

# ATV 基本技能

ATV 部分主要熟悉几个基本功能的流程：ATV 自动搜台、ATV 手动搜台、VIF 参数的调整、ATV 换台流程、ATV 非标参数调整等。由于ATV 在全球的不同地区对应不同的制式，导致不同的地区的搜台、换台、频道管理等都有一些差异, 本文档让大家简单的了解ATV 的一些基础知识。

## 1. ATV 搜台流程

### 1.1 全球各制式介绍

全球各个制式对应的行数/场频/载波频率/带宽/分布范围详见下图，  
更详细信息可以参考文档《世界各国电视制式.doc》。

Worldwide Video						
Video Standar	LINE	Fvsyn c (Hz)	FSC (MHz)	AUDIO CARRIER (MHz)	CHANNEL BANDWIDT H (MHz)	Note
NTSC M	525	59.94	3.579545	4.5	6	U.S.,man y others
NTSC J	525	59.94	3.579545	4.5	6	Japan
NTSC 4.43	525	59.94	4.43361875	4.5	6	NTSC Video [PAL color subcarrier is 4.43MHz(Son g 1070~)]
PAL B,B1,G,H	625	50	4.43361875	5.5	B=7 B1,G,H=	Europe,Many
PAL D,K	625	50	4.43361875	6.5	8	China
PAL M	525	59.94	3.57561149	4.5	6	Brazil
PAL N	625	50	4.43361875	5.5	6	many
PAL Nc	625	50	3.58205625	4.5	6	Argentin
PAL I	625	50	4.43361875	5.9996	8	Great Britain,others

SECAM D,K,K1,	625	50	Dr=4.40625 0	6.5	8	France
SECAM B,G	625	50	Db=4.25000	5.5	B=7 G=8	Russia

