

Universidad Técnica Federico Santa
María
Campus Casa Central
Valparaíso, Chile



“TRANSMISIÓN PICTURE IN PICTURE (PIP)”

Televisión digital bajo norma ISDB-Tb

DESARROLLADO POR:

Sebastián Castillo Araya

CORREO :

sebastian.castilloar@sansano.usm.cl

FECHA:

24 Enero 2023

Índice

1	Introducción	2
2	Requerimientos	2
2.1	Hardware	2
2.2	Software	2
3	Configuración script bash	3
4	Configuración EiTV Playout Professional	5
5	Prueba práctica y resultados	10
5.1	PiP con videos locales	10
5.2	PiP con video local y webcam	11
5.3	PiP con video local y transmisión en vivo	12
6	Conclusión	13
7	Referencias	13

Índice de tablas

1	Parámetros comando streamlink-ffmpeg	3
---	--------------------------------------	---

Índice de figuras

1	Interfaz log in	5
2	Nuevo elemento multimedia	5
3	Configuración contenido multimedia	6
4	Nuevo servicio	6
5	Configuración nuevo servicio	6
6	Asociación a nuevo servicio	7
7	Nuevo transport stream	7
8	Configuración general transport stream	7
9	Configuración norma ISDB-T	8
10	Configuración Capa B con su servicio	8
11	Transport stream con capa A y B configurada	9
12	Script con opción cero.	10
13	PiP videos locales.	10
14	Script con opción uno.	11
15	PiP con video local y webcam.	11
16	Script con opción uno.	12
17	PiP con video local y webcam.	12

1. Introducción

El presente documento tiene como objetivo guiar al lector paso a paso a aplicar el efecto picture in picture en una señal de video y posteriormente transmitirla a la televisión digital, bajo la norma ISDB-Tb, utilizando EiTV Playout Professional. Para eso se presenta al lector los requisitos que este necesita, preparación de script, preparación de playout y finalmente una prueba práctica. Al seguir correctamente las indicaciones, el lector será capaz de aplicar el formato picture in picture combinando distintas fuentes, por ejemplo webcam y video almacenado, y podrá transmitir el resultado a la señal abierta.

2. Requerimientos

Los requisitos de hardware y software indicados a continuación, fueron utilizados en pruebas de transmisión y lograron un resultado satisfactorio.

2.1. Hardware

Para **transmitir**:

- Computador:
 - CPU: Intel Core i3-5005U 2.00 GHz
 - Memoria Ram: 8GB
 - SSD: 240GB
- EiTV Playout Professional.
- Una antena para transmitir.

Para **recibir**(opcional para observar resultados):

- Televisión.
- Set Top-Box (si la televisión ya cuenta con compatibilidad digital entonces este dispositivo no hace falta).
- Una antena para recibir.

2.2. Software

- Sistema operativo GNU/Linux Ubuntu 22.04 LTS [Link oficial](#).
- ffmpeg 4.4.2 [Link oficial](#).
- Streamlink 3.1.1 (sólo en caso de querer usar livestreams de internet).[Link oficial](#).

3. Configuración script bash

Para aplicar el efecto picture in picture se utiliza la herramienta de software Ffmpeg. Esta proporciona la capacidad de aplicar filtros los cuales incluyen entre todas sus posibilidades la de superponer videos generando así el efecto deseado . El siguiente comando nos permite realizar dicha acción:

```
$ ffmpeg -i [fuente 1] -i [fuente 2] -filter_complex "[1]scale=iw/5:ih/5 [pip];
[0][pip] overlay=main_w-overlay_w-10:main_h-overlay_h-10" \
-c:v mpeg2video -c:a aac -pix_fmt yuv420p -s 720x480 -aspect 16:9
-streamid 0:180 -streamid 1:181 \
-b:a 128k -b:v 2298k -maxrate 2298k -minrate 2298k -bufsize 2298k \
-f mpegts -muxrate 2700000.0 udp://10.2.51.11:12345;
```

