REALTEK

Realtek Ameba 1 Firmware Image

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

Table of Contents

1 Introduction

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

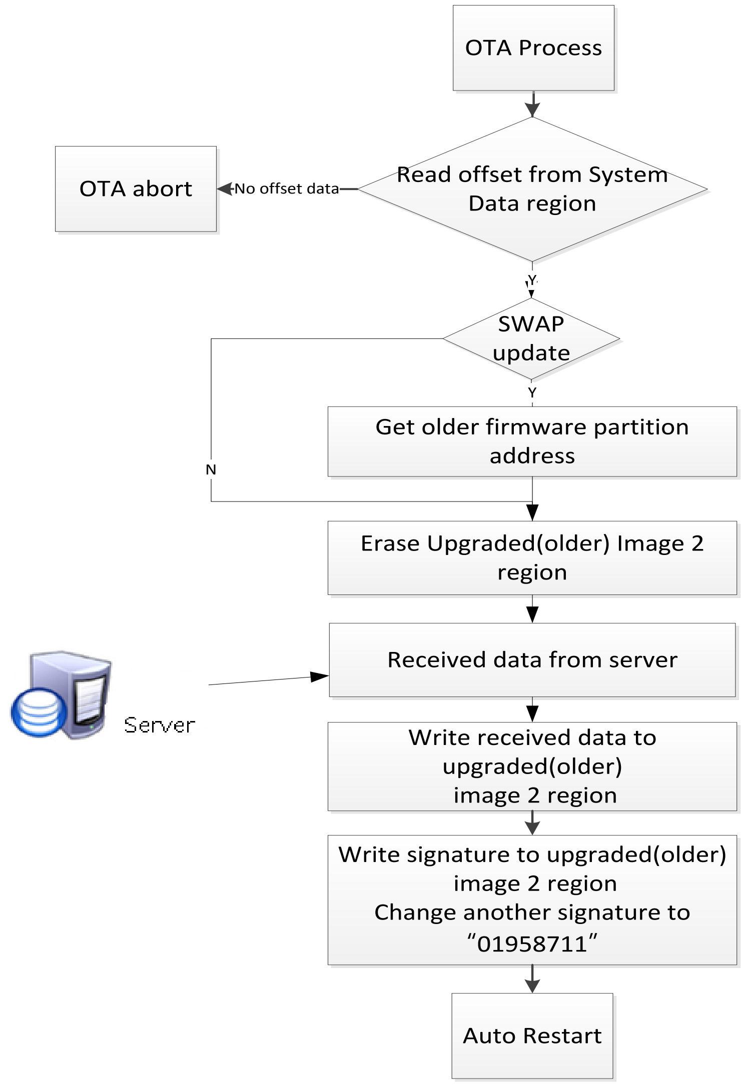
2 Firmware Image Output

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 images can be found at *SDK_folder/project/project_name/EWARM-RELEASE/Debug/Exe.*

