Realtek Ameba 1 Firmware Image

This document introduces firmware image output and how to update image over the air.

Table of Contents

1 Introduction 3
2 OTA overview 3
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.

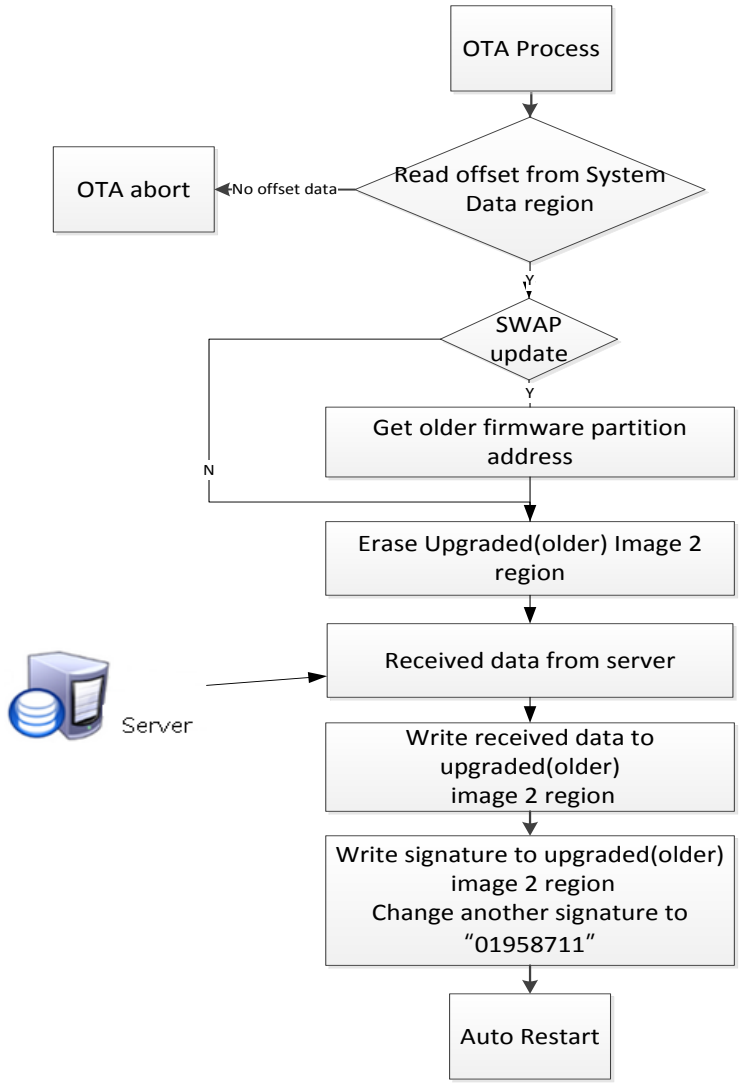
2 Firmware Image Output

After building project source files, there are 2 files will be generated automatically. The first is *ram_all.bin* that is containing boot loader and application image. And the second is *ota.bin* that is application only image. Those two image can be found at *SDK_folder/project/project_name/EWARM-RELEASE/Debug/Exe.*

