

第一步，定义一个结构体

```
#define PLMN_LEN 6

typedef struct
{
    kal_uint8 HPLMN[PLMN_LEN]; //Home PLMN
    kal_uint8 RPLMN[PLMN_LEN]; //实网下的PLMN
} Plmn_Id_Struct;

const Plmn_Id_Struct plmn_list[] =
{
    //移动卡
    {46000, 64F000}, //64F000是BCD码，转换成字符串是46000
    {46002, 64F000}, //...
    {46004, 64F000}, //...
    {46007, 64F000}, //...
    {46008, 64F000}, //...
    //联通卡
    {46001, 64F010}, //64F010是BCD码，转换成字符串是46001
    {46006, 64F010}, //...
    {46009, 64F010}, //...
};
```

第二步，AT+CIMI查询HPLMN，取前5位，比如460040689501984中前5位是46004。故该卡HPLMN是46004。

第三步，用第二步中的HPLMN在plmn_list[]中查找实网下的PLMN。比如HPLMN 46004对应的实网下PLMN是64F000。

第四步，AT+CRSM=176, 28539, 0, 0, 12查询FPLMN List，将第三步匹配的实网下PLMN，在FPLMN LIST中逐一匹配。匹配成功则执行AT+CRSM=214, 28539, 0, 0, 12, "FFFFFFFFFFFFFFFFFFFFFFFF"清除。匹配不成功，直接跳过，不操作。（此处需注意不要匹配FFFFFFF）

第五步，复位重启。

示例：

AT+CIMI

460040689501984 // 查询HPLMN是46004

OK

//在 plmn_list[] 中查找 46004 对应的实网下 PLMN 是 64F000

AT+CRSM=176,28539,0,0,12 //查询FPLMN

+CRSM: 144,0,"64F01064F000FFFFFFFFFFFF" //46004对应的实网下PLMN 64F000在里边，需要清除。

OK

AT+CRSM=214,28539,0,0,12,"FFFFFFFFFFFFFFFFFFFFFFFF" //清除

+CRSM: 144,0

OK

//复位重启