



Realtek Over-The-Air Programming

Over-the-air programming (OTA) provides a methodology of updating device firmware remotely via TCP/IP network connection.

Table of Contents

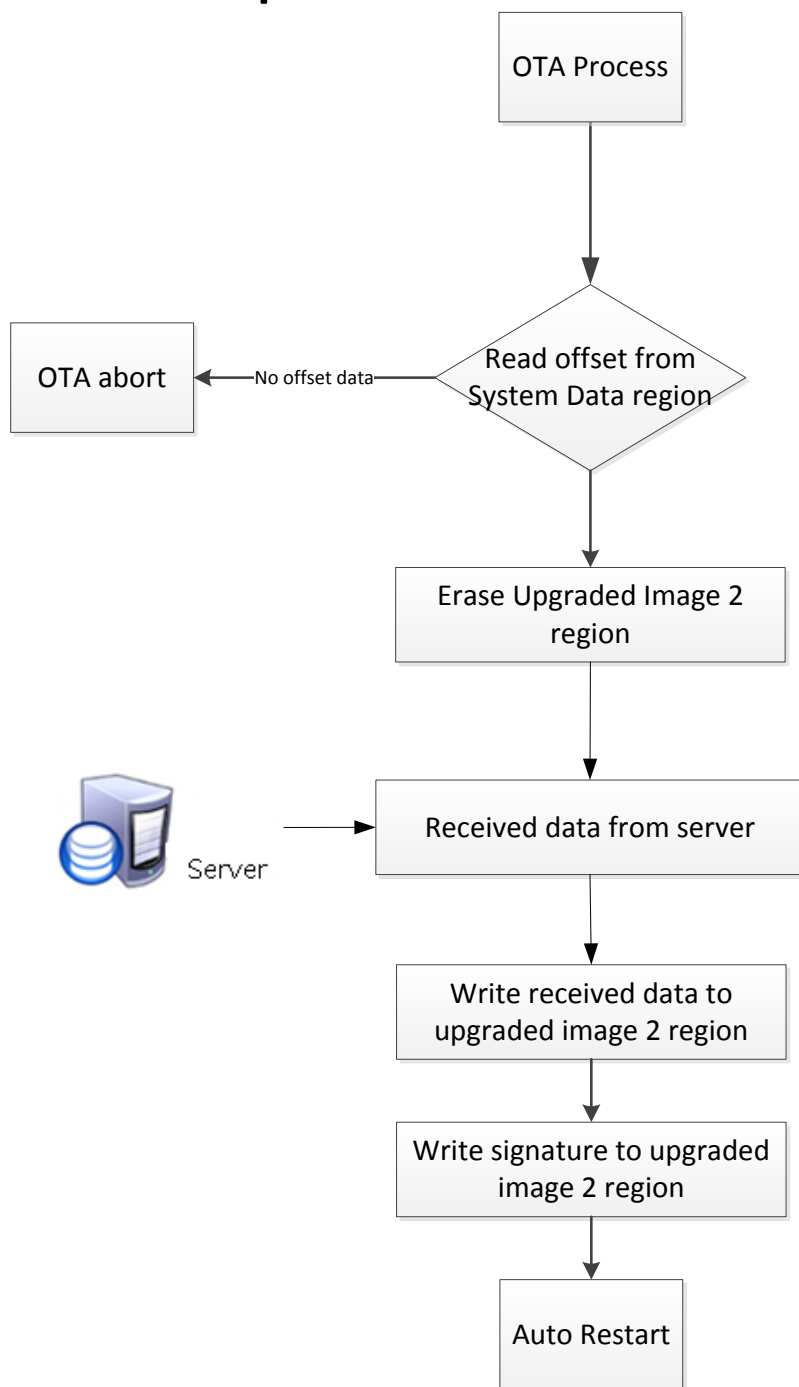
1	Introduction	3
2	OTA overview	4
2.1	OTA operation flow	4
2.2	Boot process flow	5
2.3	Upgraded Partition	6
3	Implement OTA over Wi-Fi	7
3.1	OTA using local download server	7
3.1.1	Build OTA Application image	7
3.1.2	Local download server	10
4	OTA Signature	12

1 Introduction

Over-the-air programming (OTA) provides a methodology of updating device firmware remotely via TCP/IP network connection.

