RTL8720D 常见问题

修改记录

类型	修改内容	修改人	日期	软件版本
A	初版	杨宾	2019/06/04	-

类型：A-新增 M-修改 D-删除

1. 配置和使用相关问题

1.1 环境搭建注意事项

Windows 使用 Cygwin 环境，必须使用 Cygwin 32 位版本(64 位 windows 也要使用 32 位的 Cygwin)

Example 中的 demo 参考 raw 目录转中的例子

1.2 CM0/CM4 含义

CM0 表示小核，CM4 表示大核，8720D 是一个大小核芯片

1.3 切换 AT 指令集

默认 SDK 使用的是精简指令集，该指令集实现的功能较少，适合做二次开发的用户使用。

如果主要使用 AT 指令控制模块，需要切换 AT 指令集，切换后可以通过 AT 指令实现大部分 wifi 模块的功能。

切换方法是将 project/realtek_amebaD_cm4_gcc_verification/inc/platform_opts.h 中的宏定义 CONFIG_EXAMPLE_UART_ATCMD 设置为 1 。

然后将 component/common/example/uart_atcmd/example_uart_atcmd.h 中的

elif (CONFIG_AMEBA == AMEBAD)分支中的

#define UART_TX PB_1

#define UART_RX PB_2

这个是修改 AT 串口的指令，AT 串口使用的是 LP_UART 可选位置有三处，演示使用的是 PB_1，PB_2

(1) TX:PA_12 RX:PA_13

(2) TX:PA_26 RX:PA_25

(3) TX:PB_1 RX:PB_2

log 口波特率 115200，AT 口波特率默认 38400

2. 常见操作问题

2.1 V2 AT 指令集 log 口无响应

修改使用 V2 AT 指令集后会发现 log 口 AT 指令无响应，需要修改下面的 patch 修复代码

```
diff --git a/component/common/mbed/targets/hal/rtl8721d/serial_api.c b/component/common/mbed/targets/hal/rtl8721d/serial_api.c
index 276a8a3..004aa12 100644
--- a/component/common/mbed/targets/hal/rtl8721d/serial_api.c
+++ b/component/common/mbed/targets/hal/rtl8721d/serial_api.c
@@ -56,7 +56,7 @@ typedef struct {
        VOID *RxCompCbPara; // the pointer argument for RxCompCallback
 }MBED_UART_ADAPTER, *PMBED_UART_ADAPTER;

-#define UART_NUM (3)
+#define UART_NUM (4)
 #define SERIAL_TX_IRQ_EN            0x01
 #define SERIAL_RX_IRQ_EN            0x02
 #define SERIAL_TX_DMA_EN            0x01
@@ -68,16 +68,16 @@ typedef struct {

 #define CONFIG_GDMA_EN 1

-static uint32_t serial_irq_ids[UART_NUM] = {0, 0, 0};
+static uint32_t serial_irq_ids[UART_NUM] = {0, 0, 0, 0};

 static uart_irq_handler irq_handler[UART_NUM];
-static uint32_t serial_irq_en[UART_NUM]={0, 0, 0};
+static uint32_t serial_irq_en[UART_NUM]={0, 0, 0, 0};

 static int current_baudrate;


 #ifdef CONFIG_GDMA_EN
-static uint32_t serial_dma_en[UART_NUM] = {0, 0, 0};
+static uint32_t serial_dma_en[UART_NUM] = {0, 0, 0, 0};
 #endif

 MBED_UART_ADAPTER uart_adapter[MAX_UART_INDEX+1];
```

2.2 ATPU/ATPG/ATPL 指令导致固件损坏

这个是因为写入数据覆盖了固件导致的，修改 UART_SETTING_BACKUP_SECTOR 保存的地址（RTL8720D 没有专门保留这个 config 的地址，所以放在 user data 中就可以了）

