



Slic Separate Porting SOP

Version: 1.2
Release date: 2016-01-22

2008 - 2016 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc.

Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.

Specifications are subject to change without notice.

Document Revision History

Revision	Date	Author	Description
1.1	2016-01-06	Yafei.Ren	File created
1.2	2016-01-22	Yafei.Ren	1.删除 IOCTL 方式, 增加动态库和静态库接口方式 2.直接读写 添加 spi reset 和 cfg 接口。 3.注意事项里增加 CSI 和 ISI 时隙 说明。

1. Introduction

本文档主要目的：介绍采用7526/7512/7521系列SOC，VoIP slic分离时，加载语音驱动和使用SPI读写接口的方法。

2. 加载语音驱动

VoIP 相关的驱动加载顺序如下：

```
insmod sys_mod.ko
insmod DSPCore.ko
insmod pcm1.ko
#insmod pcm2.ko
insmod lec.ko
insmod spi.ko
insmod slic3_sep.ko type=SPI fxsNum=1 fxoNum=0 devNum=1 pcm1ChNum=1
pcm2ChNum=0
insmod fxs3.ko slic1="le9641"
insmod ksocket.ko
insmod ortp.ko
insmod acodec_x.ko
insmod foip.ko
insmod ovdsp.ko
insmod pcmDump.ko
```

说明：

(1) 存在两颗slic，并且slic接口类型都为ZSI或者ISI时，需要加载pcm2.ko，且第二颗slic会挂载在pcm2上。其余情况都不需要加载pcm2.ko，slic都挂载在pcm1上；

(2) 加载 slic3_sep.ko 参数说明：

type: 可配置类型包括三种 SPI ,ZSI,ISI

fxsNum: 所有 slic 的 fxs 个数和

fxoNum: 所有 slic 的 fxo 个数和

devNum: slic 个数

pcm1ChNum: pcm1 上挂载的所有 slic 的 fxsNum+fxoNum

pcm2ChNum: pcm2 上挂载的所有 slic 的 fxsNum+fxoNum

(3) 当 slic 厂商为 microsemi 时，应加载 fxs3.ko；当 slic 厂商为 silicon labs 时应加载 fxs3_silicon.ko；

(4) fxs3.ko 或者 fxs3_silicon.ko 应带上 slic 型号的参数，例如：

单slic（每颗slic有1个fxs口）： insmod fxs3.ko slic1="le9641"

双 slic（每颗 slic 有 1 个 fxs 口）： insmod fxs3.ko slic1="le9641" slic2="le9641_1"

单slic（每颗slic有2个fxs口）： insmod fxs3.ko slic1="zl88601" slic2="zl88601"

双slic（每颗slic有2个fxs口）：

```
insmod fxs3.ko slic1="zl88601" slic2=" zl88601" slic3="zl88601_1" slic4="
zl88601_1"
```

EECONF

3. SPI 读写接口（直接读写）

spi.ko 中提供了 SPI reset,cfg 和读写接口函数，客户可以直接调用。

函数名如下：

```
int SPI_Reset(int interface_type);  
int SPI_cfg (int slic_type , int id);  
int SPI_bytes_read(int slic_type, int interface_type, int id, unsigned char ctrl, unsigned char cmd, unsigned char *data_ptr, unsigned char cmdlen);  
int SPI_bytes_write(int slic_type, int interface_type, int id, unsigned char ctrl, unsigned char cmd, unsigned char *data_ptr, unsigned char cmdlen);
```

参数说明如下：

参数	参数说明
slic_type	表示 slic 厂商， 0 表示 microsemi 产品， 1 表示 siliconlabs 产品， 2 表示 lantiq 产品
interface_type	0 表示 SPI， 1 表示 ZSI， 2 表示 ISI， 3 表示 CSI
id	0 表示第一颗 slic， 1 表示第二颗 slic
ctrl	slic 控制值 microsemi:no use; siliconlabs:read 0x70,write 0x30; lantiq: read 0xbe, write 0x7e
cmd	读写的 slic reg 地址(zarlink 用地址奇偶控制读写，偶为写，奇为读)
data_ptr	写入 data 的指针或者读取 data 的指针
cmdlen	写入 data 或者读取 data 的有效长度

4. SPI 读写接口（动态库和静态库接口方式）

动态库 libsllic_ctrl.so 和静态库 libsllic_ctrl.a 中都提供了以下三个函数接口：

```
int slic_ctrl_init (spi_cmd_struct *cmd);
```

```
int slic_ctrl_rw(spi_cmd_struct *cmd);
```

```
int slic_ctrl_pressure_test_lantiq(spi_cmd_struct *cmd);
```

参数说明：

cmd 是 spi_cmd_struct 类型结构体。

```
typedef struct _spi_cmd_struct {
    unsigned char slic_type_id; // 0 zarlink, 1 silicon lab, 2 lantiq
    unsigned char interface_type_id; // 0 spi, 1 zsi, 2 isi, 3 csi
    unsigned char slic_id; // 0 the first slic, 1 the second slic
    unsigned char ctrl;
    unsigned char cmd;
    unsigned char cmdLen;
    unsigned char data[32]; // write or read data
} spi_cmd_struct;
```

参数	参数说明
slic_type_id	表示 slic 厂商， 0 表示 microsemi 产品， 1 表示 siliconlabs 产品， 2 表示 lantiq 产品
interface_type_id	0 表示 SPI， 1 表示 ZSI， 2 表示 ISI， 3 表示 CSI
slic_id	0 表示第一颗 slic， 1 表示第二颗 slic
ctrl	slic 控制值 microsemi:no use; siliconlabs:read 0x70,write 0x30; lantiq: read 0xbe, write 0x7e
cmd	读写 slic reg 地址(zarlink 用地址奇偶控制读写，偶为写，奇为读)
data[32]	写入或者读取的数据
cmdlen	写入 data 或者读取 data 的有效长度

三个函数的使用,可以参考 slic_ctrl_test.c。

slic_ctrl_test 是可执行文档，是调用这三个函数的 demo.

以 lantiq 为例，SPI 接口时，reset_and_cfg，写，读，压力测试命令参考：

```
slic_ctrl_test init 2 0  
slic_ctrl_test 2 0 0 0x7E 0x25 2 5 6  
slic_ctrl_test 2 0 0 0xBE 0x25 2  
slic_ctrl_test test 0
```

以 lantiq 为例，CSI 接口时，reset_and_cfg，写，读，压力测试命令参考：

```
slic_ctrl_test init 2 3  
slic_ctrl_test 2 3 0 0x7E 0x25 4 1 2 1 2  
slic_ctrl_test 2 3 0 0xBE 0x25 4  
slic_ctrl_test test 3
```


5. 其它注意事项

对于 ZSI 和 CSI 接口，起始 timeSlots idx 值为 2。