AmebaZ Image Tool User Guide

This document introduces how to use Image Tool to encrypt, generate and download Images.

Version 1.0

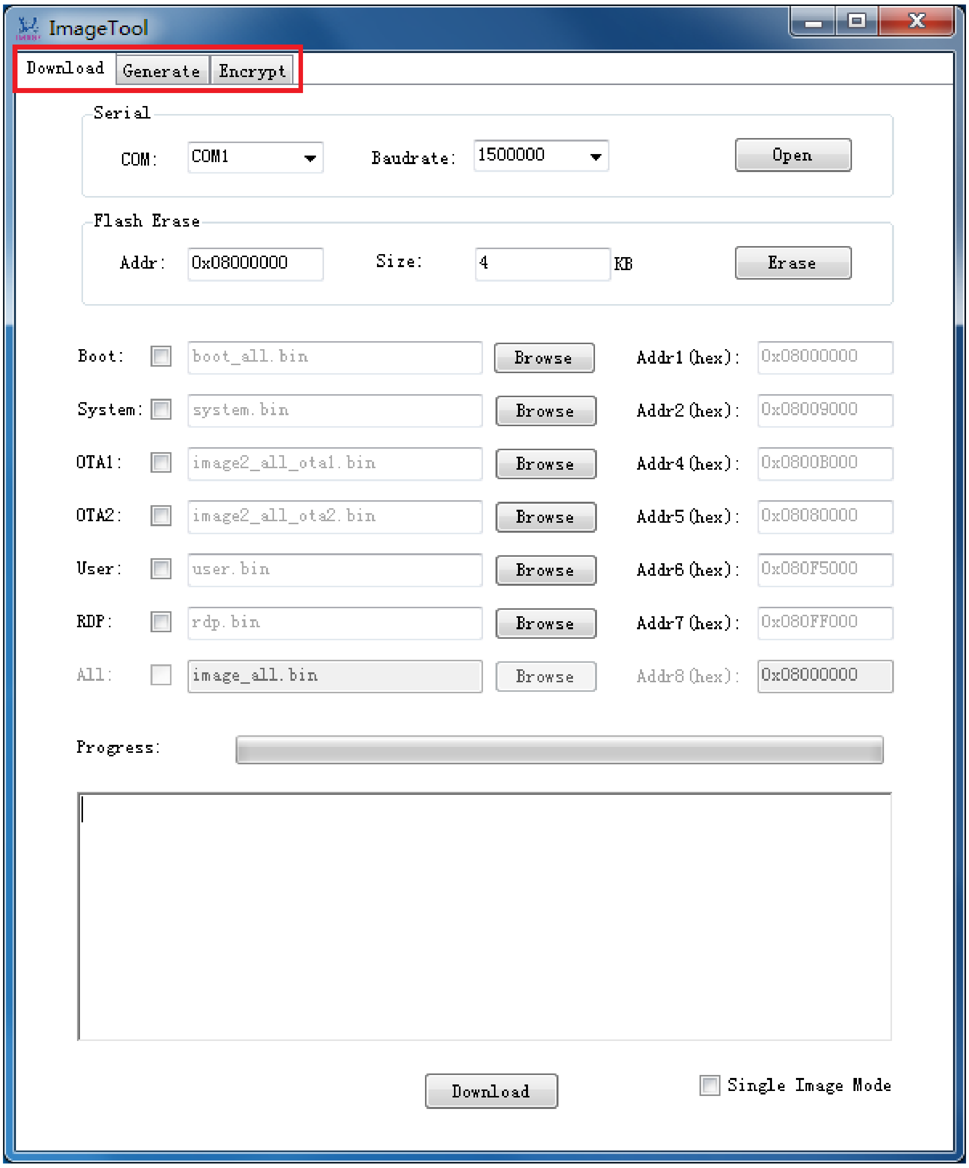
Table of Contents

1. Introduction

This document introduces how to use Image Tool to encrypt, generate and download Images.

As show in the following figure, Image Tool has three tabpages:

- Download: used as image download server to transmit images to AmebaZ through UART
- Generate: concat separate images and generate a final image
- Encrypt: encrypt images which are used for firmware protection



Serial

COM: COM1

Baudrate: 1500000

Flash Erase

Addr: 0x08000000

Size: 4 KB

Boot: □ boot_all.bin Addr1 (hex): 0x08000000

System: □ system.bin Addr2 (hex): 0x08009000

OTA1: □ image2_all_ota1.bin Addr4 (hex): 0x0800B000

OTA2: □ image2_all_ota2.bin Addr5 (hex): 0x08080000

User: □ user.bin Addr6 (hex): 0x080F5000

RDP: □ rdp.bin Addr7 (hex): 0x080FF000

All: □ image_all.bin Addr8 (hex): 0x08000000

Progress:

□ Single Image Mode

Figure 1- 1 Image Tool UI

2. Environment Setup

2.1. Hardware Setup

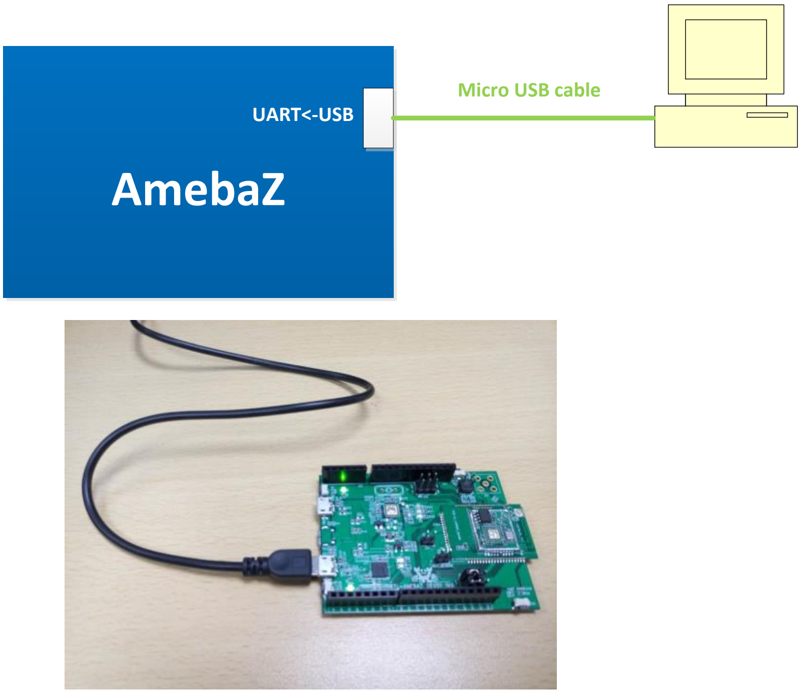
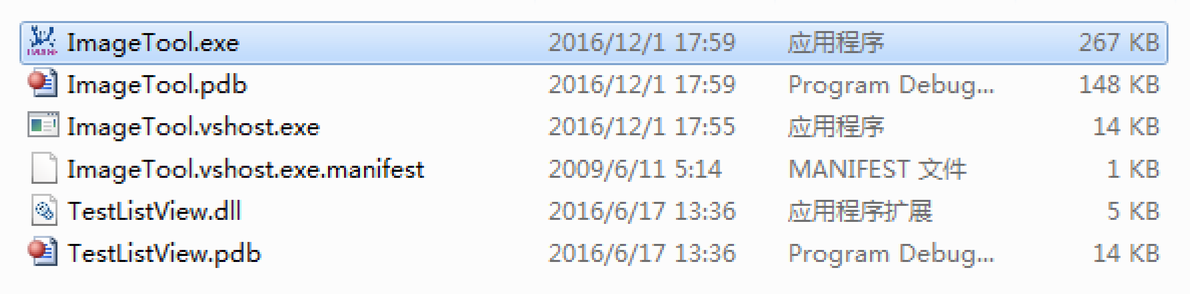


