

# AMBILIGHT

Montaje y configuración



RASPBERRY PI

OPENELEC

HYPERION

LED5 WS2801

# Índice de contenido

Introducción.....	3
Usando la RB Pi como media Center y como controlador de los leds.....	4
Tiras de leds compatibles.....	5
Esquema de conexionado.....	6
Empezando con el montaje físico.....	8
Por dónde empezar a pegar los leds.....	8
Conectando los leds a la RB Pi.....	13
El convertor de niveles de 3.3V a 5V.....	15
Esquema conexiones del GPIO en la RB Pi 2 modelo B.....	18
Todos los esquemas de conexión de las diferentes RB Pi.....	18
Dimensionado de la fuente de alimentación para los leds.....	18
Instalando y configurando el software.....	21
Instalación manual de Openelec e Hyperion.....	23
Configurando el reproductor multimedia Kodi (opcional).....	24
Adaptando el fichero de configuración de Hyperion.....	26
Generando nuestro fichero config.hyperion.json.....	27
Cómo copiar el fichero de configuración a la Pi.....	33
Usando la aplicación de configuración de Hyperion bajo Android (o iOS).....	36
Preparando el montaje para que acepte cualquier fuente de video HDMI.....	37
Capturas manuales de la imagen de vídeo externa. Ajuste de los recortes.....	44
Notas finales.....	46

## Introducción

En este tutorial se detalla todo el proceso de montaje y configuración que hay que realizar para instalar un [sistema de iluminación ambilight](#) casero similar a los de los televisores de Philips pero adaptable a cualquier TV o monitor. Además del ahorro en el coste final del proyecto si nos decidimos por hacerlo todo nosotros, no hay nada como la satisfacción de montárselo uno mismo y hacerlo funcionar.

El hardware que controla todo el sistema es una [Raspberry Pi](#) equipada con una distribución compatible (**Openelec**). El software de control de los leds será [Hyperion](#).

Además de controlar los leds, la combinación de la RB Pi junto con Kodi constituye un excelente Media Center capaz de reproducir todo tipo de contenidos de audio, vídeo e imagen, de reproducir nuestra colección multimedia almacenada en el PC o en un disco externo, e incluso de reproducir directamente contenidos on-line si la conexión es del suficiente ancho de banda.

Parto de la base de que quien se decide a meterle mano al proyecto, no tiene conocimientos técnicos avanzados. Por ello he tratado de explicarlo todo detalladamente, y por la misma razón, para los que ya tienen conocimientos de electrónica y/o informática, la mayor parte de lo que se expone les parecerá obvio y probablemente redundante. Pero la idea es ayudar precisamente a todos aquellos que de todo esto no tiene ni idea, y que vean que con cuatro conceptos básicos y **leyendo todo desde el principio hasta el final**, va a conseguir montar su propio **ambilight**. No son necesarios conocimientos de programación ni vamos a enviar una sonda a la luna... así que cualquiera con un poco de empeño y dedicación lo puede conseguir.

Existen muchas posibilidades a la hora de decidirnos por una u otra distribución para nuestra RB Pi, de si queremos instalar todo desde cero o de si preferimos una imagen ya preparada con todo lo necesario, de si lo vamos a destinar a ver la TV o si lo queremos controlar desde el PC, etc. Por ello y dependiendo de unas circunstancias u otras y de la disponibilidad de medios y conocimientos, podemos optar por diferentes combinaciones de *hardware* y de *software*. Pero como abarcar todas las posibilidades y variables haría este tutorial inacabable, nos vamos a centrar en las siguientes características:

- El procesador para todo el sistema será una **Raspberry Pi modelo 2 o 3**. Si no sabes lo que es una Raspberry Pi, pásate primero por [aquí](#). Si quieres profundizar en el tema, lo mejor es navegar por el [foro de soporte oficial](#). Además de ser el hardware de control, nos permitirá montar un excelente centro multimedia basado en Kodi. Ahora sí que tu aparato de TV se convertirá en una verdadera televisión inteligente ;)
- El software de control de los leds será [Hyperion](#). Es el componente de software encargado de calcular el efecto ambilight y de controlar los leds a través del [puerto GPIO](#) de la Raspberry Pi. Como alternativa existe [Boblight](#), pero consume muchos más recursos, sobrecarga la CPU de la Raspberry Pi y no funciona de forma independiente de Kodi.
- Usaremos una imagen preparada de [Openelec](#) que ya incluye Hyperion preinstalado, así que tendremos mucho menos trabajo que hacer.

- Opcionalmente será posible disfrutar del ambilight con fuentes de vídeo externas a la Pi añadiendo unos periféricos adicionales y modificando muy poco la configuración de nuestro sistema. Pero esto queda para la parte final del tutorial.

Si quieres comparar el ambilight 'casero' frente el original, puedes ver el siguiente vídeo en el que aparecen funcionando ambos lado a lado:

[Ver vídeo que compara el ambilight casero con el oficial de Philips](#)

Hay que puntualizar que Philips comercializa TVs con tres posibles configuraciones del ambilight: sólo en dos lados, en tres lados (en los laterales y arriba) o en los cuatro lados de la TV. En el vídeo anterior el ambilight de Philips es el de 3 laterales.

Para todos aquellos que no se sientan cómodos cacharreando y teniendo que pelearse con configuraciones, cableados, etc existen actualmente algunas alternativas sin pasar por la adquisición de un nuevo aparato de TV con el ambilight incluido de serie. En todos los casos hay que realizar como es lógico algún que otro montaje y configuración, pero muchos de ellos son casi 100% 'conectar y listo'. A modo de ejemplo y sin exponer en detalle todas sus posibilidades, existen las siguientes alternativas comerciales:

- **DreamScreen**: un simple *stick* HDMI que controla los leds a partir de lo que captura en su entrada.
- **Ambivision**: compacto y casi 'conectar y listo', aunque es necesario adquirir de algún conversor de señal ya que sólo admite VGA en su entrada.

[Estas y otras alternativas, se comentaron aquí.](#)

## Usando la RB Pi como media Center y como controlador de los leds

Si sólo pretendemos usar la RB Pi como **Media Center** y disponer del efecto **ambilight** para los contenidos reproducidos desde la misma, lo que vamos a necesitar es lo siguiente:

- **Tira de leds WS2801** o compatible de suficiente longitud para cubrir el perímetro de tu monitor o TV. Pueden ser de tipo "luces de navidad" o preferiblemente tira de leds smd.
- **Raspberry Pi** (recomendable la versión 2 o la 3) + **microSD de 8GB**.
- **Fuente de alimentación 5V y de por lo menos 6A**. Vamos a necesitar más o menos amperaje dependiendo del número final de leds conectados. Más adelante indico cómo calcularla.
- **Buffer de línea (adaptador de niveles de 3.3V a 5V)** que irá entre los pines GPIO de la RB Pi y la tira de leds WS2801.
- **Opcionalmente**: cables, conectores, soldador, estaño, macarrón termorretráctil, velcro,...
- **Software necesario**: cualquier SO que, siendo compatible con la Pi, admita la instalación de Hyperion. En este tutorial usaremos una imagen de Openelec ya preparada de antemano por los creadores del **Lightberry**, y que ya trae Hyperion preinstalado.

Como ya hemos comentado, la alternativa a usar Hyperion es Boblight, pero es mucho más pesado y lento. Boblight en la Pi anterior (la B+) iba muy justito por tener menos RAM y una CPU mucho más limitada. Actualmente Hyperion es unas 15 veces más rápido que Boblight. Funciona como servicio, por lo que es independiente de Kodi (al contrario que Boblight, que es un add-on y necesita funcionar sobre kodi). Y la carga de trabajo de Hyperion (que es el que en el fondo procesa la información de los leds) es de solo un 2% con uno solo de los cuatro 'cores' del procesador de la Pi 2 B en uso (típicamente con 50 leds conectados).

## Tiras de leds compatibles

Hay a la venta infinidad de tiras de leds, pero sólo son válidas para nuestro proyecto las que incluyan controladores **WS2801 o compatibles**. Son un tipo de tiras led controlables mediante el protocolo SPI en las que cada uno de los leds puede ser controlado de forma independiente del resto. Por el contrario, en las típicas tiras RGB más baratas todos los leds asumen el mismo color y brillo simultáneamente.

Las tiras WS2801 se distinguen fácilmente las tiras RGB en que las WS2801 incluyen chips de control, mientras que las RGB no. Ambas son de 4 pines (cables), pero el hecho de que tengan o no chips de control visibles tampoco es señal de que servirán o no para el ambilight. En la siguiente imagen tenemos una tira de leds WS2801 que sí sirve para el ambilight.



### Tira de leds WS2801 como las usadas en nuestro ambilight

Se comercializan también otras tiras -ligeramente más asequibles- en las que los leds también se controlan de forma individual, pero que no sirven para nuestros propósitos. Son las tiras del tipo WS2811/WS2812/WS2813 y sus variantes. Permiten una mayor densidad de leds por metro porque el chip de control (similar al WS2801) está integrado dentro del led. En vez de 4 pines son de 3, lo que ya nos permite saber de antemano que no valen para el ambilight aquí propuesto.

Aunque el vendedor o la página del producto indiquen que son compatibles con las WS2801, no debemos comprarlas porque con la RB Pi 2 (y presumiblemente con la RB Pi 3 también) no funcionarán. Ese tipo de tiras son controlables mediante PWM y no mediante SPI.

Como es lógico, tampoco sirven las típicas tiras de leds RGB porque no permiten el control individual de cada led. En la siguiente imagen se muestra una tira WS2812b de 3 pines, en oposición a las WS2801 que tienen 4 pines.



**Las tiras leds WS2812b y similares no valen para el ambilight aquí propuesto.**

La densidad de leds en las tiras WS2801 suele ser de 30 o 32 leds/m. Si necesitáis más leds por metro, podéis elegir los APA102, que se fabrican en mayores densidades (hasta los 144 leds/m). El efecto ambilight no va a mejorar apreciablemente. Simplemente el efecto será más brillante. Y para el visionado nocturno, que es cuando probablemente se le saque más jugo al ambilight, demasiado brillo puede ser un inconveniente (aunque siempre se puede regular el brillo mediante software). También hay que tener en cuenta que el consumo con más leds por metro se disparará, por lo que será necesario prestar especial atención al dimensionado de la fuente de alimentación. Y sobrecargaremos más la Pi, porque serán necesarios más cálculos al existir más leds que controlar.

## Esquema de conexionado

Como podemos ver en el esquema mostrado más abajo, tendremos la RB Pi conectada mediante un cable HDMI a una de las entradas libres de nuestra TV. La alimentaremos desde una fuente de 5V que puede ser la misma que se use para los leds si la hemos dimensionado adecuadamente. También puede ser alimentada mediante su propia fuente, que **deberá ser de 5V 2A mínimo en el caso de la RB Pi 2 y de 2.5A para la RB Pi 3** (si le vamos a conectar periféricos que consuman cierta cantidad de corriente.)



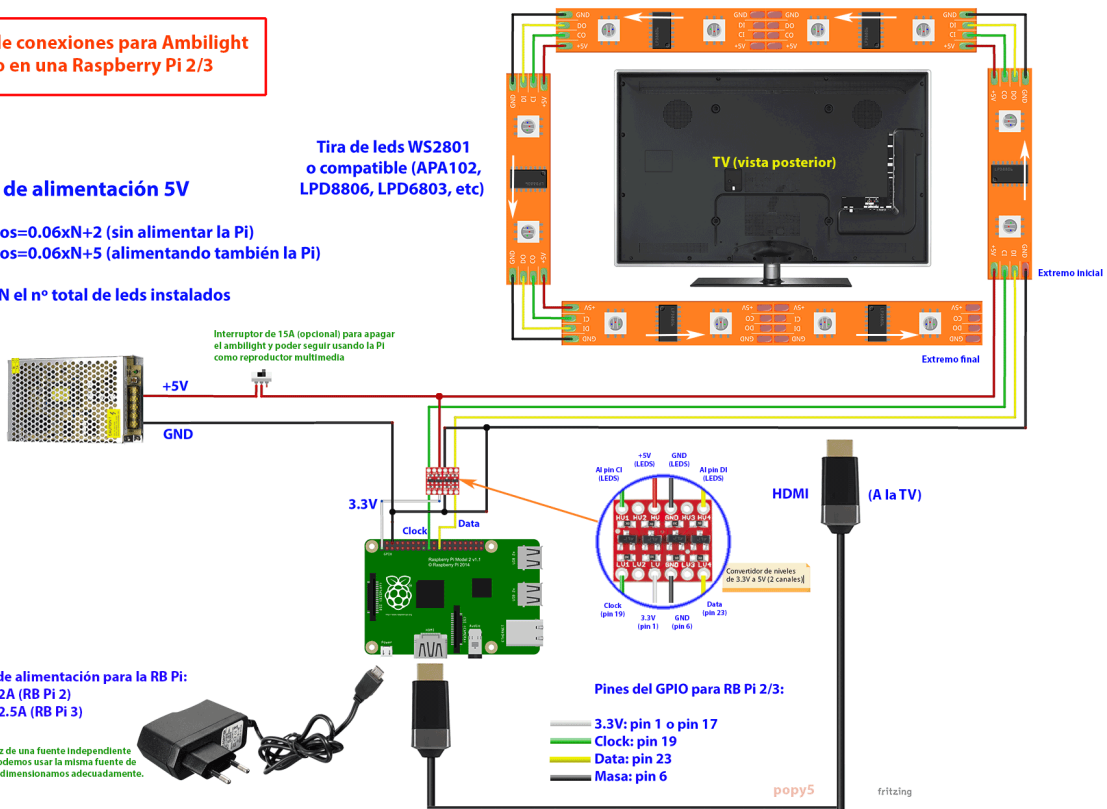
NOTA: Si la Pi la alimentamos con una fuente de 5V independiente de la usada para los leds, es obligatorio unir las dos líneas de masa (GND) de ambas fuentes con una conexión fiable. Esto implica llevar un cable desde le terminal GND de la fuente de los leds al GND de la Pi a través del GPIO por ejemplo.

**Esquema de conexiones para Ambilight basado en una Raspberry Pi 2/3**

**Fuente de alimentación 5V**

Amperios= $0.06 \times N + 2$  (sin alimentar la Pi)  
 Amperios= $0.06 \times N + 5$  (alimentando también la Pi)  
 siendo N el nº total de leds instalados

Tira de leds WS2801 o compatible (APA102, LPD8806, LPD6803, etc)



**Esquema básico de conexiones para el ambilight con la RB Pi 2/3**

En el siguiente enlace está disponible la imagen a mayor resolución: [ver imagen](#).

Para reproducir contenidos alojados en internet (vía *streaming*) o desde nuestra red local, siempre es mejor optar por una conexión de red cableada. Un simple latiguillo de red hasta el *router* o *switch* de nuestra red es suficiente. En la RB Pi 2 es posible conectar además un receptor wifi usb compatible, pero si la vamos a usar para ver contenidos por *streaming*, puede no ser capaz de trabajar con fluidez dependiendo de la calidad de la señal wifi disponible, del ancho de banda de nuestra conexión y del *bitrate* del vídeo reproducido.

No es necesario que la Pi esté conectada a internet. Podemos conectarla sólo a la LAN y reproducir nuestra colección multimedia alojada en el PC. O reproducir contenidos multimedia desde unidades externas (discos duros, memorias flash...).

