OpenCaster para SATVD-T

OpenCaster para SATVD-T

LIFIA

3 de mayo de 2011

Índice

1. Introducción

Este documento es una extensión del manual de *OpenCaster*. *OpenCaster* es un software desarrollado por *AVALPA* Digital Engineering SRL, para generación de *transport streams* MPEG2.

Dentro de las actividades de TV Digital que el *LIFIA* lleva a cabo, se nos hizo necesario disponer de transmisiones de TV específicas para poder ejecutar pruebas de calidad en nuestras investigaciones. Usamos *OpenCaster* para ello, pero lamentablemente *OpenCaster* no tiene soporte para las extensiones a DVB que definen la norma japonesa y brasileñas, bases del *Sistema Argentino de Televisión Digital Terrestre: SATVD-T*. En *LIFIA*, modificamos *OpenCaster* para agregar una parte de la norma faltante, de modo de poder continuar nuestro desarrollo. Estas modificaciones no son completas.

Ya nos hemos comunicado con *AVALPA* con el objeto de transmitirle nuestras modificaciones a *OpenCaster*, con la esperanza de que dichas extensiones cumplan con los requerimientos de calidad y que sean incorporados a la distribución oficial. Mientras tanto, *LIFIA* pone las extensiones junto con este manual a disposición del público en general.

Esta guía no tiene por objeto explicar que es un *transport stream* ni tampoco como funciona *OpenCaster*. El lector debería procurar entender la teoría de TV Digital primero ([4], [6], [5], [8]), luego leer el manual de *OpenCaster* [7], para finalmente sacar provecho de esta guía.

2. Instalando OpenCaster

2.1. Requerimientos

OpenCaster tiene muy pocos requerimientos: básicamente GNU/Linux, el compilador de C de GCC y un ambiente python.

Como referencia, estas son las versiones usadas durante la escritura de esta guía:

```
$ gcc --version
gcc (Gentoo 4.4.3-r2 p1.2) 4.4.3
Copyright (C) 2010 Free Software Foundation, Inc.
Esto es software libre; vea el código para las
condiciones de copia.  NO hay garantía; ni siquiera
para MERCANTIBILIDAD o IDONEIDAD PARA UN PROPÓSITO EN
PARTICULAR

$ python --version
Python 2.6.5
```

2.2. Bajando los fuentes

Los fuentes se pueden bajar del sitio original del *OpenCaster* en

```
http://www.avalpa.com/
```

Se requiere tener un usuario registrado, pero el registro es libre.

Es recomendable bajar también el manual de *OpenCaster* del sitio original. Muchos de los detalles del uso de *OpenCaster* no se desarrollan en este documento, solo lo particular para el *SATVD-T*.

La distribución que *LIFIA* modificó está versionada como `2.4`.

Los fuentes de dicha versión están disponibles también en

```
ftp://tvd.lifia.info.unlp.edu.ar/OpenCaster2.4/
```

Allí se encuentran los archivos:

OpenCaster.2.4.tgz Contiene la distribución original de *OpenCaster* 2.4 según distribuida por *AVAL-PA*.

OpenCaster2.4-lifia-rev362.patch Es el archivo usado para aplicar los cambios de *LIFIA* a la distribución.

README.OpenCaster2.4-lifia-rev362.txt Contiene una descripción de instalación y los cambios hechos a la distribución original.

Se deben descargar todos los archivos a algún directorio temporal para su instalación. En adelante, consideramos que se bajan en:

```
/home/tvd/OpenCaster/
```

2.3. Preparado de fuentes y compilación

Se deben seguir los siguientes pasos:

1. Descomprimir la versión original de los fuentes:

   ```
   $ cd /home/tvd/OpenCaster/
   $ tar zxf OpenCaster.2.4.tgz
   ```

2. Aplicar el patch de *LIFIA*:

   ```
   $ cd OpenCaster2.4/
   $ patch -p 1 < ../OpenCaster2.4-lifia-rev362.patch
   ```

3. Compilar:

   ```
   $ make
   ```

Esto deja las herramientas listas para instalar.

2.4. Instalación

Esta parte de la instalación requiere el uso de la cuenta *root*.

```
$ su
# make install
```

Una vez que el software está instalado, ya se pueden generar transport streams. Para probar la correcta instalación de *OpenCaster*, ejecutar el siguiente comando:

```
$ python -c "from dvbobjects.PSI.PAT import *"
```

El comando debería ejecutar sin mostrar ningún error en la salida.