Figure 2- 1 Hardware Setup

2.2. Software Setup

Environment Requirements: EX. WinXP, Win 7 Above **Microsoft .NET Framework 3.5**

ImageTool.exe Location: project\tools\AmebaZ\Image Tool\ImageTool.exe



ImageTool.exe	2016/12/1 17:59	应用程序	267 KB
ImageTool.pdb	2016/12/1 17:59	Program Debug...	148 KB
ImageTool.vshost.exe	2016/12/1 17:55	应用程序	14 KB
ImageTool.vshost.exe.manifest	2009/6/11 5:14	MANIFEST 文件	1 KB
TestListView.dll	2016/6/17 13:36	应用程序扩展	5 KB
TestListView.pdb	2016/6/17 13:36	Program Debug...	14 KB

3. Image Download

3.1. Image Download

Assuming that the Image Tool on PC is Server, which sends images files to AmebaZ(Client) through UART. Client or Server, whichever starts first will be ok.

Steps to start client are as following, which will get AmebaZ into UART_DOWNLOAD mode.

(1) For QFN32, pins marked with “1” on the figure should be connected by jumper cap.
For QFN48 & QFN68, pins marked with “2” should be connected by jumper cap.

(2) Push the Image Download Button and keep it pressed

(3) Power on the board or press the Reset Button. Now AmebaZ get into UART DOWNLOAD mode

(4) Finally Release the Image Download Button. Now the client is ready for receiving data.

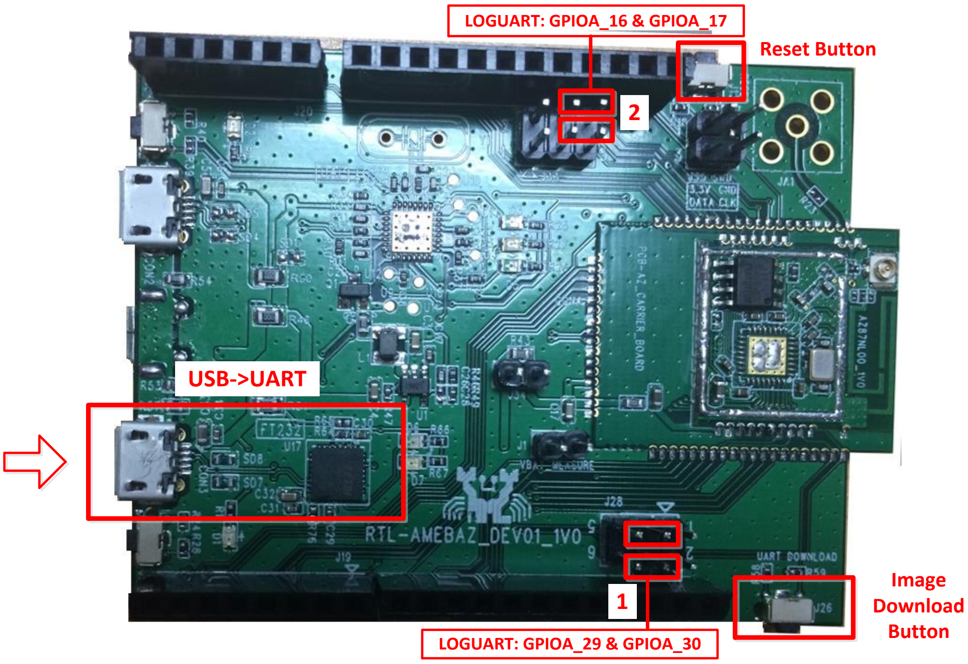


Figure 3- 1 AmebaZ DEV

Steps to start Server are as following:

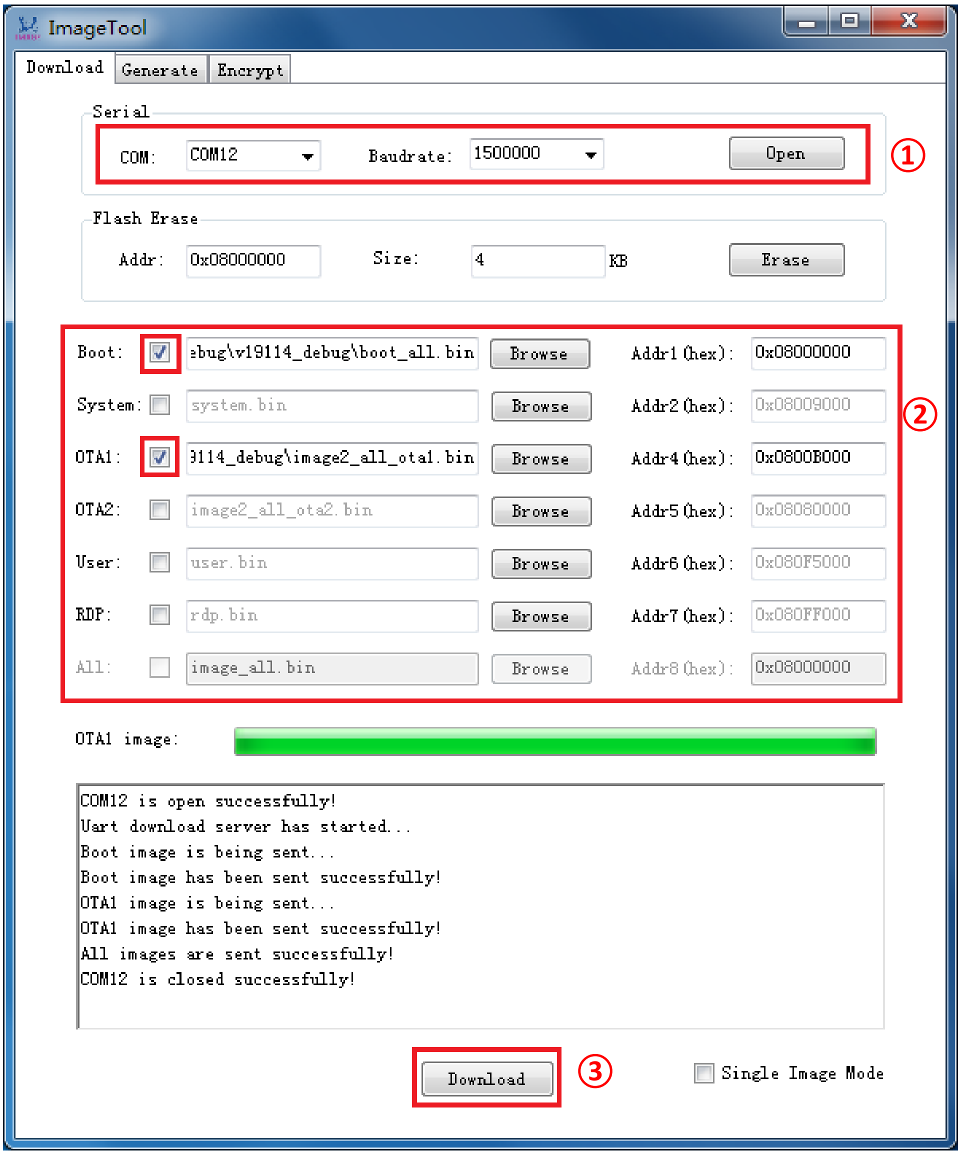
(1) Select serial port and transmission baudrate. The default baudrate is 1.5Mbps.Then open serial port

(2) Select the images to be transferred and input addresses. The address which starts with 0x08 is for flash, and 0x10 for RAM

(3) Click Download button to start. Now the server is ready to send data.

