

w217 补充适配说明

Version: 1.1

Date: 2023-06-08

**iot.10086.cn**

修订记录

编号	版本号	修订日期	修订描述	修改人
1	V1.0	2023-04-28	初稿	OneOS Team
2	V1.1	2023-06-08	添加版本包制作说明	OneOS Team
3				
4				

目 录

1 文档结构	3
2 重要配置	4
3 进度条显示	6
4 版本包制作	6

1 文档结构

W217的fota还原升级适配相关文件存在于 **external/oneos** 和 **Third-party/oneos** 两个路径下。

名称	修改日期	类型	大小
Docs	2023/4/10 15:21	文件夹	
external	2023/4/10 15:22	文件夹	
fdl_bootloader	2023/4/10 15:22	文件夹	
halo_rec	2023/4/10 15:22	文件夹	
JEE	2023/4/10 15:21	文件夹	
lib	2023/4/10 15:22	文件夹	
make	2023/4/10 15:22	文件夹	
MS_Customize	2023/4/10 15:21	文件夹	
MS_MMI_Main	2023/4/10 15:21	文件夹	
MS_Ref	2023/4/10 15:22	文件夹	
nes	2023/4/10 15:22	文件夹	
PARSER	2023/4/10 15:22	文件夹	
pri_code	2023/4/10 15:22	文件夹	
project	2023/4/10 15:22	文件夹	
RTOS	2023/4/10 15:22	文件夹	
spload	2023/4/10 15:22	文件夹	
tcad_resource	2023/4/10 15:22	文件夹	
Third-party	2023/4/10 15:22	文件夹	
tools	2023/4/10 15:23	文件夹	
ual	2023/4/10 15:23	文件夹	
UWS6121E_WC_1H00	2023/4/10 15:22	文件夹	
UWS6121E_WC_1H00_NoGNSS	2023/4/10 15:22	文件夹	
UWS6121E_WC_1H00_NoGNSS_SEC	2023/4/10 15:22	文件夹	
UWS6121E_WC_1H00_NoGNSS_USER	2023/4/10 15:22	文件夹	

其中, **external/oneos/src/flash/os_flash.c** 和 **external/oneos/inc/flash/os_flash.h** 中, 存放了 flash 操作 (读、写、擦) 的相关接口; **Third-party/oneos/bootloader** 路径下存放了还原升级的库文件以及启动升级的函数接口。

名称	修改日期	类型	大小
inc	2023/4/10 15:22	文件夹	
FAQ.md	2023/4/10 15:22	Markdown File	2 KB
libcmiot_ota.a	2023/4/10 15:22	A 文件	402 KB
os_ota_main.c	2023/4/10 15:22	C 文件	6 KB
os_ota_main.h	2023/4/10 15:22	H 文件	2 KB
README.md	2023/4/10 15:22	Markdown File	12 KB

其中,

[inc] : 包含 **ota_lib.h** 和 **ota_type.h** 两个还原库对应的头文件;

FAQ.md, README.md: 还原库相关的帮助文档, 请仔细阅读;

libcmiot_ota.a: 适用于 w217 平台的还原库;

os_ota_main.c, os_ota_main.h: 实现了启动还原升级接口 **int cmiot_ota_main(void)**, 在 **fota_bootloader_main.c/int main(void)** 接口中调用。

```

132: #ifdef FOTA_SUPPORT_CMIOT
133: {
134:     ... rprintf("cmiot fota update routin enter!\n");
135:     ... errno = cmiot_ota_main();
136:     ... rprintf("cmiot fota update routin exit!\n");
137: }
138: #endif