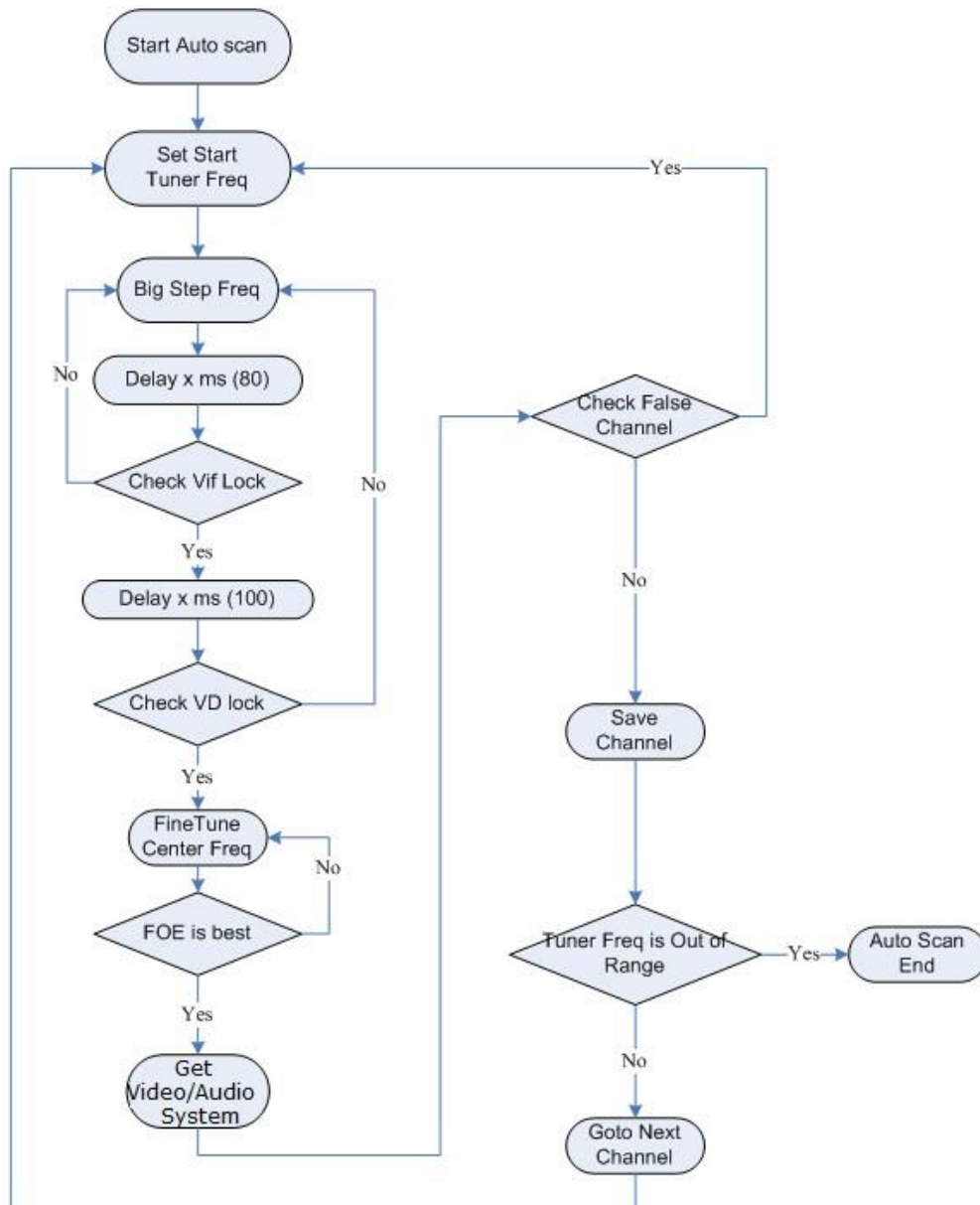
Figure 6.1

## 1.2 ATV 美规自动搜台流程

- ◆ Type:
  - ANT: Air TV
  - CATV: Cable TV
- ◆ Total 181 channel number
- ◆ Channel 2~13 is same Freq between ANT&CATV
- ◆ CATV standard:
  - STD
- ◆  $\text{Frequency\_in\_MHz} = (6 * (8 + \text{channel\_designation})) + 1.25$ 
  - IRC
- ◆  $\text{Frequency\_in\_MHz} = (6 * (8 + \text{channel\_designation})) + 1.2625$ 
  - HRC
- ◆  $\text{Frequency\_in\_MHz} = 6.0003 * (8 + \text{channel\_designation})$
- ◆ 美规 NTSC 中分为STD/HRC/IRC，因为三种规格的频道表都是固定的，所以搜台流程比欧规简单，不需要步进跳跃频点，只需要根据频道表设定频点，并查看当前频点是否有信号即可。有两种做法：
  - 做法一：分STD/HRC/IRC 三张频道表分别遍历，找到有信号的频点就存下来。
  - 做法二：只按STD 一张频道表进行遍历，每个频点上还需要检查+/- 1.5M 的范围内 是否有信号，这样可以完全覆盖到另外两张频道表。

### 1.3 ATV 内销自动搜台流程

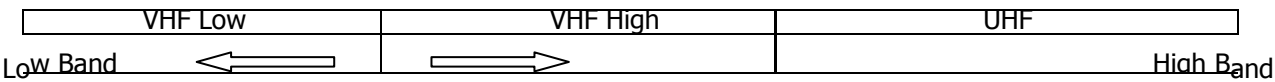
ATV 内销机自动搜台流程请参考下图，具体的 code flow 在函数 MW\_ATV\_Scan\_AsiaChina::DoScan()实现。相比美规， 频率表不固定。



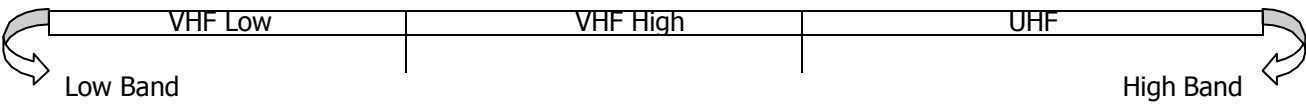
1.4 手动搜台流程

PAL 制手动搜台流程过程基本和自动搜台相同，主要有以下三点差别：

1) 手动搜台有从低频点往高频点搜索和从高频点往低频点搜索两个方向；自动搜台只有从低频点往高频点搜索一个方向。



2) 手动搜台在到达高频点（或低频点）边界时，不会自动停止，需要跨过边界，从低频频点（或高频点）继续保持原来方向搜台；自动搜台则是到达高频点边界时会自动停止。



3) 为了避免邻频假台干扰,手动搜台的起始频点是在当前频点基础上+/- 一个频道的带宽， 对于内销机，频道带宽是 8MHz。所以，起始频点=当前频点 +/- 8MHz, 但内销机为了避免 漏台，步长一般改成了+/-2MHz。