When the client and server are both ready, data transmission begins.

The progress bar will show the transmit progress of each image. You can also get the message of operation successful or errors occur from log window.



COM: COM12

Baudrate: 1500000

Addr: 0x08000000

Size: 4 KB

Boot: ☑ ebug\v19114_debug\boot_all.bin Addr1 (hex): 0x08000000

System: □ system.bin Addr2 (hex): 0x08009000

OTA1: ☑ 9114_debug\image2_all_ota1.bin Addr4 (hex): 0x0800B000

OTA2: □ image2_all_ota2.bin Addr5 (hex): 0x08080000

User: □ user.bin Addr6 (hex): 0x080F5000

RDP: □ rdp.bin Addr7 (hex): 0x080FF000

All: □ image_all.bin Addr8 (hex): 0x08000000

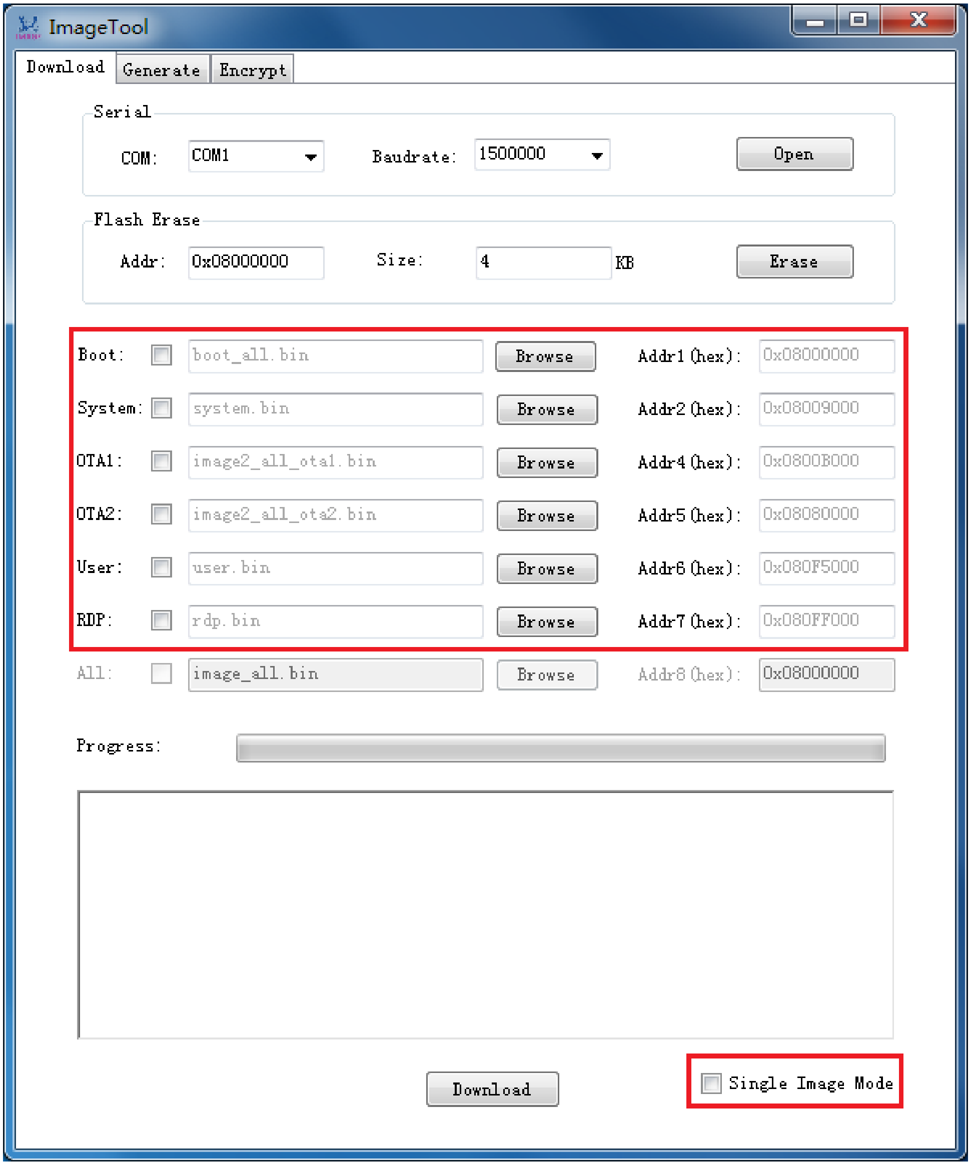
```
COM12 is open successfully!
Uart download server has started...
Boot image is being sent...
Boot image has been sent successfully!
OTA1 image is being sent...
OTA1 image has been sent successfully!
All images are sent successfully!
COM12 is closed successfully!
```

□ Single Image Mode

Figure 3- 2 Image Download Tabpage

3.2. Single Image Mode

- Normal Mode
  - Default mode
  - For development use
  - Any one of the images or combination of them can be transferred at a time.