3 Firmware Update Over the Air

3.1 Overview

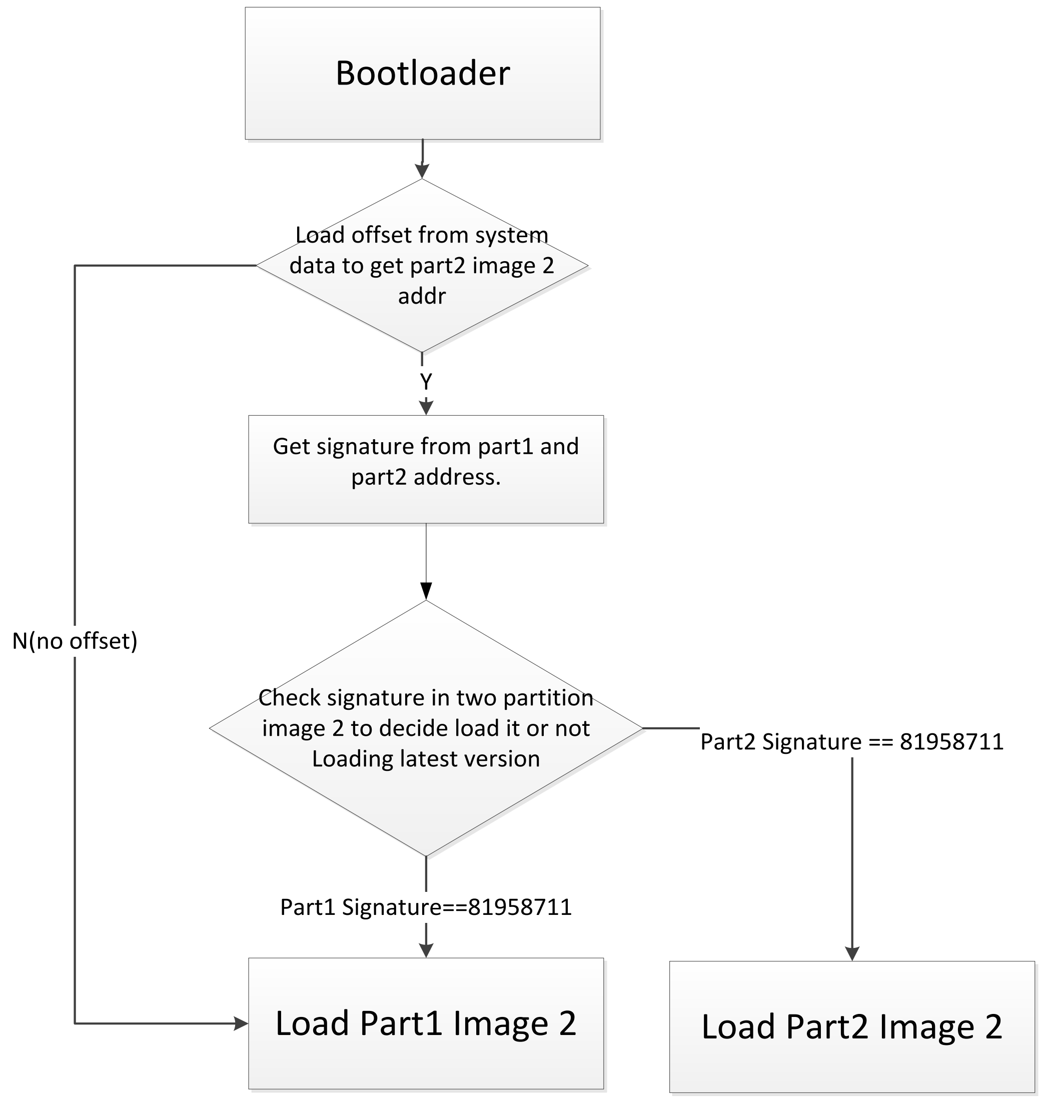
3.1.1 OTA operation flow


OTA Process
Read offset from System Data region
No offset data
OTA abort
Y
SWAP update
N
Y
Get older firmware partition address
Erase Upgraded(older) Image 2 region
Received data from server
Server
Write received data to upgraded(older) image 2 region
Write signature to upgraded(older) image 2 region
Change another signature to "01958711"
Auto Restart

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

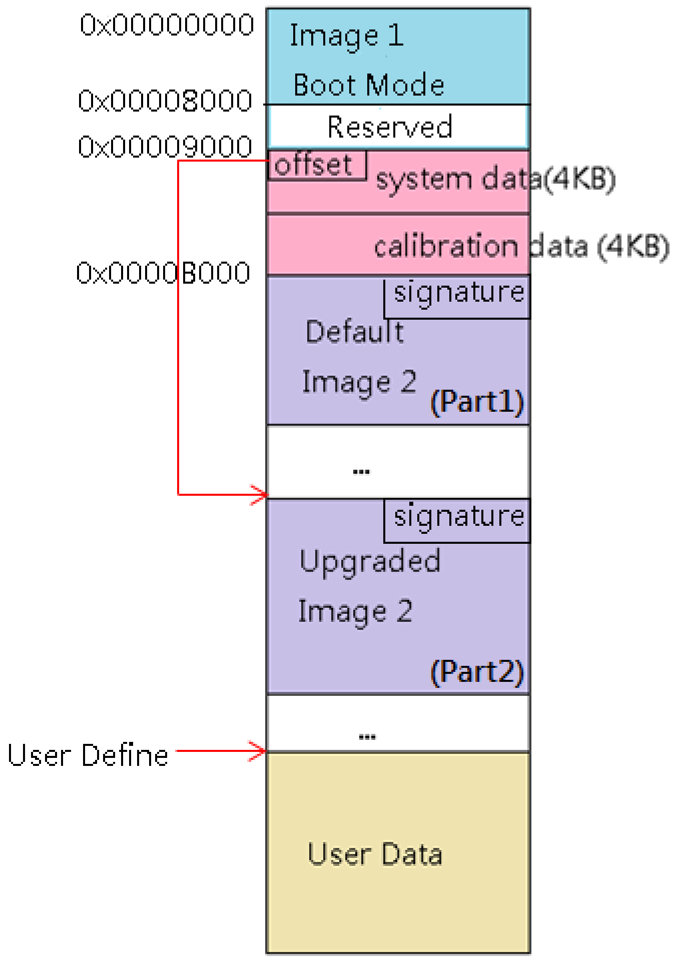
Note: OTA updater will change signature of another region from "81958711" to "01958711" when writing signature. Firmware with signature "01958711" is previous version, and with "81958711" is latest updated version.

3.1.2 Boot process flow


Bootloader
Load offset from system data to get part2 image 2 addr
Y
N(no offset)
Get signature from part1 and part2 address.
Check signature in two partition image 2 to decide load it or not Loading latest version
Part2 Signature == 81958711
Part1 Signature==81958711
Load Part1 Image 2
Load Part2 Image 2

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 ota_8195a.h. 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 base on socket

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.