Parámetro	Descripción
-i	Fuente de entrada para ffmpeg.
-filter complex	Flag para indicar filtro.
-c:v	Codificación de video.
-c:a	Codificación de audio.
-streamid	Indica PID de video y audio, por defecto primero el video luego el audio.
-b:a	Bitrate de audio.
-b:v	Bitrate de video.
-maxrate -minrate	Máximo y mínimo bitrate
-muxrate	Tasa de multiplexación
-f	Formato de salida

Tabla 1: Parámetros comando streamlink-ffmpeg.

La tabla 1 muestra el significado de algunas banderas del comando a utilizar.

Ya conocido el comando se recomienda crear un script bash para facilitar las pruebas que se quieran realizar. Para esto se crea con el editor de texto de preferencia el archivo “pip.sh” en el que se escriben tres acciones a probar:

- PiP con videos almacenados.
- PiP con video y webcam en vivo.
- PiP con video y transmisión en vivo.

```
1 #!/bin/bash
2 option=$1;
3
4 if test $option -eq 0 #PiP videos almacenados
5 then
6     ffmpeg -i bruno1.mp4 -i final.mp4 -filter_complex "[1]scale=iw
7         /5:ih/5 [pip]; [0][pip] overlay=main_w-overlay_w-10:main_h-
8         overlay_h-10" \
9     -c:v mpeg2video -c:a aac -pix_fmt yuv420p -s 720x480 -aspect
10        16:9 -streamid 0:180 -streamid 1:181 \
11    -b:a 128k -b:v 2298k -maxrate 2298k -minrate 2298k -bufsize
12        2298k \
13    -f mpegts -muxrate 2700000.0 udp://10.2.51.11:12345;
14
15 elif test $option -eq 1 #PiP con video y webcam en vivo
16 then
17     ffmpeg -i final.mp4 -i /dev/video0 -filter_complex "[1]scale=
18         iw/5:ih/5 [pip]; [0][pip] overlay=main_w-overlay_w-10:main_h-
19         -overlay_h-10" \
20    -c:v mpeg2video -c:a aac -pix_fmt yuv420p -s 720x480 -aspect
21        16:9 -streamid 0:180 -streamid 1:181 \
22    -b:a 128k -b:v 2298k -maxrate 2298k -minrate 2298k -bufsize
23        2298k \
24    -f mpegts -muxrate 2700000.0 udp://10.2.51.11:12345;
25
26 elif test $option -eq 2 #PiP con video y transmision en vivo
27 then
28     streamlink https://www.youtube.com/watch?v=Lcdi902XB4E best -O
29         | ffmpeg -i pipe:0 -i final.mp4 -filter_complex "[1]scale=iw
30         /5:ih/5 [pip]; [0][pip] overlay=main_w-overlay_w-10:main_h-
31         overlay_h-10" \
32    -c:v mpeg2video -c:a aac -pix_fmt yuv420p -s 720x480 -aspect
33        16:9 -streamid 0:180 -streamid 1:181 \
34    -b:a 128k -b:v 2298k -maxrate 2298k -minrate 2298k -bufsize
35        2298k \
36    -f mpegts -muxrate 2700000.0 udp://10.2.51.11:12345;
37
38 else
39     echo "Solo opcion 0, 1 y 2"
40     exit -3;
41 fi
```

Los videos bruno1.mp4 y final.mp4 son videos almacenados localmente, /dev/video0 corresponde a la webcam integrada y el link de YouTube corresponde a una transmisión en vivo. Aclarado esto se procede finalmente a dar permisos de ejecución al archivo creado:

```
$ chmod +x pip.sh
```

4. Configuración EiTV Payout Professional

Mediante el navegador de preferencia se accede a la interfaz de configuración que proporciona el payout (<http://10.2.51.11>), con los siguientes datos de identificación:
usuario: eitv y contraseña: payout (figura 1).