Serial

COM: COM1 Baudrate: 1500000

Flash Erase

Addr: 0x08000000 Size: 4 KB

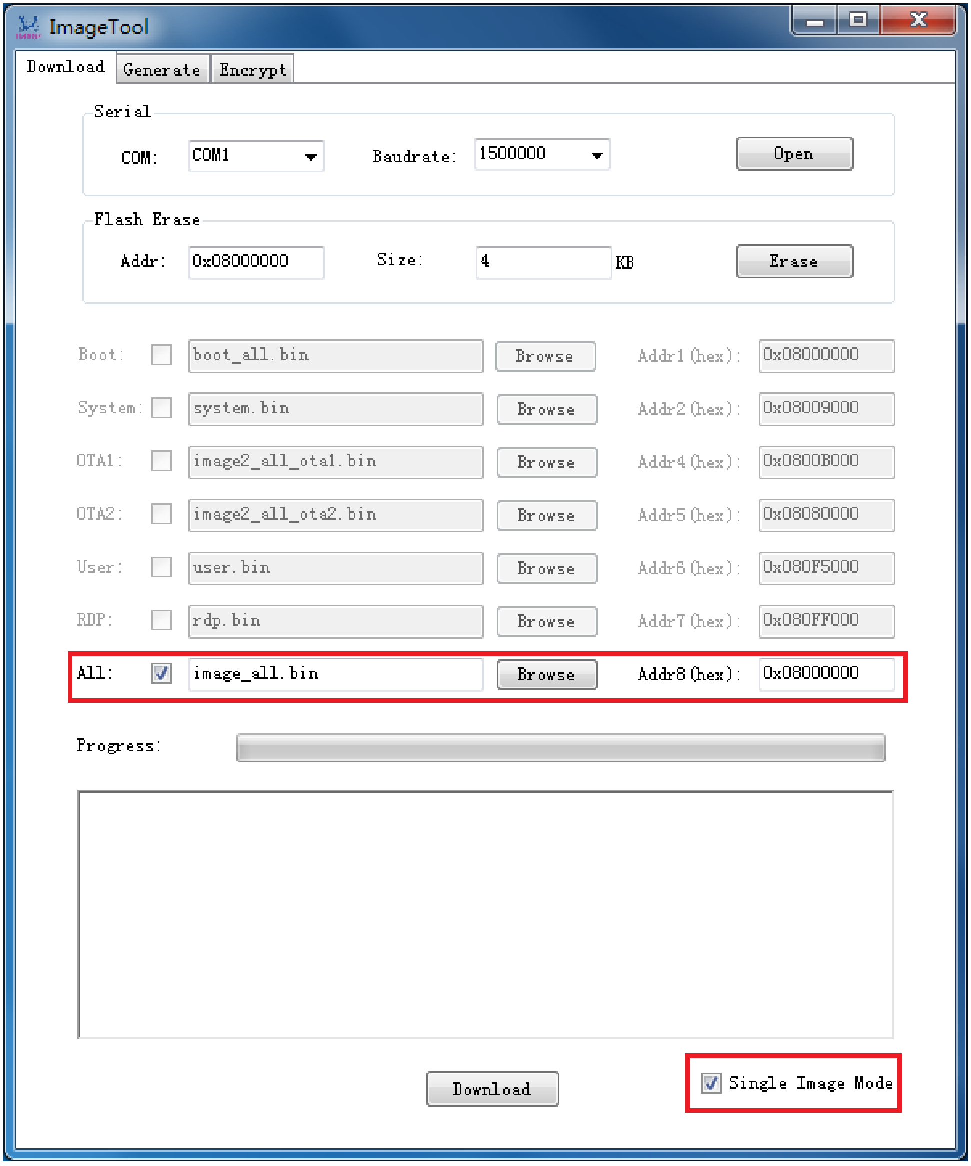
Boot:	□	boot_all.bin	Addr1(hex):	0x08000000
System:	□	system.bin	Addr2(hex):	0x08009000
OTA1:	□	image2_all_ota1.bin	Addr4(hex):	0x0800B000
OTA2:	□	image2_all_ota2.bin	Addr5(hex):	0x08080000
User:	□	user.bin	Addr6(hex):	0x080F5000
RDP:	□	rdp.bin	Addr7(hex):	0x080FF000
All:	□	image_all.bin	Addr8(hex):	0x08000000

Progress:

□ Single Image Mode

Figure 3-3 Normal Mode

- Single Image Mode:
  - For factory use
  - Only one image file can be transferred at a time
  - Besides, other image transmission bars are disable



Serial

COM: COM1 Baudrate: 1500000

Flash Erase

Addr: 0x08000000 Size: 4 KB

Boot: □ boot_all.bin Addr1 (hex): 0x08000000

System: □ system.bin Addr2 (hex): 0x08009000

OTA1: □ image2_all_ota1.bin Addr4 (hex): 0x0800B000

OTA2: □ image2_all_ota2.bin Addr5 (hex): 0x08080000

User: □ user.bin Addr6 (hex): 0x080F5000

RDP: □ rdp.bin Addr7 (hex): 0x080FF000

All: ☑ image_all.bin Addr8 (hex): 0x08000000

Progress:

☑ Single Image Mode

Figure 3- 4 Single Image Mode

3.3. Flash Erase

Steps to erase flash are as following:

(1) Start client. Steps are already mentioned in 3.1. Make sure that AmebaZ has entered UART_DOWNLOAD mode.

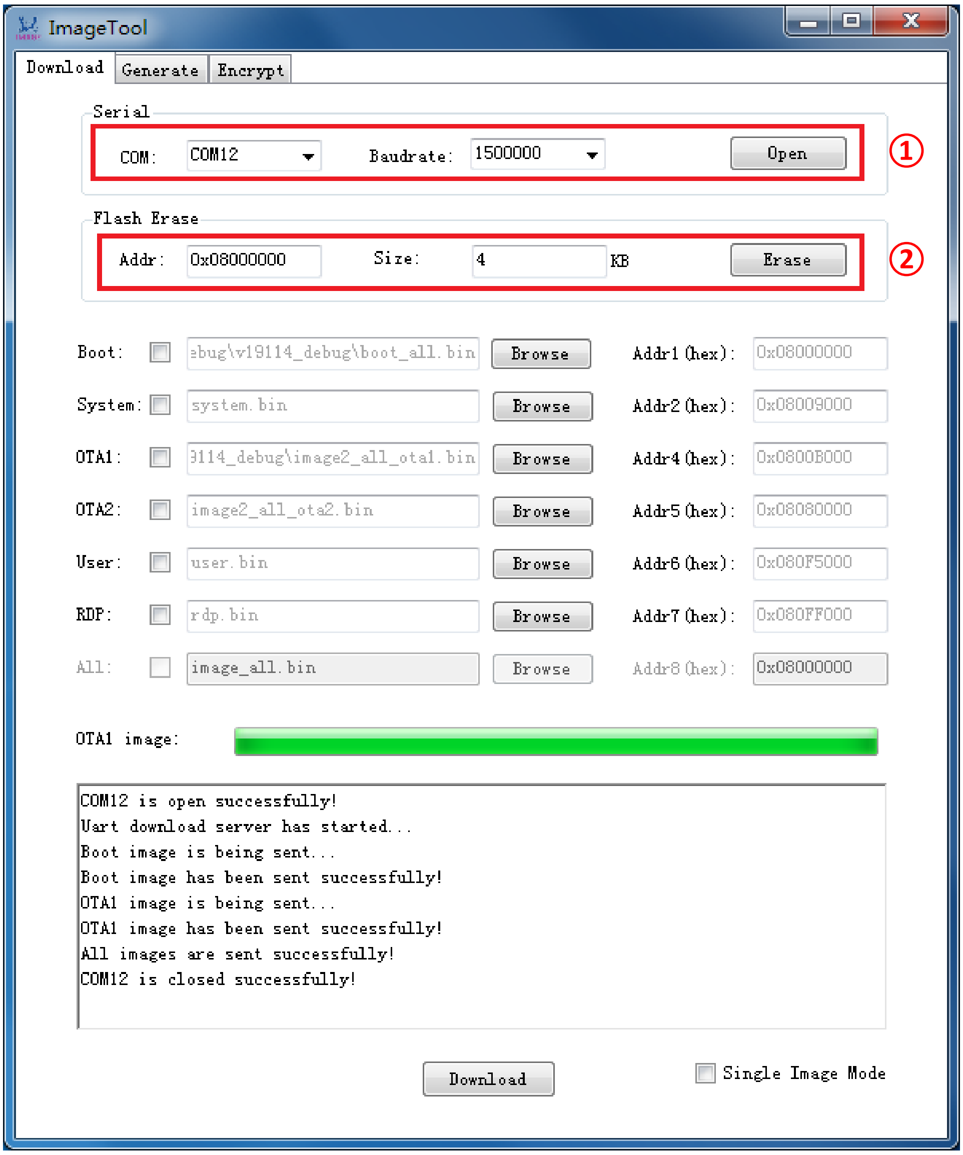
(2) Select serial port & baudrate, then open it

