

数模一体电视机通用处理方案

随着技术的不断发展，电视广播由模拟技术转换到数字技术的趋势已经成为必然。目前很多地方有数字电视的信号播出，而且大部分城市在数字电视转换过程中采取了机顶盒方式，机顶盒方式虽然可以接收数字电视信号，但是在机顶盒把接收来的数字电视信号输出给电视机时，又将其转换成了模拟信号，这样无形中损失了图像质量。而且用户在操作时不是很方便：需要用连接线连接机顶盒和电视机、同时需要两个遥控器等。数模一体电视机很好的解决了这个问题，一台电视机既可以接收传统的模拟电视信号，也可以接收最新播出的数字电视信号。但是早期的数模一体电视机是把接收数字电视的模块放在普通的逐行电视中，此种实现方式几乎和机顶盒方式效果一样，只是外表看起来是一个整体。而现在的方案基本上是单芯片方案，即数字电视解码处理和模拟通道信号处理在同一个芯片中实现。数字电视信号被解码后直接在芯片内部进行 **SCAler**(缩放)等处理后输出，最大限度地保证了图像质量不受损失，真正满足了用户的需求，将成为市场的最佳选择。

数字电视与模拟电视不同之处还在于数字电视可以加密，这样运营商就可以很方便的控制着用户，而不担心欠费现象。用户要想接收数字电视需得到运营商的许可，即 **CA**(条件接收)，这也是运营商想大力推广数字电视的根本原因。运营商给每个电视或机顶盒分配一个 **CA** 卡(俗称小卡)，里面有用户识别等信息，小卡是装在大卡里面的，大卡里面有数字电视信号加密的有关信息，负责和电视机里面相关的接口芯片通讯把数字电视信号解出来。大卡和电视机相连的接口有 **PCMCIA**、**UTI** 等。

PCMCIA 接口已经成熟并且颁布了行业标准，也是国际上通用的一种接口标准。一般的机顶盒上并没有看到大卡，这是因为机顶盒生产商是和运营商联合推广市场的，数字电视的加密算法已经确定，机顶盒生产商就可以把大卡做在机顶盒里面了。而电视机生产商是把电视机直接卖给用户的，不同于机顶盒的操作模式，为了适应不同的 **CA** 要求，本方案采取了机卡分离方式，即把 **CA** 大卡和整机分离开来，电视机上留有 **PCMCIA** 接口，对于不同 **CA** 的运营商配置相应的大卡即可，非常灵活方便，机卡分离也是数字电视的发展趋势。

数模一体机的系统

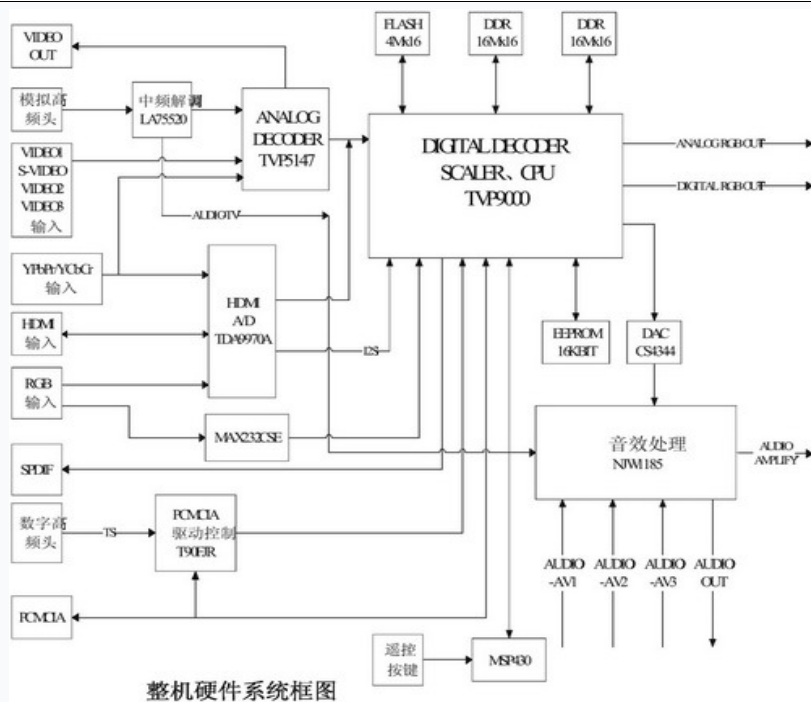
TI 公司发布的最新一款集成 **HDTV** 解码和视频处理的一体化 **SoC** 芯片 **TVP9000**，主要针对的就是数模一体电视机的市场。与其他芯片厂家提供的方案相比，其集成度较高，成熟度较佳，图像解码模块成熟，成本较低，具有很强的市场竞争力。本方案就是以 **TVP90**