Figura 1: Interfaz log in

Ya ingresado a la interfaz, se debe agregar un nuevo elemento multimedia. Para esto se hace click en Add New TS dentro de la sección A/V (figura 2).

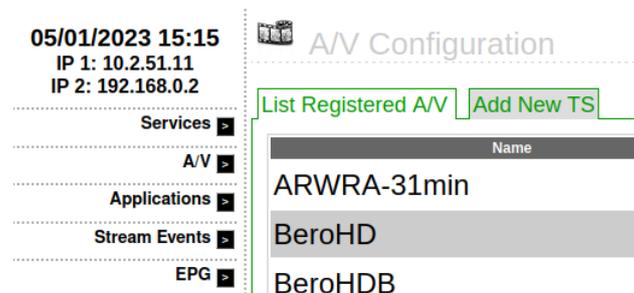


Figura 2: Nuevo elemento multimedia

La configuración para el nuevo elemento debe seguir lo dado por la figura 3. La IP debe corresponder a la del payout (a la fecha en que se realiza este informe corresponde a 10.2.51.11) y para el puerto se tiene más libertad, únicamente se debe registrar pues se usará como destino en el script bash. Respecto a los demás datos a rellenar deben corresponder a los que se utilizarán en el script bash.

Name: AV_Webcam_SC

Source: IP

Address: 10.2.51.11

IP Interface: Gb 1 Port: 12345

Protocol: UDP RTP

PID Video: 180

Video Stream Type: [0x02] MPEG-2 ITU-T Rec. H262 video

PID Audio: 181

Audio Stream Type: [0x0F] ISO/IEC 13818-7 AAC Audio (ADTS transport syntax)

PID PCR: 180

TOT time decode: Synchronize system time with TOT from this input

Bitrate Video (kbps): 2298

Bitrate Audio (kbps): 128

Bitrate TS (bps): 2700000

Figura 3: Configuración contenido multimedia

Siguiendo con la configuración, el archivo creado se procede a adjuntar a un Service. Para esto se va al apartado de Services y click en Add New Service (figura 4). Ya creado se configura (figura 5) y luego se asocia el archivo A/V creado (figura 6) al servicio.

05/01/2023 15:45
IP 1: 10.2.51.11
IP 2: 192.168.0.2

Services

List Registered Services Add New Service

Internal ID	Service Name
-------------	--------------

Figura 4: Nuevo servicio

General

Name: Service_Webcam_SC

Provider: FFmpeg

Service Type: [0x01] Serviço de televisão digital

EIT Schedule Flag:

EIT Present Following Flag:

H-EIT Flag:

M-EIT Flag:

L-EIT Flag:

Custom

The following parameters only need to be configured if custom values are needed. The value "0" (zero) can be left for defaults.

Program PIDs

Primary Video PID: 180 (decimal)

Video Descriptors: (tag:length:byte:tag:length:byte ...)

Primary Audio PID: 181 (decimal)

Audio Descriptors: (tag:length:byte:tag:length:byte ...)

PCR PID: 180 (decimal)

Figura 5: Configuración nuevo servicio

Figura 6: Asociación a nuevo servicio

Si siguiendo con la configuración de playout ahora se procede a crear un transport stream. Para eso se va al apartado de transport stream y click en Add New TS (figura 7)

Figura 7: Nuevo transport stream

Se procede a rellenar los parámetros de acuerdo a nuestro objetivo, indicando interfaz de salida, frecuencia de transmisión, norma a utilizar, modulación entre otros más (Figura 8 y 9).

Figura 8: Configuración general transport stream

Transport Stream: TS_Webcam_SC

Configuration | DVB-C | DVB-T | DVB-S | ISDB-T

ISDB-T Transmission data successfully updated!