修改之后 ATPU 可以正常使用了，但是 ATPU 无法自动重连，ATPL 也无法自动进入透传，只是不会导致固件损坏了。

```
diff --git a/component/common/api/at_cmd/atcmd_wifi.h b/component/common/api/at_cmd/atcmd_wifi.h
index 2369803..95633fb 100644
--- a/component/common/api/at_cmd/atcmd_wifi.h
+++ b/component/common/api/at_cmd/atcmd_wifi.h
@@ -119,7 +119,9 @@ typedef enum {

  //first segment for uart
  #if !defined(UART_SETTING_BACKUP_SECTOR)
-#define UART_SETTING_BACKUP_SECTOR (0x8000)
+#define UART_SETTING_BACKUP_SECTOR (0xFB000)
+
  #endif
  #define UART_CONF_DATA_OFFSET (0)
  #define UART_CONF_DATA_SIZE ((((sizeof(UART_LOG_CONF)-1)>>2) + 1)<<2)
diff --git a/project/realtek_amebaD_cm4_gcc_verification/inc/platform_opts.h b/project/realtek_amebaD_cm4_gcc_verification/inc/platform_opts.h
index 4262fa0..a67b911 100644
--- a/project/realtek_amebaD_cm4_gcc_verification/inc/platform_opts.h
+++ b/project/realtek_amebaD_cm4_gcc_verification/inc/platform_opts.h
@@ -53,6 +53,7 @@
  #define AP_SETTING_SECTOR 0x000FE000
  #define UART_SETTING_SECTOR 0x000FC000
  #define SPI_SETTING_SECTOR 0x000FC000
+//in atcmd_wifi.h #define UART_SETTING_BACKUP_SECTOR (0xFB000)
  #if defined(CONFIG_BAIDU_DUER) && CONFIG_BAIDU_DUER
  #define FAST_RECONNECT_DATA 0x1FF000
  #else
```

3. API 的使用和说明

3.1 读写 flash 和用户 flash 范围

标准的 RTL8710BX 使用的是 2M 的 flash，除去系统和固件占用的空间，现剩余给用户的有 1028K 空间

读写最好 4 字节对齐，我们用到的空间最好和系统使用的统一记录在一起，系统的记录在 sdk/project/realtek_amebaD_cm4_gcc_verification/inc/platform_opts.h

```
#define AP_SETTING_SECTOR       0x000FE000
#define UART_SETTING_SECTOR         0x000FC000
#define FAST_RECONNECT_DATA   (0x80000 - 0x1000)
#define FLASH_SECTOR_SIZE           0x1000
```

Flash 读写函数

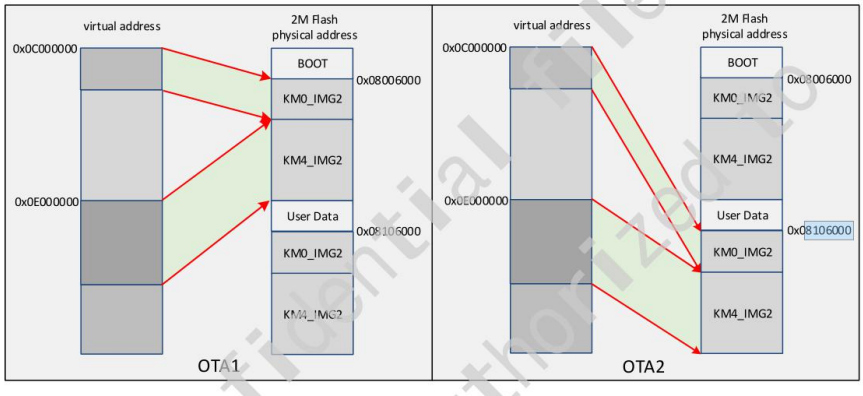
```
flash_t             flash;
uint8_t u8_buf[32] = {0};
//擦除从 AP_SETTING_SECTOR 开始的 4K 字节数据(一次最少擦除 4K 数据，且要 4K 对齐)
flash_erase_sector(&flash, AP_SETTING_SECTOR);
//从 AP_SETTING_SECTOR 写入 32 字节数据(写入数据要 4 字节对齐)
flash_stream_write(&flash, AP_SETTING_SECTOR, 32, (uint8_t*)u8_buf);
//从 AP_SETTING_SECTOR 地址读取 32 字节数据
flash_stream_read(&flash,AP_SETTING_SECTOR, 32, ((uint8_t*)u8_buf);
```

用户可用 flash 空间计算要根据 00014218-AN0400-Ameba-D-Application-Note-v09_205022.pdf 中的描述计算。注意 user data 中有一部分空间已经在 sdk/project/realtek_amebaD_cm4_gcc_verification/inc/platform_opts.h 中使用了，注意避让。

REALTEK Ameba-D Application Note

Ameba-D provides 8 MMU Entries. If virtual address is not included in the MMU entry, use virtual address as physical address. If virtual address is included in the MMU entry, physical address should be VAddress +/- MMU_ENTRYx_OFFSET.

MMU is implemented to facilitate OTA update procedure like Fig 11-3. Fig 11-3 is an example for 2M flash OTA, we need 2 MMU entries in this example.



3.2 malloc 和 free 函数

RTL8710BX 中开辟和释放内存需要使用以下两个函数
rtw_malloc(size)
rtw_free(ptr)