```

2 重要配置

1. 地址映射

W217 使用了逻辑地址映射功能，在 `oneos_config.h` 中，默认配置了 4 个可升级 bin 文件，其中 `UWS6121E_UWS6121E_WC_1H00_stone.img.bin` 实际对应 `UWS6121E_UWS6121E_WC_1H00_stone.img` 文件，由于算法仅支持文件类型为 `*.bin`，添加了“.bin”后缀。每个 bin 文件都使用了 `0x9xxxx` 或者 `0x8xxxx` 类型的逻辑地址，该地址可通过烧录工具中的信息获取。使用逻辑地址的好处是，当调整分区信息时，只需要修改分区表即可。

```

/* Fota by CMIOT */

#define FOTA_USING_CMIOT
#define CMIOT_USING_SINGLE
#define CMIOT_USING_CMS
#define CMIOT_DEFAULT_NETWORK_PROTOCOL 2
#define CMIOT_ALGORITHM_LUCKPAND
#define CMIOT_DEFAULT_TRY_TIME 5000
#define CMIOT_FOTA_ALGORITHM 1
#define CMIOT_DEFAULT_SECTOR_SIZE 0x10000
#define CMIOT_FOTA_AVAILABLE_RAM_SIZE 0x80000
#define CMIOT_UPDATE_SIZE 0xA0000
#define CMIOT_FOTA_WOSUN_VERSION "IOT4.0_R42641"
#define CMIOT_FOTA_LUSUN_VERSION "IOT5.0_LUSUN11_R50426"
#define CMIOT_FOTA_LUCKSPAR_VERSION "IOT6.0_LUCKSPAR_R010100"
#define CMIOT_FOTA_LUCKPAND_VERSION "IOT7.0_LUCKPAND_R010100"
#define CMIOT_SEGMENT_SIZE_INDEX_START 3
#define CMIOT_DEFAULT_SEGMENT_SIZE_INDEX 3
#define CMIOT_FOTA_FILENAME1 "mmi_res_240x240_child_circular.bin"
#define CMIOT_FOTA_ADDRESS1 0x90000004
#define CMIOT_FOTA_FILE_MORE1
#define CMIOT_FOTA_FILENAME2 "UWS6121E_WC_1H00_delta_nv.bin"
#define CMIOT_FOTA_ADDRESS2 0x90000100
#define CMIOT_FOTA_FILE_MORE2
#define CMIOT_FOTA_FILENAME3 "UWS6121E_UWS6121E_WC_1H00_cp.bin"
#define CMIOT_FOTA_ADDRESS3 0x90000101
#define CMIOT_FOTA_FILE_MORE3
#define CMIOT_FOTA_FILENAME4 "UWS6121E_UWS6121E_WC_1H00_stone.img.bin"
#define CMIOT_FOTA_ADDRESS4 0x80000003
#define CMIOT_FOTA_CONTRAST_BLOCK_SIZE 64
#define CMIOT_FIRMWARE_VERSION "1.0"
#define CMIOT_FOTA_OS_VERSION "ONEOS_V2.X"

```

☐ Select All Files
Select Product: PAC_NOR_FLASH_UWS6121E

FileID	FileName	Base1	Size1
<input checked="" type="checkbox"/> NOR_FDL1	D:\D2_Work\code\misc\w217-watch\tool\Dow...	0x50800000	0x14000
<input checked="" type="checkbox"/> NOR_FDL	D:\D2_Work\code\misc\w217-watch\tool\Dow...	0x80100500	
<input checked="" type="checkbox"/> BOOTLOADER	D:\D2_Work\code\misc\w217-watch\tool\Dow...	0x80000000	
<input checked="" type="checkbox"/> PS	D:\D2_Work\code\misc\w217-watch\tool\Dow...	0x80000003	
<input checked="" type="checkbox"/> FOTA_BOOTL...	D:\D2_Work\code\misc\w217-watch\tool\Dow...	0x80000002	
<input checked="" type="checkbox"/> MMIREs	D:\D2_Work\code\misc\w217-watch\tool\Dow...	0x90000004	
<input checked="" type="checkbox"/> CP	D:\D2_Work\code\misc\w217-watch\tool\Dow...	0x90000101	
<input checked="" type="checkbox"/> IMS_TOGGLE		0x90000112	0x10000
<input checked="" type="checkbox"/> CONFIG_NV		0x90000113	0x10000
<input checked="" type="checkbox"/> OperatorData	D:\D2_Work\code\misc\w217-watch\tool\Dow...	0x90000100	
<input checked="" type="checkbox"/> NV	D:\D2_Work\code\misc\w217-watch\tool\Dow...	0x90000001	0x30000
<input checked="" type="checkbox"/> PHASE_CHECK		0x90000002	0x1000
<input checked="" type="checkbox"/> FLASH		0x90000003	0xC0000
<input type="checkbox"/> ERASE UDISK		0x90000005	