3. Creando un TransportStream con 1 servicio de TV Digital

La definición de un *Transport stream* con 1 servicio requiere de los siguientes elementos:

- Archivo de video.
- Archivo de audio.

- Creación de las tablas SI/PSI.
- Multiplexado del *Transport stream*.

Para este ejemplo, usamos los archivos de audio y video que vienen en los tutoriales de *OpenCaster*: `firstvideo.ts` y `firstaudio.ts`. En el manual de *OpenCaster* se puede encontrar información de como generar el video y audio a partir de otras fuentes.

Para generar las tablas SI/PSI, vamos a escribir un script en python en un archivo `gtables.py`. En estas tablas se define la estructura del *Transport stream* que estamos armando.

Usando un editor cualquiera, vaya escribiendo el archivo `gtables.py` según se detalla en las siguientes secciones. En adelante, consideramos que lo guarda en un directorio nuevo:

```
/home/tvd/ts_1servicio/gtables.py
```

3.1. Encabezado del script que genera las tablas

Empezamos escribiendo el encabezado del archivo `gtables.py` como se muestra a continuación:

```
#!/usr/bin/env python
import os

from dvbobjects.PSI.PAT import *
from dvbobjects.PSI.NIT import *
from dvbobjects.PSI.SDT import *
from dvbobjects.PSI.PMT import *

from dvbobjects.SBTVD.Descriptors import *

tvd_ts_id             = 0x073b # ID de red.
tvd_orig_network_id   = 0x073b # ID de red original.
ts_freq               = 533    # Frecuencia de transmisión
ts_remote_control_key = 0x05   # Tecla de control remoto.

tvd_service_id_sd = 0xe760 # ID de servicio de TV Digital.
tvd_pmt_pid_sd    = 1031   # PID de la PMT del servicio.
```

Aparte de los encabezados propios de cualquier script python, se incluye la librería *OpenCaster* y se definen identificadores para el *Transport stream*. Se detallan:

- `tvd_ts_id` es el identificador del *Transport stream*.
- `tvd_orig_network_id` es el identificador de red original.
- `ts_freq` es la frecuencia en que se transmite el *Transport stream*, en este caso 533 Mhz.
- `ts_remote_control_key` es la tecla de control remoto virtual, sirve para poder usar el control remoto para elegir el canal más rápido.
- `tvd_service_id_sd` es el identificador del servicio de TV digital.
- `tvd_pmt_pid_sd` es el PID que se usará para transmitir la información que componen el servicio.

3.2. Definión de la red: NIT

En el siguiente listado se muestra el contenido de la información de red actual, mediante el uso de la tabla NIT. Es muy parecida a la versión original de *OpenCaster*, agregando datos específicos del *SBTVD-T* y *SATVD-T*.

```
nit = network_information_section(
    network_id = tvd_orig_network_id,
    network_descriptor_loop = [
```

```
        network_descriptor(network_name = "LIFIATV",),
        system_management_descriptor(
            broadcasting_flag = 0,
            broadcasting_identifier = 3,
            additional_broadcasting_identification = 0x01,
            additional_identification_bytes = [],
        )
    ],
    transport_stream_loop = [
        transport_stream_loop_item(
        transport_stream_id = tvd_ts_id,
        original_network_id = tvd_orig_network_id,
        transport_descriptor_loop = [
            service_list_descriptor(
                dvb_service_descriptor_loop = [
                    service_descriptor_loop_item (
                        service_ID = tvd_service_id_sd,
                        service_type = 1,
                    ),
                ],
            ),
            terrestrial_delivery_system_descriptor(
                area_code = 1341,
                guard_interval = 0x01,
                transmission_mode = 0x02,
                frequencies = [
                    tds_frequency_item( freq=ts_freq )
                ],
            ),
            partial_reception_descriptor (
                service_ids = []
            ),
            transport_stream_information_descriptor (
                remote_control_key_id = ts_remote_control_key,
                ts_name = "LIFIATV",
                transmission_type_loop = [
                    transmission_type_loop_item(
                        transmission_type_info = 0x0F,
                        service_id_loop = [
                            service_id_loop_item(
                                service_id=tvd_service_id_sd
                            ),
                        ]
                    ),
                    transmission_type_loop_item(
                        transmission_type_info = 0xAF,
                        service_id_loop = [],
                    ),
                ],
            )
        ],
        ),
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)
```