Broadcast type: 13-segment TV Broadcast

Frequency: Channel 23 | 527142857 Hz

Area Code: 17

Virtual Channel (Remote Control Key ID): 17

TS Name: EITV Playback

Transmission Mode: Mode 3: 8k

Guard Interval: 1/16

Partial Reception: Enabled partial reception in the Layer A (implies in Layer A with only 1 segment)

Emergency: Enabled emergency flag

Hierarchical Layers	Layer A	Layer B	Layer C
Segments	1	12	0
Modulation	QPSK	16-QAM	16-QAM
Convolutional Rate	2/3	5/6	3/4
Time Interleaving Length	4	4	4
Bitrate (bps)	440560	13216800	0
Total Bitrate (bps)	13657360		

Save

Figura 9: Configuración norma ISDB-T

AVISO: Se configuró el streaming para utilizar la capa B de la norma ISDB-Tb, para que pueda ser reproducido en TV (a través de set top-box), pero aún así es necesario adjuntar un servicio en la Capa A. En la figura 10 se adjunta servicio a capa B, el procedimiento para la capa A es igual. Se aprovechó un servicio ya creado previamente para adjuntar a transport stream (figura 11).

05/01/2023 15:47
IP 1: 10.2.51.11
IP 2: 192.168.0.2

Services Layers Classification

Linked Services

No linked services have been found for this TS.

Services

A/V

Applications

Stream Events

EPG

EWBS

Transport Stream

TS Service Association

Service: [FFMPEG] Service_Webcam_SC

Layer: A B C

Service ID: 32 Suggested value [32]

Associate

Figura 10: Configuración Capa B con su servicio

05/01/2023 15:48
IP 1: 10.2.51.11
IP 2: 192.168.0.2



Services Layers Classification

Linked Services

Service ID	Service Name	Actions
32	[FFMPEG] Service_Webcam_SC [Layer B]	
56	[ffmpeg] LabTVD_Eric_os [Layer A]	

TS Service Association

Service

Associate

Figura 11: Transport stream con capa A y B configurada

Con esto se da por finalizada la configuración de playout, tan solo queda dar play a transport stream y seguir con los siguientes pasos.

5. Prueba práctica y resultados

Se realizan tres pruebas prácticas, una para dos videos almacenados localmente, otra para un video local y webcam, y finalmente para video local y transmisión en vivo. La forma en que estas se realizan es haciendo uso del script bash “pip.sh” creado anteriormente. La sintaxis que este utiliza es la siguiente:

```
$ ./pip.sh [opcion prueba]
```

5.1. PiP con videos locales

En primer lugar se escoge la opción cero que según lo programado es la opción de videos locales, para esto se deben tener los videos en la misma carpeta que se encuentra el script. La figura 12 muestra como se ve el script en consola al escoger la opción cero. Ya preparado el script se procede a ejecutar.

```
sebita@sebita-HP-Notebook:~/Workspace/practica$ ./pip.sh 0
```

Figura 12: Script con opción cero.

La figura 13 muestra la sintonización de la señal, en el que se logra recibir la transmisión y además se logra observar el efecto picture in picture. En grande se tiene un video de Bruno Mars en un concierto mientras que en pequeño se tiene una recopilación de la Final del mundial de fútbol 2022.

