

# ODROID

Año Seis  
Num. #69  
Sep 2019

Magazine

## ¡GO-Tank, Go!

CONTROLA UN  
ROBOT QUE SE  
DESPLAZA CON  
TU ODROID-GO



**ODROID-N2:**  
UN ESTUDIO AL  
DETALLE DEL  
BUQUE INSIGNIA  
DE HARDKERNEL

**MULTI-ARRANQUE C2:**  
IMAGE MULTIARRANQUE PARA ODROID-C2

**VIDEO COMPUESTO:**  
UN ANTIGUO MONITOR CON EL ORDENADOR  
MAS POTENTE DE HARDKERNEL



## Cinco Minutos de Diversión con tu Monku R1: Una Tarjeta SD Portátil Sega Genesis

© September 1, 2019

Ahora que ya tienes configurada tu nueva y estupenda consola de juegos retro y disfrutas jugando algunos juegos, detengámonos un momento y pensemos qué más podemos hacer con nuestro dispositivo ODROID. Necesitarás un dispositivo Monku Retro, como el que ya hemos enseñado a fabricar (R1, R2), y en te caso(R3). ▶



## Juegos Linux: PC-Engine/TurboGrafx - Parte 6

© September 1, 2019

Con este artículo, llegamos al final de la serie PC-Engine / TurboGrafx, que admito que ha durado bastante más de lo que esperaba. También ha sido mucho más interesante de lo que inicialmente pensaba. Veamos qué trae esta última entrega, si continua la serie de los famosos shooters, o si ▶



## ¡GO-Tank, Go! - Controla un Robot Tanque con tu ODROID-GO

© September 1, 2019

Usar un sistema de juegos portátil ODROID-GO para controlar un par de pequeños motores DC ubicado dentro de un robot tanque es más fácil de lo que piensas. Esta tarea es aún más llamativa cuando te das cuenta que solo es necesario utilizar dos de los pines de entrada/salida de ▶



## CoreELEC para el ODROID-N2

© September 1, 2019

Kodi ha sido durante años uno de los reproductores multimedia más populares y con bastante frecuencia es el que se suele recomendar cuando la gente pregunta por alguna solución multimedia, ya que está disponible para múltiples sistemas operativos y una amplia gama de dispositivos. La mayoría de los dispositivos de ▶



## Cómo Crear una Consola de Juegos Retro Monku - Parte 3: Dando las Ultimas Pinceladas

© September 1, 2019

Esta es la continuación del artículo sobre la consola de juegos retro del mes pasado, donde aprendimos cómo configurar el software de una consola de juegos retro.



## Conexión de Video Compuesto ODROID-N2: Usa tu Antiguo Monitor con el Ordenador más Potente de Hardkernel

© September 1, 2019

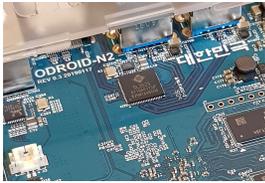
El video compuesto es un sistema de transmisión de video analógico que fue muy popular antes de la era de la televisión digital. El ODROID-N2 soporta esta señal de video.



## El Punto G: Tu Destino para Todas las Cuestiones Relacionadas con Juegos Android

© September 1, 2019

La noticia más destacada del mundo de los juegos de agosto viene de la mano del lanzamiento del 29 de julio de Gigantic X para Android.

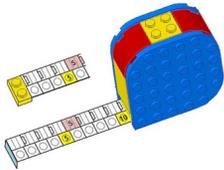


## Análisis del ODROID-N2

© September 1, 2019

El ODROID-N2 tiene un enorme potencial y es el SBC ARM más potente que he visto. Se adapta a muchos posibles usos, desde un servidor doméstico/mini hasta un centro multimedia con todas las funciones o un ordenador de escritorio que puede ejecutar

casi cualquier carga de trabajo instalada o en [▶](#)



## Cinco Minutos de Diversión con tu Monku R1: Redimensionar la Partición de la Tarjeta SD

© September 1, 2019

Este tutorial te mostrará cómo ajustar el tamaño de las las particiones en una tarjeta SD hecha a partir de una imagen de una tarjeta SD de menor tamaño. Por ejemplo, tiene una nueva tarjeta SD de 32 GB lista para tu Monku1000/ODROID-GO



## Gestiona el Tiempo de Ordenador de tu Hijo con mqttNanny

© September 1, 2019

En mi último artículo configuré mi ODROID-H2 para que fuera el primer ordenador (ejecutando Linux) de mi hijo de 7 años. Como sabes, un gran poder conlleva una gran responsabilidad, esto significa que tengo que ser capaz de imponer algunos límites de tiempo de uso del ordenador.