(3) Input erase start address which has to be 4-byte aligned

(4) Input erase size which will be cast to a multiple of 4KB

(5) Click Erase button

(6) You will get the message of operation from log window



COM: COM12 Baudrate: 1500000

Addr: 0x08000000 Size: 4 KB

Boot: □ ebug\v19114_debug\boot_all.bin Addr1 (hex): 0x08000000

System: □ system.bin Addr2 (hex): 0x08009000

OTA1: □ 3114_debug\image2_all_ota1.bin Addr4 (hex): 0x0800B000

OTA2: □ image2_all_ota2.bin Addr5 (hex): 0x08080000

User: □ user.bin Addr6 (hex): 0x080F5000

RDP: □ rdp.bin Addr7 (hex): 0x080FF000

All: □ image_all.bin Addr8 (hex): 0x08000000

```
COM12 is open successfully!
Uart download server has started...
Boot image is being sent...
Boot image has been sent successfully!
OTA1 image is being sent...
OTA1 image has been sent successfully!
All images are sent successfully!
COM12 is closed successfully!
```

Figure 3- 5 Flash Erase

4. Image Generation

The Generate tabpage has two functions:

- Concat separate images and generate a final image named Image_All.bin
- Concat OTA1 and OTA2 images and generate a cloud OTA image named OTA_All.bin

4.1. Image All Generation

Steps to generate a final image are as follows:

(1) Select Image_All as Generate Target type, just as marked with a red box in the following figure.

(2) Select images to be concated and input corresponding address. The Memory Layout bar will show the relative positions of the selected images. If the contiguous images overlap, the overlapped area is in red color for warnning.

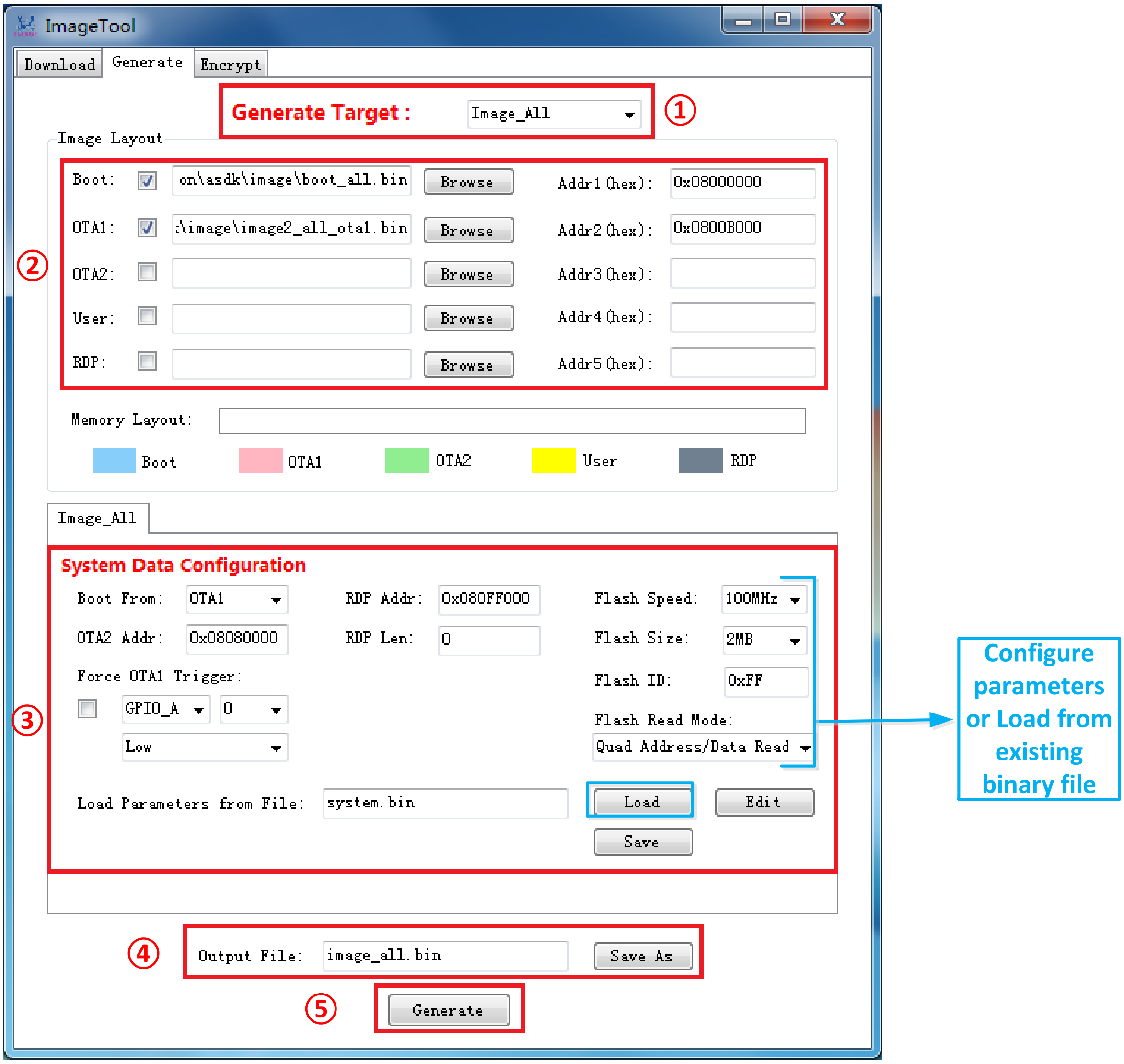
(3) Configure boot and flash initialization options in System Data. Or you can also load an existing binary file by clicking Browse and Load button.By the way, system data can be saved to system.bin in tool directory after click Save button.

You will see the introduction of each configuration in system data in 4.1.1.

(4) Specify output file name and path.

(5) Click Generate button.

(6) After the operation is done, the final image named *image_all.bin* is generated in the tool directory by default or in specified directory.



Generate Target : Image_All

Image Layout

Boot: ☑ on\asdk\image\boot_all.bin Addr1 (hex): 0x08000000

OTA1: ☑ :\image\image2_all_ota1.bin Addr2 (hex): 0x0800B000

OTA2: □ Addr3 (hex):

User: □ Addr4 (hex):

RDP: □ Addr5 (hex):

Memory Layout:

🟦 Boot 🟥 OTA1 🟩 OTA2 🟨 User ⬛ RDP

Image_All

System Data Configuration

Boot From: OTA1

OTA2 Addr: 0x08080000

Force OTA1 Trigger: □ GPIO_A 0 Low

RDP Addr: 0x080FF000

RDP Len: 0

Flash Speed: 100MHz

Flash Size: 2MB

Flash ID: 0xFF

Flash Read Mode: Quad Address/Data Read

Load Parameters from File: system.bin

Output File: image_all.bin

Configure parameters or Load from existing binary file

Figure 4- 1 Image All Generation

4.1.1. System Data

- Boot From
  - Select OTA1 or OTA2 as Image2
- OTA2 Addr
  - Flash address for OTA2.
  - If 0xFFFFFFFF, OTA2 is invalid and select OTA1
- Force OTA1 Trigger

- GPIO force OTA1 as image2.
- This function has a higher priority than "Boot From" on Image2 selection.