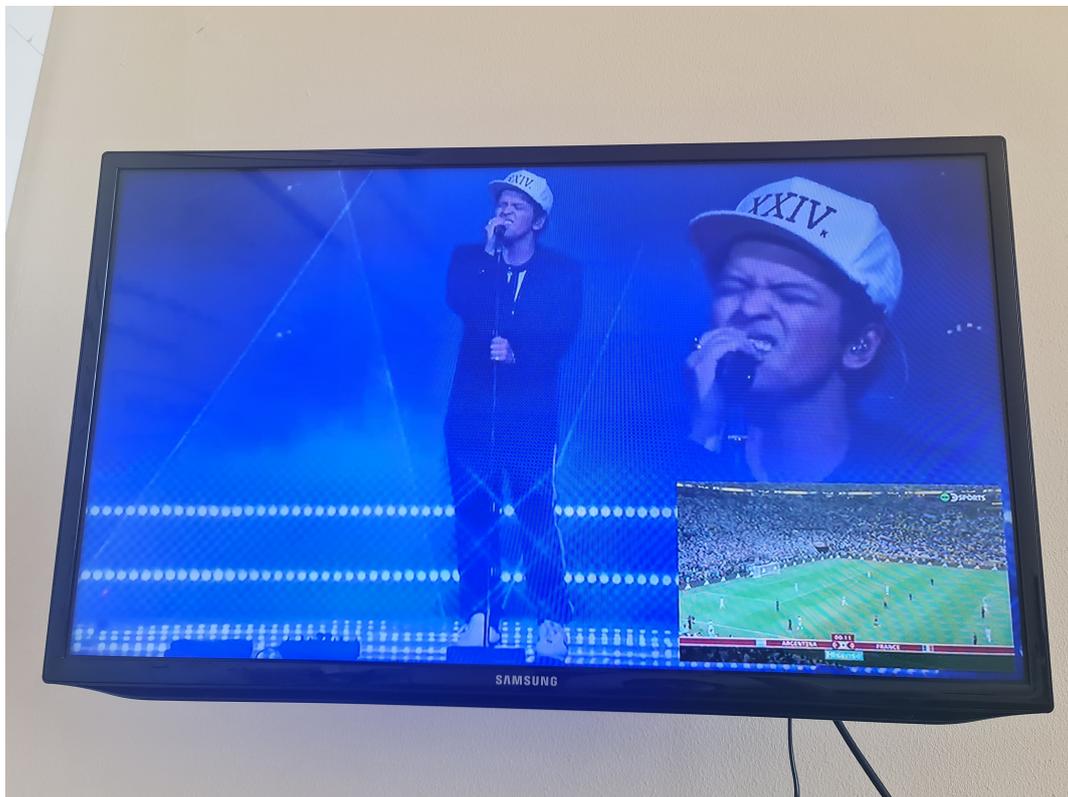


Figura 13: PiP videos locales.

5.2. PiP con video local y webcam

Para lograr el efecto picture in picture con video local y webcam, los pasos no son distintos a los ya realizado anteriormente, incluyendo el paso de tener el video en la misma carpeta que el script. La figura 14 muestra el script seleccionando la opcion uno, la cual corresponde a PiP de video y webcam. Ya preparado el script se procede a ejecutar.

```
sebita@sebita-HP-Notebook:~/Workspace/practica$ ./pip.sh 1
```

Figura 14: Script con opción uno.

La figura 15 muestra la sintonización de la señal, en el que se logra recibir la transmisión y además el efecto picture in picture. De fondo en grande se tiene la recopilación de la Final del mundial de fútbol 2022 y en pequeño se tiene la señal en vivo de la webcam integrada del pc utilizado.



Figura 15: PiP con video local y webcam.

5.3. PiP con video local y transmisión en vivo

Finalmente para lograr el efecto picture in picture con video local y livestream, los pasos siguen a los ya realizado anteriormente, incluyendo el paso de tener el video en la misma carpeta que el script. La figura 16 muestra el script seleccionando la opción dos, la cual corresponde a PiP de video y livestream. Ya preparado el script se procede a ejecutar.

```
sebita@sebita-HP-Notebook:~/Workspace/practica$ ./pip.sh 2
```

Figura 16: Script con opción uno.

La figura 17 muestra la sintonización de la señal, en el que se logra recibir la transmisión y además el efecto picture in picture. De fondo en grande se tiene la transmisión en vivo rescatada de YouTube y en pequeño se tiene la recopilación de la Final del mundial de fútbol 2022.