Entre las cosas específicas del *SBTVD-T* y *SATVD-T* están:

- `system_management_descriptor` que define propiedades del sistema, definiendo que es el sistema ISDB, y que está transmitiendo ahora.
- `terrestrial_delivery_system_descriptor` que define propiedades de la modulación, como intervalos de guarda, frecuencia de transmisión, etc.
- `partial_reception_descriptor` que define la lista de servicios de recepción parcial. Esta lista tendría que tener la lista de servicios *1-Seg*.
- `transport_stream_information_descriptor` que define otras propiedades del *Transport stream* que estamos creando, como la tecla de control remoto, el nombre del *Transport stream*, información de los tipos de servicios ofrecidos, etc.

3.3. Definición de los servicios: SDT

En el siguiente listado se muestra el contenido de la tabla SDT que sirve para especificar los servicios disponibles en el *Transport stream*.

```
sdt = service_description_section(
    transport_stream_id = tvd_ts_id,
    original_network_id = tvd_orig_network_id,
    service_loop = [
        service_loop_item(
            service_ID = tvd_service_id_sd,
            EIT_schedule_flag          = 0,
            EIT_present_following_flag = 0,
            running_status = 4,
            free_CA_mode = 0,
            service_descriptor_loop = [
                service_descriptor(
                    service_type = 1,
                    service_provider_name = "",
                    service_name = "LFASD",
                ),
            ],
        ),
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)
```

3.4. Definición del mapa de programas: PAT

En el siguiente listado se muestra el contenido de la tabla PAT. La tabla PAT tiene el mapa de servicios en PMTs. Las PMTs son las tablas que definen como se compone cada servicio y se detalla en la próxima sección.

```
pat = program_association_section(
    transport_stream_id = tvd_ts_id,
    program_loop = [
        program_loop_item(
            # Programa especial para la tabla NIT
            program_number = 0,
            PID = 16,
        ),
        program_loop_item(
            program_number = tvd_service_id_sd,
            PID = tvd_pmt_pid_sd,
        ),
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)
```

3.5. Definición de los componentes del servicio: PMT

En el siguiente listado se muestra el contenido de la tabla PMT.

A diferencia de las tablas anteriores, tiene que haber una PMT por cada servicio que se transmita. En nuestro caso, hay una sola.

Esta tabla define que flujos de datos componen el servicio. En ella se detallan cual es el stream de audio, de video, etc. que componen el servicio. También se define en que PID se encuentra el PCR que es la marca de reloj del servicio.

En esta tabla PMT estamos definiendo que nuestro servicio se compone de un stream de video MPEG2 en el PID 2064, y de un stream de audio MPEG2 en el PID 2068.

Además se detalla que el PID de PCR es el 2064, el mismo que el de video.

```
pmt_sd = program_map_section(
```

```
    program_number = tvd_service_id_sd,
    PCR_PID = 2064,
    program_info_descriptor_loop = [],
    stream_loop = [
        stream_loop_item(
            stream_type = 2, # mpeg2 video stream type
            elementary_PID = 2064,
            element_info_descriptor_loop = [
            ]
        ),
        stream_loop_item(
            stream_type = 3, # mpeg2 audio stream type
            elementary_PID = 2068,
            element_info_descriptor_loop = []
        ),
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)
```

3.6. Escribiendo las tablas a archivos

En el siguiente listado se muestra el código necesario para escribir las tablas anteriores en archivos. Estos archivos son parte de los fuentes necesarios para generar el multiplexado final del *Transport stream*.

```
out = open("./nit.sec", "wb")
out.write(nit.pack())
out.close()
os.system("sec2ts 16 < ./nit.sec > ./nit.ts")

out = open("./pat.sec", "wb")
out.write(pat.pack())
out.close()
os.system("sec2ts 0 < ./pat.sec > ./pat.ts")

out = open("./sdt.sec", "wb")
out.write(sdt.pack())
out.close()
os.system("sec2ts 17 < ./sdt.sec > ./sdt.ts")

out = open("./pmt_sd.sec", "wb")
out.write(pmt_sd.pack())
out.close()
os.system("sec2ts " + str(tvd_pmt_pid_sd) +
          " < ./pmt_sd.sec > ./pmt_sd.ts")
```

3.7. Creando las tablas

Una vez que el archivo `btables.py` está completo, tenemos que correr el script para generar las secciones que componen las tablas.