3 Firmware Update Over the Air

3.1 Overview

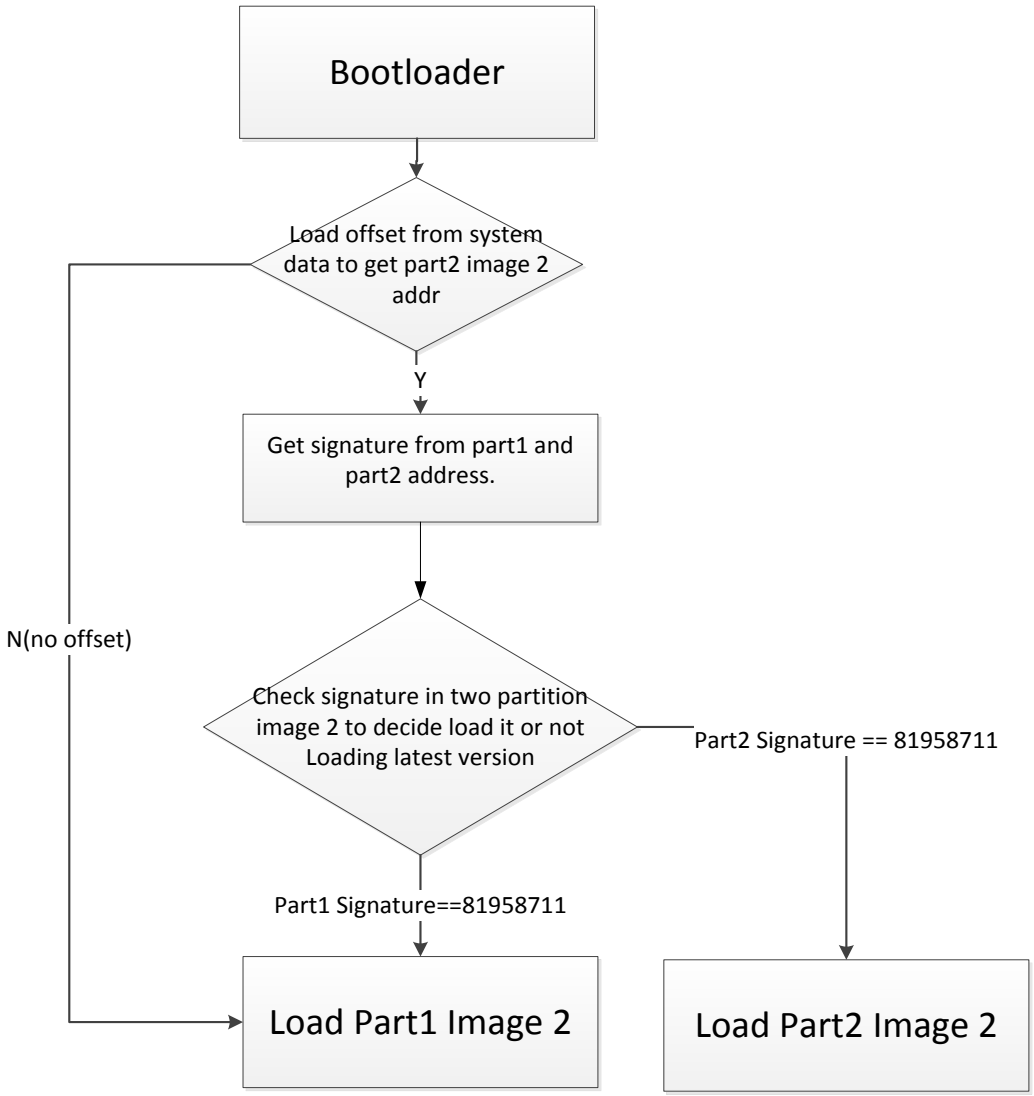
3.1.1 OTA operation flow



Note: During the step of "Erase Upgraded (Older) Image2 region", the signature is set to 0xffffffff, which is invalid signature.