Sí que será necesario, al menos durante la fase de configuración del software, que tengamos acceso a la Pi a través de una conexión de red (cableada o mediante wifi).

Si adquirimos la RB Pi más reciente (modelo 3 en el momento de escribir esta guía), ya dispondremos de conectividad wifi de serie, y podemos prescindir del adaptador wifi externo. Si la conexión wifi en este caso en concreto os da problemas de estabilidad, de retardos, desconexiones, etc, se recomienda desactivar la gestión de energía del módulo wifi.

## Empezando con el montaje físico

Lo primero que deberemos hacer es pegar la tira de leds a la parte trasera de nuestra TV. Tanto si los leds son del tipo "luces árbol de navidad" o son tipo tira de leds SMD, deberemos elegir la forma en la que serán pegados a la parte posterior de la TV.

Los leds tipo "luces árbol de navidad" son bastante más liosos de adherir a la trasera de la TV, de orientarlos correctamente hacia la pared y estéticamente bastante menos discretos que una tira de leds smd. En contrapartida no requieren realizar soldaduras en las esquinas o emplear conectores para ello.

Por su parte, las tiras de leds suelen venir con una cinta adhesiva de doble cara ya instalada, por lo que basta con limpiar convenientemente la zona en la que deberá ir adherida y pegarla con cuidado. Como las tiras de leds son bastante delicadas, hay que definir previamente dónde las pegaremos, ya que luego dependiendo de la calidad del adhesivo, puede ser complicado despegarlas y volverlas a pegar para tratar de corregir errores o hacer alteraciones.

Otra posibilidad pasa por utilizar unas tiras de velcro a modo de adhesivo para los leds. Pegaremos una de las dos partes de la tira de velcro a la parte trasera de la TV en donde deberá ir nuestra tira de leds, y la otra cinta de velcro la pegaremos al adhesivo que trae la tira de leds. Luego bastará con fijar la tira de velcro de los leds sobre la base de velcro ya pegada antes en la TV.

## Por dónde empezar a pegar los leds

Antes de empezar a pegar la tira de leds, debemos identificar el extremo inicial de la misma. Dependiendo del fabricante, la serigrafía en la tira de leds puede variar. La tira es de 4 pines y generalmente los pines superior e inferior de esos 4 corresponden a los de alimentación de los leds (+5V o Vcc y GND). Los dos del medio serán por norma general los de datos.

Debemos tratar de localizar en uno de los dos extremos de la tira los textos **DI/CI** (data input/clock input). Estos serán los pines de entrada de los datos. Por lógica el otro extremo de la tira debe venir identificado como **DO/CO** (data output/clock output), es decir, será el extremo final.

Normalmente el aislante que traen sobre las soldaduras de los extremos tapa la serigrafía, por lo que en el peor de los casos será necesario cortarlo para identificar qué extremo es. Pero en general basta con fijarse en cómo va la secuencia de pines a lo largo de toda ella. Empezando por un extremo, si vemos que la secuencia a lo largo de toda la tira es:

... | entrada → salida | entrada → salida | entrada → salida | ...

donde el carácter | indica la línea de corte entre cada dos leds, ése extremo por el cual hemos empezado es el de entrada (porque la señal va siempre desde la entrada hacia la salida). Si por el contrario la secuencia que vemos es:

... | salida ← entrada | salida ← entrada | salida ← entrada | ...

donde el carácter | indica la línea de corte entre cada dos leds, ése extremo por el cual hemos empezado es el de salida.

Lo normal es que a lo largo de toda la tira se repitan las secuencias de los textos anteriores de entradas y salidas. Cada tramo en el que hay un led y un chip de control puede trabajar de forma independiente del resto. Por eso van a existir multitud de entradas y salidas a lo largo de toda la tira, y la salida de un tramo va siempre conectada a la entrada del siguiente, y así sucesivamente.

Puede que en la serigrafía aparezcan identificados los pines de control como **CK** y **SD** (Clock y Serial Data respectivamente), y en este caso deberían ir acompañados de los textos **IN** (entrada) y **OUT** (salida).

Algunas tiras sólo identifican uno de los extremos a lo largo de toda su longitud, es decir, que sólo indican cuáles son los pines de entrada por ejemplo. Por lógica, los contactos justo al lado de los de entrada que no tienen identificación son los de salida. Otras fabricantes los identifican como **CK/SD IN** para la entrada y **CK/SD OUT** para la salida, o como **CK/SI**, etc.

A lo largo de la tira deben existir unas líneas de corte que a veces aparecen identificadas con el símbolo de unas tijeras. A un lado de la línea de corte está el final de un tramo (DO/CO), y al otro lado, el inicio del siguiente (DI/CI), y así sucesivamente hasta el final de la tira.

Otra forma de identificar el extremo inicial podría ser viendo si aparece una pequeña flecha impresa. La flecha siempre apunta hacia el extremo final de la misma (indica el sentido en el que se propagan los datos desde el punto de entrada). **Los cortes de la tira sólo se deberán hacer por esas líneas que separan la salida de un tramo de la entrada del siguiente.**

La flecha serigrafiada (si existe) indica el sentido en el que se propagan las señales



### Esquema simplificado de una tira de leds WS2801 o compatibles

El punto de partida desde el cual empezaremos a pegar en la trasera de la TV el extremo inicial o de entrada de la tira de leds puede ser cualquiera. Lo normal es hacerlo, viendo la TV desde atrás, por la esquina inferior derecha de la TV, y continuar en el sentido antihorario (siempre viendo la TV desde detrás), es decir, primero hacia arriba por el lateral derecho, luego hacia la izquierda por la parte superior de la TV, a continuación hacia abajo por el lateral izquierdo, y finalmente hacia la derecha por la parte inferior, hasta llegar al punto desde el que hemos iniciado el pegado.

También podemos iniciar el pegado de la tira de leds desde la parte central inferior (a la altura del soporte de la TV) y continuar hacia la derecha (o hacia la izquierda, porque se puede después configurar en el software el sentido de giro que hemos elegido), luego hacia arriba, a izquierdas nuevamente, hacia abajo y de nuevo a derechas hasta el centro de la parte inferior. Quizás este sea el modo más adecuado en algunas TVs dependiendo de cómo tengamos hechas las conexiones traseras en la misma, porque es ahí en la parte central inferior donde suelen pasar todos los cables, y el pegado de la tira empezando en la zona central puede molestar menos.

Para facilitar el pegado y para que no nos estorben los cables, es mejor poner la TV apoyada sobre una superficie horizontal (protegiendo adecuadamente la pantalla) como la de una mesa o un sofá. Así podremos pegar los velcros y/o las tiras de leds más cómodamente, y si necesitamos realizar soldaduras, se facilitará mucho el trabajo.

Deberemos situar los leds cerca del borde de la TV pero de tal forma que no sean visibles desde la parte frontal de la misma aunque estemos algo desplazados lateralmente de la perpendicular a la TV. En caso contrario el efecto ambilight no será como el pretendido, y el brillo de los leds acabarán por molestar más que ayudar a una experiencia inmersiva.

Si es posible, orientaremos los leds de forma que apunten hacia la parte posterior pero ligeramente hacia afuera para que se abra un poco el cono de iluminación. La línea perpendicular a los leds es preferible que no llegue también perpendicular a la pared. La idea es iluminar con el

ambilight la zona de la pared que es visible desde el frontal de la TV. Por eso si la base del cono de iluminación de cada led es perpendicular a la pared, la mitad del mismo quedará oculto por la TV.

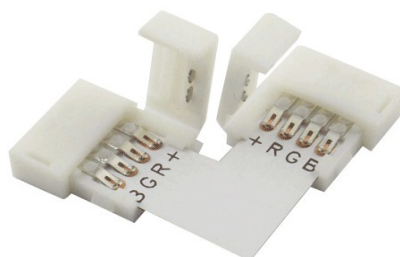
En la web de Lightberry indican que los leds HD (las tiras de leds smd) se deben colocar de forma que apunten no a la pared, sino completamente hacia los laterales ([ver aquí](#)), y de hecho incluyen unos soportes/escuadras plásticas para colocarlos así. Desde mi punto de vista esta disposición sólo será efectiva si la TV está muy cerca de la pared, pero si la tienes alejada 30cm o más, es mucho mejor que los leds apunten ligeramente abiertos hacia la superficie reflectante porque se perderá menos luz. La única ventaja de este sistema que ellos proponen es que te evitas cortar la tira en las esquinas y usar conectores o soldaduras.

Antes de nada habrá que tomar medidas para determinar las longitudes de cada tramo de leds para la parte superior, inferior, y los dos laterales de la pantalla. Los dos tramos horizontales deberán tener ambos el mismo número de leds (y lo mismo los dos tramos verticales entre sí). Cortaremos los trozos de la tira de leds por las líneas de corte que aparecen en ella (entre los contactos de salida de un tramo y los de entrada del siguiente).

Es posible que el tramo inferior nos veamos obligados a pegar menos leds que en el superior por culpa de la forma irregular en la zona del soporte/peana. O incluso a optar por dividir la tira inferior en dos trozos con menos leds en conjunto que en la superior. Pero no tiene importancia porque desde el configurador le podemos indicar el número final de leds a usar. Si hemos pegado uno o dos más permanecerán apagados, y si en el centro hemos pegado menos, se le indica el tamaño del hueco sin cubrir.

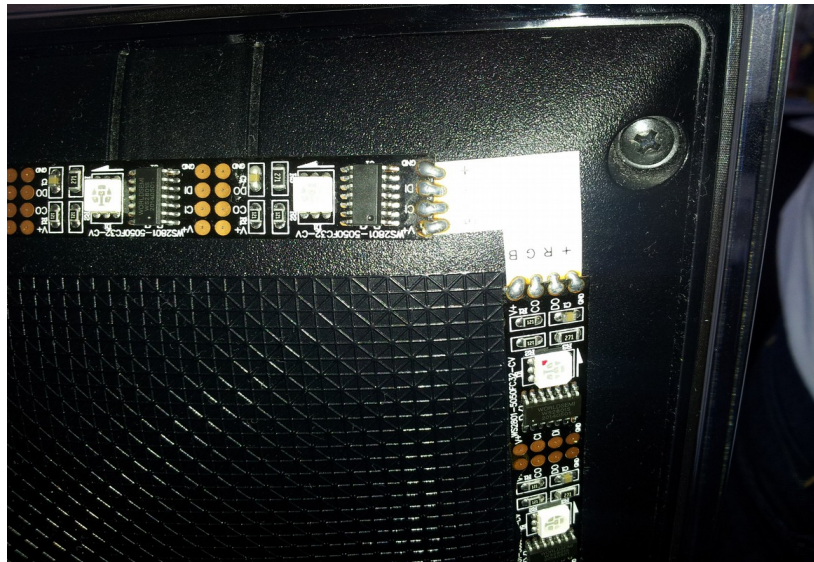
A continuación debemos elegir la forma en la que se harán las conexiones en cada una de las esquinas, y aquí tenemos básicamente tres posibilidades:

- **Usar conectores cableados o en L para las esquinas:** se intercalan entre el final de un tramo y el inicio del siguiente. Pueden ir a presión, aunque recomiendo abrirlos y soldarlos una vez colocados en su sitio definitivo. Verificad a la hora de comprarlos que son para el ancho de vuestra tira de leds (habitualmente las tiras son de 12mm de ancho, y la mayoría de conectores en L suelen ser de 10mm de ancho). Si no los conseguís de 12mm, mejor optar por soldar los contactos internos de los conectores (ver la siguiente solución propuesta).



**Conector a presión en L**

- Usar pequeños cables soldados entre el final de una tira y el inicio de la siguiente. O aprovechar la tira en "L" del interior de los conectores antes mencionados para realizar las soldaduras en vez de usar cables, como se ve en la foto siguiente.



Tira en L de un conector soldada en las esquinas

- Plegar la tira de leds siguiendo el siguiente esquema:



Plegado de la tira de leds para los giros de 90° en las esquinas

### Plegado de la tira de leds para crear un cambio de dirección de 90° sin cortarla

De los tres posibles modos de "acabado" para las esquinas, el más simple a primera vista

probablemente parezca el plegado de la tira, aunque dependiendo del grosor o del tipo de acabado, puede no ser posible (los componentes smd en la tira tampoco dejan mucho margen de espacio por el que plegarla con garantía de éxito). Además, el plegado debería hacerse a la primera y sin forzar demasiado, porque las pistas de cobre sobre la tira flexible de soporte son muy finas, y si plegamos y desplegamos varias veces por el mismo sitio, se acabarán por cortar.

La solución que peor calidad de conexionado suele dar es la del conector a presión a la vista de los comentarios aportados por otros usuarios: o no son de las medidas exactas y no hacen buen contacto o ni siquiera encajan, o a la larga acaban por dar muchos problemas de fiabilidad.

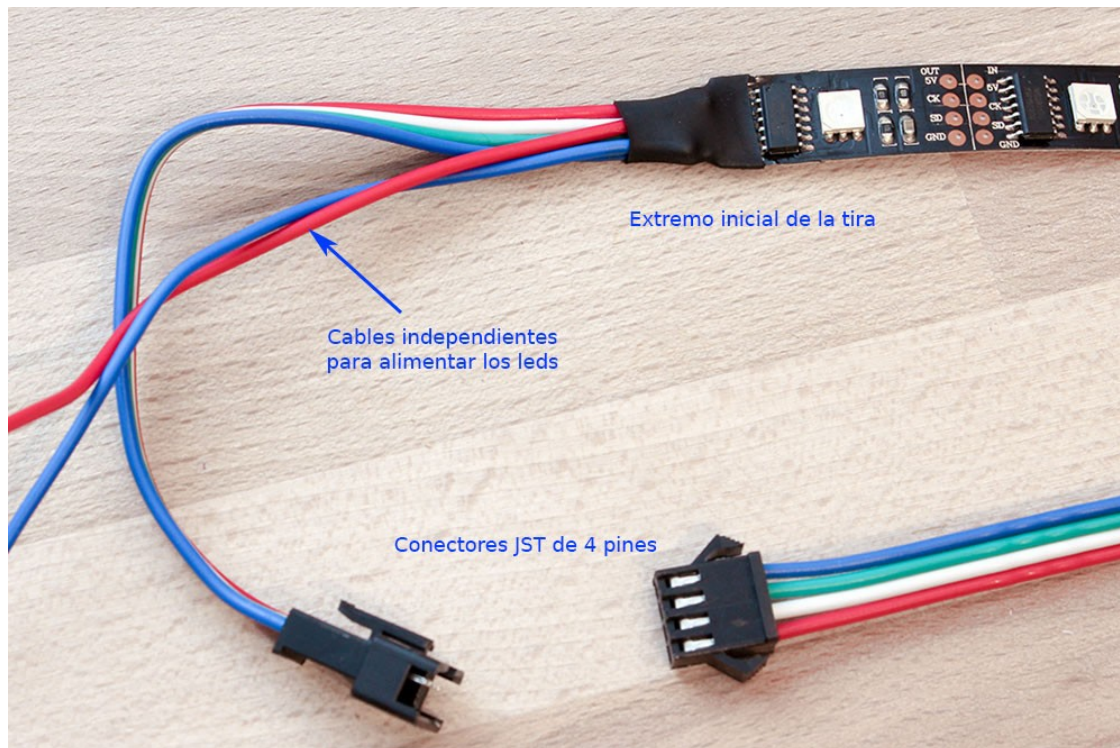
Recomiendo optar por la segunda solución: comprar los conectores en L, abrirlos y aprovechar sólo la tira en forma de L del interior de los mismos, y soldarla. En cualquier caso el método finalmente elegido dependerá de las habilidades y medios disponibles.