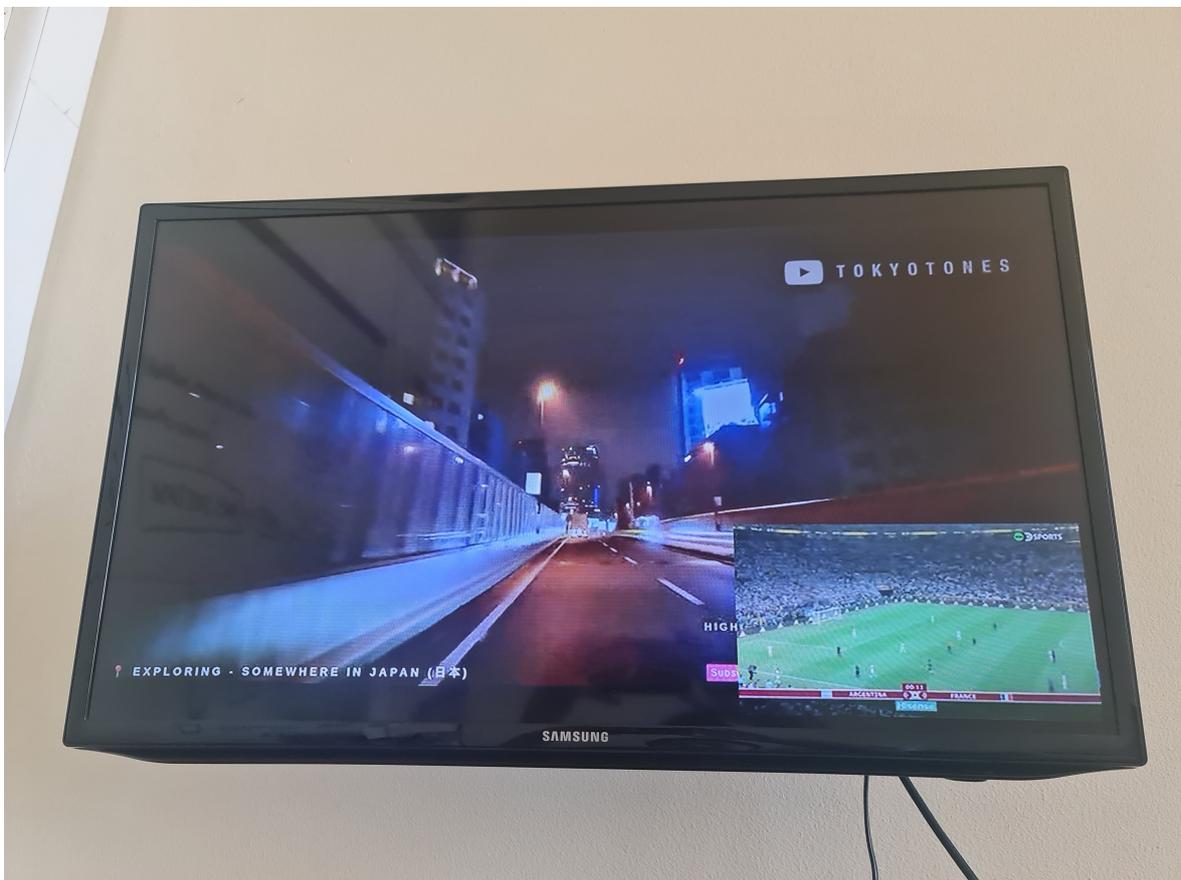


Figura 17: PiP con video local y webcam.

6. Conclusión

En este documento se abarcaron los pasos para transmitir una señal de video, aplicando el formato picture in picture, a la televisión digital de manera satisfactoria.

- Se expusieron los requisitos para realizar el experimento.
- Se explicó cómo preparar un script con tal de facilitar el experimento.
- Se explicó cómo configurar el servidor playout.
- Se explicaron tres formas de aplicar el formato picture in picture junto a pruebas prácticas.

7. Referencias

- PICTURE IN PICTURE effect using FFMPEG. [Fuente.](#)
- FFmpeg Filters Documentation. [Fuente.](#)