N 制的话我们就做的比较简单，直接根据频道号改变频点，然后其他大致和自动搜台时的遍历相同。

1.5 常见搜台问题分析

1) 声音制式检测错误

可能原因：

A. 信号不稳定，建议在 audio 制式检测之前可以加一些delay；如果确定是前端信号不稳定，要HW 人员从前端做参数调整和改善。

B. Audio 的 threshold 不合理。这部分目前很少遇到了，但是如果遇到，请调整 mapi\_audio\_customer.cpp 里面的

AuSiflnitThreshold\_PAL\_SIF[],AuSiflnitThreshold\_PAL\_VIF[],AuSiflnitThreshold\_BTSC[]

具体调整方法请找HQ audio team 咨询(Mstar 方面)。

2) 图像制式检测错误

A. 信号不稳定，建议在video 制式检测之前可以加一些delay；如果确定是前端信号不稳定，要HW 人员从前端做参数调整和改善。

B. 可以通过mapi\_interface::Get\_mapi\_vd()->GetVerticalFreq()获得场频来直接判别PAL 系统和NTSC 系统（仅限内销机使用）。

C. 对于某个频点总是识别错误，可以先把频点微调过去，然后直接读寄存器看一下结果

Check Video Standard	AFEC_CC[1]: Burst on/off AFEC_CC[7:5]: FSC AFEC_CD[4]: V Total line number, 1:625 Lines, 0:525 Lines AFEC_CD[1]: PAL Switch, 1:PAL Group, 0:NTSC Group
----------------------	---

D. 确认有打开burst detection。

## ◆ Burst Detection Result

```
AFEC_CC[1]: 1:Burst On, 0: No Burst
AFEC_CC[7:5]: VD FSC Type,
000: 4.43M, PAL or NTSC443
001: 4.285M, SECAM
010: 3.5795M, NTSC
100: 3.5756M, PAL-M
110: 3.5820M, PAL-Nc
```

## ◆ Burst Detection Enable/Disable

```
AFEC_CB[0]: FSC PAL 1: disable 0:enable
AFEC_CB[1]: FSC SECAM 1: disable 0:enable
AFEC_CB[2]: FSC NTSC 1: disable 0:enable
AFEC_CB[3]: FSC NTSC443 1: disable 0:enable
AFEC_CB[4]: FSC PAL-M 1: disable 0:enable
AFEC_CB[6]: FSC PAL-Nc 1: disable 0:enable
```

Figure 6.4

3) 假台多 假台和漏台是搜台的两个矛盾问题。如果锁频的灵敏度太弱，就会导致信号锁不住漏

台；如果锁频的灵敏度太强，就会导致假台太多。为了保证不漏台，我们一般会设置锁频灵敏度偏强一些，所以会导致出现一些假台出现。

所谓假台，就是频域上的强信号干扰，或者强信号台的倍频辐射。正常出现一两个假台 是没有问题的，如果假台太多，就需要利用check false channel 的流程来判断当前台是否符合真台的一些特征如果不符合就把它剔除掉因此如果假台多就说明 check false channel 的流程还不够严苛。

可以通过调整软件严格卡关真台的特征条件来过滤假台，真台的特征有：  
VIF/VD 一定是lock 的、一定有audio 负载波且单边带、一定符合AFC 曲线。

4) 漏台

首先打开MW\_ATV\_Scan\_Customer.cpp 里面的 debug message对比message 查看code flow。

```
#define ATV_Scan_Cus_DBG(fmt, arg...)      printf((char *)fmt, ##arg)
#define ATV_Scan_Cus_FLOW_DBG(fmt, arg...) printf((char *)fmt, ##arg)
#define ATV_Scan_Cus_IFO(fmt, arg...)      printf((char *)fmt, ##arg)
```

漏台情况可分为下面几种：

A. VIF 没有 lock 住，就跳走了。 检查方法：

① 查看log 中是否在PAL\_SCAN\_SIGNAL\_CHECK\_BY\_VIF 步骤之后，就跳回PAL\_SCAN\_START 了。

(3) 停止搜台，将频点微调值漏台的频点，确认此频点的信号好坏。

可以通过读取VIF 的CR lock 寄存器来确认是否有lock:

CR lock: DBB1\_39[0] = 0 (unlock); 1 (lock)

(4) 如果step2 的检查是unlock，则说明进到VIF 的信号确实太烂，看是否可以调整前端来改善。此时可以联系HW 人员协助(Mstar 方面)。

(5) 如果step2 的检查是lock，则需要在搜台流程中再增加打印追查为什么搜台过程中到达此频点时VIF unlock。此时可以联系VIF team RD 协助(Mstar 方面)。