Recordad que las **tiras de leds tienen orientación**, es decir, que las señales de control entran por un extremo y salen por el otro, **y no son intercambiables**, por lo que **en las esquinas el final de un tramo debe conectarse con el inicio del siguiente**. Las flechas o textos serigrafiados en la tira nos sirven para determinar qué extremo es de entrada y cuál de salida. El extremo inicial es el que deberá conectarse a los pines GPIO de la RB Pi a través del convertidor de niveles de 3.3V a 5V que comentaré más adelante.

## Conectando los leds a la RB Pi

Una vez que tengamos la tira de leds pegada a nuestra TV, procederemos a conectarla al convertidor de niveles. En el extremo inicial de la tira de leds tenemos 4 conexiones: +5V, masa (GND), DI (Data Input) y CI (Clock Input). Algunas tiras incluyen además dos cables sueltos (suelen ser rojo y negro) en uno de los extremos, o incluso en ambos, y van soldados a los pines de +5V y GND. Si es así, esos son los cables que deberemos usar para darle alimentación a los leds. Podemos alimentar la tira por cualquiera de los dos extremos, o incluso por los dos simultáneamente.

En la siguiente imagen podemos ver como suelen venir preparadas las tiras de leds: con un **conector JST de 4 pines** (un macho y una hembra) y dos cables adicionales para la alimentación.



Los conectores no suelen ser de mucha calidad, y a la larga pueden dar problemas. Tampoco suelen venir siempre de igual forma soldados: unas tiras traen sólo uno de ellos soldado (generalmente en el extremo inicial) y el otro viene suelto; otras traen los dos conectores soldados -uno en cada extremo-. Y por si fuera poco, el conector de entrada puede ser el macho o la hembra... por eso no hay que confiar ni en los colores de los cables ni esperar que el conector de entrada sea siempre el macho.

Las conexiones de alimentación de los leds deben ir a los correspondientes bornes de la fuente de 5V como es lógico. Opcionalmente podemos incluir un pequeño interruptor de 15 o 20A en la línea de 5V que va hasta los leds. De esta forma podemos apagar el ambilight cuando no sea necesario, y mientras seguir usando la Pi como Media Center.

Otra alternativa más sofisticada pasa por programar un script para poder activar y desactivar el ambilight a voluntad con el mando a distancia de la TV. Los detalles los publicó el usuario **Bite\_Your\_Idols** en [éste mensaje](#).

A la hora de hacer las conexiones, verificad varias veces que lo hacéis con la polaridad correcta antes de alimentar nada. **No os fiéis del código de colores de los cables que salen de la tira de leds.** Aunque en circuitos de CC el rojo suele ser el positivo, mejor certificarse de que sí lo es. En mi caso por ejemplo, hay un cable rojo con el extremo libre que es para alimentar la tira, y otro cable rojo en el conector macho y en el hembra de cada extremo que es para una de las señales de control. Como suele decirse, "se mide dos veces y se corta una".

Si habéis adquirido una fuente de alimentación de tipo industrial, antes de aplicarle cualquier

carga, verificad con un polímetro en la escala de medición de voltaje de CC que la salida da los 5V requeridos. Si no es así, habrá que ajustarla al valor correcto girando el potenciómetro de ajuste. Después podéis fijarlo con una pequeña gota de barniz de uñas/cera derretida/pegamento tipo "supergen" para que no se desajuste accidentalmente.



**Fuente de alimentación industrial de 5V.**  
**Al lado del led verde aparece el potenciómetro de ajuste del voltaje de salida.**

La otra punta de la tira de leds (el extremo final) puede llevar o no cables y/o conector. En el caso de que instalemos 4 o 5 metros de leds, puede ser recomendable "inyectar" la alimentación por los dos extremos de la tira. Para ello sólo hay que conectar los cables +5V y masa (GND) del extremo final a los mismos cables del extremo inicial para que, de esta forma al inyectar corriente por ambos extremos, se minimicen las pérdidas por caída de tensión en una tira de leds demasiado larga.

## El convertidor de niveles de 3.3V a 5V

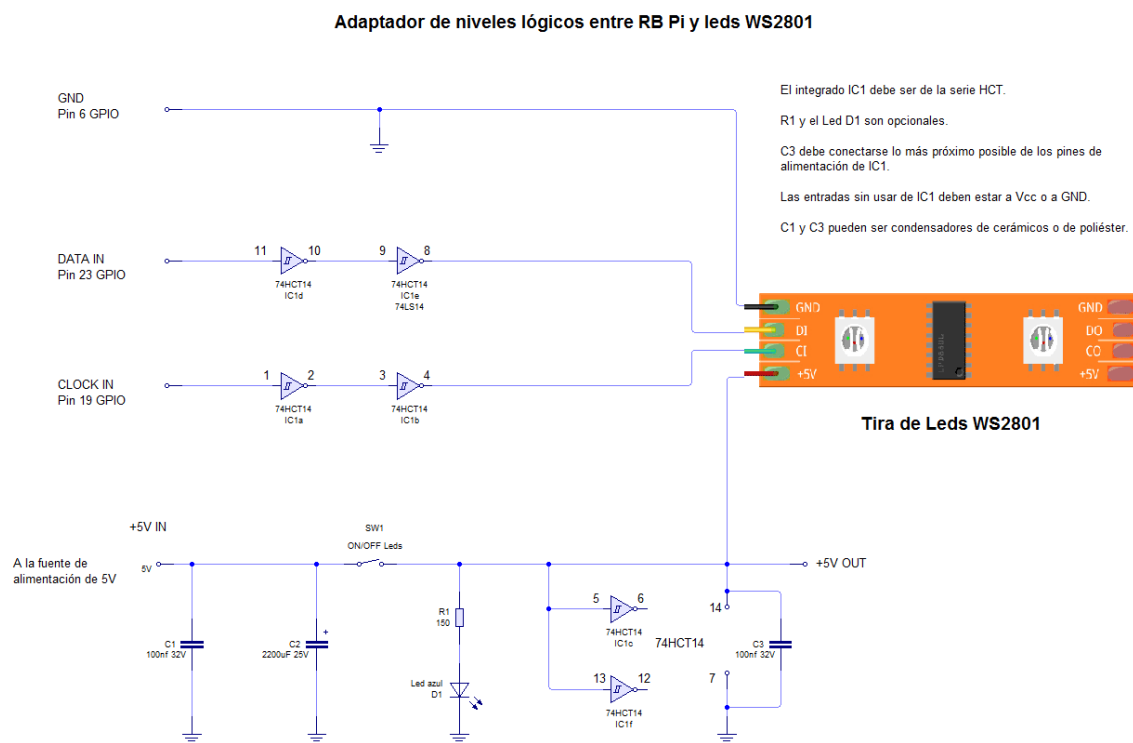
El convertidor de niveles realiza la función de adaptar las señales lógicas de 3.3V del GPIO de la Pi hasta los 5V de la lógica empleada en los chips WS2801 que controlan los leds. [Según las especificaciones de los chips WS2801](#), un nivel alto deberá alcanzar un valor de tensión en las entradas DI/CI de  $0.8 \times V_{cc}$ , siendo  $V_{cc} = 5V$  la tensión con la que se alimentan los chips.

Por ello, el nivel mínimo de tensión que teóricamente admiten los WS2801 para considerar una señal de entrada un "1" lógico debe ser de  $0.8 \times 5 = 4V$ . Sin embargo, la Pi trabaja con lógica a 3.3V,

por lo que un "1" lógico en el GPIO alcanzará como máximo un nivel de tensión de 3.3V, lo que está bastante alejado del mínimo aceptado de 4V de los chips WS2801. Por eso es recomendable instalar el convertidor: **nos garantiza señales lógicas compatibles entre ambos dispositivos.**

En la práctica, casi nadie lo instala y el ambilight funciona, lo que indica que las tolerancias de fabricación de los WS2801 son, por decirlo finamente, más anchas que Castilla (o en caso contrario sería del todo imposible que la comunicación funcionase).

Podemos montar nosotros mismos el adaptador de niveles usando integrados de la serie 74HCT. Por ejemplo, podemos usar un 74HCT00 (4 puertas NAND de dos entradas), un 74HCT32 (4 puertas OR de dos entradas), un 74HCT08 (4 puertas AND de dos entradas) o incluso un 74HCT04 (6 inversores) ([más info](#)). Bastará con combinar las puertas lógicas de tal forma que al final obtengamos un *buffer* de línea sin inversión de la señal para cada una de las dos señales a transmitir. Y es importante recalcar que el chip debe ser de la serie HCT para que funcione sin problemas con las señales de 3.3V de entrada.

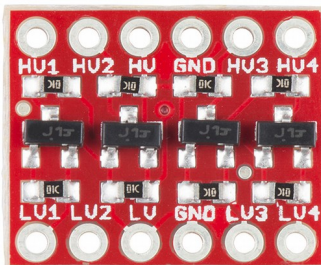


En el siguiente enlace está disponible la imagen anterior con más resolución: [descargar imagen](#).

Otra alternativa que dará menos trabajo que montar desde cero el convertidor es construirlo usando las típicas plaquitas adaptadoras de niveles (de 3.3V a 5V y viceversa) como las que se usan para conectar periféricos de 3.3/5V a los arduinos. Cuestan pocos céntimos de euro y sólo necesitamos 2 canales unidireccionales, aunque si el conversor es bidireccional o de más de 2 canales, servirá igualmente.

Los dos hilos que nos faltan por conectar (ya hemos conectado los de alimentación de los leds a la

fuente) son las señales de control **DI (Data Input)** y **CI (Clock Input)**. Estas dos líneas de datos deben ir al GPIO de la Pi, pero pasan antes por el convertor de niveles. Vamos a suponer que el convertor lo habéis montado con una plaquita de las usadas en los arduinos, como ésta:



### Convertor de 3.3V a 5V como los empleados en los arduinos

Este tipo de convertidores pueden adaptar los niveles de tensión de varios canales o líneas de datos. Para nuestro caso con 2 canales es suficiente (uno para la línea Clock y otro para Data). Si es de más canales os servirá igualmente aunque desaprovecharéis varios.

Generalmente estos convertidores tiene un lado de baja tensión (marcado como LV) y un lado de 'alta tensión', marcado como HV. El lado de baja tensión o LV es aquel en el que debemos conectar los pines que llegan desde el GPIO de la Pi (son los que trabajan a 3.3V).

El lado de alta tensión o HV debe ser conectado a los pines de entrada de la tira de leds (es lado que trabaja a 5V). Los pines de GND del lado de baja (LV) y del de alta (HV) deben conectarse entre ellos si no lo están en la propia plaquita.

**La línea de +5V para el lado de alta debe provenir de la misma alimentación de 5V proporcionada a los leds.** Sólo si usamos una única fuente de alimentación para los leds y la Pi podemos aprovechar uno de los pines de +5V del GPIO para conectarlo al lado de HV. **Si no estamos en esa situación, la línea de 5V a él conectada tiene que ser obligatoriamente la de alimentación de los leds.**

Algunas plaquitas conversoras incluyen dos líneas de conversión por canal, esto es, una para enviar datos y otra para recibirlos. Revisad las especificaciones para saber dónde debemos conectar cada pin en este caso, teniendo en consideración que sólo usamos uno de los dos posibles sentidos para las comunicaciones (es decir, sólo enviamos datos desde la Pi hacia los leds, pero no los recibimos). [Más info aquí.](#)

Por tanto, el pin 19 del GPIO de la Pi deberá ir al lado de baja de uno de los canales del convertor (al LV1 por ejemplo), y el pin 23 del GPIO a otra entrada del lado de baja (al LV2 o cualquier otro libre). Por el otro extremo del convertor, llevaremos la línea que sale del pin HV1 a la entrada CI o Clock de la tira de leds. La salida del lado de alta de la otra línea (Data) será el pin HV2 (o el correspondiente al que hayamos usado como entrada), e irá al pin DI (Data) de la tira de leds.

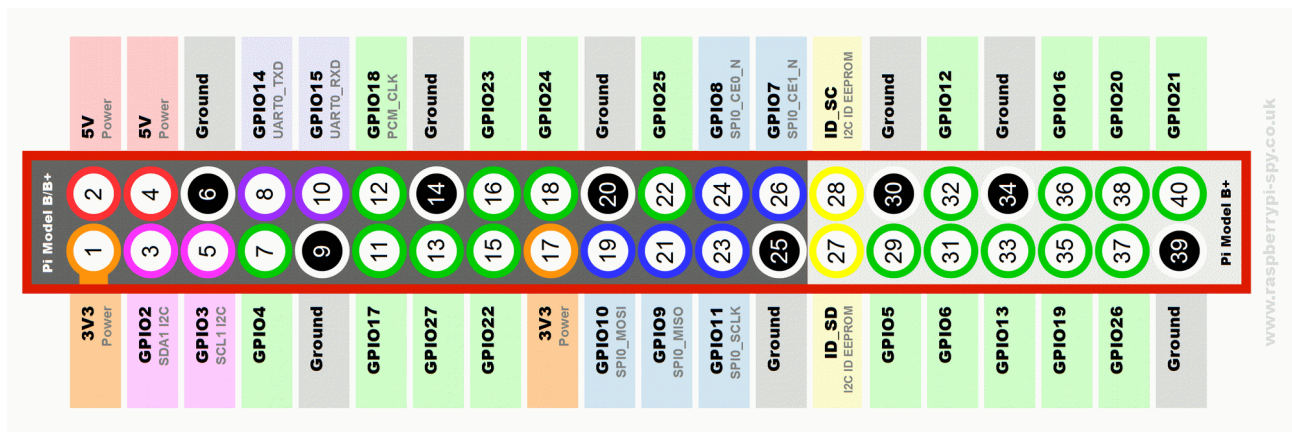
La tercera alternativa es adquirir el convertidor en forma de kit (para montarlo tú mismo) o ya montado. Lo podéis encontrar en este enlace:

<https://hackerspaceshop.com/collections/ambipi-tv/products/ambipi-tv>

En [éste](#) mensaje explico las variantes del kit.

## Esquema conexiones del GPIO en la RB Pi 2 modelo B

La Pi 2 modelo B tiene el mismo *pinout* que los modelos A+ y B+.



## Todos los esquemas de conexión de las diferentes RB Pi

Todos los esquemas de conexiones del GPIO de las diferentes versiones de la RB Pi están disponibles en [ésta página](#).

## Dimensionado de la fuente de alimentación para los leds

Una tira de leds del tipo WS2801 incorpora leds smd con encapsulado 5050. Un led 5050 está constituido por tres leds independientes (uno rojo, uno verde y uno azul) integrados dentro de un encapsulado común. El consumo máximo de cada led RGB 5050 en la peor de las circunstancias, esto es, cuando los tres leds individuales están encendidos a máximo brillo, es de 60mA. Por tanto, conocido el total de leds instalados e independientemente de la densidad de leds por metro de la

tira elegida, el consumo máximo de nuestros leds determinará la potencia que debe suministrar nuestra fuente de alimentación.