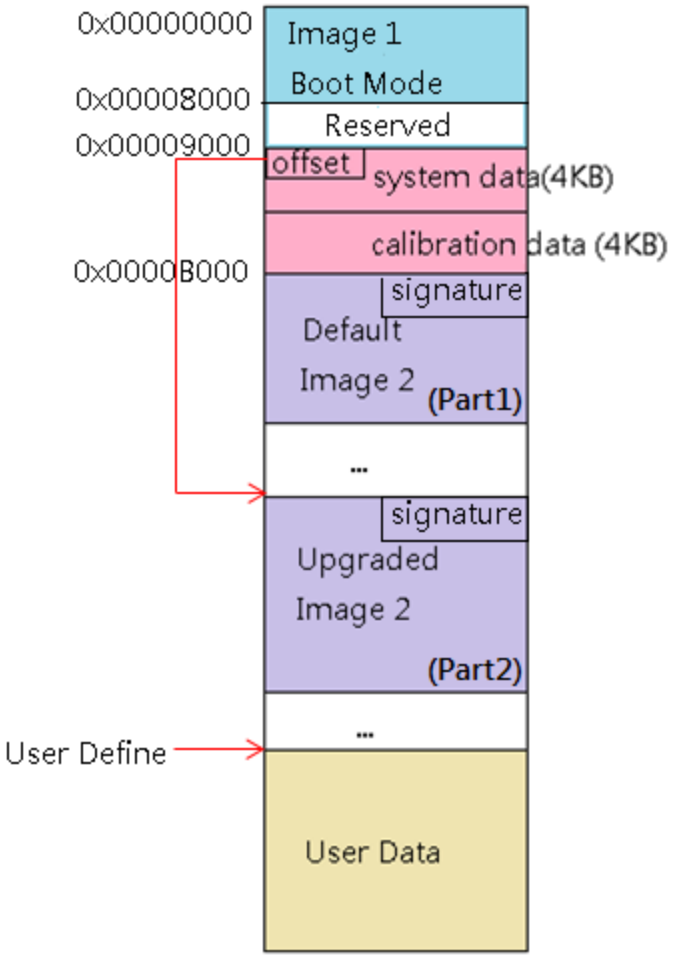
Note: OTA updater will change signature of another region from "81958711" to "01958711" when writing signature. Firmware with signature "01958711" is previous version, and with "81958711" is latest updated version.

3.1.2 Boot process flow



Boot loader will select latest (signature == “81958711”) updated image2 and load it to SRAM.

3.1.3 Upgraded Partition



In most case, we suggest only updating Upgraded Image 2.

Default Image2 can be updated if set SWAP_UPDATE in update.c. This behavior may damage another region data, please use this at your own risk.

NOTE: Signature “81958711” is mean latest updated version. “01958711” is previous version. Boot loader will load latest version by default.

3.2 Implement OTA over Wi-Fi

3.2.1 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.

3.2.2 Build OTA Application image

Turn on OTA command

The flag defined in \project\realtek_ameba1_va0_example\inc\platform_opts.h

```
//Config in platform_opts.h
#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
```

Define SWAP_UDPATE in update.c file

```
//Config in update.c
#define SWAP_UPDATE 1
```

Enable this will update OTA image to Default (Part1) Image2 region. This behavior may damage another region data, please use this at your own risk.

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 & 0xfff) != 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 & 0xfff) != 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.

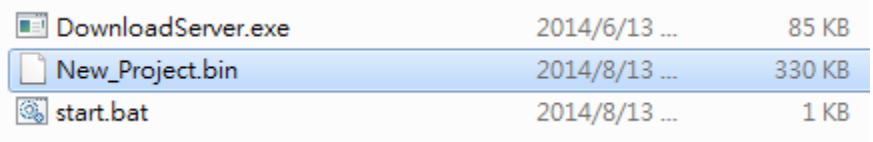
Define custom signature

```
//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
…
}
```

3.2.3 Local download server

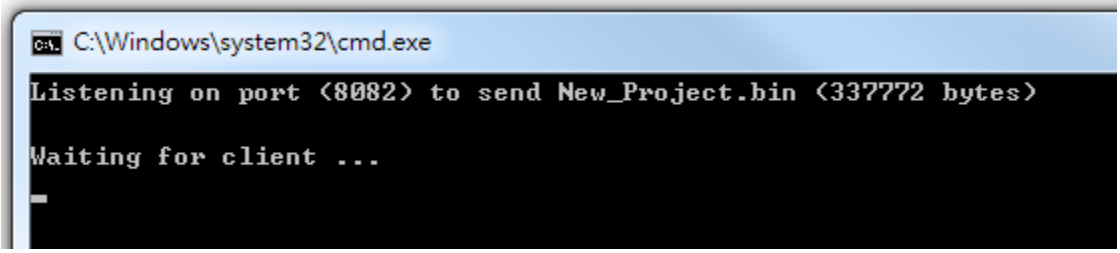
Build new image New_Project.bin in DownloadServer folder (path: tools\DownloadServer\).