B. VD Hsync 没有 lock 住，就跳走了。

检查方法:

(1) 查看 log 中是否在PAL\_SCAN\_SIGNAL\_CHECK\_BY\_VD 步骤之后，就跳回PAL\_SCAN\_SIGNAL\_WEAK 了

(2) 检查VD detect 的灵敏度是否有问题，可以尝试调节灵敏度 参数设定在vd.ini 里面:

```
[VD_SENSIBILITY]
VD_HSEN_NORMAL_DETECT_WIN_BEFORE_LOCK =
0x08; VD_HSEN_NOAMRL_DETECT_WIN_AFTER_LOCK
=                                0x08;
VD_HSEN_NORMAL_CNTR_FAIL_BEFORE_LOCK = 0x0F;
VD_HSEN_NORMAL_CNTR_SYNC_BEFORE_LOCK =
                                0x10;
VD_HSEN_NORMAL_CNTR_SYNC_AFTER_LOCK =
                                0x1C;
VD_HSEN_CHAN_SCAN_DETECT_WIN_BEFORE_LOCK = 0x06; #TVOS
                                0x06,SN 0x04
VD_HSEN_CHAN_SCAN_DETECT_WIN_AFTER_LOCK = 0x06;      #TVOS
                                0x06,SN 0x04
VD_HSEN_CHAN_SCAN_CNTR_FAIL_BEFORE_LOCK = 0x0F;      #TVOS
                                0x0F,SN 0x08
VD_HSEN_CHAN_SCAN_CNTR_SYNC_BEFORE_LOCK = 0x18;      #TVOS
                                0x18,SN 0x30
VD_HSEN_CHAN_SCAN_CNTR_SYNC_AFTER_LOCK = 0x0E;
```

如果尝试调节无效果，请联系HQ VD team RD (Mstar 方面)。

C. 满足了 check false channel 的条件，真台被当作假台过滤掉了 如果真台信号太强，导致信号有扭曲，在某一时刻，真台的特征条件被破坏，这时候用 check false channel 的条件去卡关，就会被认为是假台。

可以改善的措施有两个:

(1) 适当放宽卡关条件。(此方法要慎用，如果用不好会有副作用，导致假台激增)

(2) 增加卡关次数，并统计通过率，这样可以降低单次判断的误差。(此方法有副作用，会加长搜台时间，需要适当调整)

5) 搜台死循环 从软件逻辑设计的角度来讲，搜台流程是不存在死循环的，但是在实际情

况中，如果出现强信号干扰的假台，或者邻频干扰的时候，信号不稳定导致时而能lock，时而unlock 的状况，这时候可能会超出预期，变成死循环。

重点通过log 检查code flow 在PAL\_SCAN\_SIGNAL\_CHECK\_BY\_VIF 到 PAL\_SCAN\_FINE\_INCREASE，PAL\_SCAN\_FINE\_DECREASE 之间为什么来回跳转。如果有必要可以在检查完VIF 和VD lock 之后进行超时强制退出死循环。

## 2、ATV 换台流程

ATV 换台流程如下图所示。主要分两阶段，第一阶段是换台前的准备，比如黑屏等；第二阶段是设置频点/彩色制式/伴音制式/Set XC window 等，具体 code flow 在函数 StartChangeChannel()和ProgramChangeProcess()实现。