Mediante la siguiente fórmula podemos dimensionar la fuente sin complicarnos demasiado:

**$I_{max} = 0.06 \times N + 2$**  si sólo vamos a alimentar los leds con la fuente

**$I_{max} = 0.06 \times N + 5$**  si además de los leds, alimentamos la Pi con la misma fuente

siendo N el nº total de leds instalados.

Si el vendedor nos indica en los datos de la tira que cada led individual tiene un consumo máximo diferente de 60mA, sustituiríamos el valor 0.06 por los amperios que consumiese un único led. Pero para todas las tiras que empleen leds 5050, ése es el valor máximo de corriente (por led) esperado.

Como medida de seguridad le añadimos el factor 2 (o 5, según el caso) con la finalidad de dejar cierto margen para posibles picos, y por si luego alimentamos algún pequeño periférico con la misma fuente, como el hub USB, el convertidor de HDMI2AV o el *splitter*.

La fuente tiene que ser obligatoriamente de una tensión de salida de 5V para alimentar la Pi y la mayoría de tiras de leds disponibles. Hay que revisar con cuidado el voltaje de nuestros leds. Casi todas las tiras trabajan a 5V, pero también se venden de 12V y de 24V.

Si hemos comprado una tira de 12 o 24V, el adaptador de niveles de 3.3V hasta el voltaje de trabajo de los leds es obligatorio, pero tendréis que diseñarlo desde cero (no valen los hasta ahora propuestos)... Pero una cosa sí está clara: sin él, no funcionará. Habrá que analizar con cuidado los posibles circuitos de conversión, ya que no todos servirán... [Éste por ejemplo](#) es compatible con lógicas de 12V, 2.8V y 1.8V (además de 3.3V y 5V).

Lógicamente deberemos adquirir una fuente de alimentación del voltaje requerido por la tira de leds. Además hay que tener mucho cuidado con la alimentación de los leds si la tira es de 12/24V para que accidentalmente no se derive esa tensión hacia la Pi. Un error al conectar el conversor o al usar el canal de datos que convierte desde el lado de alta al de baja en vez del opuesto y te cargarás sin remedio el GPIO de la Pi.

Es posible usar un alimentador independiente para la Pi y la fuente de alta potencia usarla sólo para alimentar los leds. Si ese es nuestro caso, debemos interconectar entre sí las dos masas (GND) de las dos fuentes de alimentación usadas. Si no conectamos entre sí ambas masas, el ambilight no funcionará.

**Ejemplo:** *Nuestro montaje tiene en total 84 leds y alimentaremos con la fuente de 5V también la Pi y los periféricos externos.*

La fuente necesaria para alimentar todo el montaje deberá ser -según la fórmula antes expuesta- de, por lo menos,  $0.06 \times 84 + 5 = 10.04A$ . Elegiríamos entonces la fuente de 5V que nos dé esa corriente de 10A. En caso de no existir de ese valor, elegiríamos la del valor disponible inmediatamente superior (12A por ejemplo).

Si la fuente que conectamos puede dar (muchos) más amperios que los que requiere nuestro montaje, servirá perfectamente... simplemente funcionará mucho más holgada al no llegar a su máximo teórico nunca, pero funcionará sin problemas (más vale que sobre que no que falte, como suele decirse).

El margen de seguridad añadido en la fórmula anterior es bastante generoso, por lo que con una fuente de 8A también trabajaría sin problemas en el caso del ejemplo anterior (si no le conectamos ninguna otra carga más que los leds). Hay que tener en cuenta que en el funcionamiento normal del ambilight no es muy probable que los leds estén al 100% de brillo e iluminando siempre en color blanco -que es el caso más desfavorable- durante horas seguidas, por lo que con estos márgenes la fuente no se debería recalentar demasiado (ni aún después de varias horas funcionando). Además, dada la dudosa calidad de algunas fuentes, es mejor contar con un margen suficientemente amplio para evitarnos posibles sorpresas.



**IMPORTANTE: no es posible alimentar los leds desde los pines de +5V del GPIO de la Pi. La Pi es incapaz de suministrar la corriente necesaria que demandarán los leds.**

Es necesario usar cables de alimentación independientes para los leds, y deben ir directamente hasta la fuente. Los debemos seleccionar de la sección adecuada para la corriente máxima estimada. No es recomendable usar cables demasiado finos (y a veces, los del conector de datos son demasiado finos) para alimentar los leds si las intensidades son muy elevadas y/o las distancias hasta la fuente de alimentación son grandes.

Si la corriente es excesiva, corremos el riesgo de quemar los cables, o que directamente el montaje no funcione como debe por las caídas de tensión introducidas por unos cables inadecuados. Un cable que puede servirnos -siempre que no sea demasiado fino- es el típico que se usa para conectar cajas acústicas y altavoces (y además viene ya con el cable 'positivo' bien diferenciado del negativo).



**Si usamos fuentes de alimentación independientes para los leds y la Pi, es obligatorio interconectar las masas (polos GND o V-) de ambas fuentes entre sí (¡pero no los polos positivos!!).**

Por último, sólo resta conectar mediante un cable HDMI nuestra RB Pi a la TV, alimentarla mediante los 5V requeridos y conectarle el cable de red para poder acceder a ella desde nuestro ordenador.

Con el montaje ya acabado y después de que le hayamos enviado a la Pi el fichero de configuración de los leds como se explica en los siguientes apartados, ya podremos disfrutar del ambilight para todos los contenidos que reproduzcamos desde la Pi, es decir, para aquellas señales de vídeo que salen por el cable HDMI que une la Pi con la TV.

Para poder disfrutar del ambilight con otras fuentes externas de vídeo HDMI (como una consola de juegos, un reproductor de BluRay/DVD, un TDT, etc) es necesario adquirir una serie de componentes auxiliares y modificar ligeramente el código del fichero de configuración. Pero esto lo comentaremos hacia el final del tutorial.

## Instalando y configurando el software

Para que nuestra RB Pi funcione como un potente Media Center necesitamos, como ya se indicó al inicio, descargar una distribución de Kodi (antes XBMC), y adicionalmente, algún software que nos permitirá controlar la tira de LEDs. En cuanto al software para el control de los leds tenemos dos posibilidades: **Hyperion** o **Boblight**. El más recomendable, en especial si tienes una RB Pi model B de 512MB, es Hyperion porque consume muchos menos recursos que Boblight, y no la llevará tan al límite.

La instalación de **Hyperion** puede hacerse vía terminal de comandos, pero es más rápido y cómodo descargarnos alguna de las distribuciones ya existentes al efecto desde la web de [Lightberry](#) basadas en Raspbmc, OSMC o bien Openelec.

En esta guía vamos a usar la versión de **Openelec** más actualizada de las disponibles en la web de Lightberry (actualmente, la versión 6.0.3). No suele ser la versión de Openelec más reciente, pero en contrapartida ya **trae Hyperion preinstalado**.

Quien prefiera disponer de la última versión de Openelec podrá descargarla de [aquí](#), y deberá posteriormente instalar de forma manual Hyperion (y muy probablemente realizar algunos retoques en la configuración para hacerla funcionar).

- Primero descargamos la imagen correcta de Openelec para nuestro montaje (más info [aquí](#)).

### Para la RB Pi 1:

- OpenELEC 6.0.0 for RPI 1: [descargar](#).
- OpenELEC 5.0 with STK1160/UTV007 compatible kernel RPI 1: [descargar](#).

### Para la RB Pi 2/RB Pi 3:

- OpenELEC 6.0.3 for RPi2 / RPi3: [descargar](#).
- Descargaremos e instalaremos también la utilidad **Win 32 Disk Imager** que nos va a permitir grabar de forma sencilla cualquier imagen en la tarjeta microSD:

**Win 32 Disk Imager:** [descargar](#).

*A pesar de su simplicidad, hay que fijarse muy bien en lo que se hace con esta herramienta, porque si nos equivocamos al elegir la unidad de destino para la escritura, podemos borrar por completo una partición o disco de nuestro PC.*

- Por si fuese necesario formatear la microSD, podemos descargar el programa **SDFormatter**. El enlace de descarga que adjunto incorpora todas las versiones de SDFormatter, por lo que una vez descomprimido instalaremos la que se adapte a nuestro caso:

**SDFormatter - Todas las versiones:** [descargar](#).

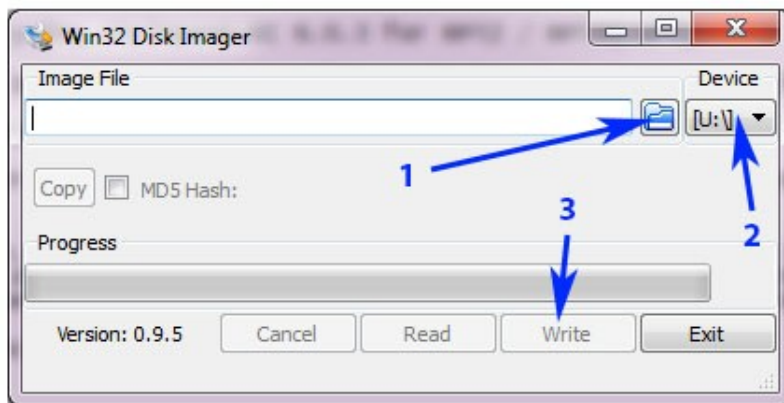
Más info: [aquí](#).

Todas estas utilidades las podemos obtener en una única descarga bajándonos de la web de **Lightberry** el [tutorial en español](#). Una vez descargado el tutorial, lo descomprimimos, y en la carpeta \PROGRAMAS tendremos el Win 32 Disk Imager y restantes utilidades.

Tras descargar la imagen de Openelec correspondiente a nuestro caso, la descomprimimos y obtendremos un fichero de extensión **.img**, que será el que debemos grabar en la microSD.

Insertaremos la microSD en un lector de tarjetas, lo conectaremos al bus USB de nuestro equipo y esperaremos a que éste reconozca la tarjeta y le asigne una letra de unidad (si ya está formateada). Si la tarjeta no tiene formato, procederemos a dárselo con el programa **SDFormatter**. A continuación nos debemos fijar en el nombre de la unidad que nuestro sistema operativo le habrá asignado (por ejemplo, la unidad H:).

Ejecutaremos la utilidad **Win 32 Disk Imager** y en la pantalla principal haremos clic en el botón de carpeta para cargar nuestra imagen de Openelec en formato .img que estará en la carpeta donde la hemos descomprimido, luego pulsaremos sobre el botón de la unidad de destino ('**Device**') -que en nuestro ejemplo sería la H:-, y finalmente en el botón '**Write**' para iniciar la escritura de la imagen de Openelec (o de la distribución elegida) a la microSD.



- 1: Clic en la carpeta para buscar el fichero de imagen de Openelec (.img)
- 2: Clic en Device para elegir la unidad destino
- 3: Clic en el botón Write para iniciar la escritura de la imagen a la microSD

**IMPORTANTE:** comprobad que la unidad de destino elegida es la tarjeta microSD de vuestra Pi. Un error en este punto y puedes borrar por completo una unidad del equipo.

Transcurridos unos minutos, tendremos nuestra microSD ya preparada con la imagen de elegida. Ahora sólo resta meter la microSD en la RB pi, conectar el cable HDMI entre la Pi y la TV y alimentarla para que arranque de forma automática Openelec y nos muestre, tras unos segundos de espera, la pantalla principal del reproductor multimedia Kodi.

**NOTA:** Si no conectamos el cable HDMI entre la Pi y la TV **antes de alimentarla**, será necesario reiniciarla de nuevo, pero esta vez con el cable HDMI conectado. Por defecto, si en el arranque no detecta ningún dispositivo HDMI conectado, la Pi no activará dicha salida de vídeo.

## Instalación manual de Openelec e Hyperion

Si hemos instalado la imagen de Openelec descargada de la web de Lightberry, este paso no será necesario y podemos continuar con el de la configuración de Kodi.

Para los que prefieran usar una versión más actual de Openelec, es posible realizar una instalación completa manualmente. En este caso, tras instalar Openelec (OSMC, etc), como la imagen que habremos usado no vendrá preparada para trabajar inmediatamente con el ambilight, será necesario instalar a continuación manualmente Hyperion. Muy resumidamente el proceso se reduce a los pasos siguientes ([más info](#)):

- Instalación de **hyperiond** e **hyperion-remote** (por defecto se instalarán en '/usr/bin').
- Preparar el fichero de configuración **hyperion daemon** (por defecto será '/etc/hyperion.config.json').
- Añadir **hyperiond** a los servicios del SO (añadir hyperion.conf a '/etc/init').
- Iniciar el servicio de **hyperion** mediante el comando 'initctl start hyperion'.

En vez de hacer todo esto de forma manual, es posible ejecutar un *script* que nos instalará Hyperion más cómodamente. Debemos seguir los pasos siguientes:

- Para Openelec, [ver aquí](#).
- Para OSMC, [ver aquí](#).

El script para instalar Hyperion de forma más automatizada probablemente no funcione en las versiones más recientes de Openelec y de otras distribuciones. En este caso o recurrimos a una instalación manual o tendremos que usar otra versión de Openelec menos reciente en la que sí funcione el script.

Sólo en el caso de la RB Pi 2 -desconozco si a la RB Pi 3 también le pasa lo mismo- y para las versiones de Openelec anteriores a la 5.0.3, debido a un problema de compatibilidad con Hyperion, será necesario editar el fichero **/flash/config.txt** accediendo remotamente a la Pi y añadirle el siguiente código al final del mismo y antes del cierre del fichero:

```
"dtparam=spi=on"
```

## Configurando el reproductor multimedia Kodi (opcional)

A estas alturas deberemos ya tener visible en nuestra TV la pantalla principal de Kodi. Si durante el funcionamiento de Kodi nos aparece un cuadrado pequeño con los colores del arco iris superpuesto en la esquina superior derecha de la pantalla, es una indicación de que tenemos problemas con la fuente de alimentación de nuestra Pi: voltaje demasiado bajo, consecuencia de un consumo excesivo que la fuente no es capaz de suministrar. Probablemente esto nos pase si estamos usando un alimentador independiente para la Pi. Si este es el caso, o procedemos a alimentarla de la misma fuente de los leds (siempre que la hayamos dimensionado con holgura) o tendremos que recurrir a otro alimentador de 5V y 2A de más calidad (los que solemos tener tirados por casa de algún móvil en desuso, y a pesar de que indiquen que suministran 2A, pueden no ser de suficiente calidad).

No es recomendable alimentar la Pi inyectando los 5V directamente al GPIO. En este caso nos estamos saltando todas las protecciones, filtros y el fusible que hay en la placa base. Es mucho mejor y más seguro alimentar la Pi siempre desde el conector microUSB previsto para ello.

En mi caso y con el mando a distancia de la TV, puedo moverme a través de los menús de Kodi. Si no funcionase el control de este modo conviene revisar si nuestra TV soporta [CEC](#), y si además estamos conectando el cable HDMI en una entrada de la TV que lo soporte, ya que puede ser que no todas las entradas HDMI sean compatibles. Incluso podría ser problema del cable HDMI elegido. Si soporta CEC, lo más cómodo es usar el mando de la TV para movernos a través de Kodi (antes llamado XMBC).