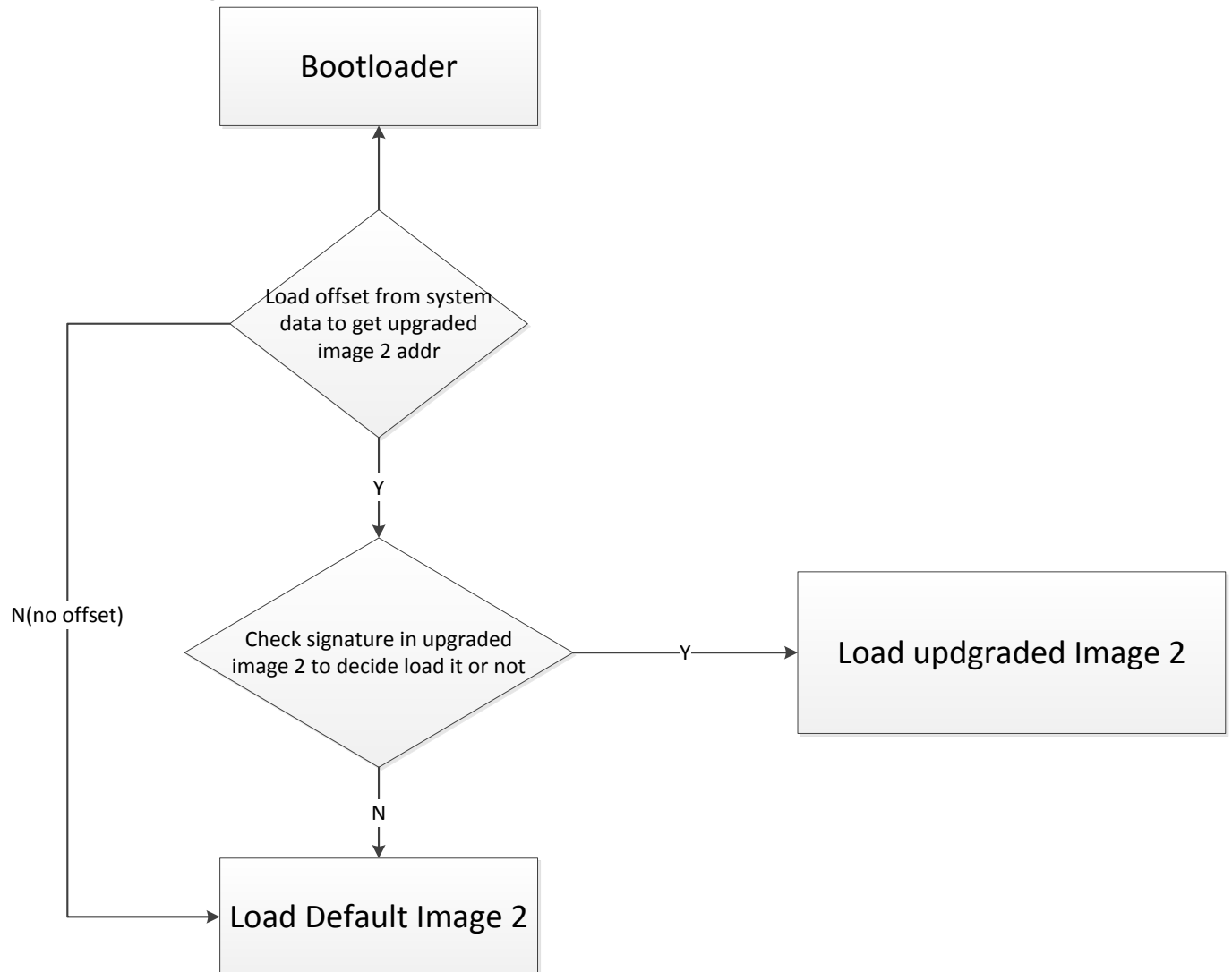
2 OTA overview

2.1 OTA operation flow

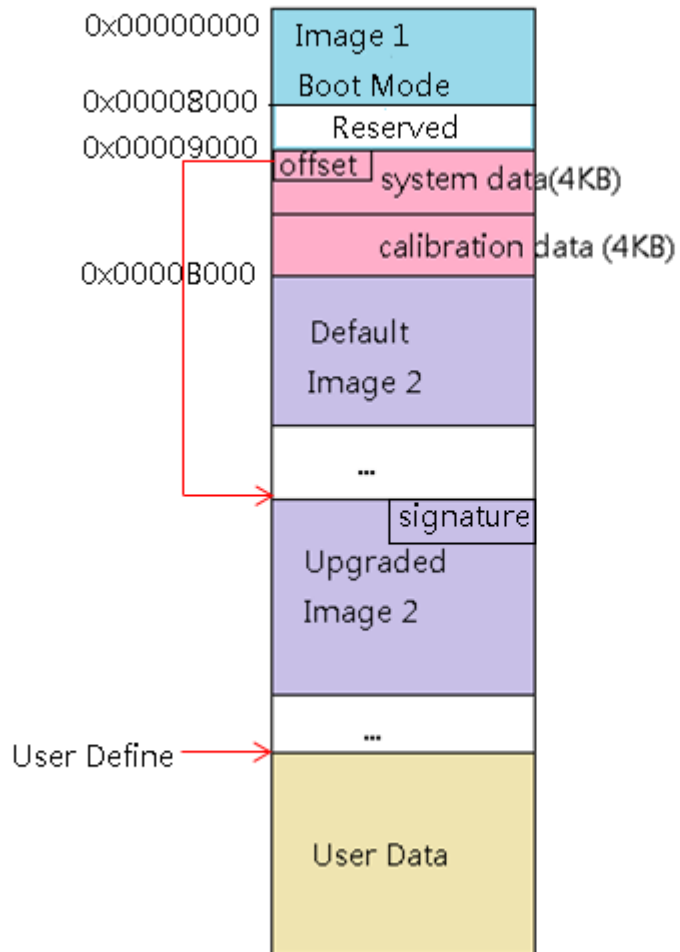


Note: During the step of “Erase Upgraded Image2 region”, the signature is set to 0xffffffff, which is invalid signature.

2.2 Boot process flow



2.3 Upgraded Partition



According to the figure above, only Upgraded Image 2 is updated. The offset of Upgraded Image 2 is get from the system data area locates in the flash section 0x00009000.

3 Implement OTA over Wi-Fi

3.1 OTA using local download server

The example shows how device updates image from a local download server. The local download server send image to device based on network socket.

Make sure both device and PC are connecting to the same local network.

3.1.1 Build OTA Application image

Turn on OTA command

The flag defined in \component\common\api\at_cmd\atcmd_wifi.c

```
//Config in atcmd_wifi.c
#define CONFIG_OTA_UPDATE    1
```

Define server type = SERVER_LOCAL in update.c file (path: tools\DownloadServer\).

```
//Config in update.c
#define SERVER_LOCAL        1
#define SERVER_CLOUD        2
#define SERVER_TYPE          SERVER_LOCAL
```

Write the address of the upgraded image 2 to system data.