3.2.1.1 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 SWAP_UDPATE in ota_8195a.h file

```
//Config in ota_8195a.h
#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_OTA_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 ota_8195a.c
uint32_t update_ota_prepare_addr(void)
{
   …
   //Get upgraded image 2 addr from offset
   device_mutex_lock(RT_DEV_LOCK_FLASH);
   flash_read_word(&flash, OFFSET_DATA, &NewImg2Addr);
   device_mutex_unlock(RT_DEV_LOCK_FLASH);
   if((NewImg2Addr > IMAGE_3) && ((NewImg2Addr < (IMAGE_3+Img3Len))) ||
      (NewImg2Addr < IMAGE_3) ||((NewImg2Addr & 0xfff) != 0)|| (NewImg2Addr == ~0x0)){
          return -1;
   }
```

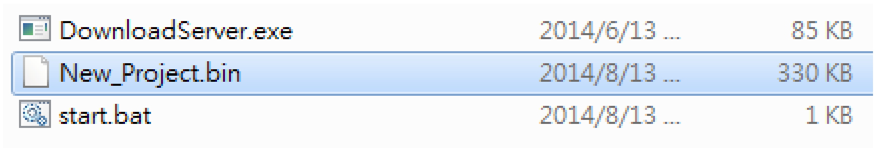
The address of OFFSET_DATA is 0x9000, and the address of upgraded image 2 is the first 4 byte from this address. If the address was not qualified, then the OTA process will be stopped.

Define custom signature

```
//Configuration in ota_8195a.h
1. turn on the marco as follows:
#define CONFIG_CUSTOM_SIGNATURE 1
2. Define your own signature.
//Define in ota_8195a.c

#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.
int update_ota_checksum(_file_checksum *file_checksum, uint32_t flash_checksum, uint32_t
NewImg2Addr)
{
…
#if CONFIG_CUSTOM_SIGNATURE
        && !strcmp(read_custom_sig,custom_sig)
#endif
…
}
```

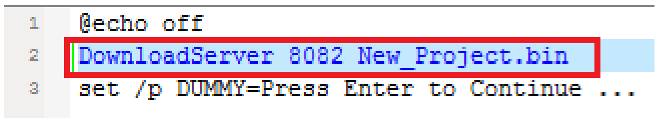
3.2.1.2 Local download server

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



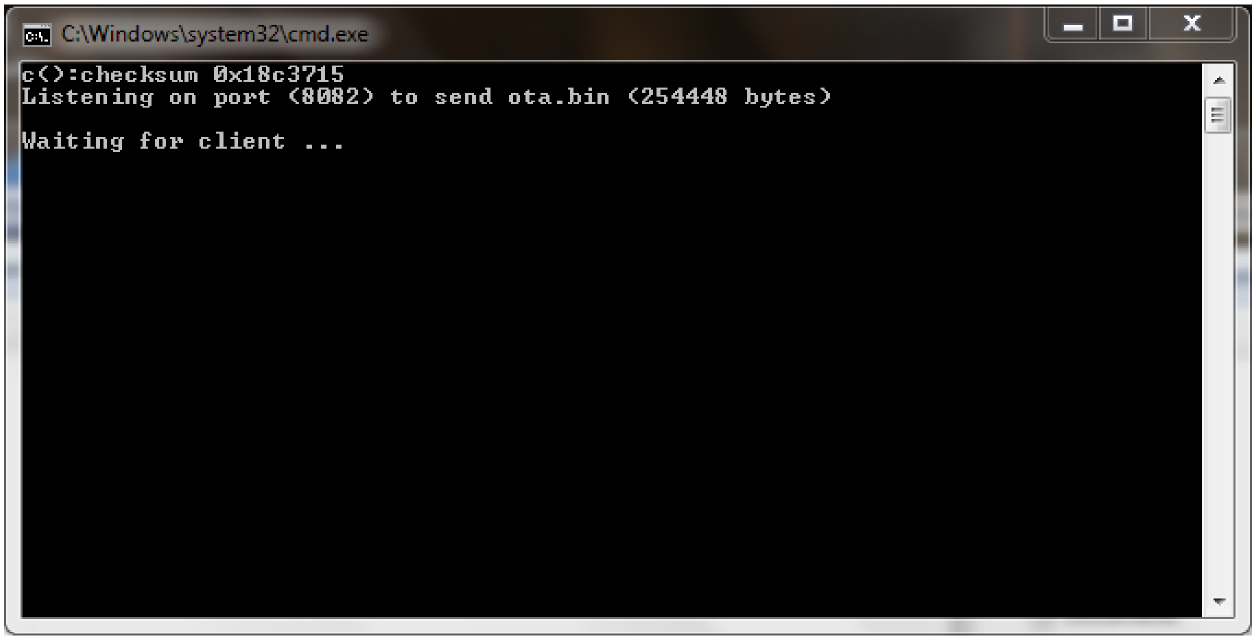
DownloadServer.exe	2014/6/13 ...	85 KB
New_Project.bin	2014/8/13 ...	330 KB
start.bat	2014/8/13 ...	1 KB

