

## OTP 录音说明文档

2011-12-5

OTP 版本录音程序在 AC309N\_103 版本上完成。OTP 录音程序支持一款 led 数码管显示，一款 fm 收音模块，加多一个 mic 模式。

### 1、fm 模块宏开关

在下面定义想要的 fm 模块，如果 fm 模块都定义为 0 的话，程序就会没有 fm 模式

//fm 模块选择

```
#define RDA5807      1
#define CL6017G      0
#define BK1080       0
#define KT0830EG     0
#define QN8035       0
```

```
#if ((1 == RDA5807) || (1 == CL6017G) || (1 == BK1080) || (1 == KT0830EG) || (1 == QN8035))
    ///<是否有 fm 模块
```

```
#define FM_MODULE      1
#else
#define FM_MODULE      0
#endif
```

### 2、显示模块宏开关

把 monitor 定义为数码管或 led 灯，程序就编译用那种显示，led 灯的显示函数定义在 main.c 文件中，修改 led 灯的显示闪烁时要在宏定义区域内修改。

//显示屏选择

```
#define DISP_LED5X7    1
#define DLED           2
#define NO_DISP        3
#define monitor        DISP_LED5X7
```

```
#if (monitor == DISP_LED5X7)                ///<是否为数码管 LED 屏
#define DISP_SEG_LED      1
#else
#define DISP_SEG_LED      0
#endif
```

```
#if (monitor == DLED)                                //是否为 LED 灯
#define DISP_DLED                                     1
#else
#define DISP_DLED                                     0
#endif
```

显示部分不用 `my_printf` 函数来统一处理，把 AC309N\_103 上的显示全换回直接调用底层显示函数。

### 3、sd 卡检测

otp 录音版本的 sd 卡检测默认都使用 `sd clk` 查询模式

这种检测模式下必须需在 SD 卡座上的 CLK 和 DECTL 两引脚之间串联一个 3.3K 电阻，将 `config.h` 中的 `SDMMC_CLK_MODE` 打开，其他两项关闭掉就可以了。

### 4、虚拟识别功能

在 `config.h` 文件中，虚拟识别功能宏定义如下

```
#define VIRTUAL_ENABLE 1 //<虚拟识别使能 1: enable; 0: disable;
当 VIRTUAL_ENABLE 定义为 1 时，开虚拟识别使能，所谓的虚拟识别就是把 u 盘、
u 盘内的录音文件夹、sd 卡和 sd 卡内的录音文件夹这四个部分看成不同的设备。
设备的定义
```

```
enum
{
    NO_DEVICE = 0,
    DEVICE_SDMMC0 = 1,
    //DEVICE_SDMMC1,
    DEVICE_UDISK = 2,
    DEVICE_SDMMC0_REC = 0x41, //sd 卡的录音文件夹
    DEVICE_UDISK_REC = 0x42, //u 盘的录音文件夹
    DEVICE_AUTO_NEXT = 0x81,
    DEVICE_AUTO_PREV = 0x82,
    MAX_PLAY_DEVICE = 2,
};
```

还定义了 `VIRTUAL_DEVICE` 用来判断设备是不是录音文件夹

```
#define VIRTUAL_DEVICE (DEVICE_SDMMC0_REC & (~DEVICE_SDMMC0))
```

当 `VIRTUAL_ENABLE` 定义为 0 时，关虚拟识别使能。设备内的录音文件和音乐文件不分开处理。

## 5、找设备

找设备的消息 MSG\_MUSIC\_SELECT\_NEW\_DEVICE 放在热键函数 void ap\_handle\_hotkey(u8 key) 里面处理, u8 find\_device(u8 select) 因为加入了虚拟功能, 因而找设备函数就有所改变, 在找上下一个设备时就用查表的方法来找上下设备, 在判断设备时留意 device\_active 是要 & VIRTUAL\_DEVICE 还是 & (~VIRTUAL\_DEVICE)。

## 6、设备初始化

在文件系统初始化后就是要记录下录音文件夹里第一个文件号和总文件数, 并且记录该设备的文件总数 logic\_fileTotal。

## 7、获取文件号函数 u8 fs\_get\_filenum(u8 playmode, u8 searchMode)

在此函数中用到的 given\_file\_number 就是该设备的逻辑文件号, 在函数里面要留意文件号的转换, 在播录音文件夹和音乐文件分别用不同的播放模式变量 play\_mode\_rec 和 play\_mode 来记录当前的播放模式。虚拟识别开后, 录音文件夹有倒序播放功能。

## 8、根据文件序号找文件

在调用 fs\_getfile\_bynumber 函数来找文件时, 逻辑文件号 given\_file\_number 要传到 filenum\_logic\_phy (u16 filenumber) 函数中转换成物理文件号再传到 fs\_getfile\_bynumbe (u8 playmode, u8 searchMode) 函数中找文件。

## 9、断点记忆

在各断点记忆相关的函数中先调用 void get\_rec\_mem\_info(void) 函数来为设备断点记忆做数据准备, 把 4 个设备的断点记忆都分开来记忆, 如果用虚拟识别的话, 要用 eepROM 来记忆。

### 10、文件号转换

u16 filenum\_phy\_logic(u16 filenumber) 文件号从物理文件号转换为逻辑文件号  
 u16 filenum\_logic\_phy(u16 filenumber) 文件号从逻辑文件号转换为物理文件号

1
2
3
4
5
6
7
8
9
10

在设备初始化时，扫描文件顺序如左图，把这种排序后的文件号定义为物理文件号，假如 u 盘的录音文件夹的文件为 5、6、7，其他的都为 u 盘的音乐文件号，那当选择录音文件夹为当前设备时，录音文件夹的物理文件号 5、6、7 转换为相应逻辑文件号为 1、2、3，而 u 盘的音乐文件 1、2、3、4、8、9、10 转换为相应逻辑文件号为 1、2、3、4、5、6、7。文件号的转换主要运用以上的两个函数。

### 11、录音部分

录音功能嵌入到 music 模式、fm 模式和 linein 模式，而 mic 的录音会跳到 mic 模式去录音。主要是把录音的各种功能按键消息放在热键函数 void ap\_handle\_hotkey(u8 key) 中处理，在录音的过程中定义了录音过程中各种状态

```
enum
{
    RECODE_STOP    = 0,    ///<未录音状态
    RECODE_PLAY,        ///<播放录音文件状态
    RECODE_INIT,        ///<录音初始化
    RECODE_WORKING,    ///<正在录音
    RECODE_PAUSE,    ///<录音暂停
};
```

录音的主要函数

u8 f_write_init(u8 dev)	录音设备初始化，dev:录音设备
u8 start_encode(u8 type)	开始录音                      type: 录音文件格式，otp 录音只支持 mp3 格式
void stop_encode(void)	结束录音
void pause_encode(void)	暂停录音
void continue_encode(void)	暂停录音后继续录音

录音按键和暂停按键跟据录音各种状态，发送开始录音、结束录音、暂停录音和继续录音的消息，在开始录音前先要找给定设备，初始化各种录音的参数，然后就调用 `start_encode()` 函数来开始录音，当结束录音后会跳到刚录完的文件播放。

## 12、增加的功能

在 `music` 中加入了删除文件功能和在线设备的切换功能，如果只想删除录音文件夹内的文件，就在删除消息一开始判断是不是在播录音文件夹的判断拦截