En el peor de los casos, no queda otra alternativa que adquirir un teclado y ratón sin hilos que sea compatible con la Raspberry Pi, y cuyo receptor USB deberemos pinchar en uno de los puertos USB de la Pi. Antes de adquirir un ratón y teclado inalámbrico conviene hacer una búsqueda en foros de la RB Pi o en Google para determinar cuáles son compatibles.

Si no te desenvuelves en inglés con soltura, lo primero que deberás hacer es configurar Kodi para que se muestre en español. Para ello debes acceder a **SYSTEM > Settings > Appearance > International > Language**, configurar el idioma en español de España. De esta forma ya veremos todos los textos y ayudas en español.

También puede suceder que la imagen descargada de Openelec de la web de Lightberry arranque automáticamente en polaco. En este caso tendremos que poner Kodi en español antes de proseguir con la configuración básica. Para ello debemos ir al menú **SYSTEM > Ustawienia > Wyglad > Regionalne** y en la parte derecha en el campo **Jezyk**, cambiar **Polish** por **Spanish**.

Lo siguiente que deberemos hacer es configurar el vídeo para que Kodi quede bien adaptado al tamaño de nuestra pantalla en el caso en que no se vean por completo los iconos y textos. Lo haremos desde el menú de **SISTEMA > Ajustes > Sistema > Hardware de vídeo**. En el apartado **Resolución** definiremos la resolución de salida en 1920x1080p si nuestra Tv es Full HD, o la ajustaremos a nuestro caso particular.

Bajamos hasta el menú **Calibración de vídeo...**, donde nos presentará varias pantallas que deberemos configurar una tras otra. La primera es para el ajuste del '**overscan**' en la esquina superior izquierda (0,0). Pulsaremos repetidas veces la tecla de flecha a la derecha del mando a distancia o del teclado hasta que se visualice en la esquina superior izquierda una línea vertical de varios píxeles de ancho. La dejaremos ajustada lo necesario para que se vea en su ancho completo (si es necesario pulsamos la tecla flecha a izquierdas para dejar la marca justo al borde de la pantalla). Seguidamente pulsaremos la tecla de flecha abajo del mando o teclado hasta que aparezca en la misma esquina una línea horizontal y se vea en su altura completa. Pulsamos en el botón OK o de aceptar de nuestro mando/teclado.

Ahora repetiremos el mismo proceso pero para la esquina inferior derecha, moviendo la pantalla primero con la tecla de flecha izquierda y luego con la de tecla arriba, hasta que las marcas de dicha esquina estén perfectamente visibles. Volvemos a pulsar en OK para aceptar y pasar a la siguiente pantalla. Es el momento de ajustar la altura en la que se mostrarán los subtítulos cuando sean necesarios, lo que se indica mediante una línea horizontal centrada en la parte inferior de la pantalla. Con las teclas flecha arriba/abajo ajustaremos la posición de la visualización de los subtítulos.

Aceptamos nuevamente y ahora aparecerá un cuadrado centrado en la pantalla. A ojo o usando una regla por ejemplo ajustaremos con las flechas del mando/teclado el cuadrado mostrado de forma que sea un cuadrado perfecto. Este ajuste minimizará las distorsiones por una posible diferencia de escala entre el eje horizontal y el vertical. Si volvemos a pulsar OK, reiniciaremos de nuevo el proceso de calibración. Pero si todo quedó correctamente configurado, simplemente le damos al botón atrás del mando/teclado y salimos nuevamente al menú de nivel anterior.

Ahora ya conseguiremos ver el 100% de la pantalla de Kodi con los iconos de Ajustes (en la esquina superior izquierda) y del reloj (en la esquina superior derecha), así como los iconos de pantalla anterior y Home en la esquina inferior derecha.

Es muy recomendable que, al menos para la fase inicial de configuración del Hyperion/Kodi/etc, tengamos la Pi conectada a nuestra red local y con acceso a internet. Si no puedes hacerlo con cable -que es lo más recomendable para poder ver contenidos on-line o reproducir contenidos alojados en nuestro PC sin problemas de tirones por falta de ancho de banda- deberás añadir al conjunto un receptor Wifi USB compatible con la Pi (o usar el adaptador wifi integrado de la RB Pi 3), y configurar la conexión para que acceda a tu red wifi.

Llegados a este punto, podemos movernos por los diferentes menús de Kodi y configurar todo a nuestro gusto. Tómatelo con calma porque tiene mucho jugo, y en caso de querer exprimirlo bien, te recomiendo te leas estos enlaces (o que busques uno que te guste más):

[Tutorial sobre Kodi en la Pi](#)  
[Convierte tu RB en un media center](#)

Una vez que ya hayamos configurado Kodi a nuestro gusto y comprobado que accedemos sin problemas a nuestros contenidos multimedia vía red o directamente conectados a uno de los puertos USB de la Pi, podemos meternos a personalizar la configuración del Hyperion.

## Adaptando el fichero de configuración de Hyperion

En la página de descargas de [Lightberry](#) podemos descargar ficheros adaptados a las diversas configuraciones base de los kits que comercializan, y que dependen básicamente del número de leds montados.

Si has instalado leds tipo pixel (o árbol de navidad), hay ficheros de configuración predefinidos para 38 leds (12x7), 40 leds (13x7), 42 leds (13x8 y 14x7), 44 Leds (14x8), 46 leds (15x8), 48 leds (15x9), 52 leds (17x9), 56 leds (18x10) y 64 leds (20x12). Descarga el genérico o el de OpenELEC en función de tus necesidades. El fichero de configuración descargado (**config.hyperion.json**) deberá ser copiado a la ruta adecuada de la Pi:

- Si usas **Raspbmc**, debes copiarlo a **/etc/**
- Si usas **OpenELEC**, debes copiarlo a **/storage/.config/**

Más abajo se explica con detalle cómo copiar el .json a la ruta correcta.

Si tu configuración no se ajusta a ninguna de las predefinidas, entonces debes descargar la utilidad **[Hyperion Configuration Tool](#)** a tu ordenador. Para conocer a fondo qué hace cada parámetro de Hyperion, recomiendo el siguiente enlace de la [página oficial](#) (en inglés).

**Hyperion Configuration Tool** es un archivo java ejecutable (con extensión .jar) que nos permitirá de forma muy visual ir rellenando los valores de todos y cada uno de los parámetros que afectan al funcionamiento del ambilight.

Si tienes soporte para ejecutar aplicaciones de Java -que es lo habitual bajo Windows-, simplemente lo copias a una carpeta cualquiera de tu equipo y lo ejecutas con un doble clic (no hay que descomprimirlo). A continuación pasaremos a configurar cada uno de los apartados según nuestras necesidades.

Si no quiere usar el configurador en el PC o no disponemos de soporte para aplicaciones de Java, la otra opción pasa por usar el **add-on de Hyperion en Kodi** que viene incluido en las últimas imágenes del SO que Lightberry disponibiliza a través de su web.

Localizaremos el add-on en Kodi y desde el mismo es posible configurar toda la instalación con el mando a distancia de la TV. Sin embargo en este tutorial asumiremos que la configuración será hecha desde la utilidad **Hyperion Configuration Tool** y no con el add-on mencionado.

Para quien quiera tener todas las particularidades de las diversas configuraciones en función del tipo de leds, del tipo de conversor de niveles y del tipo de capturadora, deben consultar la siguiente página:



**Para conocer todas las particularidades de las diversas configuraciones en función del tipo de leds, del tipo de conversor de niveles y del tipo de capturadora, deben consultar la siguiente página de la web de Lightberry:**

<http://lightberry.eu/getting-started/>

## Generando nuestro fichero config.hyperion.json

En la utilidad **Hyperion Configuration Tool** existen varias pestañas de configuración, y dentro de ellas varios apartados que deberemos personalizar para ajustar el efecto "ambilight" a nuestro gusto y a nuestra instalación particular. Lo mejor es dejar todos los parámetros en el valor por defecto que muestra la primera vez que lo ejecutamos, y sólo modificar aquellos que realmente necesitamos. Y si con una determinada configuración que has enviado a la Pi te funciona el ambilight, antes de ponerte a hacer cambios para tratar de mejorarlo, **haz una copia de seguridad**

de ese **config.hyperion.json**, porque al tercer cambio que hayas hecho no te vas a acordar lo que has modificado, y si después no te funciona, te desesperas por no recordar cómo lo tenías cuando funcionaba.

### Pestaña Hardware:

En esta primera pestaña deberemos especificar en primer lugar el tipo de leds que tenemos instalados. Ahí por ejemplo, si tenemos una tira con el chip WS2801, deberemos configurar **Device=WS2801**; si es la tira de leds píxel de Lightberry, pues deberemos poner **Device=Lightberry**, y así sucesivamente. Si no sabes exactamente que tipo de leds tienes, consulta la información del vendedor, o en el peor de los casos, vete probando uno a uno los diversos tipos hasta acertar con el tuyo particular.

En **Output** deja el valor por defecto: `/dev/spidev0.0`

En **Baudrate**: 1.000.000 (en el caso de usar leds APA102, deberemos poner en el Baudrate un valor de 500.000).

En **RGB Byte Order** debes indicar el tipo de ordenación "RGB" que realmente tiene la tira de leds. Lo más habitual es que sea RGB -es decir, Rojo, Verde y Azul-, pero también puede ser cualquiera de las otras combinaciones (RBG, BGR,...).

Como siempre, primero consulta la información que te ha proporcionado el vendedor. Ahí debería indicarlo. Si no lo pone claramente, empieza por asignarle el valor RGB, y luego lo confirmas o descartas mediante la aplicación **Hyperion de Android**, por ejemplo. Basta con seleccionar en esa aplicación -una vez conectada remotamente a Hyperion- el color rojo puro en la rueda de colores y ver si los leds se iluminan de rojo. Repetimos con el verde y el azul. Si vemos, por el contrario, que cuando elegimos el color rojo la luz de los leds es azul por ejemplo, los canales Rojo (R) y Azul (B) están invertidos, y el orden correcto es probablemente BGR. En el peor de los casos, vamos cambiando el orden y probando hasta acertar.

Hay que tener en cuenta que cada modificación realizada en el .json para determinar el orden correcto de los colores de la tira RGB requiere que enviemos el .json generado a la ruta correcta de la Pi y la reiniciemos, ya que desde la aplicación de android no es posible alterar el orden del color de los leds.

Más abajo, en la sección **Construction**, tenemos que indicarle el número de leds, la orientación y el punto inicial u origen de nuestra tira. Si viendo la TV desde la parte frontal, nuestra tira va en el sentido de las agujas del reloj, deberemos asignarle a **Direction=clockwise**. En caso contrario, será **counter\_clockwise**.

Si además en las esquinas tenemos posicionados leds, deberemos indicárselo en los apartados '**Led in top corners**' y '**Led in bottom corners**' poniendo dicho parámetro en '**true**', pero aunque no lo configuremos correctamente, el efecto será igual de bueno.

En **Horizontal #** debemos indicar el nº total de leds que hay en el tramo superior. Hyperion asume que en tramo inferior habrá el mismo nº de leds también. En **Vertical #** debemos indicarle el nº de leds que hay en uno de los laterales (y por lógica, el mismo nº de leds instalados deberá haber en

el opuesto).

En **Bottom Gap #** indicaremos el nº de leds de la parte central del tramo inferior que no están instalados. Esta situación hace referencia a que usualmente en la parte inferior y por culpa de la protuberancia del soporte/peana en la carcasa trasera, no instalamos algunos leds. Con este parámetro especificamos de cuántos leds es el hueco sin cubrir, es decir, cuántos faltan. Hyperion asume que los leds que faltan siempre están en la parte central de la tira inferior (donde habitualmente pasan los cables o está el soporte de la TV).

En **1st LED offset** deberemos indicar dónde está situado nuestro primer led de la tira (el led nº 0, el origen de coordenadas). La forma más fácil de determinarlo es ver el esquema simulado de la Tv con los leds que la utilidad nos muestra en la parte derecha de la pantalla. Una vez que ya hemos indicado el nº total de leds de nuestro montaje, localizamos el que marca con el nº 0, y vamos variando en un sentido u otro el valor de **1st LED offset** hasta que veamos que el led 0 aparece posicionado justo donde tenemos nuestro primer led físico en la TV. Tened en cuenta que el esquema de la TV que muestra Hyperion Configuration Tool es visto desde el frontal de la TV. En mi caso, por ejemplo, que tengo 27 leds en horizontal y 15 en vertical, y tengo el primer led de la tira (led 0) situado abajo en el centro, el valor de '1st LED offset' es de -29.

Finalmente, en la sección **Image Process** es donde debemos especificar los parámetros que adaptan el efecto a nuestros gustos. Así por ejemplo, en **Horizontal depth [%]** asignamos qué % del ancho de pantalla se usará para el cálculo de los efectos de ambientlight. Y lo mismo con **Vertical depth [%]**.

Valores muy altos en estos parámetros hacen que contribuyan al valor final de la luz los colores de los píxeles más hacia el interior de la imagen. Esto puede hacer que, en función de la imagen mostrada, dé como resultado un valor demasiado amplio de píxeles que, en promedio, harán que la luz emitida sea muy clara o blanca, perdiéndose el efecto buscado de que el color mostrado sea lo más parecido a los píxeles del borde de la imagen. Podemos imaginarnos que en este caso se 'diluye' el efecto al calcular el promedio de muchos más píxeles de colores dispares en una mayor superficie. Es decir, hay pocos píxeles con colores similares contribuyendo al efecto, y por tanto, este será débil y alejado del color de los píxeles del borde de la imagen. En contrapartida, cuando los píxeles que definan el color no estén en el borde de la imagen sino más adentro, un valor elevado de profundidad ayuda a que los leds no se apaguen.

Por el contrario, un valor bajo hace que los colores se 'capturen' más fielmente porque menos píxeles contribuyen al color que debe ser mostrado, y además están más cerca del borde de la imagen. Y en general en una superficie más reducida capturada en la imagen de la TV, los píxeles tienden a ser del mismo color o muy similar.

Como vemos hay que establecer una solución de compromiso entre la profundidad establecida para el cálculo del color de los leds y la precisión del color deseado.

Podemos empezar con **Horizontal depth [%]=10** y **Vertical depth [%]=8** por ejemplo. Se puede probar con un valor entre 5...20 para el horizontal y 3...15 para el vertical, pero como siempre, lo mejor es ir experimentando hasta adaptarlo a nuestros gustos.

Los parámetros de **Horizontal gap** y **Vertical gap** sirven para indicarle el % de pantalla en ambas direcciones que no debe ser tomado en consideración a efecto de los cálculos -siempre a contar desde el correspondiente borde de la pantalla-. Es decir, ese % de píxeles desde los bordes no entrarán a formar parte de los que determinan el valor del color calculado. En algunas circunstancias podría ayudar a limitar el efecto de los bordes negros si el correspondiente parámetro de detección explicado más abajo no funcionase del todo bien.

En **Overlap** especificaremos qué % de solapamiento tendrá cada área asignada a cada led con respecto a sus vecinas a efectos del cálculo del color. Por ejemplo, si aquí ponemos un valor del 100% quiere decir que el área del led N se solapa al 50% con la de su vecino N-1 (led anterior) y al 50% restante con su vecino N+1 (led posterior). Como es lógico, valores muy altos hacen que se mezclen demasiado los colores de los píxeles y que áreas contiguas tiendan a armonizar el color calculado, dando como resultado un color muy similar y haciendo que las transiciones entre áreas contiguas sean suaves.