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



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

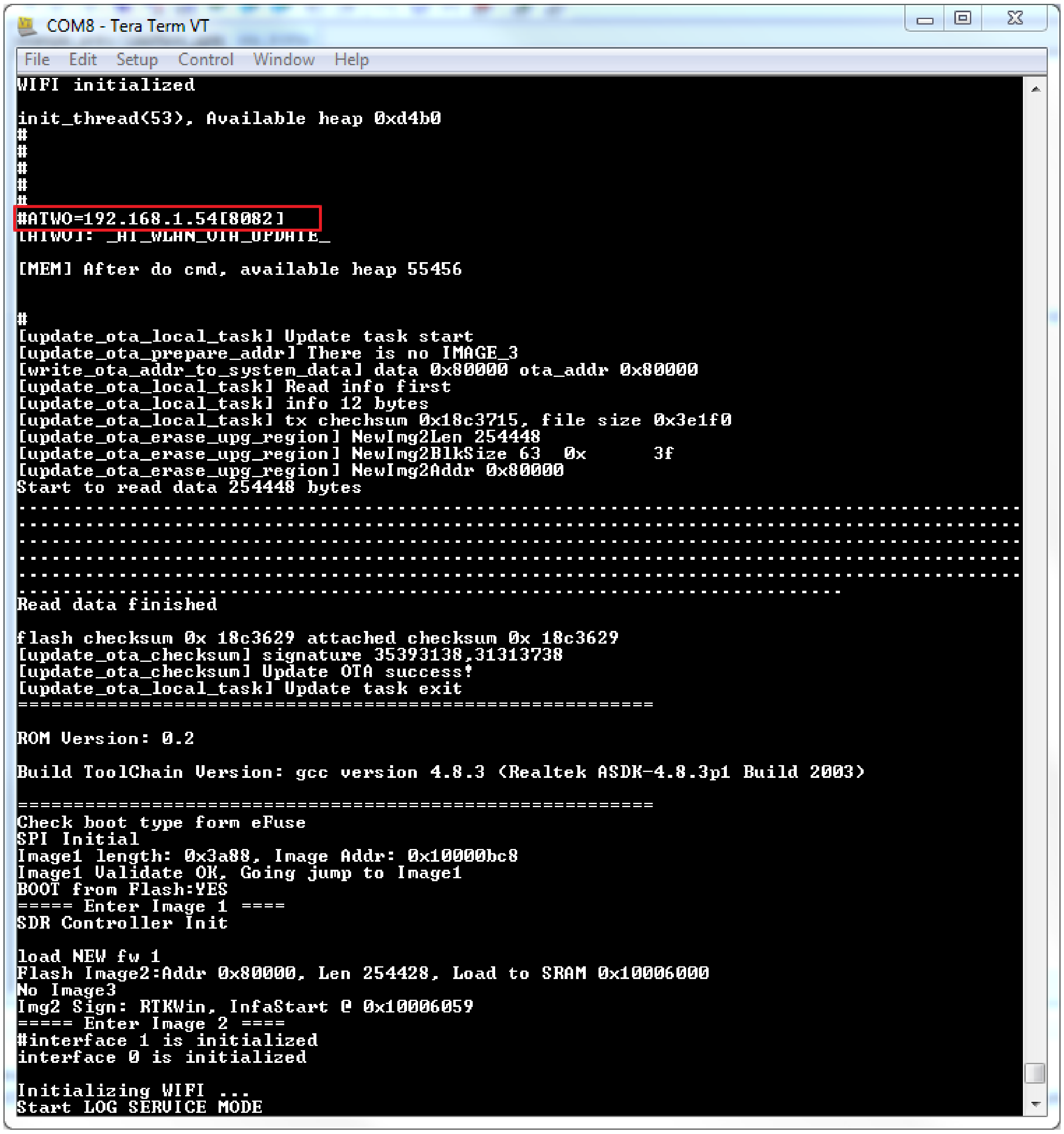
Execute start.bat



```
c():checksum 0x18c3715
Listening on port (8082) to send ota.bin (254448 bytes)

Waiting for client ...
```