## Una Potente Imagen Multiarranque para el ODROID-C2: Ejecuta Android, Ubuntu MATE y LibreELEC desde un Único Dispositivo

© September 1, 2019

Inicié este viaje con un dispositivo Raspberry Pi 2 y un gestor de arranque llamado: "Berryboot" (no soy su desarrollador) que en realidad todavía está activo y es mantenido por la comunidad Raspberry Pi. Luego oí hablar de este increíble dispositivo llamado ODROID-C2 más potente que un dispositivo Raspberry Pi [▶](#)

# Cinco Minutos de Diversión con tu Monku R1: Una Tarjeta SD Portátil Sega Genesis

© September 1, 2019 By Brian Ree Juegos, ODROID-C1+, ODROID-C2, ODROID-XU4, Mecaniquero



Ahora que ya tienes configurada tu nueva y estupenda consola de juegos retro y disfrutas jugando algunos juegos, detengámonos un momento y pensemos qué más podemos hacer con nuestro dispositivo ODROID. Necesitarás un dispositivo Monku Retro, como el que ya hemos enseñado a fabricar ([R1](#), [R2](#)), y en te caso([R3](#)). Puedes usarlo para crear una tarjeta SD de expansión para la línea AtGames de dispositivos portátiles Sega Genesis. Te mostraremos cómo preparar la tarjeta SD, que únicamente se puede hacer en un sistema basado en Linux, que viene a ser tu ODROID. Esto es realmente genial. El dispositivo no es tan asombroso como un [ODROID-GO](#) pero ejecuta muy bien los juegos de Sega Genesis y tiene soporte para ampliar la lista de juegos con tus propias ROM. Prepara tu dispositivo ODROID y empecemos el tutorial.

## Herramientas necesarias

- Un Monku Retro 1, 2, 3 / [ODROID-C1+](#), [ODROID-C2](#), [ODROID-XU4](#). Se entiende que estos dispositivos están configurados con Ubuntu y MATE. Haga clic [aquí](#) para dispositivos R1 y R2, y [aquí](#) para dispositivos R3.
- Un dispositivo portátil AtGames Sega Genesis - [53.99\\$](#)
- Una tarjeta Micro SD de 8GB o más - [8.00\\$](#)
- Convertidor de tarjeta SD a USB - [9.99\\$](#)

## La tarjeta SD

De hecho, no puedes usar una máquina Mac o Windows para crear una tarjeta SD compatible con la serie de dispositivos portátiles AtGames de Sega Genesis. También sucede que tu dispositivo ODROID con Ubuntu es uno de los únicos dispositivos que puede crear una que funcione correctamente. Hazte con las herramientas que necesitarás. A continuación, tienes todo para seguir este tutorial.



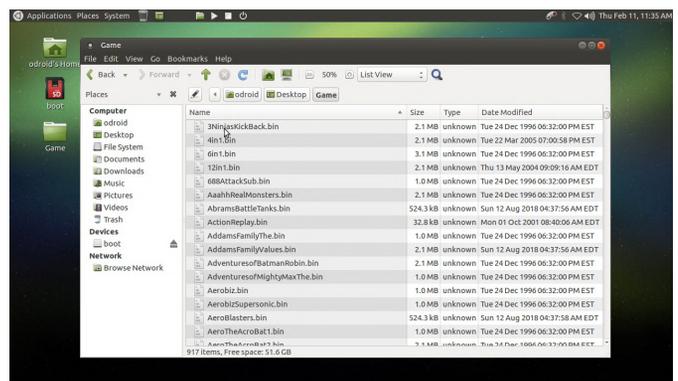
**Figura 1: Todas las herramientas que necesitarás para este proyecto**

Para este pequeño proyecto necesitaremos un dispositivo portátil AtGames Sega Genesis con una ranura de expansión para la tarjeta SD, tal y como se muestra a continuación. También necesitaremos una tarjeta SD de algún tipo. Nosotros usaremos una tarjeta micro SD con adaptador de tarjeta SD. También será necesario un lector de tarjetas SD. En nuestro caso usaremos un lector USB dual para tarjetas SD/micro SD.



**Figura 2 - Lector USB tarjetas SD/micro SD**

Haz una copia de tu colección ROM Sega Genesis. Necesitarás realizar algunos cambios en la forma en que se nombran los archivos. Las ROM del juego deben copiarse en una carpeta en la raíz de la tarjeta SD preparada llamada Game. Las ROM del juego deben tener una extensión de archivo .bin y un estilo de escritura camelCase con los caracteres y espacios no alfanuméricos eliminados.