Ejecutando el script, se generan archivos `.sec` que son las secciones de las tablas con la información detallada, y archivos `.ts` con las secciones embebidas en paquetes de 188 bytes que después vamos a usar para generar el *Transport stream*:

```
$ cd /home/tvd/ts_1servicio/
$ chmod u+x gtables.py
$ ./gtables.py
```

3.8. Multiplexado del TS

Se debe copiar al directorio con las tablas generadas los archivos de audio y video, junto con el archivo `null.ts` que sirve para que el *Transport stream* generado sea del ancho de banda requerido por la norma *ISDB-T*.

```
$ cd /home/tvd/ts_1servicio
$ cp /home/tvd/OpenCaster/OpenCaster2.4/tutorials/OCTutorial2/firstvideo.ts .
$ cp /home/tvd/OpenCaster/OpenCaster2.4/tutorials/OCTutorial2/firstaudio.ts .
$ cp /home/tvd/OpenCaster/OpenCaster2.4/tutorials/OCTutorial2/null.ts .
```

Finalmente, se multiplexa con el siguiente comando:

```
$ tscbrmuxer \
        600000 \
        b:15040 pat.ts \
        b:15040 pmt_sd.ts \
        b:3008 sdt.ts \
        b:3008 nit.ts \
        b:2300000 firstvideo.ts \
        b:188000  firstaudio.ts \
        b:27434198 null.ts > prueba1.ts
```

Un poco de detalle acerca de los números:

- *600000* Es la cantidad de paquetes a multiplexar. El sistema ISDB-T transmite cerca de 20000 paquetes por segundo, con lo cual, el archivo generado es de cerca de 30 segundos. [1]
- *b:15040* Tanto la PAT como la PMT deben ser enviadas al menos 10 veces por segundo. Sabiendo que cada una de las tablas entra en un solo paquete de 188 bytes, tenemos que enviar 10 paquetes por segundo. Y como cada paquete es de $188\ bytes \times 8 = 1504\ bits$, queremos que el ancho de banda sea de $1504\ bits \times 10 = 15040\ bps$. Un análisis similar se necesita para la NIT y SDT.
- *b:2300000*, *b:188000* Son el ancho de banda del audio y del video respectivamente. Más detalle se puede encontrar en el manual de *OpenCaster*.
- *b:27434198* Es el ancho de banda de paquetes nulos. El sistema ISDB-T tiene un ancho de banda fijo de 29.958.294 bps, y como estamos usando: $15040+15040+3008+3008+2300000+188000 = 2524096$ en total, tenemos que completar el *Transport stream* con paquetes nulos. El ancho de banda requerido se deriva de: $29958294 - 2524096 = 27434198$ bps.

Tener en cuenta que el ancho de banda del sistema ISDB-T es muy grande. Generar *Transport streams* muy largos es una de las formas más rápidas que conozco para agotar el espacio en disco.

Ahora tendríamos que tener un archivo `prueba1.ts` que tiene nuestro primer multiplexado. Sin embargo, hay un problema con el archivo generado: el PCR. El PCR es la referencia del reloj del sistema que trabaja en 27 Mhz, y en nuestro caso, viaja en el stream de video. Como estuvimos cambiando la posición de los paquetes del video, la llegada de los mismos al STB es imprecisa. Para arreglar esto, usamos la herramienta `tsstamp`:

```
$ tsstamp prueba1.ts 29958294 > prueba1.fixed.ts
```

Ahora, el archivo `prueba1.fixed.ts` está casi completo. Para adherir a la norma, falta agregarle la fecha actual usando la tabla TOT o TDT, y agregarle información de programación usando la tabla EIT. Más detalles de esto se pueden ver en el manual de *OpenCaster*. De todos modos, se puede usar el *Transport stream* como está, la mayoría de los STB soportan *Transport streams* con estos faltantes.

4. Agregando una aplicación Ginga NCL al servicio

El *SATVD-T* soporta el envío de aplicaciones GINGA/NCL usando el carousel de objetos y se señaliza con la tabla AIT, según [1].