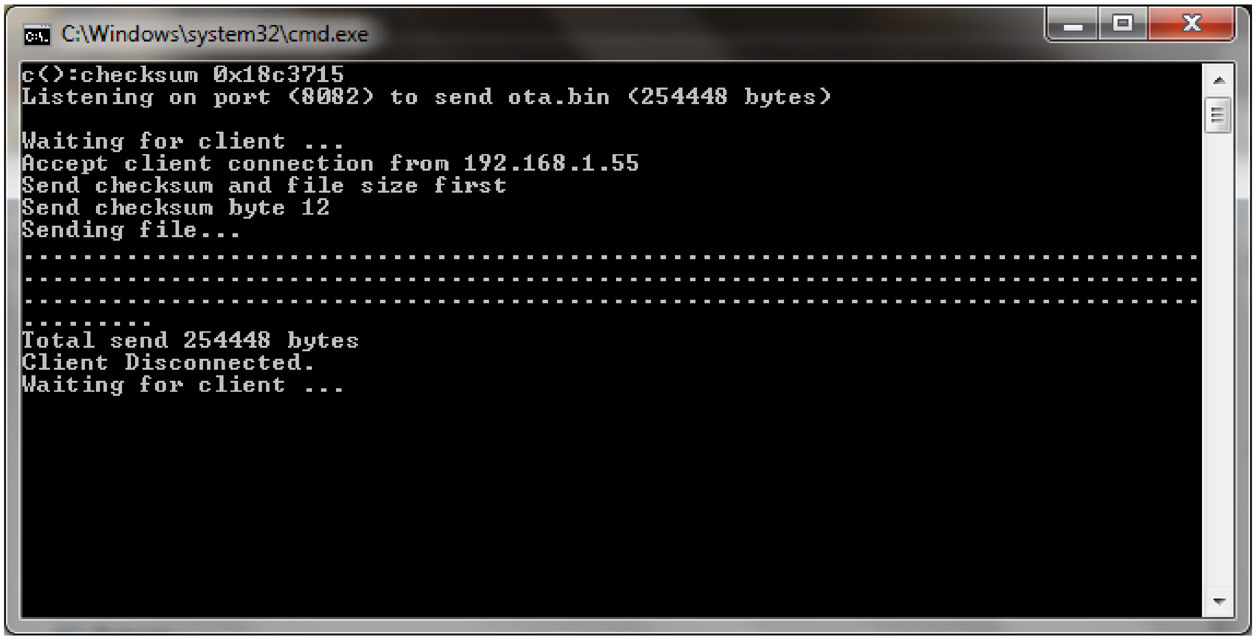
Reboot device and connect to AP.

Enter command: ATWO=IP[PORT].



```
WIFI initialized

init_thread(53), Available heap 0xd4b0
#
#
#
#
#
#ATWO=192.168.1.54[8082]
[ATWO]: _AT_WLAN_OTA_UPDATE_

[MEM] After do cmd, available heap 55456


#
[update_ota_local_task] Update task start
[update_ota_prepare_addr] There is no IMAGE_3
[write_ota_addr_to_system_data] data 0x80000 ota_addr 0x80000
[update_ota_local_task] Read info first
[update_ota_local_task] info 12 bytes
[update_ota_local_task] tx chechsum 0x18c3715, file size 0x3e1f0
[update_ota_erase_upg_region] NewImg2Len 254448
[update_ota_erase_upg_region] NewImg2BlkSize 63  0x      3f
[update_ota_erase_upg_region] NewImg2Addr 0x80000
Start to read data 254448 bytes
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................
Read data finished

flash checksum 0x 18c3629 attached checksum 0x 18c3629
[update_ota_checksum] signature 35393138,31313738
[update_ota_checksum] Update OTA success!
[update_ota_local_task] Update task exit
=========================================================

ROM Version: 0.2

Build ToolChain Version: gcc version 4.8.3 (Realtek ASDK-4.8.3p1 Build 2003)

=========================================================
Check boot type form eFuse
SPI Initial
Image1 length: 0x3a88, Image Addr: 0x10000bc8
Image1 Validate OK, Going jump to Image1
BOOT from Flash:YES
===== Enter Image 1 ====
SDR Controller Init

load NEW fw 1
Flash Image2:Addr 0x80000, Len 254428, Load to SRAM 0x10006000
No Image3
Img2 Sign: RTKWin, InfaStart @ 0x10006059
===== Enter Image 2 ====
#interface 1 is initialized
interface 0 is initialized

Initializing WIFI ...
Start LOG SERVICE MODE
```

Local download server success message:



C:\Windows\system32\cmd.exe

```
c():checksum 0x18c3715
Listening on port (8082) to send ota.bin (254448 bytes)

Waiting for client ...
Accept client connection from 192.168.1.55
Send checksum and file size first
Send checksum byte 12
Sending file...
................................................................................
................................................................................
................................................................................
.........
Total send 254448 bytes
Client Disconnected.
Waiting for client ...
```

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

3.2.2 OTA using local download server base on HTTP

This example shows how device updates image from a local http download server. The local http download server will send the http response which data part is ota.bin after receiving the http request.