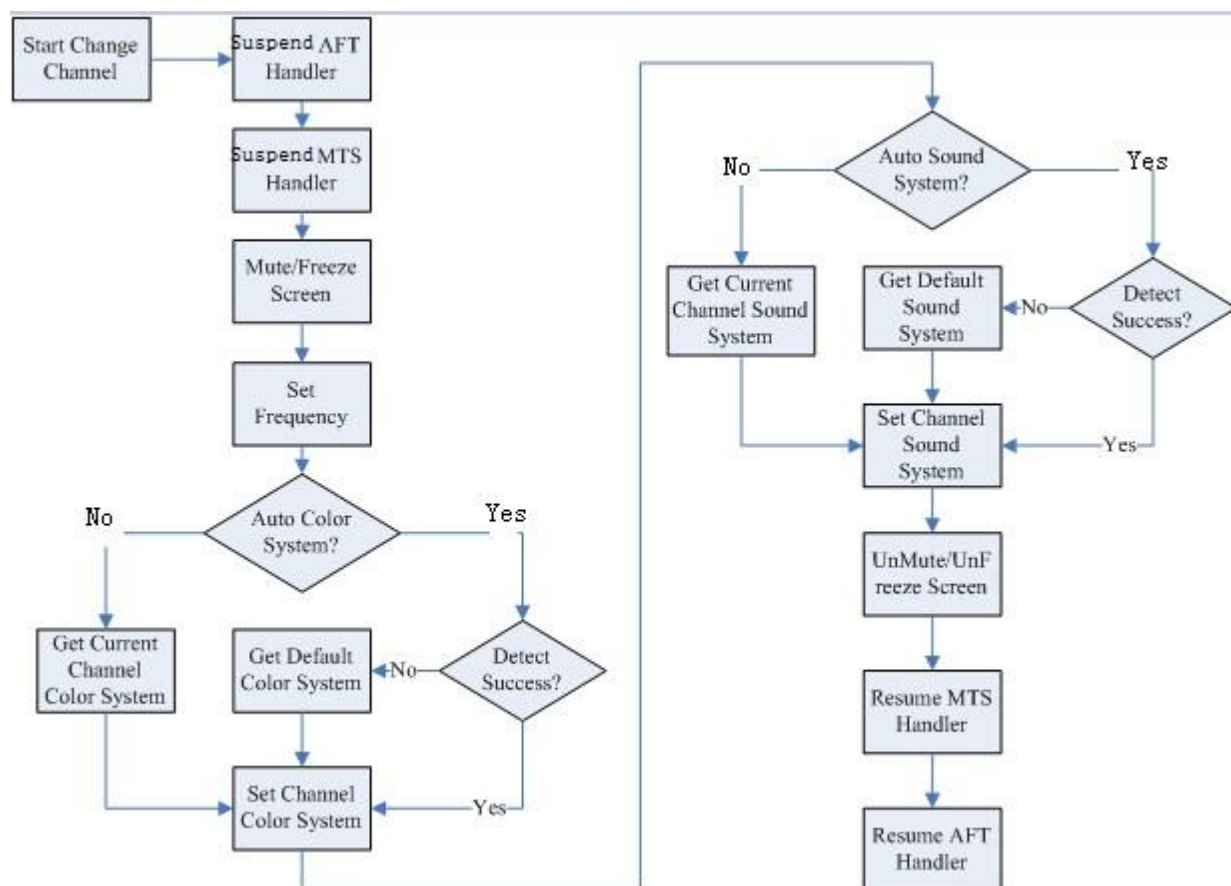


Figure 6.5



### 3. ATV 节目信息管理

#### 1) 单个节目信息数据结构

ATV 基本的节目信息定义在结构体ST\_ATV\_PROGRAM\_DATA，它包含了频道频点，频道属性，频道微调步长，频道名称等关键信息。具体请参考MSrv\_ATV\_Database.h。

```
/// the structure of one program
typedef struct
{
    /// Tuner PLL Value
    U16 wPLL;
    /// Program misc info
    ST_ATV_MISC Misc;
    /// is program sort?
    U8 u8Sort;
    /// program fine tune frequency
    S8 s8FineTune;
    /// Program Name
    U8 sName[MAX_STATION_NAME];
    /// to save the list page of this program
    U8 u8ListPage[MAX_LISTPAGE_SIZE];
} ST_ATV_PROGRAM_DATA;
```

```
/// the channel attributes
typedef struct
{
    /// the audio standard
    U16 eAudioStandard : 4;
    /// is skip or not
    U16 bSkip : 1;
    /// is hide or not
    U16 bHide : 1;
    /// the video standard
    U16 eVideoStandard : 4;
    /// is dual audio selected or not
    U16 bWasDualAudioSelected : 2;
    /// Volume compensation for every channel.
    U16 eVolumeComp : 4;
    /// the audio mode
    U16 eAudioMode : 4;
    /// is realtime audio standard detection or not
    U16 bIsRealtimeAudioDetectionEnabled : 1;
    /// Favorite setting value
    U16 u8Favorite : 8;
    /// is Medium type Cable or Air
    U16 eMedium : 1;
    /// is lock or not
    U16 bIsLock : 1;
    /// the channel number
    U16 u8ChannelNumber : 8;
    /// is auto frequency tuning enable
    U16 bAFT : 1;
    /// is Mode direct tune or manual tune?
    U16 bIsDirectTuned : 1;
    /// To record the offset of AFT
    U16 u8AftOffset : 8;
    #if (ISDB_SYSTEM_ENABLE == 1)
    /// is Auto color or not
    U16 bIsAutoColorSystem : 1;
    /// is Auto color or not
    U16 Unused : 5;
    #endif
} ST_ATV_MISC;
```