Los pasos para componer la aplicación al servicio son:

- Generar el carousel de objetos con la aplicación.

[1] El ancho de banda del sistema ISDB-T es de 32.507.937 bps (bits por segundo), en paquetes de 204 bytes. Esto da un ancho de banda de aproximadamente 29.958.294 bps en paquetes de 188 bytes, dejando cerca de 19.919 paquetes por segundos.

- Generar las secciones de la tabla AIT con la información de la aplicación enviada.
- Multiplexar el *Transport stream* de nuevo con los agregados.

4.1. Generando el carousel de objetos

Creamos una carpeta conteniendo la aplicación en nuestro directorio de trabajo:

```
$ cd /home/tvd/ts_1servicio
$ mkdir app_ginga
$ cp -r <path a la aplicación>/* app_ginga/
```

Después se usa la herramienta `oc-update.sh` que viene con *OpenCaster* para generar el carousel de objetos.

```
$ oc-update.sh app_ginga 0x0C 1 2004 2
```

Esto va a generar un archivo `app_ginga.ts` que contiene los paquetes que llevan las secciones del carousel. Los parámetros significan:

- *app_ginga* El directorio donde está la aplicación. También define el nombre del archivo generado.
- *0x0C* El *association_tag* del carousel generado.
- *1* El número de versión de los módulos generados.
- *2004* El PID en el que se envía el carousel.
- *2* El *carousel_id*.

4.2. Agregando la AIT

Vamos a modificar el archivo `gtables.py` que generamos anteriormente para agregar la tabla AIT. También tenemos que modificar la PMT en dicho archivo para agregar el envío de la AIT y el carousel entre los streams del servicio de TV Digital.

Primero, agregamos los encabezados necesarios en la primera parte del archivo:

```
from dvbobjects.MHP.AIT import *
from dvbobjects.MHP.Descriptors import *
```

Después agregamos la definición de la AIT. Notar que los campos carousel_id, association_tag, el pid y demás se corresponden con los que usamos cuando ejecutamos la herramienta `oc-update.sh`. También, en `ginga_ncl_application_location_descriptor`, tenemos que asegurarnos que `initial_class` se corresponda con el *NCL* que está en el directorio, en este caso es `main.ncl`. Lo mismo con el `base_directory`.

```
ait = application_information_section(
    application_type = 0x0009, # GINGA-NCL
    common_descriptor_loop = [],
    application_loop = [
        application_loop_item(
            organisation_id = 0x0000000A,
            application_id  = 0x64,
            application_control_code = 0x01, # AUTOSTART

            application_descriptors_loop = [
            transport_protocol_descriptor(
                protocol_id = 0x0001,
                transport_protocol_label = 0,
                remote_connection = 0,
                component_tag = 0x0C, # association_tag
            ),
            application_descriptor(
                application_profile = 0x0001,
```

```
                version_major = 1,
                version_minor = 0,
                version_micro = 0,
                service_bound_flag = 1,
                visibility = 3,
                application_priority = 1,
                transport_protocol_labels = [ 0 ],
            ),
            application_name_descriptor(
                application_name = "APP_GINGA"
            ),
            ginga_ncl_application_descriptor(
                parameters = [ ]
            ),
            ginga_ncl_application_location_descriptor (
                base_directory = "/",
                class_path_extension = "",
                initial_class = "main.ncl", # nombre del archivo NCL
                                            # a ser ejecutado.
            ),
            ]
        ),
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)
```