■ RDP Addr

- Flash address for RDP

■ RDP Len

- RDP image length, 16 bytes alignment.
- Not include checksum 4 bytes.

■ Flash Speed

- The frequency of SPIC clock.

■ Flash Size

- The capacity of flash chip.

■ Flash ID

- Reserved, use it when flash ID cannot get from flash ID cmd.

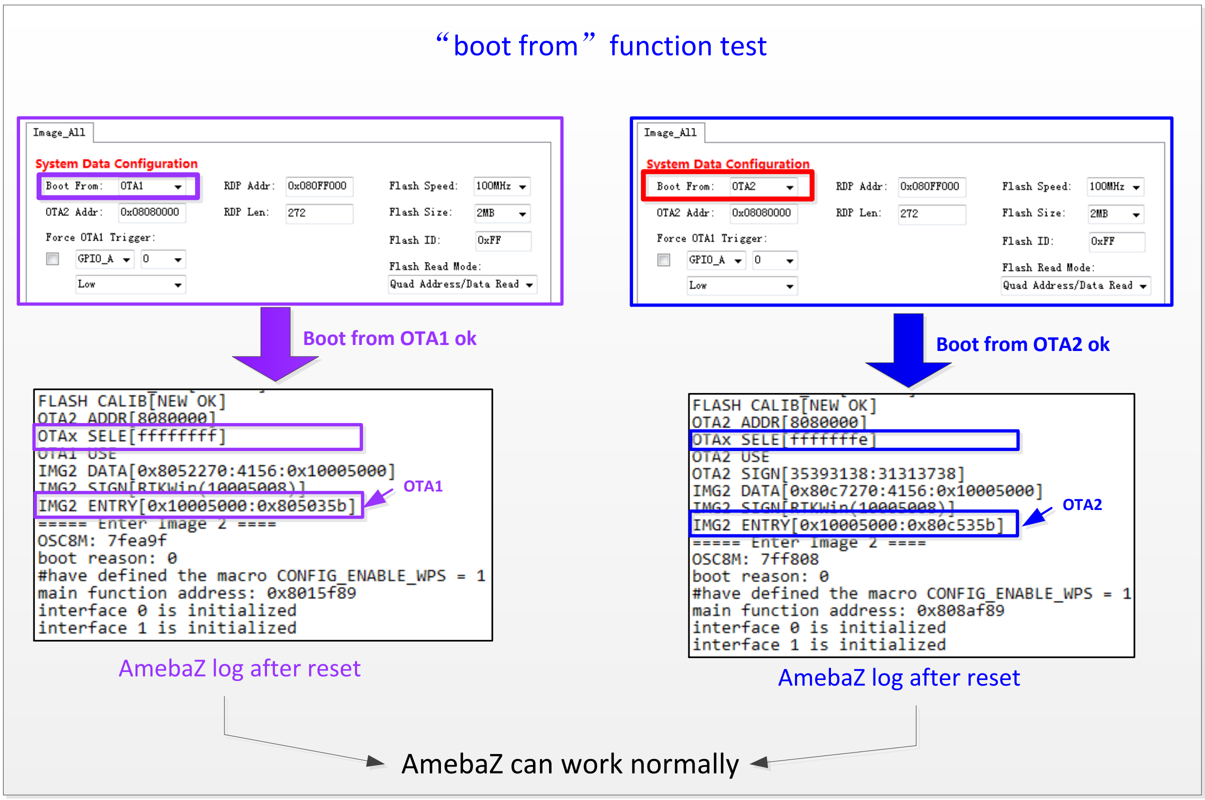
■ Flash Read Mode

- Specify flash read mode, quad IO/quad O/dual IO/dual O/1 Bit mode.

This is an example of how the "Boot From" and "Force OTA1 Trigger" functions effect on OTA image selection.

(1) "Boot From" function:

In this example, "Force OTA1 Trigger" is not enabled and only "Boot From" effects.



Boot From: OTA1
OTA2 Addr: 0x08080000
Force OTA1 Trigger: □ GPIO_A 0 Low
RDP Addr: 0x080FF000
RDP Len: 272
Flash Speed: 100MHz
Flash Size: 2MB
Flash ID: 0xFF
Flash Read Mode: Quad Address/Data Read

```
FLASH CALIB[NEW OK]
OTA2 ADDR[8080000]
OTAx SELE[ffffffff]
OTA1 USE
IMG2 DATA[0x8052270:4156:0x10005000]
IMG2 SIGN[RTKWin(10005008)]
IMG2 ENTRY[0x10005000:0x805035b]
===== Enter Image 2 ====
OSC8M: 7fea9f
boot reason: 0
#have defined the macro CONFIG_ENABLE_WPS = 1
main function address: 0x8015f89
interface 0 is initialized
interface 1 is initialized
```

Boot From: OTA2
OTA2 Addr: 0x08080000
Force OTA1 Trigger: □ GPIO_A 0 Low
RDP Addr: 0x080FF000
RDP Len: 272
Flash Speed: 100MHz
Flash Size: 2MB
Flash ID: 0xFF
Flash Read Mode: Quad Address/Data Read