Por último, debemos indicar cómo tratar el 'problema' de las bandas negras en los formatos panorámicos. Cuando estamos visualizando contenidos en formato panorámico, suelen salir en la parte superior e inferior de la pantalla unas bandas negras. Si no queremos que por culpa de las mismas los efectos de ambient light se pierdan -recordad que píxeles en negro o muy oscuros implican leds apagados-, es necesario activar el detector de bordes negros poniendo **Blackborder Detector en ON**. En el siguiente parámetro, **Blackborder Threshold [%]**, especificaremos qué % de umbral será tenido en cuenta para considerar el borde negro.

Un valor adecuado de partida es el del 1%. Si Hyperion detecta píxeles más oscuros que el valor umbral configurado, los interpretará como bandas negras y no los tendrá en cuenta para el cálculo del color de los leds. Pero si los píxeles oscuros no superan el valor umbral configurado, serán interpretados como lo que son: píxeles oscuros, pero no negros, y como tales, entrarán a formar parte del cálculo del color del correspondiente led contribuyendo en la práctica a reducir (oscurecer) la intensidad del efecto en ese lateral.

El detector de bordes negros es quizás el parámetro que más quebraderos de cabeza nos dará. No siempre funciona bien, unas veces porque no detecta las bandas negras y otras porque tarda mucho en detectarlas.

Además al proceso de detección de las bandas negras contribuyen varios factores que complican aún más el problema. En general las bandas negras en las señales de vídeo no forman parte de la propia trama de vídeo, es decir, son añadidas en tiempo real por el reproductor multimedia o la TV para 'rellenar' de negro los píxeles de la pantalla sin señal. Pero otras veces, las menos probablemente, sí que serán parte de la señal de vídeo. Y en estas circunstancias si el color de los píxeles de las bandas negras no es realmente negro puro -aunque a nuestros ojos sí lo parezca-, el detector de bordes negros puede fallar. Asumirá que esos píxeles, siendo parte integrante de la imagen y a la vez tan oscuros -pero no negros-, deben apagar los leds. Por eso habrá que ir incrementando con el valor de **Blackborder Threshold [%]** hasta encontrar el punto de equilibrio que nos satisfaga.

Una solución de compromiso para minimizar este problema es aumentar la profundidad de captura de las bandas superior e inferior, es decir, subir el valor de **Horizontal depth [%]**. De esta forma en el cálculo del color de los leds intervendrán más píxeles no tan oscuros situados más al interior de la imagen. Y como el promedio del color calculado ya no será tan oscuro, los leds superiores e inferiores no se apagarán completamente (aunque probablemente tampoco brillen al máximo). Pero como efecto colateral, cuando la señal de vídeo no tenga bandas negras, estaremos abarcando un área mucho mayor de píxeles alejados del borde de la imagen para determinar precisamente el color promedio que tienen los del borde. Como resultado, el valor medio del color calculado tenderá a alejarse del valor real de los píxeles que están más al borde de la imagen, es decir, perderemos precisión en el color mostrado.

### **Pestaña Process:**

Los valores por defecto en esta pestaña son los siguientes:

#### **- Frame Grabber -**

Enabled: Sí

Width: 64

Height: 64

Interval (ms): 100

#### **- Smoothing -**

Enabled: Sí

Type: Linear smoothing

Time (ms): 200

Update Freq. (Hz): 20

Si activamos este parámetro, las transiciones de colores de los leds serán más suaves. Los restantes parámetros determinan el algoritmo de suavizado, el tiempo para el efecto y la frecuencia de la transición.

#### **- Colors -**

default

#### **- Transform -**

Indices: \*

RED Thres.: 0

GREEN Thres.: 0

BLUE Thres.: 0

RED Gamma: 1

GREEN Gamma: 1

BLUE Gamma: 1

RED Blacklvl: 0

GREEN Blacklvl: 0

BLUE Blacklvl: 0

RED Whitelvl: 1

GREEN Whitelvl: 1

BLUE Whitelvl: 1

HSV Saturation gain: 1.0

HSV Value Gain: 1.0

### **Pestaña external:**

En esta pestaña dejaremos marcado el XBMC Checker si queremos que durante el funcionamiento de Kodi (antes llamado XBMC) trabaje el ambilight. Cada uno de los apartados podemos configurarlo para que se muestre el efecto o no en los diferentes modos de trabajo de Kodi. Por ejemplo, si no queremos que en el menú de Kodi los leds funcionen, pues ponemos el parámetro **XBMC Menu** en Off, y lo mismo con los restantes modos.

En cuanto a la IP del servidor, deberá ser **127.0.0.1**.

A continuación debemos activar el **Json server** y dejar el número de puerto configurado en el 19444. Si modificamos el número del puerto aquí, tenemos que cambiarlo igualmente en la configuración de la app [Hyperion Free de Android](#) (si la usamos), o no conseguiremos conectarnos remotamente.

Los otros dos servidores (**Proto server** y **Boblight server**) podemos dejarlos en sus valores por defecto... Personalmente desactivo el de Boblight porque no lo utilizo, así que cuantos menos recursos se reserven para tareas inútiles, mejor para la Pi.

Finalmente en el apartado **Effect Engine** deberemos dejar como ruta para los efectos la siguiente:

```
/storage/hyperion/effects
```

Y si queremos que cada vez que arranque Hyperion se muestre la animación del arcoiris o cualquier otra de las que soporta, marcaremos la casilla **Bootsequence**, pondremos el efecto deseado en el campo **Type** (por defecto, es *Rainbow swirl fast*) y la cantidad de milisegundos que deberá verse (3000, o 3 segundos por ejemplo).

Finalmente ya con todo configurado en la utilidad **Hyperion Configuration Tool**, es hora de pulsar el botón **Create Hyperion Configuration**. Se habrá generado un fichero llamado **hyperion.config.json** y que deberemos enviar a la Pi.

## Cómo copiar el fichero de configuración a la Pi

El método más adecuado por la rapidez y comodidad desde mi punto de vista requiere lo siguiente:

- Tener la Pi conectada en red y conocer su IP (ver nota)
- Instalar **Filezilla** y **Putty** en nuestro PC.

NOTA: para saber qué IP tiene la Pi, con Kodi en ejecución en la TV accedemos al menú **Sistema > Info del sistema**. Ahí aparecerá la actual IP. Dependiendo de la configuración del *router* de nuestra LAN, podría suceder que tras cada reinicio de la Pi, el *router* le asigne una IP diferente. Si así fuese, lo mejor por comodidad sería modificar la configuración para que el *router* le asigne siempre la misma IP a la Pi en nuestra red local.

Para comunicarnos con la Pi y enviarle el fichero **.json**, los dos programas que vamos a necesitar son los siguientes:

- Un cliente de FTP como **Filezilla**, que podemos descargar de [aquí](#). Como alternativa al FileZilla podemos usar [WinSCP](#).
- Un emulador de terminal con soporte SSH como Putty, que descargaremos de [aquí](#) e instalaremos en nuestro PC.

Con la Pi conectada a la red local y encendida, ejecutamos **Putty** y configuramos una nueva conexión usando los datos siguientes:

**Hostname:** ponemos la IP de nuestra Pi en la red local.

**Port:** 22

**Connection type:** SSH

**Saved sessions:** este campo es opcional. Sirve para ponerle un nombre a la conexión y memorizarla. Podemos ponerle *Hyperion Server* por ejemplo, y después pulsamos en el botón **Save** para guardar la conexión así configurada.

Ahora basta hacer doble clic sobre el nombre de la conexión que le hemos asignado o directamente clic en el botón **Open** para abrir la ventana de comandos con la conexión SSH con nuestra Pi.

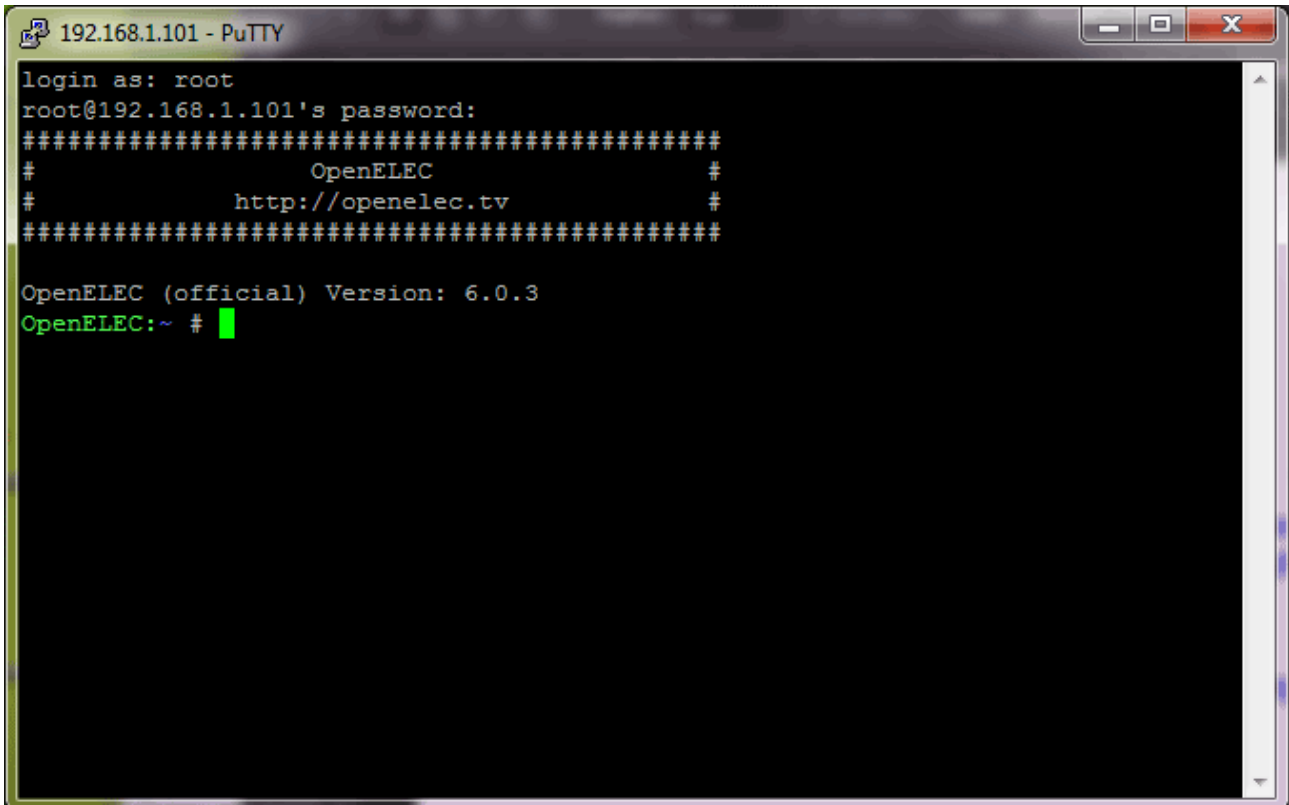
En la ventana del terminal nos va a solicitar los datos para hacer 'login'. Debemos entrar los

siguientes datos de usuario/contraseña para conectarnos (pulsando Intro tras cada dato):

**Login as: root**

**root@192.168.1.101's password: [openelec](#)**

La IP mostrada ahí cuando solicite la contraseña no tiene porqué ser ésta... variará dependiendo de la que le haya sido asignada a vuestra Pi. Si todo va bien, nos acabamos de conectar y veremos algo como esto:



```
192.168.1.101 - PuTTY
login as: root
root@192.168.1.101's password:
#####
#                               #
#           OpenELEC             #
#           http://openelec.tv   #
#####

OpenELEC (official) Version: 6.0.3
OpenELEC:~ # █
```

### **Pantalla de la conexión SSH (Putty) con la Pi**

En el caso de estar bajo Raspbmc, los datos de acceso serán user: root y pass: raspbmc.

Una vez conectados remotamente a la Pi, detenemos Hyperion con este comando:

```
killall hyperiond
```

Ahora tenemos que copiar el fichero .json que hemos preparado a la ruta correcta en la Pi. Podemos hacerlo desde la terminal de Putty mediante una serie de comandos, pero es más fácil hacerlo a través del cliente de FTP Filezilla que antes hemos descargado e instalado.

Abrimos **Filezilla** y configuramos una nueva conexión hacia la Pi. Para ello hacemos lo siguiente: menú **Archivo > Gestor de sitios** y pulsamos sobre **Nuevo sitio**. Como nombre le ponemos el que nos dé la gana (*Raspberry Pi - Hyperion*, por ejemplo). Los datos para configurar la conexión son los siguientes:

**Servidor:** ponemos la IP de nuestra Pi (por ejemplo, en mi caso es 192.168.1.101)

**Puerto:** 22

**Protocolo:** SFTP - SSH File Transfer Protocol

**Modo de acceso:** Normal

**Usuario:** **root**

**Contraseña:** **openelec** (bajo Raspbmc es *raspbmc*)

En el resto de pestañas no necesitamos tocar nada. Pulsamos sobre el botón **Conectar** y se iniciará la conexión (y además se guardará para disponer de ella otra vez nuevamente cuando sea preciso).

Una vez conectados, veremos que en la parte derecha inferior donde pone "**Sitio remoto**" nos muestra la estructura de directorios de la Pi. Nos posicionaremos en la carpeta destino adecuada (bajo Raspbmc es */etc/*, y **bajo OpenELEC es */storage/.config/***) y copiaremos -subiremos- nuestro **config.hyperion.json** personalizado que antes habremos localizado en la parte izquierda de la pantalla de trabajo del Filezilla y que se identifica por "**Sitio local**", o directamente mediante arrastrar y soltar, lo soltamos sobre la carpeta destino correcta.

En la siguiente imagen puede verse la estructura de carpetas de la Pi colgando de la raíz *\storage*. Nuestro fichero **hyperion.config.json** deberemos copiarlo a la carpeta *\storage\.config* reemplazando al que ahí ya exista. Una vez hecho esto deberemos reiniciar Hyperion para que asuma la nueva configuración, y podemos hacerlo de dos formas:

- por las bravas, apagando la alimentación de la Pi y volviéndola a alimentar (la menos recomendable)
- o desde la sesión de Putty aún abierta, reiniciamos Hyperion con este comando:

```
/storage/hyperion/bin/hyperiond.sh /storage/.config/hyperion.config.json </dev/null >/dev/null 2>&1 &
```

La primera alternativa no es muy recomendable porque podemos cargarnos la estructura de ficheros de la microSD si en ese momento estuviésemos accediendo a ella (y por desgracia la Pi tiene cierta 'predisposición' para cargarse las tarjetas microSD). Además si lo hacemos así, se perderán las conexiones abiertas de Putty y del Filezilla, obligándonos a reconectarnos tras cada reinicio hecho de esta forma. Por ello, y salvo que no haya otra alternativa, es mejor la segunda opción.

Si el fichero de configuración es correcto, Hyperion se habrá reiniciado y habrá mostrado la animación inicial del efecto arcoiris. Ya estamos preparados para probarlo con cualquier vídeo. Para ello ponemos a reproducir un vídeo o película desde Kodi y los leds deberían encenderse en

consonancia con lo mostrado en la pantalla de la TV. Si todo va bien, solo falta adaptar los efectos, colores, etc a tus gustos, pero lo básico ya está hecho. Ahora solo queda entretenerse muchas horas jugando con los parámetros hasta dar con la configuración que más te agrade.