Modificamos la PMT, para que el servicio incluya la AIT y el carousel. Para esto se agregan 2 *elementary streams*. Reemplace la definición anterior de la PMT en el archivo `gtables.py` con la que sigue:

```
pmt_sd = program_map_section(
    program_number = tvd_service_id_sd,
    PCR_PID = 2064,
    program_info_descriptor_loop = [],
    stream_loop = [
        stream_loop_item(
            stream_type = 2, # mpeg2 video stream type
            elementary_PID = 2064,
            element_info_descriptor_loop = [
            ]
        ),
        stream_loop_item(
            stream_type = 3, # mpeg2 audio stream type
            elementary_PID = 2068,
            element_info_descriptor_loop = []
        ),
        stream_loop_item(
            stream_type = 5, # AIT stream type
            elementary_PID = 2001,
            element_info_descriptor_loop = [
                data_component_descriptor (
                    data_component_id = 0xA3, # sistema AIT
                    additional_data_component_info = ait_identifier_info(
                        application_type = GINGA_NCL_application_type,
                        ait_version      = 0
                    ).bytes(),
                ),
                application_signalling_descriptor(
                    application_type = 9, # 9 GINGA-NCL
                    AIT_version      = 1, # current ait version
                ),
            ]
        ),
        stream_loop_item(
            stream_type = 0x0B, # DSMCC stream type
            elementary_PID = 2004,
            element_info_descriptor_loop = [
                association_tag_descriptor(
                    association_tag = 0x0C,
                    use = 0,
                    selector_lenght = 0,
                    transaction_id = 0x80000000,
```

```
                    timeout = 0xFFFFFFFF,
                    private_data = "",
                ),
                stream_identifier_descriptor(
                    component_tag = 0x0C,
                ),
                carousel_identifier_descriptor(
                    carousel_ID = 2,
                    format_ID = 0,
                    private_data = "",
                ),
                data_component_descriptor (
                    data_component_id = 0xA0, # sistema GINGA
                    additional_data_component_info = additional_ginga_j_info(
                        transmission_format = 0x2,
                        document_resolution = 0x5,
                        organization_id     = 0x0000000A,
                        application_id      = 0x0064,
                        carousel_id         = 2,
                    ).bytes(),
                ),
            ]
        )
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)
```

Finalmente, agregamos el código que escribe las secciones y los paquetes que las transportan:

```
out = open("./ait.sec", "wb")
out.write(ait.pack())
out.close()
os.system('sec2ts ' + str(2001) + ' < ./ait.sec > ./ait.ts')
```

Se generan las tablas de nuevo con:

```
$ ./gtables.py
```

4.3. Multiplexado del *Transport stream*

Multiplexando con la aplicación:

```
$ tscbrmuxer \
        600000 \
        b:15040 pat.ts \
        b:15040 pmt_sd.ts \
        b:3008 sdt.ts \
        b:3008 nit.ts \
        b:3008 ait.ts \
        b:400000  app_ginga.ts \
        b:2300000 firstvideo.ts \
        b:188000  firstaudio.ts \
        b:27031190 null.ts > prueba1.ts
$ tsstamp prueba1.ts 29958294 > prueba1.fixed.ts
```

Un detalle a tener en cuenta es el tamaño de la aplicación. Si la aplicación es muy grande, el archivo de carousel generado (`app_ginga.ts`) será grande también. Como la velocidad de transmisión está fija en 400.000 bps, hay que asegurar que la cantidad de segundos multiplexados son suficientes para transmitir todo el carousel al menos una vez. Si esto no sucede, el carousel va a estar incompleto y por lo tanto la aplicación nunca se va a poder bajar.

A modo de ejemplo, si el archivo es de 1200 KiB (kilobytes), su tamaño es 9.830.400 bits. A 400.000 bps, el archivo tardará 24,576 segundos para transmitirse por completo, casi los 30 segundos para un archivo de 1,2 MiB.

Si quiere transmitir archivos más grandes, puede o bien aumentar la velocidad de transmisión, o generar un multiplexado de mayor duración.

5. Agregando un archivo de actualización de firmware

Esta sección muestra como usar *OpenCaster* para agregar actualizaciones de firmware en el *Transport stream*.

Más detalle de la estructura necesaria del *Transport stream* se pueden consultar en [3] y en [2]

5.1. Copiar el archivo de firmware

```
$ cp <firmware_nuevo.dat> .
```

5.2. Definiciones previas

Agregar los siguientes imports al principio de los imports en el archivo `gtables.py` con el que estábamos trabajando:

```
from dvbobjects.DVB.Descriptors import *
from dvbobjects.DVB.DataCarousel import *
from dvbobjects.SBTVD.SDTT import *
from datetime import *
```

Agregar las siguientes definiciones, cambiando lo que corresponda:

```
# Segun norma ARIB TR-B14 - pagina 1-24
tvd_service_id_eng = 0xFFF0
tvd_pmt_pid_eng    = 8100

# PID del carousel de datos.
tvd_dsmcc_pid  = 2003

# Valores para el compatibility descriptor.
# Hay que poner los que correspondan con
# el equipo que se quiere actualizar.
maker_id = 0x00
model_id = 0x01
group_id = 0x0
targetversion_id  = 0x00e
newversion_id     = 0x00f
download_level    = 0x01
version_indicator = 0x02

# Datos del carousel de datos.
dsmcc_association_tag = 0xA
dsmcc_carousel_id     = 1

# Obtener fecha actual, usado
# para la planificacion.
timenow = datetime.now()
```