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

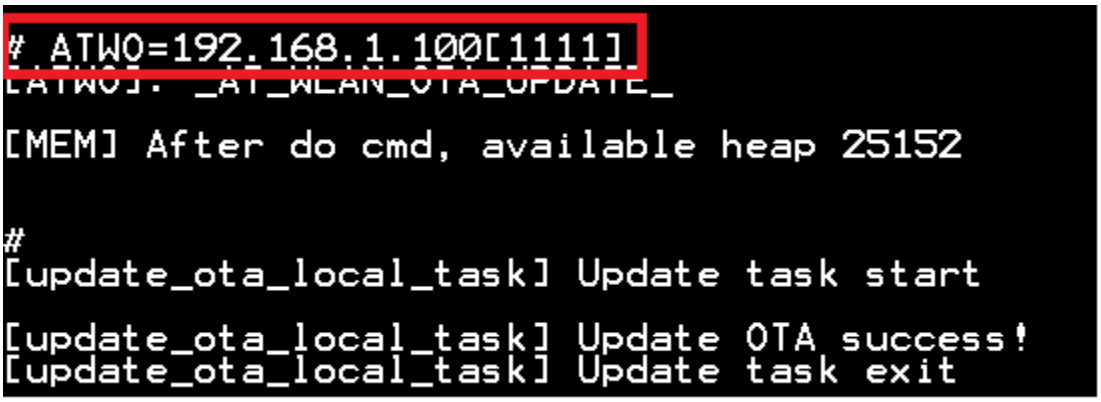
```
1  @echo off
2  DownloadServer 8082 New_Project.bin
3  set /p DUMMY=Press Enter to Continue ...
```

Execute start.bat

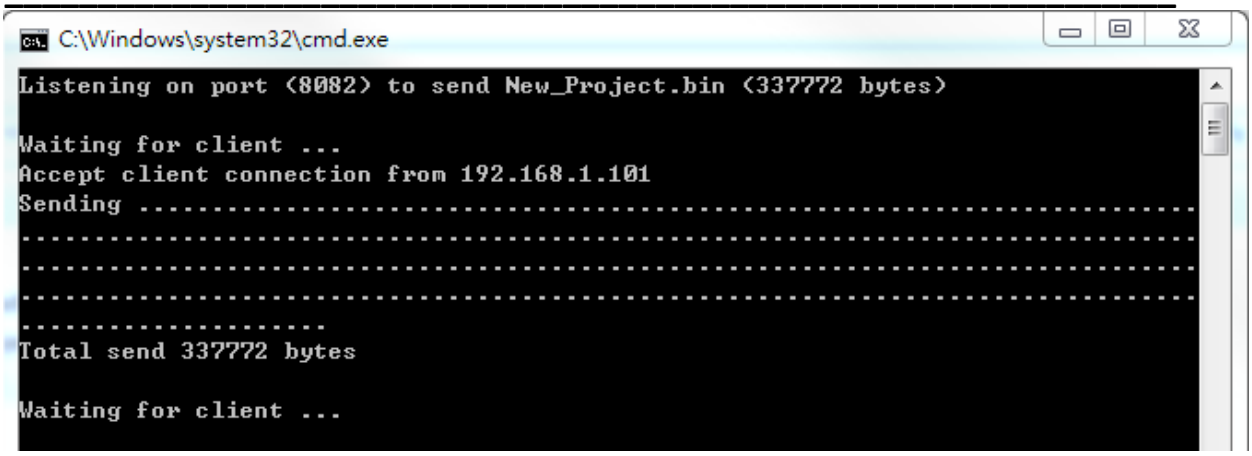


Reboot device and connect to AP.

Enter command: ATWO=IP[PORT].



Local download server success message:



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

3.3 OTA Signature

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