Figure 6.7



## 2) 节目信息数据存储结构

ATV 的节目信息存放在平台的/tvdatabase/Database 下，atv\_cmdb.bin 存放 air 的节目，atv\_cmdb\_cable.bin 存放cable 的节目。内销机只有air，所以存放在atv\_cmdb.bin。bin 档内 部的数据排布和结构体ST\_ATV\_PROGRAM\_DATA\_STRUCTURE 对应。

```
/// the structure of programs saved in
database typedef struct
{
    /// last program number
    U8          u8LastPRNumber;
    /// program index table
    U8          u8ATVPRIndexTable[MAX_NUMBER_OF_ATV_PROGRAM];
    /// program table map
    U8          u8ATVPRTableMap[MAX_PRTABLEMAP];
    /// the program data
    ST_ATV_PROGRAM_DATA ATVProgramData[MAX_NUMBER_OF_ATV_PROGRAM];
    /// the tag for mark the DB has been
    initialized U8    u8DBInitTag;
```

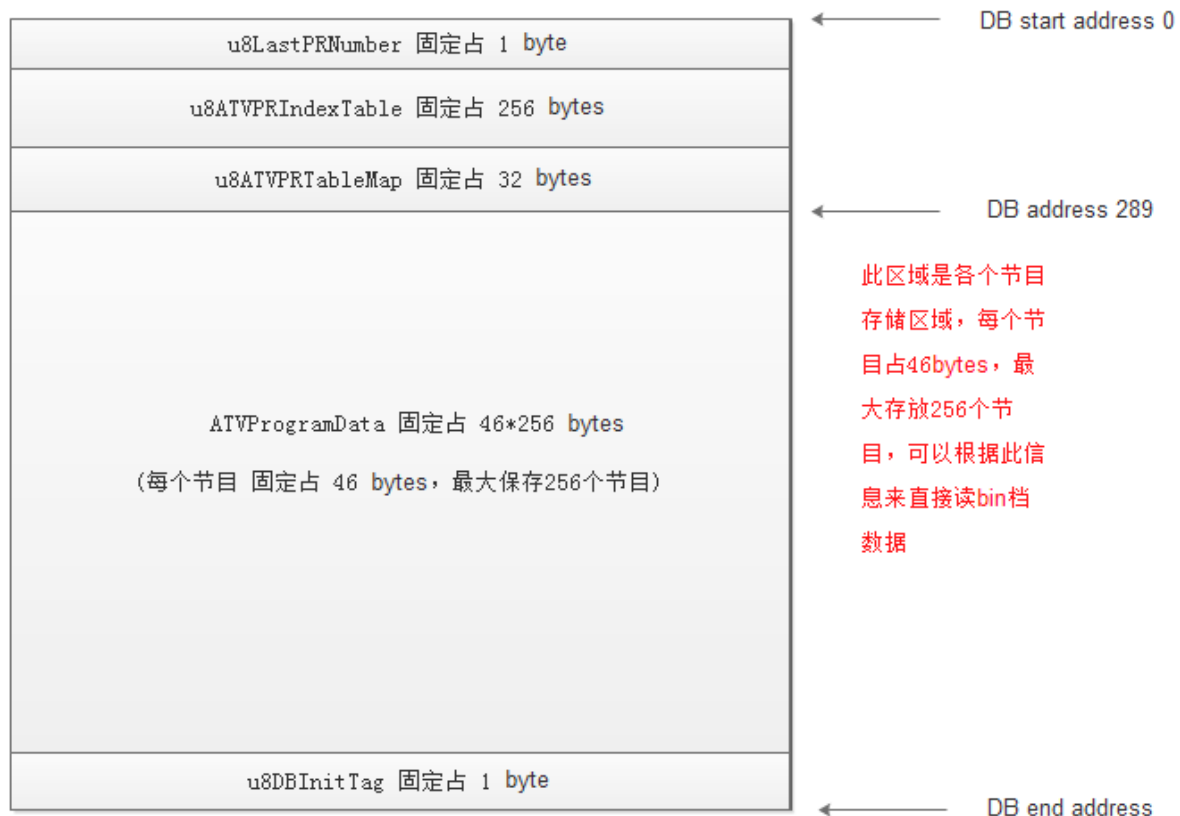


Figure 6.8

### 3) 节目信息读取和映射

下面是ATV DB 的初始化和读取流程图：

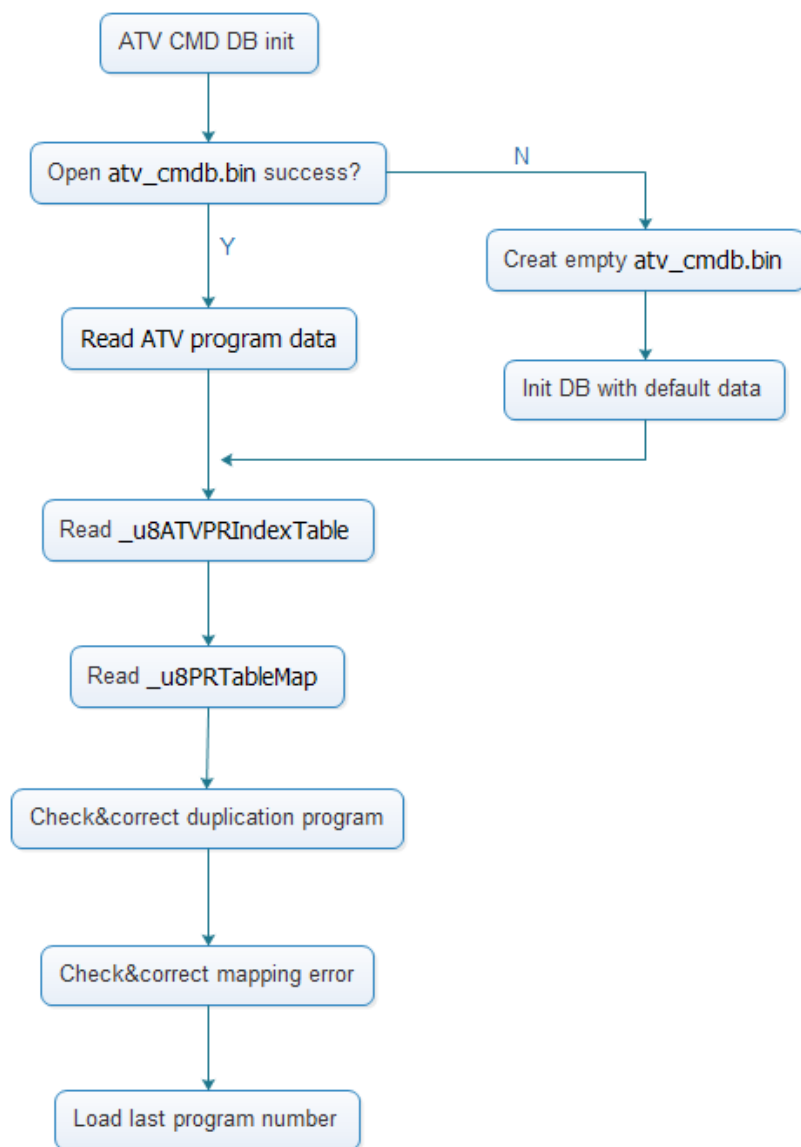


Figure 6.9

开机init 时，在Get\_mapi\_storage\_atv\_cmdb()->Init()里去 fopenatv\_cmdb.bin，若不存在， 则创建一个空白 bin 档,并把 ST\_ATV\_PROGRAM\_DATA\_STRUCTURE 各个成员赋值成 0 存到 bin 档里。

之后再用fread 去读取bin 档的节目数据，并存到ST\_ATV\_PROGRAM\_DATA\_STRUCTURE 结 构体变量stProgramData 里。

紧接着调用InitATVDataManager(), 从DB 里读出\_u8ATVPRIndexTable 和 \_u8PRTTableMap。

\_u8ATVPRIndexTable 是一个 MAX\_NUMBER\_OF\_ATV\_PROGRAM 长度的数组它用来保存每个节 目的index。\_u8PRTTableMap 是一个MAX\_PRTABLEMAP 长度的数字，它的每一个bit 和每一个 节目index 对应起来，若此index 有节目，对应的\_u8PRTTableMap 比特位会被置成1。其