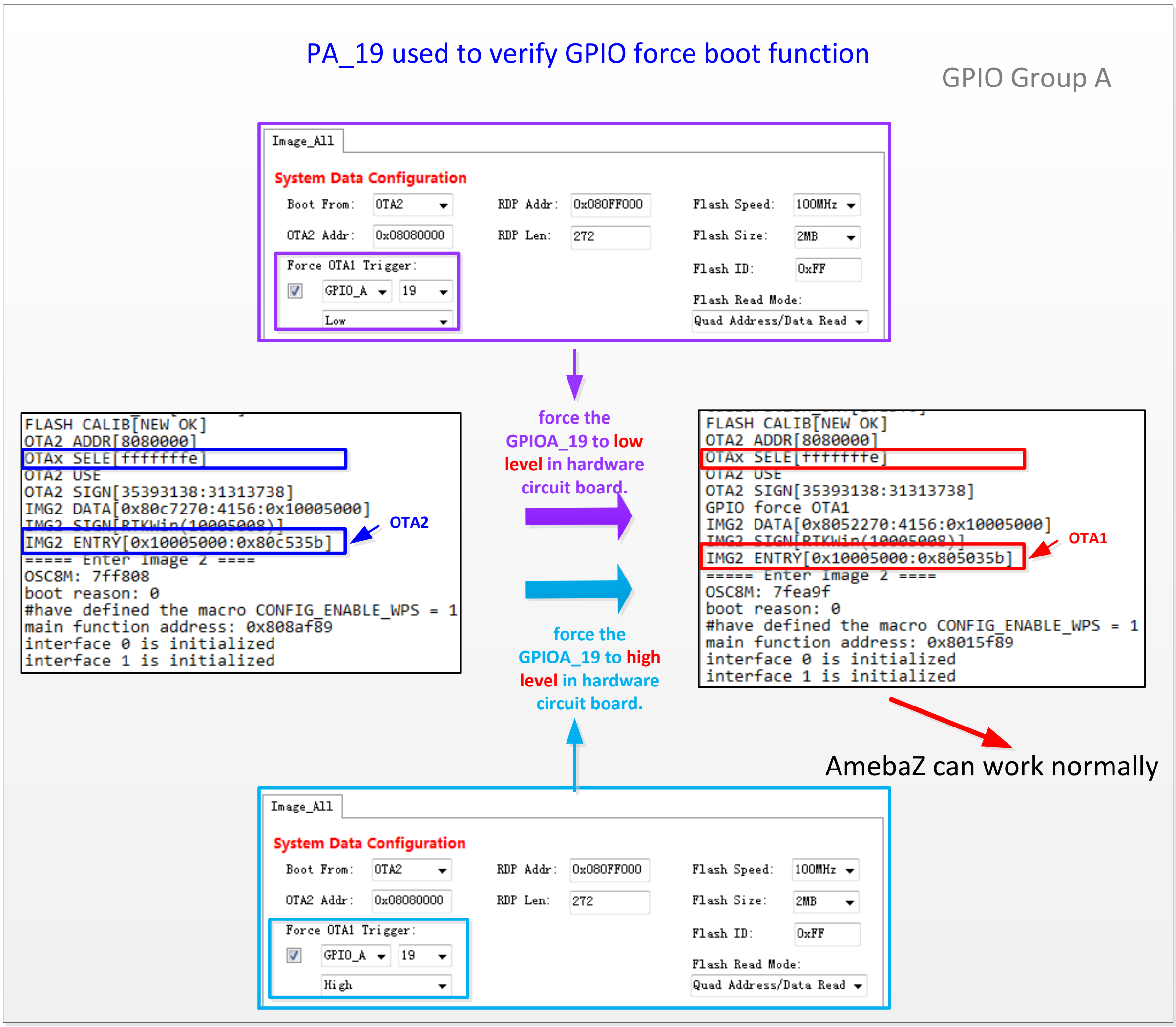
```
FLASH CALIB[NEW OK]
OTA2 ADDR[8080000]
OTAx SELE[fffffffe]
OTA2 USE
OTA2 SIGN[35393138:31313738]
IMG2 DATA[0x80c7270:4156:0x10005000]
IMG2 SIGN[RTKWin(10005008)]
IMG2 ENTRY[0x10005000:0x80c535b]
===== Enter Image 2 ====
OSC8M: 7ff808
boot reason: 0
#have defined the macro CONFIG_ENABLE_WPS = 1
main function address: 0x808af89
interface 0 is initialized
interface 1 is initialized
```

Figure 4- 2 "Boot From" function

(2) "Force OTA1 Trigger" function:

In this example, "Force OTA1 Trigger" is enabled and PA_19 used as trigger pin.

As "Force OTA1 Trigger" has a higher priority than "Boot From" on Image2 selection, no matter "Boot From" is configured OTA1 or OTA2, AmebaZ will boot to OTA1 if the level of trigger pin matches.


PA_19 used to verify GPIO force boot function
GPIO Group A
force the GPIOA_19 to low level in hardware circuit board.
force the GPIOA_19 to high level in hardware circuit board.
OTA2
OTA1
AmebaZ can work normally

Image_All

System Data Configuration

Boot From: OTA2 | RDP Addr: 0x080FF000 | Flash Speed: 100MHz
OTA2 Addr: 0x08080000 | RDP Len: 272 | Flash Size: 2MB
Force OTA1 Trigger: ☑ GPIO_A 19 Low | Flash ID: 0xFF
Flash Read Mode: Quad Address/Data Read

```
FLASH CALIB[NEW OK]
OTA2 ADDR[8080000]
OTAx SELE[fffffffe]
OTA2 USE
OTA2 SIGN[35393138:31313738]
IMG2 DATA[0x80c7270:4156:0x10005000]
IMG2 SIGN[RTKWin(10005008)]
IMG2 ENTRY[0x10005000:0x80c535b]
===== Enter Image 2 ====
OSC8M: 7ff808
boot reason: 0
#have defined the macro CONFIG_ENABLE_WPS = 1
main function address: 0x808af89
interface 0 is initialized
interface 1 is initialized
```

```
FLASH CALIB[NEW OK]
OTA2 ADDR[8080000]
OTAx SELE[fffffffe]
OTA2 USE
OTA2 SIGN[35393138:31313738]
GPIO force OTA1
IMG2 DATA[0x8052270:4156:0x10005000]
IMG2 SIGN[RTKWin(10005008)]
IMG2 ENTRY[0x10005000:0x805035b]
===== Enter Image 2 ====
OSC8M: 7fea9f
boot reason: 0
#have defined the macro CONFIG_ENABLE_WPS = 1
main function address: 0x8015f89
interface 0 is initialized
interface 1 is initialized
```

Image_All

System Data Configuration

Boot From: OTA2 | RDP Addr: 0x080FF000 | Flash Speed: 100MHz
OTA2 Addr: 0x08080000 | RDP Len: 272 | Flash Size: 2MB
Force OTA1 Trigger: ☑ GPIO_A 19 High | Flash ID: 0xFF
Flash Read Mode: Quad Address/Data Read

Figure 4- 3 "Force OTA1 Trigger" function

4.2. Cloud OTA Image Generation

Steps to generate a Cloud OTA image are as follows:

(1) Select OTA_All as Generate Target type, just as marked with a red box in the following figure.

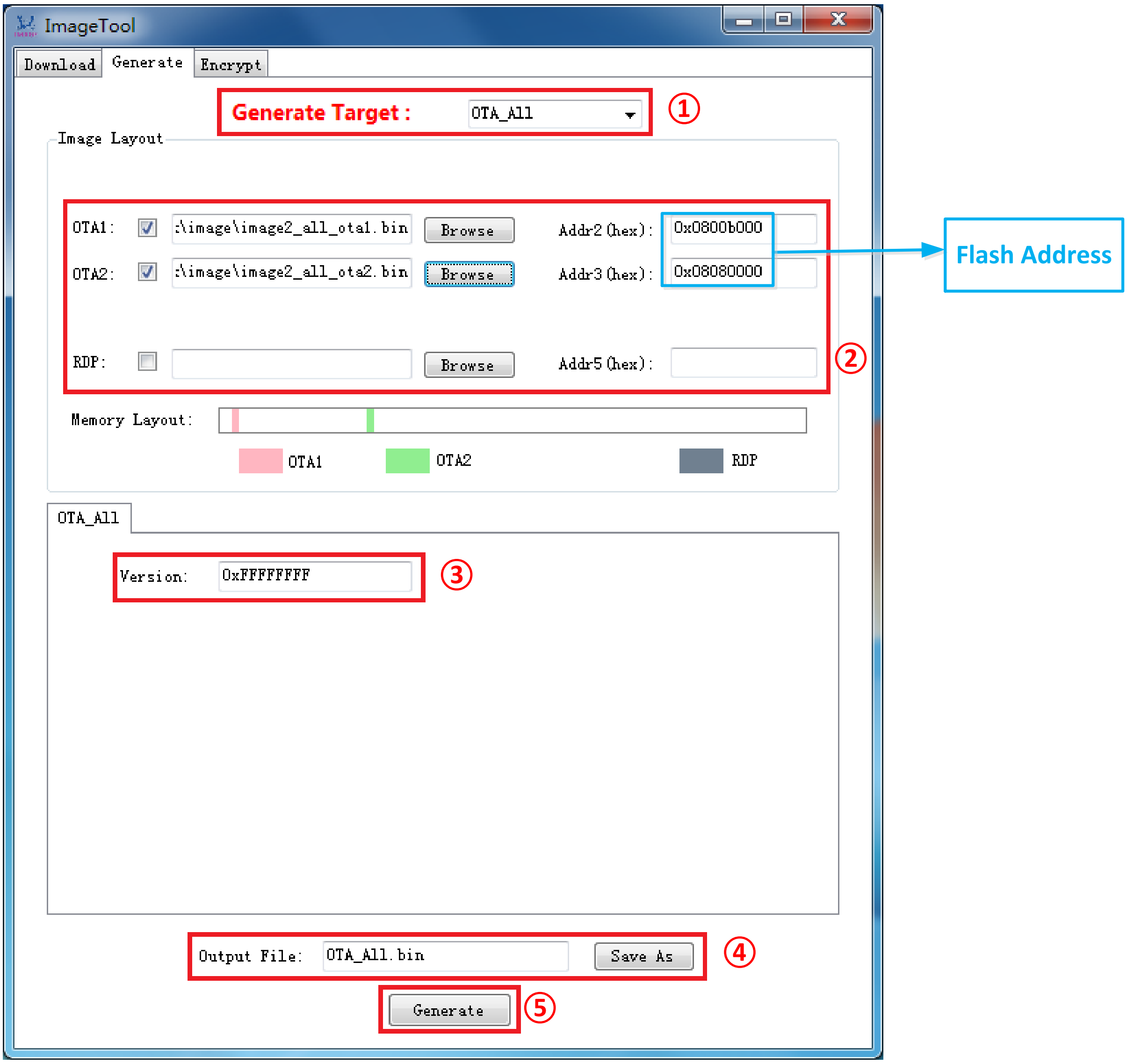
(2) Select images and input corresponding address. The address is flash address where the image is located. The Memory Layout bar will show the relative positions of the two images. If they overlap, the overlapped area is in red color for warnning.