Use the following sample code to write the upgraded image 2 address to system data flash section.

Sample code:

```
#include "flash_api.h"
#define WRITE_OAT_ADDR 1

flash_t flash;
//address:0x00080000
Uint32_t ota_addr = 0x00080000;
//boundary check
if((ota_addr > IMAGE_3) && ((ota_addr < (IMAGE_3+Img3Len))) ||
    (ota_addr < IMAGE_3) ||
    ((ota_addr & 0xff) != 0) ||
    (ota_addr == ~0x0)){
    printf("\n\r[%s] illegal ota addr 0x%x", __FUNCTION__, ota_addr);
    goto update_ota_exit;
}else
    write_ota_addr_to_system_data(&flash, ota_addr);
```

Read upgraded image 2 address from system data and verify this address

```
//Config in update.c
static void update_ota_local_task(void *param)
{
    ...
    //Get upgraded image 2 addr from offset
    flash_read_word(&flash, OFFSET_DATA, &NewImg2Addr);
    flash_read_word(&flash, IMAGE_2, &Img2Len);
    if((NewImg2Addr > IMAGE_3) && ((NewImg2Addr < (IMAGE_3+Img3Len))) ||
        (NewImg2Addr < IMAGE_3) ||
        ((NewImg2Addr & 0xff) != 0) ||
        (NewImg2Addr == ~0x0))
        goto update_ota_exit;
```


The address of OFFSET_DATA is 0x9000, and the address of upgraded image 2 is the first 4 byte from this address. If the address was not qualified, then the ota process will be stopped.