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

3.2.2.1 Build OTA Application image

Turn on OTA HTTP example

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

The http ota flag defined in \component\soc\realtek\8195a\misc\platform\ota_8195a.h

```
//Config in platform_opts.h
#define CONFIG_OTA_UPDATE      1
//Defined in ota_8195a.h
#define HTTP_OTA_UPDATE
```

Define Server IP and PORT in example_ota.c file (In \component\common\example\ota_update\example_ota.c)

```
//Defined in example_ota.c
#define PORT              8082
#define IP               "192.168.1.54"
#define RESOURCE         "ota.bin"
```

Example: SERVER: http://m-apps.oss-cn-shenzhen.aliyuncs.com/051103061600.bin

```
Setting:      #define PORT          80

              #define HOST          "m-apps.oss-cn-shenzhen.aliyuncs.com"

              #define RESOURCE      "051103061600.bin"
```

Define SWAP_UDPATE in ota_8195a.h file

```
//Config in ota_8195a.h
#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_OTA_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 ota_8195a.c
uint32_t update_ota_prepare_addr(void)
{
   …
   //Get upgraded image 2 addr from offset
   device_mutex_lock(RT_DEV_LOCK_FLASH);
   flash_read_word(&flash, OFFSET_DATA, &NewImg2Addr);
   device_mutex_unlock(RT_DEV_LOCK_FLASH);
   if((NewImg2Addr > IMAGE_3) && ((NewImg2Addr < (IMAGE_3+Img3Len))) ||
      (NewImg2Addr < IMAGE_3) ||((NewImg2Addr & 0xfff) != 0)|| (NewImg2Addr == ~0x0)){
         printf("\n\r[%s] Invalid OTA Address 0x%x", __FUNCTION__, NewImg2Addr);
         return -1;
   }
```

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

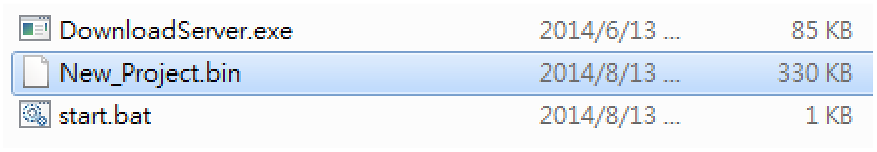
```
//Configuration in ota_8195a.h
1. turn on the marco as follows:
#define CONFIG_CUSTOM_SIGNATURE 1
2. Define your own signature.
//Define in ota_8195a.c

#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.
int update_ota_checksum(_file_checksum *file_checksum, uint32_t flash_checksum, uint32_t NewImg2Addr)
{
…
#if CONFIG_CUSTOM_SIGNATURE
          && !strcmp(read_custom_sig,custom_sig)
#endif
…
}
```

3.2.2.2 Communication with Local HTTP download server

1. In http_update_ota_local_task(), after connecting with server, Ameba will send a HTTP request to server : "GET /RESOURCE HTTP/1.1\r\nHost: host\r\n\r\n"
2. The local HTTP download server will send the HTTP response after receiving the request. The response header contains the "Content-Length" which is the length of the ota.bin The response data part is just ota.bin
3. After ameba receiving the HTTP response, it will parse the http response header to get the content length to judge if the receiving ota.bin is completed.

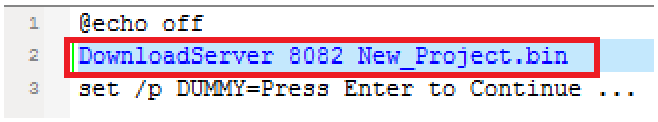
3.2.2.3 Local HTTP download server

Build new image New_Project.bin in DownloadServer(HTTP) folder.



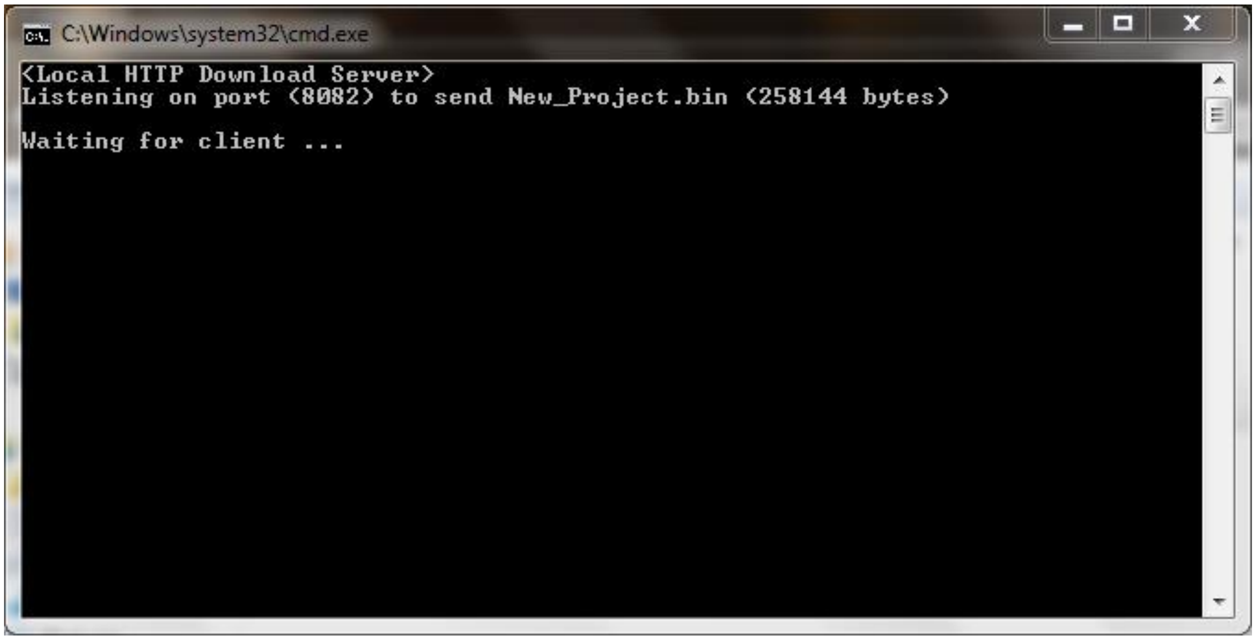
DownloadServer.exe	2014/6/13 ...	85 KB
New_Project.bin	2014/8/13 ...	330 KB
start.bat	2014/8/13 ...	1 KB