El proceso para ello es sencillo: modificamos un parámetro, generamos el config.hyperion.json correspondiente, detenemos Hyperion, copiamos el fichero de configuración a la ruta correcta de la Pi y reiniciamos Hyperion (o la Pi). Tras el reinicio comprobamos cómo afectan los cambios. Y vuelta a repetir el proceso. Y así hasta que nos quedemos satisfechos.

Como todo este proceso lleva su tiempo y requiere cierto método, se recomienda hacerlo cuando no hay nadie pendiente de que acabes para poder ver la TV... lo mejor, al menos en mi caso, es por la noche.

## Usando la aplicación de configuración de Hyperion bajo Android (o iOS)

A mi modo de ver, la opción más cómoda para modificar valores de parámetros y ver cómo afectan a los efectos de iluminación es descargar e instalar en nuestro móvil/tableta Android la aplicación oficial de Hyperion (también existe una versión para iOS, [más info aquí](#)).

Para Android hay dos versiones de Hyperion: la gratuita, llamada [Hyperion Free](#) y que incorpora publicidad, y la 'completa', de pago pero libre de anuncios llamada simplemente [Hyperion](#).

Para personalizar la configuración, con la gratuita es más que suficiente, pero si quieres controlarlo todo desde el móvil/tableta sin publicidad, te recomiendo la versión de pago que, por lo que cuesta (0.99€), vale la pena. Y además es una forma simple y barata de agradecer a los desarrolladores su trabajo, gracias al cual podemos disfrutar de un excelente efecto 'ambilight' casero a un coste contenido.

Con esta aplicación en el móvil, nos evitamos el tener que andar haciendo los tediosos cambios en el Hyperion Configuration Tool del PC, generar el fichero config.hyperion.json, detener Hyperion, copiar el fichero anterior mediante Filezilla y reiniciar Hyperion de nuevo para comprobar cómo afectan al efecto generado.

Cuando queremos probar como afectan las alteraciones al efecto generado, llega a hacerse muy tedioso todo este proceso de copia > pega > genera > transfiere > reinicia. Así que, gracias a la aplicación Hyperion de Android, podemos toquetear todo lo que queramos y ver, al instante, cómo influyen los cambios.

Una vez que hayamos encontrado la configuración ideal, simplemente copiamos los parámetros modificados a mano al programa **Hyperion Configuration Tool** y generamos el fichero de configuración definitivo. Luego lo transferimos a su ruta correcta en la Pi. Como es lógico, esto tenemos que hacerlo siempre que pretendamos conservar determinada configuración personalizada ya que todos los cambios que se hacen mediante la aplicación de Android, son

temporales, esto es, se pierden al apagar o reiniciar la Pi. De ahí la necesidad de generar el fichero de configuración definitivo en el PC con los datos personalizados y copiarlo después a la Pi. No está nunca de más hacer copias de seguridad de los diversos ficheros de configuración y documentarlos para saber más adelante qué hace cada uno de ellos y poder, si fuese necesario, recuperarlo rápidamente.

Si cuando ejecutamos el Hyperion para Android en el móvil/tableta y tras configurarle la IP correcta de la Pi en el menú **Settings** muestra siempre el mensaje de **"Unable to connect to IP:19444"**, quiere decir con casi total seguridad que no está el servidor de Hyperion funcionando. Una posibilidad es que no se haya iniciado de forma automática -como debería haberlo hecho- al arrancar la Pi. En este caso tendrás que iniciarlo manualmente con el comando mostrado unas páginas más atrás y analizar porqué no lo ha hecho. Ten presente que Hyperion trabaja en segundo plano y no tiene una *interface* de usuario visible ni ningún indicador de que está en ejecución. Sólo cuando está funcionando y tienes la IP del mismo bien configurada en la aplicación de Android (el puerto por defecto en el que se comunica es el 19444 y no debes cambiarlo a no ser que lo modifiques también en el .json), verás que tienes disponibles los diversos parámetros de configuración así como los diferentes efectos preconfigurados.

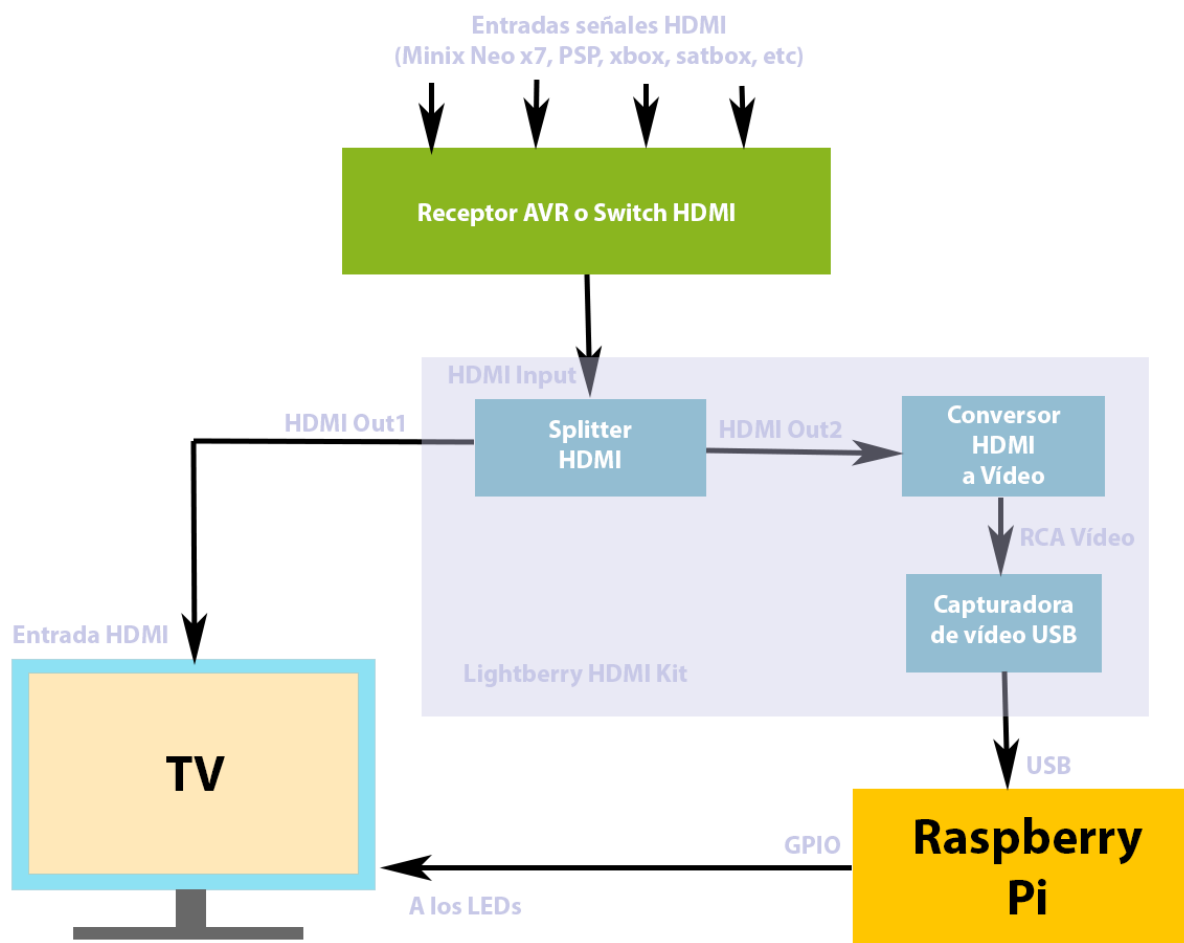
Tampoco conseguiremos conectarnos desde el móvil si en la pestaña **"External"** de Hyperion Configuration Tool no hemos activado el **Json server** en el puerto TCP 19444. Si este es el caso, activamos el **Json server**, generamos el fichero .json personalizado nuevamente y, mediante Filezilla, lo copiamos a la ruta adecuada. Tras reiniciar Hyperion (o la Pi) ya debería funcionar en segundo plano el servidor de Hyperion. Ahora sí que se debería conectar remotamente el móvil y no mostrará el mensaje de error de conexión.

## Preparando el montaje para que acepte cualquier fuente de video HDMI

Una vez que tenemos funcionando el sistema de ambilight para los contenidos que son reproducidos desde la Pi, pasaremos a prepararlo para que haga lo mismo con otras señales externas, como pueden ser un reproductor de DVD/BluRay, una consola (PSP/Xbox), un media player externo, o cualquier otra fuente de vídeo que genere la señal mediante una salida HDMI.

En este caso, la RB Pi no reproducirá los contenidos, y por tanto, no podrá saber cómo calcular los colores. Y ello es así porque la RB Pi por sí misma no puede capturar video externo, ni hay forma de conectarle una señal HDMI de entrada. Recordemos que el conector HDMI de la RB pi es para enviar la señal de vídeo generada internamente hacia un monitor externo, es decir, es de salida.

La siguiente imagen resume a modo de esquema las conexiones necesarias para que la Pi genere el ambilight a partir de fuentes de vídeo externas:



### Extendiendo el ambilight a señales HDMI externas

Como vemos será necesario adquirir a mayores **una capturadora de vídeo USB, un *splitter* o duplicador de HDMI y un conversor de HDMI a vídeo compuesto.**

Tanto el *splitter* como el conversor de HDMI a vídeo deben ser compatibles con señales Full HD de 1080p. El *splitter* suele ser HDMI v1.3, y el conversor v1.3 o incluso v1.4. Ojo con los conversores porque se venden de tres 'calidades': los que sólo soportan 480p, los que llegan hasta 720p (HD) y los que de verdad necesitamos, que son los que soportan 1080p (Full HD).

La señal HDMI que sale del dispositivo externo (consola, BluRay, etc), la conectaremos a la entrada del *splitter* (duplicador de HDMI). Una de las dos salidas del mismo la llevaremos a una entrada HDMI libre de nuestra TV. Y la otra salida la llevamos a la entrada del conversor de HDMI a vídeo compuesto. De los conectores RCA de salida del conversor de vídeo, sólo necesitamos el de color amarillo (la señal de vídeo compuesto). Lo deberemos conectar a la entrada RCA del mismo color (amarillo) de la capturadora. Y la capturadora la tendremos que conectar a bus USB de la Pi preferiblemente a través del hub USB autoalimentado.

En el conversor de HDMI a Vídeo hay un pequeño interruptor para seleccionar el formato de la

señal. Salvo que estemos usando algún dispositivo con salida en formato NTSC, deberemos posicionar el interruptor en PAL.

Tanto el *splitter* como el conversor de vídeo necesitan alimentación de 5V. El consumo de los dos dispositivos es pequeño, y en muchos casos es suficiente con la alimentación de +5V que reciben a través de los cables HDMI. Si no fuese el caso, tendríamos que conectarle por ejemplo cables USB desde el hub autoalimentado para que reciban la tensión de 5V que requieren.

Por tanto, para que podamos generar el ambilight a partir de señales HDMI externas hay que capturar esa señal video HDMI de alguna forma y enviar las capturas a la RB Pi para que las analice como hace cuando es ella la que genera la señal HDMI. Por eso primero convertimos la señal HDMI a vídeo compuesto y después la enviamos a la entrada de la capturadora.

Finalmente la capturadora realizará capturas de la señal de vídeo compuesto a suficiente velocidad para que se pueda generar el efecto ambilight a partir de los cálculos sobre los colores de los píxeles de las capturas.

En el mercado existen multitud de capturadoras USB, siendo en general conocidas bajo la marca o denominación de **Easycap**. A día de hoy, sólo dos tipos de chipsets son compatibles con el ambilight. Por eso es importante seleccionar una capturadora de este tipo que internamente use uno de los siguientes chipsets:

- **STK1160** (el más antiguo)
- **USBTV007** (el más reciente)

Las últimas imágenes de OpenELEC son compatibles con ambos chipsets, pero deberemos certificarnos que efectivamente la imagen que tenemos en la Pi soporta el chipset de la capturadora conectada. **Recomiendo adquirir el USBTV007 (también reconocido por Fushicai) porque funciona muchísimo mejor que el obsoleto STK1160.**

En este punto es importante hacer un inciso: **el consumo de corriente de este tipo de capturadoras podría superar la capacidad del puerto USB de la RB Pi** (en especial de los primeros modelos de RB Pi).

Acertar en la compra de la capturadora es el quid del éxito. Hay multitud de variantes, todas ellas conocidas genéricamente por EasyCap, pero no todas nos van a servir. La opción de ir por lo seguro es adquirir la capturadora en la tienda de Lightberry. Acertarás al 100% porque ellos ya han seleccionado las que efectivamente valen para el ambilight. Actualmente solo comercializan las USBTV007.

Pero analizando con detalle la información de algunos vendedores, es posible comprar en eBay, Aliexpress y sitios similares capturadoras que, en principio, funcionarán, y a precios generalmente más asequibles.

No podemos confiar al 100% en lo que el vendedor publicita. Muchas veces el vendedor no sabe/no entiende de lo que vende, y otras muchas te dirá que sí, que tiene exactamente lo que

bucas... y luego te llevas la sorpresa de que no vale. Suelo recomendar la que yo compré porque sí es una USBTV007, y el vendedor parece de fiar por la información que aporta en la web. Pero aclaro que no tengo ninguna relación con él, y que confiar o no en él es cuestión de fe. Por eso que cada cual decida dónde comprarla. Si queréis el enlace del producto, en el foro lo encontraréis ([aquí](#), por ejemplo).

Si cuando conectamos la capturadora directamente a la Pi vemos que en la pantalla de Putty y durante el proceso de captura manual aparece el mensaje "**Frame to small**", eso quiere decir que Hyperion ha recibido pocos datos de la capturadora o que la capturadora no recibe suficiente energía para funcionar correctamente.

Si se tratase del primer caso, podría deberse a que hayamos saturado el bus USB. La causa es inherente al diseño del hardware de la RB pi, y ahí no tenemos mucho que hacer.

Lo que sucede es que el bus USB (2.0) de la RB Pi es compartido por todos los dispositivos USB conectados y por la conexión de red (LAN). Por eso, si tenemos varios dispositivos USB pinchados y al mismo tiempo estamos usando también la conexión de red, podríamos haber ocupado todo el ancho de banda disponible. Y en estas circunstancias puede que no quede suficiente ancho de banda en el bus para que la capturadora haga su trabajo.

Posibles soluciones:

- Desconectar todos los dispositivos USB que no estemos usando.
- Viendo los conectores USB de la PI de frente, no situar ningún dispositivo USB en la misma columna en la que conectemos la capturadora.
- No usar la conexión de red cableada.
- No usar la conexión wifi (si le pinchamos el adaptador, consumiremos ancho de banda en el bus USB).

Pero también podría deberse a que la capturadora no recibe suficiente energía a través del bus USB. Y tenemos entonces dos posibilidades:

- Conectamos la capturadora a la Pi pero la alimentamos mediante dos cables adicionales que conectaremos a la fuente externa de +5V.
- Intercalamos entre la Pi y la capturadora un hub USB autoalimentado capaz de entregar la suficiente potencia.