(3) Input Image Version.

(4) Specify output file name and path.

(5) Click Generate button.

(6) After the operation is done, the cloud image named *OTA_All.bin* is generated in the tool directory by default or in specified directory



Generate Target : OTA_All

OTA1: ☑ :\image\image2_all_ota1.bin Addr2 (hex): 0x0800b000

OTA2: ☑ :\image\image2_all_ota2.bin Addr3 (hex): 0x08080000

RDP: □ Addr5 (hex):

Version: 0xFFFFFFFF

Output File: OTA_All.bin

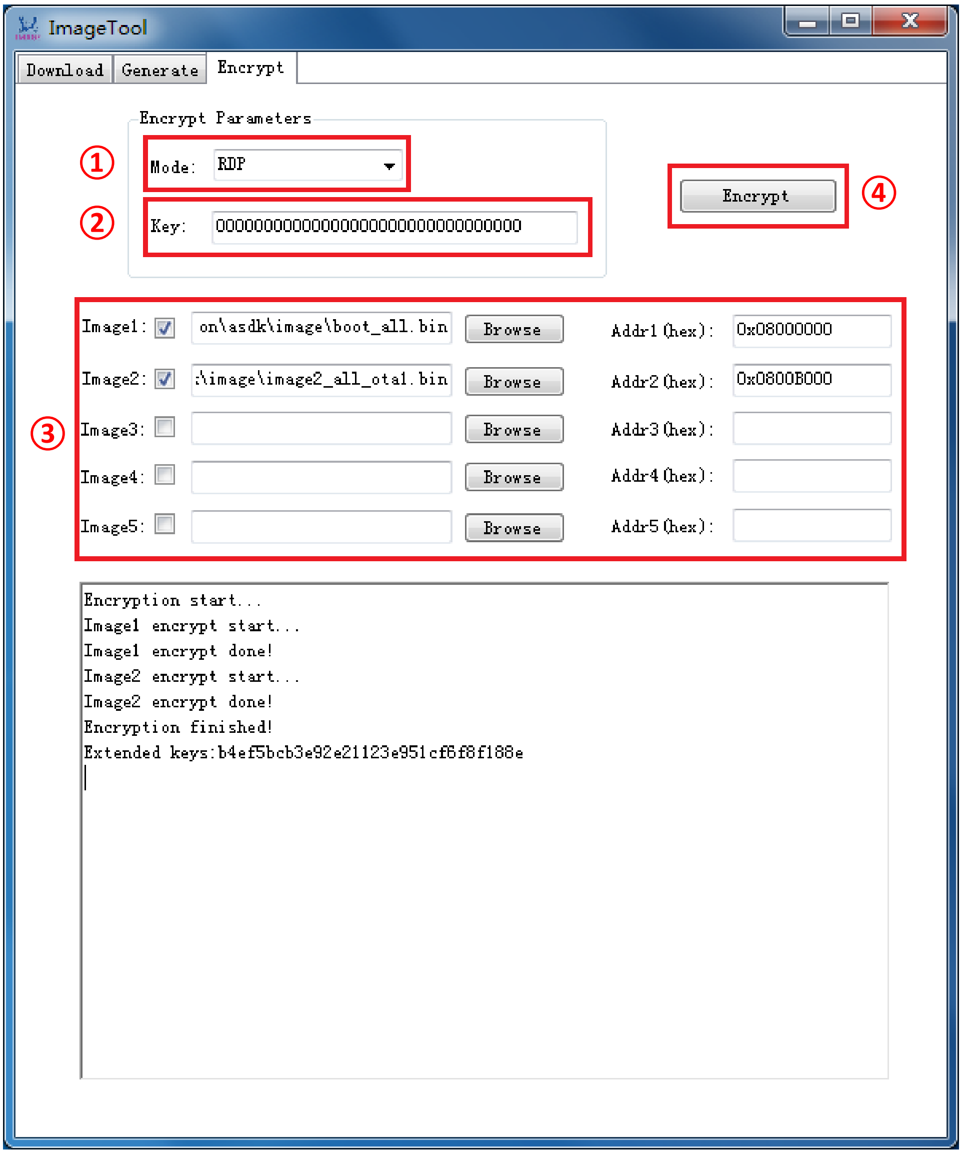
.

Figure 4- 4 Cloud OTA Image Generation

5. Encryption

The Encrypt tabpage is for encrypting images which is used in firmware protection.
Steps to encrypt images are as follows:

(1) Select Encrypt mode
(2) Input Key
(3) Select Images that need to be encrypted and input corresponding address
(4) Click Encrypt button
(5) You will get the message of encryption from log window. The encrypted images are named after the original file with "-en", which located in the same directory.



Mode: RDP

Key: 00000000000000000000000000000000

Image1: ☑ on\asdk\image\boot_all.bin Addr1 (hex): 0x08000000

Image2: ☑ :\image\image2_all_ota1.bin Addr2 (hex): 0x0800B000

Image3: ☐ Addr3 (hex):

Image4: ☐ Addr4 (hex):

Image5: ☐ Addr5 (hex):

```
Encryption start...
Image1 encrypt start...
Image1 encrypt done!
Image2 encrypt start...
Image2 encrypt done!
Encryption finished!
Extended keys:b4ef5bcb3e92e21123e951cf6f8f188e
```

Figure 5- 1 Encrypt Tabpage