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



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

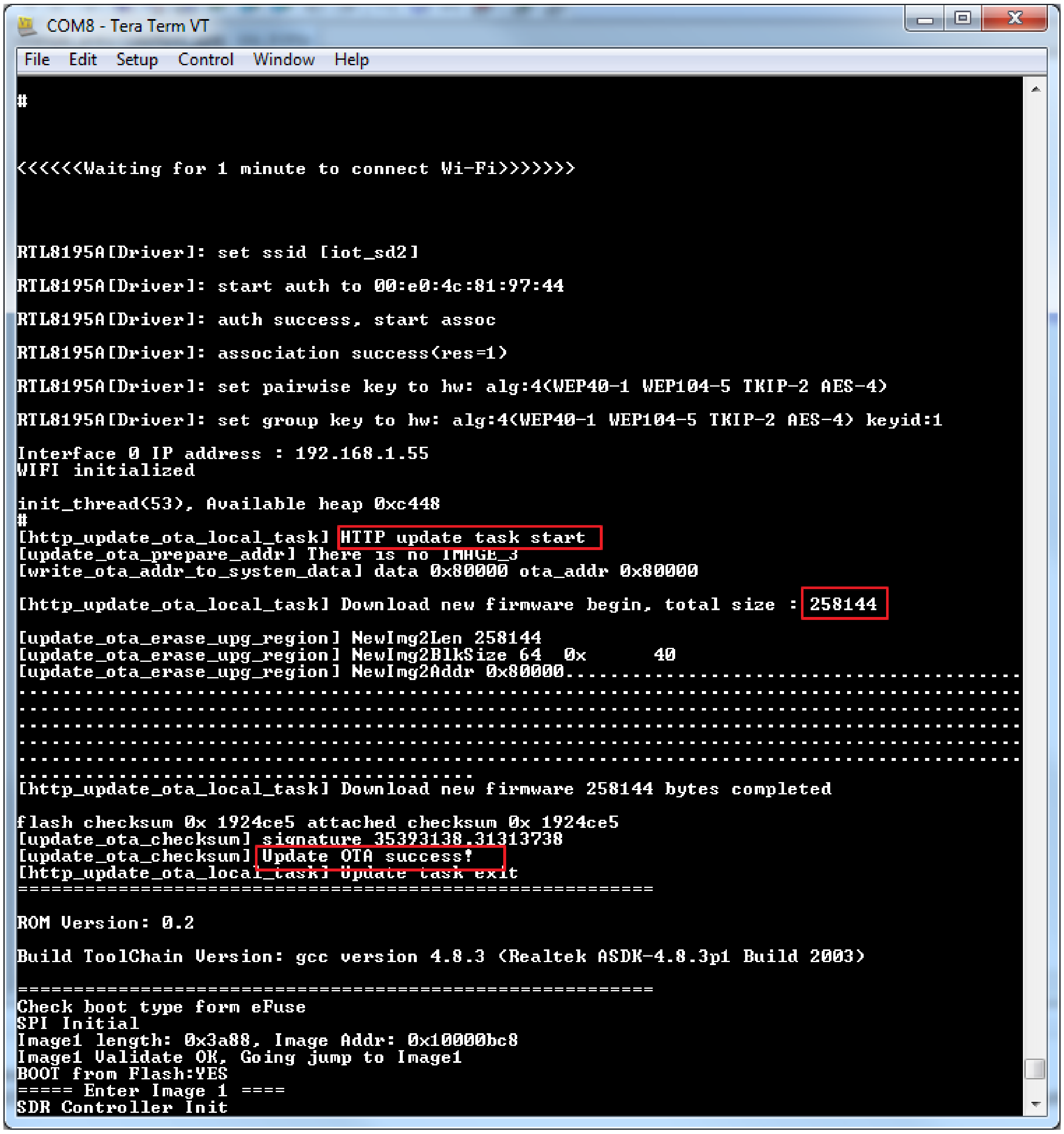
Execute start.bat



```
<Local HTTP Download Server>
Listening on port (8082) to send New_Project.bin (258144 bytes)

Waiting for client ...
```