En el primer caso tenemos que asegurarnos que sólo alimentamos la capturadora desde la fuente de +5V. Esto implica abrir la capturadora y realizarle algunas modificaciones en la pista de alimentación del bus USB: es necesario cortar la línea de 5V que sale por el conector USB de la capturadora. Adicionalmente tendríamos que soldarle un par de cables que llevaríamos a la fuente de 5V para alimentar el resto del circuito de la misma. Esta modificación puede no estar al alcance de todos los usuarios, y además anulará la garantía de la capturadora. Por eso lo más fácil y seguro es intercalar un hub USB autoalimentado entre la Pi y la capturadora.

En este punto es importante comprobar si el HUB no deriva también la alimentación externa que

recibe hacia la Pi 'aguas arriba' a través del conector USB. Se comprueba simplemente conectando el hub USB a la Pi, no alimentando a ésta y sí alimentando sólo al hub USB. Si en estas circunstancias se encienden algún led de la placa base de la Pi, mala suerte... el hub USB está inyectando corriente desde el USB hacia la Pi, lo que no es una situación deseable. En mi caso he resuelto el problema cortando el cable de +5V que desde el hub USB iba hacia la Pi. De esta forma, la tensión que alimenta al hub ya no se inyecta hacia la Pi a través del puerto USB.

Una vez conectada y alimentada la capturadora tenemos que comprobar que la Pi efectivamente la detecta, es decir, que es una de las compatibles. Para ello abrimos una sesión remota mediante Putty a la IP de la Pi, en el puerto 22 y con los datos **user: root** y **pass: openelec**. Luego ejecutamos los siguientes comandos (pulsando Intro después de cada uno):

```
lsusb  
ls /dev/video*
```

El primer comando (**lsusb**) nos devolverá un listado de todos los dispositivos USB que la RB Pi detecta conectados al bus USB. En el caso de una capturadora con chipset STK1160, devolverá un mensaje similar al siguiente:

**"Bus 001 Device 005: ID 05e1:0408 Syntek Semiconductor Co., Ltd STK1160 Video Capture Device"**

Si nuestra capturadora es una USBTV007, el mensaje será parecido a éste:

**"Bus 001 Device 006: ID 1b71:3002 Fushicai USBTV007 Video Grabber [EasyCAP]"**

Estos mensajes confirman que la Pi detecta una capturadora compatible.

El segundo comando (**ls /dev/video\***) nos devolverá la ruta de montaje a la capturadora: */dev/video0* en la mayoría de los casos. Si os sale otra ruta diferente debéis tomar nota de cuál es para modificar el código del *grabber* en consecuencia.

Seguidamente deberemos editar el fichero de configuración de Hyperion y añadir la sección "**grabber**" referente a la captura de vídeo, para lo que deberemos hacer lo siguiente:

- Manualmente -o volviendo a generar el fichero con el Hyperion Conf. Tool-, desactivamos la sección "**framegrabber**" (ver nota).
- Añadimos el código del *grabber* recomendado según el tipo de capturadora que tengamos instalada (si no sabemos cuál es, debemos identificarla usando el comando **lsusb** antes comentado):

## Si tu capturadora es una STK1160 usa este código:

```
/// SECCIÓN GRABBER - SE AÑADE SI QUEREMOS GENERAR EL EFECTO DE AMBILIGHT DESDE
/// CUALQUIER ENTRADA DE VÍDEO HDMI. SE NECESITA UNA CAPTURADORA DE VÍDEO USB
/// COMPATIBLE CONECTADA A LA RASPBERRY PI Y ALGÚN HARDWARE MÁS ADICIONAL
/// (CONVERSION DE HDMI2AV Y SPLITTER).

    /// Configuration for the embedded V4L2 grabber
    /// * device : V4L2 Device to use [default="/dev/video0"]
    /// * input : V4L2 input to use [default=0]
    /// * standard : Video standard (no-change/PAL/NTSC) [default="no-change"]
    /// * width : V4L2 width to set [default=-1]
    /// * height : V4L2 height to set [default=-1]
    /// * frameDecimation : Frame decimation factor [default=2]
    /// * sizeDecimation : Size decimation factor [default=8]
    /// * priority : Hyperion priority channel [default=800]
    /// * mode : 3D mode to use 2D/3DSBS/3DTAB (note: no autodetection) [default="2D"]
    /// * cropLeft : Cropping from the left [default=0]
    /// * cropRight : Cropping from the right [default=0]
    /// * cropTop : Cropping from the top [default=0]
    /// * cropBottom : Cropping from the bottom [default=0]
    /// * redSignalThreshold : Signal threshold for the red channel between 0.0 and 1.0 [default=0.0]
    /// * greenSignalThreshold : Signal threshold for the green channel between 0.0 and 1.0 [default=0.0]
    /// * blueSignalThreshold : Signal threshold for the blue channel between 0.0 and 1.0 [default=0.0]
"grabber-v4l2" :
{
    "device" : "/dev/video0",
    "input" : 0,
    "standard" : "PAL",
    "width" : 240,
    "height" : 192,
    "frameDecimation" : 2,
    "sizeDecimation" : 2,
    "priority" : 800,
    "mode" : "2D",
    "cropLeft" : 0,
    "cropRight" : 0,
    "cropTop" : 0,
    "cropBottom" : 0,
    "redSignalThreshold" : 0.2,
    "greenSignalThreshold" : 0.2,
    "blueSignalThreshold" : 0.2
},
```

## Si tu capturadora es una USBTV007 usa este otro código:

```
/// SECCIÓN GRABBER - SE AÑADE SI QUEREMOS GENERAR EL EFECTO DE AMBILIGHT DESDE
/// CUALQUIER ENTRADA DE VÍDEO HDMI. SE NECESITA UNA CAPTURADORA DE VÍDEO USB
/// COMPATIBLE CONECTADA A LA RASPBERRY PI Y ALGÚN HARDWARE MÁS ADICIONAL
/// (CONVERSOR DE HDMI2AV Y SPLITTER).

/// Configuration for the embedded V4L2 grabber
/// * device : V4L2 Device to use [default="/dev/video0"]
/// * input : V4L2 input to use [default=0]
/// * standard : Video standard (no-change/PAL/NTSC) [default="no-change"]
/// * width : V4L2 width to set [default=-1]
/// * height : V4L2 height to set [default=-1]
/// * frameDecimation : Frame decimation factor [default=2]
/// * sizeDecimation : Size decimation factor [default=8]
/// * priority : Hyperion priority channel [default=800]
/// * mode : 3D mode to use 2D/3DSBS/3DTAB (note: no autodetection) [default="2D"]
/// * cropLeft : Cropping from the left [default=0]
/// * cropRight : Cropping from the right [default=0]
/// * cropTop : Cropping from the top [default=0]
/// * cropBottom : Cropping from the bottom [default=0]
/// * redSignalThreshold : Signal threshold for the red channel between 0.0 and 1.0 [default=0.0]
/// * greenSignalThreshold : Signal threshold for the green channel between 0.0 and 1.0 [default=0.0]
/// * blueSignalThreshold : Signal threshold for the blue channel between 0.0 and 1.0 [default=0.0]
"grabber-v4l2" :
{
  "device" : "/dev/video0",
  "input" : 0,
  "standard" : "PAL",
  "width" : 720,
  "height" : 576,
  "frameDecimation" : 2,
  "sizeDecimation" : 8,
  "priority" : 800,
  "mode" : "2D",
  "cropLeft" : 5,
  "cropRight" : 5,
  "cropTop" : 5,
  "cropBottom" : 5,
  "redSignalThreshold" : 0.2,
  "greenSignalThreshold" : 0.2,
  "blueSignalThreshold" : 0.2
},
```



**No olvidarse de la coma ',' que hay al final del trozo de código del *grabber*. Sin ella el fichero de configuración modificado no será interpretado correctamente por Hyperion.**

(\*): En la página de información detallada sobre el funcionamiento del Grabber v4l2 se indica que es necesario **desactivar toda referencia al "framegrabber" que pueda haber antes del código del grabber v4l2 que hemos añadido**. En caso contrario, se producirá un conflicto y los leds parpadearán ([Más info](#)). En mi experiencia no he observado parpadeos aún dejando activada la sección 'framegrabber', por lo que primero se puede dejar activada y si después da problemas, la desactivamos (comentamos sus líneas de código).

Guardamos las alteraciones y enviamos de nuevo el fichero de configuración a la Pi, y la reiniciamos.

## Capturas manuales de la imagen de vídeo externa. Ajuste de los recortes

El proceso completo así como el listado de modificadores de captura está expuesto en el siguiente enlace:

<http://support.lightberry.eu/#how-to-take-screenshot-from-grabber-using-hyperion-v4l2>

Ahora vamos a comprobar que efectivamente capturamos lo que está en pantalla (la señal de vídeo que pasa por el duplicador de HDMI y el convertor de HDMI a vídeo compuesto). La idea es determinar si en la captura de la señal externa hay bandas negras introducidas por la propia capturadora para recortarlas y que no afecten al cálculo del efecto ambilight. Para ello, desde una sesión en Putty hacemos lo siguiente:

Primero detenemos Hyperion para poder capturar manualmente (en caso contrario dará “**error de dispositivo en uso**”):

```
killall hyperiond
```

Seguidamente ejecutamos una captura manualmente:

```
LD_LIBRARY_PATH=/storage/hyperion/bin /storage/hyperion/bin/hyperion-v4l2 --screenshot
```

Si todo ha ido como debe, en la ruta desde la que hemos ejecutado el comando (este punto es importante), deberá existir un fichero de imagen llamado '**screenshot.png**'. Si desde Putty no hemos cambiado de directorio tras entrar en la Pi, la captura estará siempre en la raíz de OpenELEC, es decir, en **\storage**. Cada vez que hagamos una captura se reescribirá el citado fichero, por lo que sólo habrá uno aunque hagamos muchas capturas.

Ahora transferiremos la captura **screenshot.png** al PC mediante el Filezilla, lo abrimos con cualquier visor de imágenes y vemos si ha capturado correctamente la señal de entrada. Si en la imagen capturada aparecen pequeñas bandas negras (es normal que así sea con todas las

capturadoras), sólo nos resta hacer repetidas capturas manuales mediante el comando mostrado más abajo para determinar cuánto hay que cortar en cada lado a efectos de eliminar las bandas negras.

Una vez que mediante el método de prueba y error hayamos determinado los valores a usar en “**cropLeft, cropRight, cropTop, cropBottom**”, los deberemos copiar a la sección del **grabber v4I2** que antes hemos añadido reemplazando los valores ahí existentes, y volviendo a enviar el fichero de configuración de vuelta a la Pi. Reiniciamos Hyperion (o la Pi) otra vez y ya deberemos conseguir capturar señales de vídeo externas sin que interfieran las posibles bandas negras de los laterales.

El comando para realizar capturas manuales pero con valores experimentales de “**cropLeft, cropRight, cropTop, cropBottom**”, es el siguiente:

```
LD_LIBRARY_PATH=/storage/hyperion/bin /storage/hyperion/bin/hyperion-v4I2 /storage/.config/hyperion.config.json --crop-top 0 --crop-left 0 --crop-bottom 0 --crop-right 0 --size-decimator 8 --frame-decimator 2 --screenshot
```

En el comando anterior, debemos reemplazar los '0' (ceros) tras cada crop-xxx por el valor en píxeles estimado para el recorte que deseamos hacer. Podemos empezar en 5 o menos, por ejemplo, e ir subiéndolo poco a poco hasta que veamos que en las capturas ya no hay bandas negras en ninguno de los laterales de la imagen. Anotaremos entonces los valores experimentales que hemos determinado para “**cropLeft, cropRight, cropTop, cropBottom**” y los pondremos en el código del *grabber* antes mostrado.

Finalmente, con todo ya configurado, reenviamos de nuevo el .json a la Pi, y reiniciamos Hyperion manualmente desde Putty con la orden siguiente:

```
/storage/hyperion/bin/hyperiond.sh /storage/.config/hyperion.config.json </dev/null >/dev/null 2>&1 &
```



**TRUCO:** cada vez que sea necesario escribir en Putty una orden o comando muy largo, para evitar equivocaciones y posibles errores simplemente copiamos la orden completa al portapapeles, y luego, sobre la ventana de sesión de Putty, hacemos click con el botón derecho del ratón. La orden completa se pegará en Putty, faltando sólo pulsar Enter para ejecutarla.

Las órdenes anteriormente entradas en Putty -mientras no cerremos la sesión- se pueden recuperar mediante las teclas de cursor arriba/abajo.

**¡¡Y esto es todo amigos!!**

## Notas finales

Espero que este tutorial te permita montar y configurar con éxito un ambilight casero, pero sobre todo, que aprendas y disfrutes de todo este gratificante proceso. Este tutorial de montaje y configuración se ofrece tal cual, con la esperanza de que pueda ser útil, pero no ofrezco garantías de ninguna clase.

Gracias a las opiniones y contribuciones de muchas personas de este foro me animé a montar el ambilight. He aprendido mucho durante el proceso y creo justo querer contribuir por mi parte a que otros se animen. En el fondo he escrito esta guía por puro egoísmo... a partir de ahora cada posible duda que me planteéis en el foro, será redirigida automáticamente a este libro gordo de Petete =P

Sé que la explicación ha sido larga, y para muchos probablemente demasiado obvia, pero como comenté al principio, ha sido pensada especialmente para quien no tiene suficientes conocimientos técnicos. Esperemos que este tocho-manual les ayude a llevar a buen puerto este proyecto.

He revisado el texto a conciencia, pero seguro que debe haber más de un error... Probablemente Murphy se esté ya frotando la manos... En consecuencia, no ofrezco garantía alguna y no me hago responsable de posibles daños que sufra tu instalación por fallos de montaje ajenos a mi persona o por errores no premeditados que pudiese contener este tutorial. En el mejor de los casos disfrutarás con tu nuevo ambilight, y en el peor, tendrás la satisfacción de enviar una nueva Pi al cielo de las Pis Churruscadas.

Agradezco de antemano vuestras sugerencias y correcciones, y en la medida de lo posible, se podrán ir añadiendo nuevas secciones si así fuese necesario. Agradezco también a todos los sitios, foros y personas que desinteresadamente han publicado y siguen publicando tutoriales en internet en beneficio de todos. Gracias especiales a los integrantes de [Lightberry.eu](http://Lightberry.eu). Nunca he contactado con ellos y no tengo, por tanto, ninguna relación ni interés comercial, pero es justo reconocer que sus productos y el soporte que prestan a través de su sitio web han allanado el camino y contribuido a que muchos nos hayamos decidido a montarnos nuestro ambilight casero.

Para cualquier sugerencia o reporte de errores, podéis contactarme en la siguiente dirección de correo electrónico: **[popysilver@gmail.com](mailto:popysilver@gmail.com)**

Por último ruego que las dudas sean redirigidas hacia el foro, porque así las respuestas ayudarán a otros muchos que se puedan encontrar ante las mismas dificultades. No responderé por email ni por MP a dudas que deberían ser resueltas a través del foro o a cuestiones ya aclaradas en este manual. Sé que es más fácil que te den la respuesta directa a tus dudas que leerse este tocho, pero en esta vida en algún momento hay que empezar a masticar por uno mismo, y aquí tienes para masticar un buen rato ;P.

Todo el contenido aquí publicado se ampara bajo la licencia [Creative Commons CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)