00 为核心的数模一体电视机的解决方案，可广泛应用于 CRT 电视、平板电视、背投电视等。TVP9000 集成了 162MHZ 的 ARM926EJ CPU，HDTV-MPEG2 解码，图像处理，Scaler(缩放)等模块，功能强大。

本方案能够接收数字的电视信号(支持高清 HDTV 及标清 SDTV)和模拟电视信号，数字电视部分符合 DVB-C 标准，接收 16-256QAM 制式信号；多路输入接口：其中三路 AV 输入、一路 S 端子输入、一路 AV 输出、一路分量(YPbPr/YCbCr)输入、一路 RGB 输入、一路 HDMI 输入；支持多种信号格式如：480I、480P、576I、576P、720P、1080I、1080P 及 VGA、SVGA、XGA 等信号格式；能够通过 RGB 接口实现本地软件升级和更新，方便售后服务；满足 MPEG1/MPEG2 layers1、2 音频解码；SPDIF 光纤和左右声道音频同时输出；标准电子节目单(EPG)功能；支持 NVOD(准视频点播)；支持基于 DVB-C-CI 接口标准的付费电视节目服务。

如图 1 所示，模拟电视信号由模拟高频头输入经过 LA75520 中频解调输出视频和音频，视频信号送到 TVP5147 进行处理；音频信号送到 NJW1185 进行处理。各路 AV 输入的视频和 YCbCr 信号一起送到 TVP5147 和电视视频信号作切换后进行 A/D 变换、彩色解码后转换成 ITU.656 格式的信号送给 TVP9000，如果当前输入是模拟电视信号和 AV 时，TVP5147 同时输出一个复合视频信号作为监视用。

HDMI 信号送到 TDA9970A 变换成数字的 RGB 信号；模拟 RGB 信号、YpbPr 信号送到 TDA9970A 作切换、A/D 转换处理和 HDMI 信号再切换一次后进行格式变换，以双 ITU.656 的信号格式送到 TVP9000 和 TVP5147 送来的信号作切换。当选择 TVP5147 送来的信号时，要对信号进行去隔行(Deinterlace)、图象增强、缩放(Scaler)等处理；当选择 TDA9970A 输入的信号时，要对信号进行图象增强、缩放等处理。YCbCr 和 YpbPr 共用一个输入端子，自动识别格式后选择正确的输入通道。



数字电视信号由数字高频头输入，经过信道解调后输出加密的 TS 流信号送给 T90FJR，T90FJR 把加密的 TS 流信号通过 PCMCIA 接口送到大卡，大卡对加密的 TS 流信号解密后再进行二次加密通过 PCMCIA 接口和 T90FJR 再送到 TVP9000，TVP9000 对二次加密的 TS 流信号进行解密，然后进行数字解码、缩放处理后输出视频信号，其中第二次解密的算法由大卡向 TVP9000 提供。为什么要进行二次加密的原因是防止出现透明的 TS 流而被非法拷贝，从而有效的保证了节目的版权。