5.3. Agregando un servicio de ingeniería

Agregar el programa a la PAT, debajo de la definición del servicio de TV Digital:

```
program_loop_item(
    program_number = tvd_service_id_eng,
    PID = tvd_pmt_pid_eng,
),
```

Agregar el servicio en la NIT, en el `service_list_descriptor`, debajo del anterior:

```
service_descriptor_loop_item (
    service_ID = tvd_service_id_eng,
    # ARIB Data broadcast type
    service_type = 0xA4,
),
```

Agregar el servicio en la SDT, en el loop de servicios, debajo del anterior:

```
service_loop_item(
    service_ID = tvd_service_id_eng,
    EIT_schedule_flag = 0,
    EIT_present_following_flag = 0,
    running_status = 4,
    free_CA_mode = 0,
    service_descriptor_loop = [
        service_descriptor(
            # ARIB Data broadcast type
            service_type = 0xA4,
            service_provider_name = "",
            service_name = "ENGSERVICE",
        ),
    ],
),
```

Agregar la PMT del servicio de ingeniería:

```
pmt_eng = program_map_section(
    program_number = tvd_service_id_eng,
    PCR_PID        = tvd_dsmcc_pid,
    program_info_descriptor_loop = [
        data_component_descriptor (
            # ARIB-STD TR-B14 - ver 2.8-E2 - page 1-12
            data_component_id = 0x0009,
            additional_data_component_info = "",
        )
    ],
    stream_loop = [
        stream_loop_item(
            # DSM-CC data stream type
            stream_type = 0x0D,
            elementary_PID = tvd_dsmcc_pid,
            element_info_descriptor_loop = [
                stream_identifier_descriptor(
                    component_tag = dsmcc_association_tag,
                ),
            ]
        ),
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)
```

5.4. Definición del descriptor de compatibilidad

El descriptor de compatibilidad le informa al STB cual es el equipo a actualizar.

```
comp_desc = compatibility_descriptor (
    compatibility_descriptor_loop = [
        compatibility_descriptor_loop_item (
            # Segun ARIB-STD-B21, tabla 12-4
            descriptor_type = 0x02,
            specifier_type = 0xFF,
            specifier_data = 0x819282,
            model   = (maker_id <<  8) | model_id,
            version = (group_id << 12) | targetversion_id,
            compatibility_descriptor_subloop = [],
        ),
    ]
)
```

5.5. Generación de carousel de datos

El carousel de datos se genera a partir de un archivo que contiene el software nuevo. En este caso, el software se saca del archivo `firmware_nuevo.dat`.

El STB lo recibirá como un stream de datos, pero el uso del `name_descriptor` le indica el nombre de archivo que queremos que use.

```
group = Group(
    PATH="DII.sec",
    transactionId = 0x80000002,
    downloadId    = 0x00000001,
    blockSize     = 4066,
    version   = 1,
)

group.set(
    compatibilityDescriptor = comp_desc.pack(),
    modules = [
         Module(
             INPUT="firmware_nuevo.dat",
             moduleId = 0x0001,
             moduleVersion = 0x00,
             descriptors = [
                 type_descriptor (mime_type="application/x-download"),
                 name_descriptor (name     ="firmware.20100908.dat"),
             ],
         ),
    ],
)

# Generates dsmcc sections.
group.generate("carousel")
```

5.6. Agregando la SDTT

La SDTT (Software Download Trigger Table) sirve para informar al STB la planificación de transmisión de las descargas de software.