Define custom signature

```
//Config in update.c
1. turn on the marco as follows:
#define CONFIG_CUSTOM_SIGNATURE 1
2. Define your own signature.

#if CONFIG_CUSTOM_SIGNATURE
/* -----
 * Customized Signature
 * -----*/
// This signature can be used to verify the correctness of the image
// It will be located in fixed location in application image
#pragma location=".custom.validate.rodata"
const unsigned char cus_sig[32] = "Customer Signature-modelxxx";
#endif
3. compare it while complete flashing.
static void update_ota_local_task(void *param)
{
...
#if CONFIG_CUSTOM_SIGNATURE
    && !strcmp(read_custom_sig,custom_sig)
#endif
...
}
```

3.1.2 Local download server

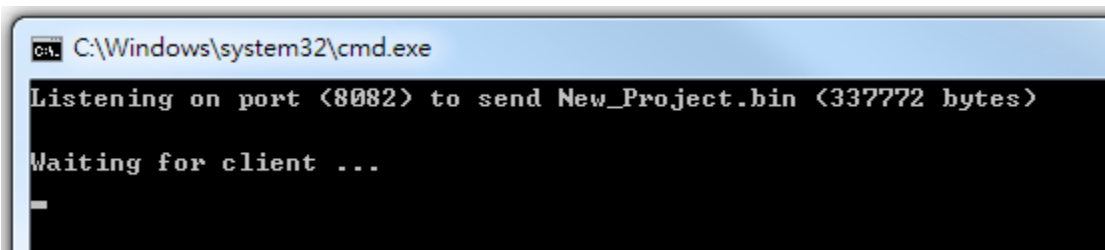
Build new image New_Project.bin in DownloadServer folder (path: tools\DownloadServer\).

	DownloadServer.exe	2014/6/13 ...	85 KB
	New_Project.bin	2014/8/13 ...	330 KB
	start.bat	2014/8/13 ...	1 KB

Edit start.bat file. Port = 8082, file = New_Project.bin

```
1 @echo off
2 DownloadServer 8082 New_Project.bin
3 set /p DUMMY=Press Enter to Continue ...
```

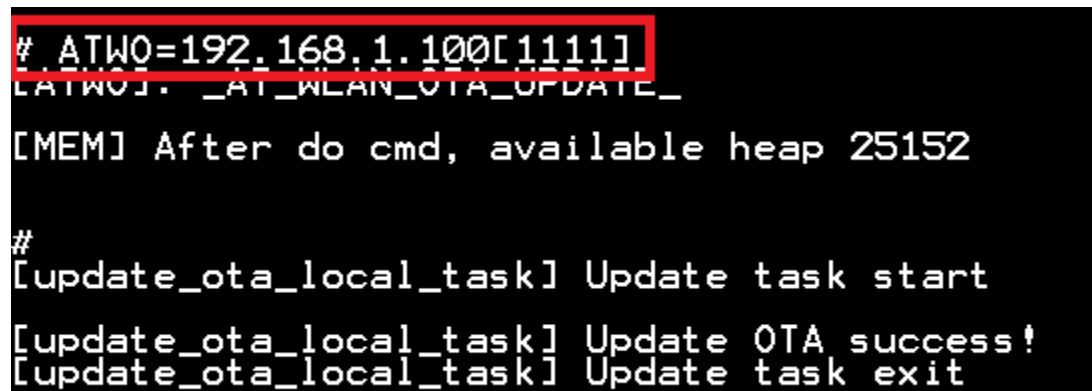
Execute start.bat



```
C:\Windows\system32\cmd.exe
Listening on port <8082> to send New_Project.bin <337772 bytes>
Waiting for client ...
```

Reboot device and connect to AP.

Enter command: ATW0=IP[PORT].



```
# ATW0=192.168.1.100[1111]
ATW0]. _AT_WLAN_OTA_UPDATE_
[MEM] After do cmd, available heap 25152
#
[update_ota_local_task] Update task start
[update_ota_local_task] Update OTA success!
[update_ota_local_task] Update task exit
```

Local download server success message:

A screenshot of a Windows Command Prompt window titled "C:\Windows\system32\cmd.exe". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The background is black with white text. The output shows a netcat listener on port 8082 waiting for a client. It successfully accepts a connection from 192.168.1.101 and sends a file named New_Project.bin (337772 bytes). The transmission is indicated by several lines of dots. Finally, it reports "Total send 337772 bytes" and returns to the "Waiting for client ..." state.

```
C:\Windows\system32\cmd.exe
```

```
Listening on port <8082> to send New_Project.bin <337772 bytes>
```

```
Waiting for client ...
```

```
Accept client connection from 192.168.1.101
```

```
Sending .....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
Total send 337772 bytes
```

```
Waiting for client ...
```

After finishing downloading image, device will be auto-rebooted, and the bootloader will load new image 2 if it exist.

4 OTA Signature

To Clear or Recover OTA signature for verification via UART at command, please refer to AN0025.