HDMI 声音由 HDMI 接口芯片 TDA9970A 输出到 TVP9000 和数字电视的声音切换后输出到 CS4344，同时输出一路数字音频信号 SPDIF 供外接的声音功放使用。CS4344 把输入的数字声音信号数模转换后送给 NJW1185 和送来的模拟电视声音、各路 AV 的声音信号切换、经过高音、低音、平衡、环绕声效果处理后送给功率放大器。如果当前输入是模拟电视信号和 AV、RGB、YCbCr/YpbPr 时，NJW1185 同时输出声音信号作为监视用。

软件架构

随着软件在整个电视产品中扮演着越来越重的角色，数字电视的发展趋势导致编写电视软件的复杂程度和工作量都比以往有了相当大的提高，已经不是个人能编写和完成的了。现阶段电视生产厂商的电视软件开发，无论是平台还是系统架构都是基于方案提供商提供，芯片供应商的千差万别导致开发的平台的不可重用性。每个供应商都有自己的一套系统架构，

这就造成在每遇到一个新的平台情况下,以前平台的功能和相关代码基本上没有可以重复利用的地方,都需要重新学习和编写新平台下的各种电视功能和功能逻辑,这种无法重利用以前的劳动成果是一种对开发的浪费,尤其在数字电视软件工作量急剧提升的情况下。

本方案的软件的特点是基于方案提供商提供的软件基础上再开发一个中间层软件,本中间层通过封装和重新架构底层电视功能函数和操作系统函数形成一个完整的数字电视上层框架。在尽可能的情况下,能够最快的移植到相关的电视环境中,并使模块能够最大化的复用到其他电视方案。设计并实现 **OSD** 库以及和底层的相关接口,模块化数据存储的定义以及和底层的相关接口,编写电视的声音,图像,搜台,信道切换等各种基本功能模块以及和底层驱动的中间层封装。软件代码具有独立性,耦合度低,具有通用性,经简单修改接口函数即可移植到其他的数模一体机平台上。具有自己独立的 **GUI** 库和行为架构将大大节省 **GUI** 的开发周期和开发复杂度,软件工程师在技术积累的基础上可以重复利用以前 **GUI** 的资源,兼容性和稳定性都将有很大的提高。电视行为控制的上层封装相当于拥有自己的 **SDK** 平台,基于此 **SDK** 平台的修改和扩展可以逐步形成自己的平台技术积累。模块用 **C** 语言开发,使用标准的 **ANSI C** 语言库,**GUI** 库使用 **C++** 语言开发,中间层是在底层库的基础上实现图像,声音,搜台,功能,节目管理等电视基本功能和逻辑。

该方案的软件是基于 **LINUX UCOS** 的平台,是具有多线程,多任务管理的操作系统。平台基本可以分为三层:固件,**API** 层,应用软件层。固件和 **API** 层由 **TI** 公司提供,用于底层的驱动和 **TVP9000** 解复用。在此基础上进行应用软件层的开发,主要是提供一些和用户接口的功能部分,用户菜单主要分为以下 5 个部分,分别是图像,声音,搜台,频道管理和服务菜单,其中服务菜单是在数字电视下才有的,**TVP9000** 的软件框架如图 2。