中  $\text{MAX\_PRTABLEMAP} = ((\text{MAX\_NUMBER\_OF\_ATV\_PROGRAM} + 7) / 8)$ 。

之后会去判断节目是否重复和 Mapping 是否有错，并会自动纠正过来。最后取出当前的节目号，即最后一次播放的节目号。

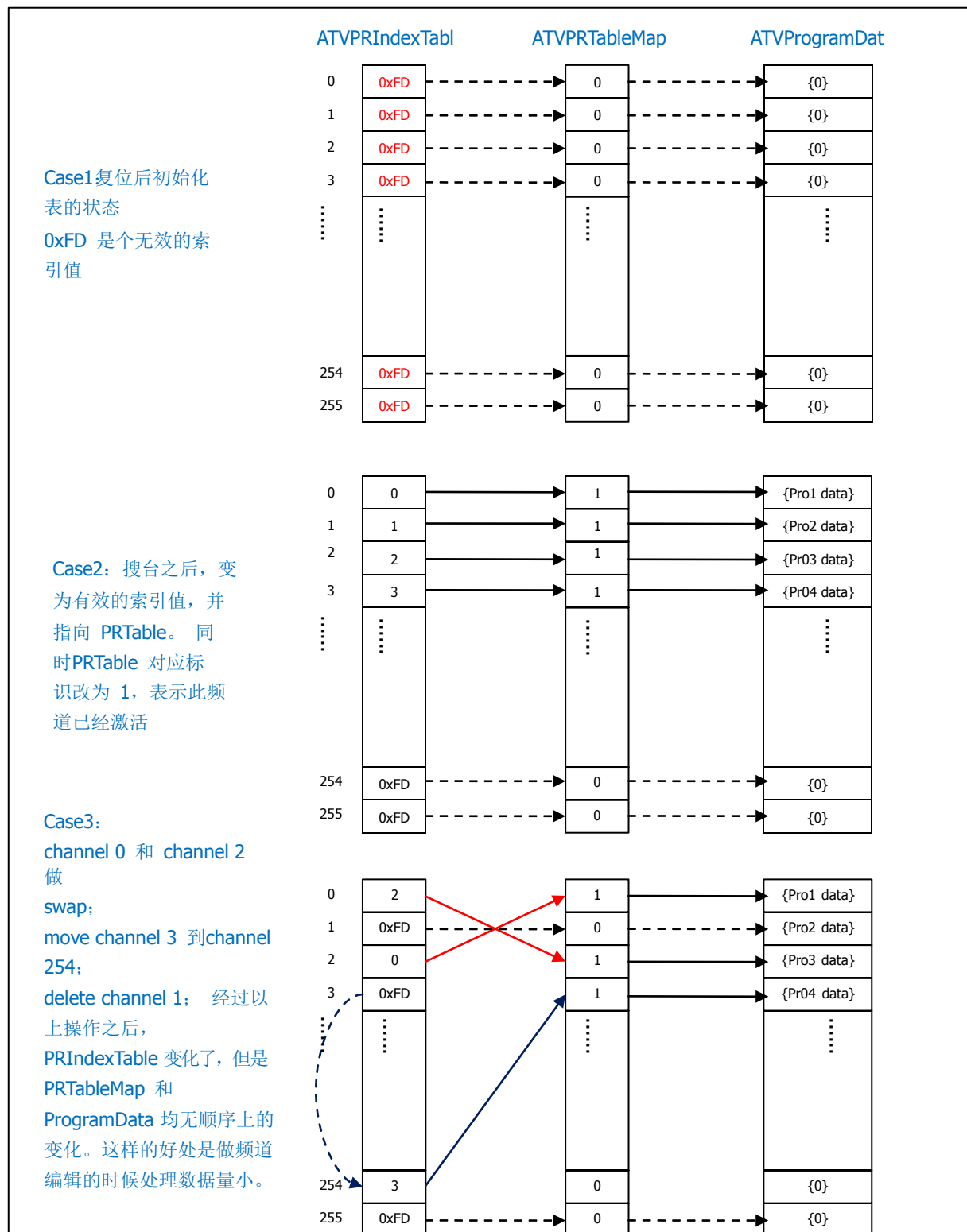


Figure 6.10

4) 节目编辑 常用的节目编辑操作包括： 频道交换：

ATVSwapProgram() 频道移动：ATVMoveProgram() 频道复制：

ATVCopyProgram() 频道删除：ATVDeleteProgram() 频道排序：

ATVSortProgram() 频道锁定：ATVLockProgram() 频道命名：

ATVSetStationName() 频道喜好设置：ATVSetFavoriteProgram() 频道隐

藏：ATVHideProgram() 频道跳跃：ATVSkipProgram() 频道微调：

ATVSetAftOffset()