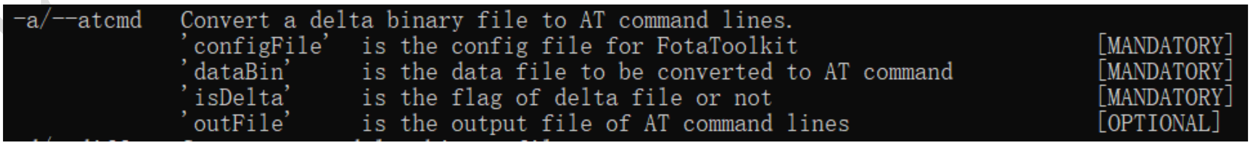
[FUNCTION]

To convert a diff/delta file into a series of AT command lines.

[FORMAT]

FotaToolkit.exe -a configFile dataBin isDelta [outFile]

[USAGE]



```
-a/--atcmd   Convert a delta binary file to AT command lines.
             'configFile'  is the config file for FotaToolkit                [MANDATORY]
             'dataBin'     is the data file to be converted to AT command    [MANDATORY]
             'isDelta'     is the flag of delta file or not                  [MANDATORY]
             'outFile'     is the output file of AT command lines            [OPTIONAL]
```

[EXAMPLE]

FotaToolkit.exe -a config/ec618.json delta.par 1 atcmd.txt

5. FAQ

5.1 Question 1

Q: What are the meanings of different ‘loading address’ options in the GUI?

A: As what you see, the option contains 3 pieces of important information: the starting address of image, the attribute of image and the chip family that the image belongs to.

Taking ec618 for example, the meaning of attribute is explained as follows:

1) BINPKG is usually used for the binary package combined by multiple images in a certain format, such as the image combination of ‘ap_at_command.bin’ & ‘cp-demo-flash.bin’.

2) AP is only used for the image of ‘ap_at_command.bin’.

3) CP is only used for the image of ‘cp-demo-flash.bin’.

4) APP is reserved for the customized image, which is not used now!

6. Terms

Table 6-1. Terms and Interpretation

Terms	Interpretation
diff	differential
par	patch archive

7. Version

Version	Date	Author	Comments
1.0	2022-3-22	Xu.Wang	Initiated by XuWang
1.1	2023-2-20	Xu.Wang	New feature for delivering data to an extensional storage device
1.2	2023-4-19	Xu.Wang	New feature for reading data out of a specific memory zone
2.0	2023-11-11	Xu.Wang	New feature for HLS and resizing the loading zone