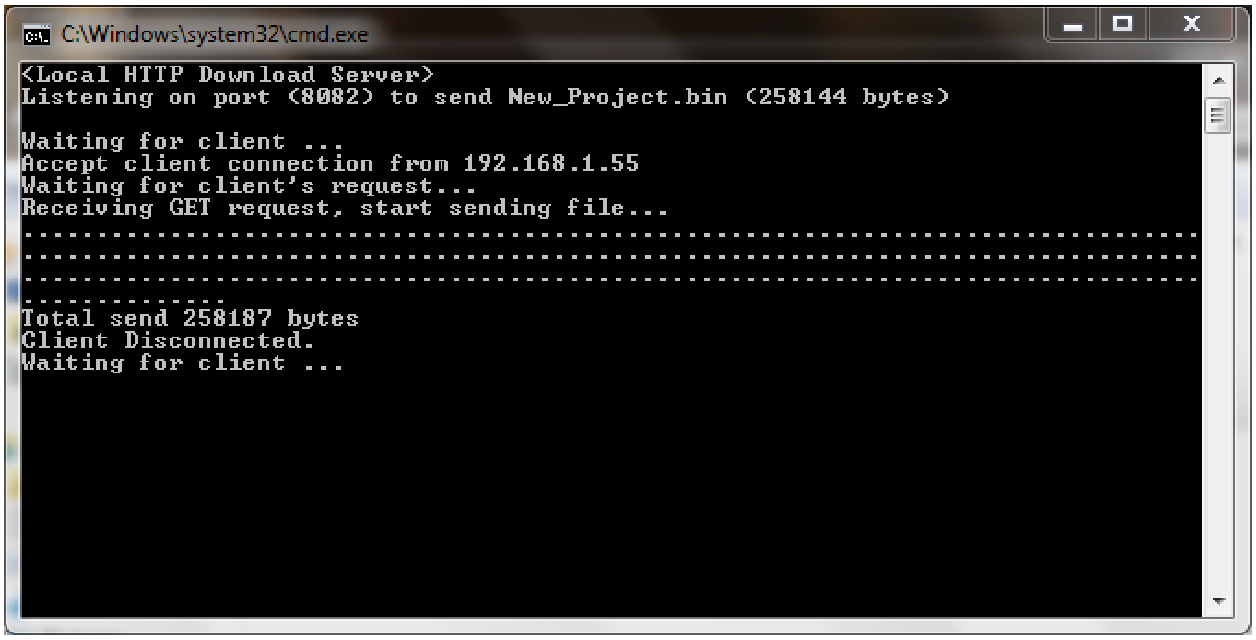
Reboot device and connect to AP.

After 1 minute, the OTA update through HTTP protocol will start.



```
#


<<<<<<Waiting for 1 minute to connect Wi-Fi>>>>>>>



RTL8195A[Driver]: set ssid [iot_sd2]

RTL8195A[Driver]: start auth to 00:e0:4c:81:97:44

RTL8195A[Driver]: auth success, start assoc

RTL8195A[Driver]: association success(res=1)

RTL8195A[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4)

RTL8195A[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4) keyid:1

Interface 0 IP address : 192.168.1.55
WIFI initialized

init_thread(53), Available heap 0xc448
#
[http_update_ota_local_task] HTTP update task start
[update_ota_prepare_addr] There is no IMAGE_3
[write_ota_addr_to_system_data] data 0x80000 ota_addr 0x80000

[http_update_ota_local_task] Download new firmware begin, total size : 258144

[update_ota_erase_upg_region] NewImg2Len 258144
[update_ota_erase_upg_region] NewImg2BlkSize 64  0x      40
[update_ota_erase_upg_region] NewImg2Addr 0x80000.....................................
...........................................................................................
...........................................................................................
...........................................................................................
...........................................................................................
...........................................................................................
..............................................
[http_update_ota_local_task] Download new firmware 258144 bytes completed

flash checksum 0x 1924ce5 attached checksum 0x 1924ce5
[update_ota_checksum] signature 35393138,31313738
[update_ota_checksum] Update OTA success!
[http_update_ota_local_task] Update task exit
=========================================================

ROM Version: 0.2

Build ToolChain Version: gcc version 4.8.3 (Realtek ASDK-4.8.3p1 Build 2003)

=========================================================
Check boot type form eFuse
SPI Initial
Image1 length: 0x3a88, Image Addr: 0x10000bc8
Image1 Validate OK, Going jump to Image1
BOOT from Flash:YES
===== Enter Image 1 ====
SDR Controller Init
```

Local download server success message:



```
<Local HTTP Download Server>
Listening on port (8082) to send New_Project.bin (258144 bytes)

Waiting for client ...
Accept client connection from 192.168.1.55
Waiting for client's request...
Receiving GET request, start sending file...
................................................................................
................................................................................
................................................................................
..............
Total send 258187 bytes
Client Disconnected.
Waiting for client ...
```

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

3.3 OTA Signature

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