La SDTT tiene información del equipo que el STB usa para saber si la descarga le corresponde o no.

```
sdtt = software_download_trigger_table (
    transport_stream_id = tvd_ts_id,
    original_network_id = tvd_orig_network_id,
    service_id          = tvd_service_id_eng,
    maker_id = maker_id,
    model_id = model_id,
    contents            = [
        sdtt_content_loop_item (
            group             = group_id,
            target_version    = targetversion_id,
            new_version       = newversion_id,
            download_level    = download_level,
            version_indicator = version_indicator,
            schedule_timeshift_information = 0x00,
            schedule_loop_items = [
                # Hoy, desde la 01:00:00 hasta las 23:59:59
                sdtt_schedule_loop_item (
                    start_year  = timenow.year - 1900, # since 1900
                    start_month = timenow.month,
                    start_day   = timenow.day,
                    start_hour  =  0x01,
                    start_minute=  0x00,
                    start_second=  0x00,
                    duration_hours   = 0x23,
                    duration_minutes = 0x59,
                    duration_seconds = 0x59,
                ),
            ],
            descriptors = [
                download_content_descriptor (
```

```
                    reboot = 0,
                    add_on = 1,
                    component_size = 0x00,
                    download_id = 0x00000001,
                    time_out_value_DII = 36600000,
                    leak_rate = 0x00,
                    component_tag = dsmcc_association_tag,
                    compatibility_descriptor_bytes = comp_desc.pack(),
                    modules_info_bytes = "",
                    privateData = "",
                    text_ISO639_lang = "",
                    text_data        = "",

                )
            ],
        )
    ],
    version_number = 0,
    section_number = 0,
    last_section_number = 0,
)
```

5.7. Agregar la escritura de las tablas nuevas

```
out = open("sdtt.sec", "wb")
out.write(sdtt.pack())
out.close()
# PID 35 (0x23): Low protection layer
os.system('sec2ts 35 < sdtt.sec > sdtt.ts')

out = open("pmt_eng.sec", "wb")
out.write(pmt_eng.pack())
out.close()
os.system('sec2ts ' + str(tvd_pmt_pid_eng) + ' < pmt_eng.sec > pmt_eng.ts')
```

5.8. Actualización de las tablas

De nuevo, se ejecuta el `gtables.py`. Antes tenemos que crear el directorio donde se guardan las secciones del carousel:

```
$ mkdir carousel
$ ./gtables.py
```

5.9. Creación del carousel

Una vez que se ejecuta el script de python, se generan las secciones que componen el carousel de datos. Falta componer estas secciones en paquetes:

```
$ cd carousel
$ for i in *.sec ; do sec2ts 2003 < $i >> ../carousel.ts.unfixed ; done
$ cd ..
$ tsfixcc carousel.ts.unfixed > carousel.ts
```

5.10. Multiplexado del *Transport stream*

Multiplexando con la actualización de software:

```
$ tscbrmuxer \
        600000 \
        b:15040 pat.ts \
        b:15040 pmt_sd.ts \
        b:15040 pmt_eng.ts \
        b:3008 sdt.ts \
        b:3008 nit.ts \
        b:3008 sdtt.ts \
```

```
        b:1000000 carousel.ts \
        b:2300000 firstvideo.ts \
        b:188000  firstaudio.ts \
        b:26414150 null.ts > prueba1.ts

$ tsstamp prueba1.ts 29958294 > prueba1.fixed.ts
```

Recordar nuevamente que la cantidad de segundos multiplexadas tiene que asegurar el transporte del carousel completo.

GNU Free Documentation License

Version 1.3, 3 November 2008

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "**publisher**" means any person or entity that distributes copies of the Document to the public.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Referencias

[1] ABNT. Televisão digital terrestre – codificação de dados e especificações de transmissão para radiodifusão digital. parte 3: Especificação de transmissão de dados. Versión 07.04.2008, disponible en http://www.forumsbtvd.org.br/materias.asp?id=112.

[2] ARIB. Operational guidelines for digital terrestrial television broadcasting. Versión 2.8, disponible en http://www.arib.or.jp/english/html/overview/sb_ej.html.

[3] ARIB. Receiver for digital broadcasting. Versión 4.6, disponible en http://www.arib.or.jp/english/html/overview/sb_ej.html.

[4] W. Fisher. *Digital Video and Audio Broadcasting Technology*.

[5] ISO/IEC. *ISO/IEC 13818-1 Information technology - Generic coding of moving pictures and associated audio information: Systems*.

[6] Steve Morris. *Interactive TV standards*.

[7] AVALPA Digital Engineering SRL. Avalpa broadcast server user manual. Versión del 14/03/2010, disponible en http://www.avalpa.com.

[8] European Broadcasting Union. *Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.3*.