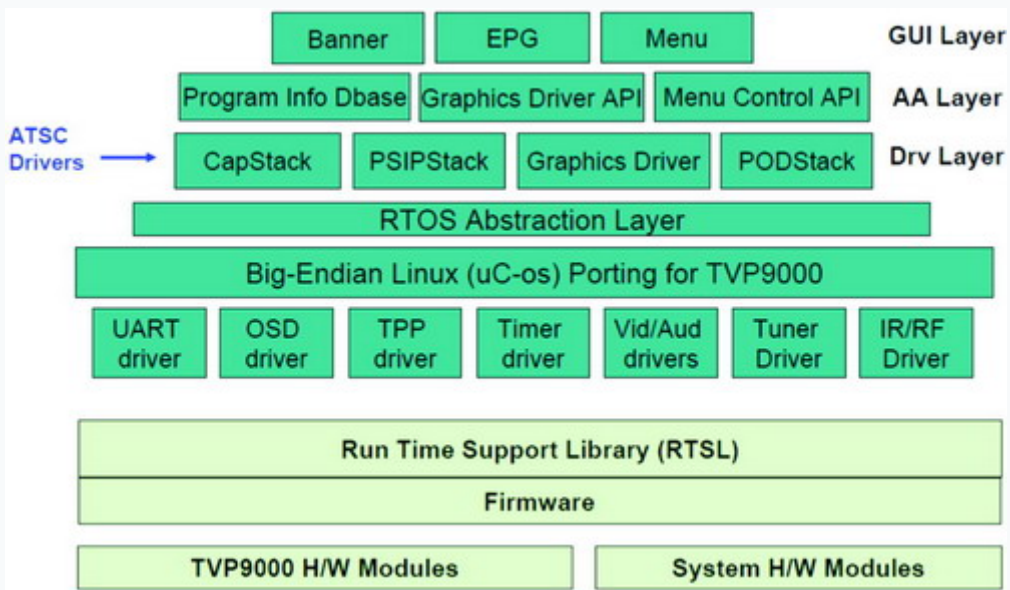


图 2 TVP9000 的软件构架

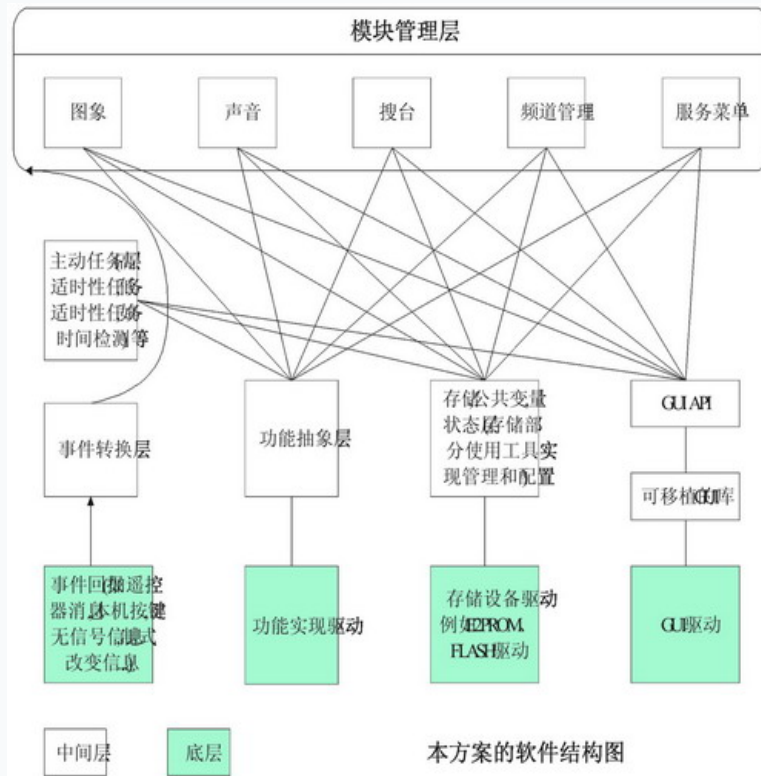


图 3 软件结构

本方案的软件结构如图 3 示，中间层包含了以下几个层：

模块管理层，模块可以控制一个菜单，或者多个菜单，比如图像模块，声音模块。理论上模块管理器可以管理的模块是无限多的，不过基于链表的效率而言，推荐的模块数目不要超过 50 个，一般电视机而言不会有超过 50 个模块的。每个模块都有自己特定的功能，比如图像，声音，搜台等模块。

事件转换层，将底层的不同的事件都经过事件转换层转换为中间层可识别的事件，比如将遥控码事件转化为中间层的 key 事件。

功能抽象层，将底层基于硬件功能的部分都抽象为中间层标准的 API 函数，客户按照相应的功能填写这些函数。

存储、公共变量层，存储(如 EEPROM, FLASH)和公共变量的管理和接口层。

主动任务层，用于作为实时检测，和定时功能实现的模块。