在 os_ota_main.c 中，flash 信息初始化时，完成逻辑地址和物理地址信息的映射

```

94: static void flash_part_init(void)
95: {
96:     map_info[IDX_MMI].logic_addr = MMI_RES_LOGIC_ADDR;
97:     map_info[IDX_MMI].phy_addr = s_platform_partition_config.mmi_res;
98:     fal_part_table[IDX_MMI].size = s_platform_partition_config.mmi_res_size;
99:
100:    map_info[IDX_NV].logic_addr = DELTA_NV_LOGIC_ADDR;
101:    map_info[IDX_NV].phy_addr = s_platform_partition_config.operator_data_addr;
102:    fal_part_table[IDX_NV].size = s_platform_partition_config.operator_data_size;
103:
104:    map_info[IDX_CP].logic_addr = CP_LOGIC_ADDR;
105:    map_info[IDX_CP].phy_addr = s_platform_partition_config.cp_addr;
106:    fal_part_table[IDX_CP].size = s_platform_partition_config.cp_size;
107:
108:    map_info[IDX_PS].logic_addr = PS_LOGIC_ADDR;
109:    map_info[IDX_PS].phy_addr = s_platform_partition_config.ps_addr;
110:    fal_part_table[IDX_PS].size = s_platform_partition_config.fota_bootloader_addr - s_platform_partitii
111:
112:    /*download part no logic addr*/
113:    map_info[IDX_DL].logic_addr = s_platform_partition_config.fota_addr;
114:    map_info[IDX_DL].phy_addr = s_platform_partition_config.fota_addr;
115:    fal_part_table[IDX_DL].size = s_platform_partition_config.fota_size;
116:
117:    for (int i = IDX_MMI; i < IDX_MAX; i++)
118:    {
119:        /*init fal part*/
120:        fal_part_table[i].address = map_info[i].logic_addr;
121:    }
122: } /*end flash_part_init*/
123:

```

2. 内置内存管理

W217 在 bootloader 阶段，内存堆不可用，因此启用了还原库内置的内存管理。

```

43: #define HEAP_BLOCK_SIZE (128)
44: #define HEAP_BUF_MAX_SIZE (CMIOT_FOTA_AVAILABLE_RAM_SIZE)
45: static ota_uint8 g_work_heap_buf[HEAP_BUF_MAX_SIZE] = {0};

```

可以通过配置 HEAP_BLOCK_SIZE 和 HEAP_BUF_MAX_SIZE 来设置还原算法可用的内存空间大小。

3 进度条显示

由于升级还原时间可能在几分钟到几十分钟，在升级过程中，务必保持屏幕点亮，并通过 void ota_update_proc_bar(int percentage)接口实时显示升级进度，以免黑屏造成死机假象，从而导致用户反复强制开关机，打断升级过程。

4 版本包制作

在 w217 平台上，提供了自动打包脚本，路径为：./Third-party/oneos/pack_cmd.bat，

编译生成 pac 后，运行该脚本可以自动生成版本包，默认使用 WinRAR 压缩工具，若使用其他工具，替换该工具路径即可。

```

1  @title:make package
2  @echo:off
3
4  set version=%1
5
6
7  if "%version%" == "" (
8      echo ***version is empty***
9      echo usage: .\pack_cmd.bat [version]
10     echo example: .\pack_cmd.bat 1.0.0
11     goto EOF
12 ) else (
13     echo package source: %version%
14 )
15
16 if exist %version% (
17     echo "package source folder %version% exist"
18 ) else (
19     md %version%
20 )
21
22 ::copy new files
23 copy ..\..\build\UWS6121E_WC_1H00_build\img\*.* %version%
24
25 ::rename image files
26 cd %version%
27 if exist UWS6121E_UWS6121E_WC_1H00_stone.img.bin (
28     del UWS6121E_UWS6121E_WC_1H00_stone.img.bin
29     echo clear old UWS6121E_UWS6121E_WC_1H00_stone.img.bin
30 ) else (
31     echo UWS6121E_UWS6121E_WC_1H00_stone.img.bin
32 )
33 copy UWS6121E_UWS6121E_WC_1H00_stone.img UWS6121E_UWS6121E_WC_1H00_stone.img.bin
34 cd ..
35
36 ::generate package
37 ota\cmiot\pack\cmiot_pack.bat "C:\Program Files\WinRAR\WinRAR.exe" oneos_config.h %version%
38
39 :EOF

```