**Figura 3 - Los archivos deben estar en camelCase con espacios y caracteres especiales eliminados**

Conecta tu tarjeta SD y lector de tarjeta SD en tu dispositivo ODROID. Si la tarjeta SD tiene un sistema de archivos admitido, deberías ver un nuevo icono en el escritorio. Ignóralo por ahora. Si no tienes GPARTED

instalado, ejecuta los siguientes comandos. Abre el terminal mate desde esta ubicación del menú: Applications -> System Tools -> MATE Terminal.

```
$ sudo apt-get install gparted -y
```

A continuación, ejecuta GParted. Abre esta ubicación del menú: System -> Administration -> GParted. Debería aparecer una aplicación como la que se muestra a continuación. Si te solicita una contraseña, introduce la contraseña de root por defecto, odroid o la contraseña que estés utilizando.

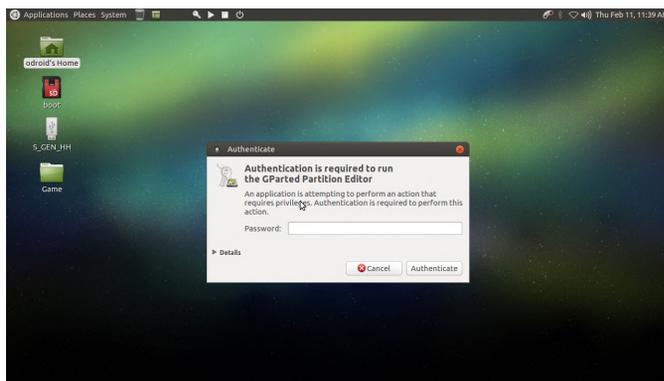


Figura 4 - Introduce tus credenciales

Selecciona tu tarjeta SD de la lista desplegable en la esquina superior derecha de la ventana de la aplicación. Compruébalo dos veces para asegurarte de que estás haciendo referencia a la nueva tarjeta SD y no a la que estas arrancando el ODROID. Haz clic derecho en cualquier partición de la tarjeta SD y selecciona Unmount. Haga clic derecho nuevamente y selecciona delete. Repite este proceso hasta que no haya particiones disponibles, solo espacio sin asignar.

Crea una nueva partición primaria con la configuración que se muestra en la siguiente captura de pantalla. Ajusta los siguientes parámetros.

```
Free Space Preceding: 1
New Size: maximum available
Free Space Following: 0
Align To: MiB
File System: Fat32
Create As: Primary Partition
Label: S_GEN_HH
```

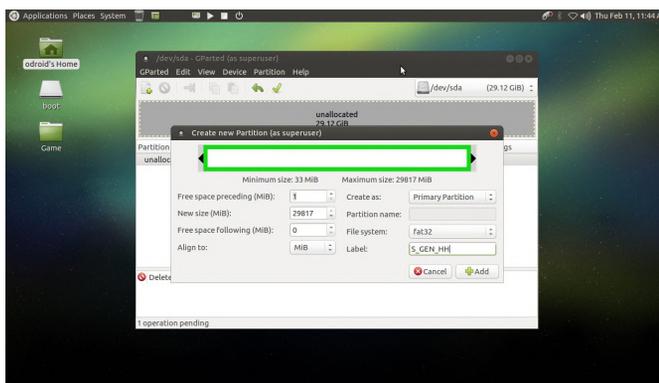


Figura 5: Utiliza esta configuración para crear tu partición

GParted ha registrado los cambios que queremos realizar en las particiones de la nueva tarjeta SD, pero aún no ha hecho nada. Para ejecutar los pasos que hemos descrito, debes hacer clic en la casilla de verificación verde "Apply All Operations" en la parte superior de la ventana, tal y como se muestra a continuación. Los pasos se ejecutarán y terminaremos con una única partición desmontada con los parámetros que hemos configurado anteriormente.

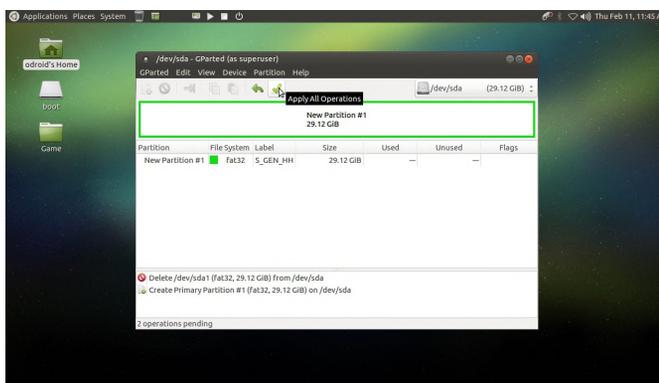


Figura 6 - Aplicar todas las operaciones

Espera a que se complete el proceso, luego haz clic en el botón Close. Cierra GParted, luego desconecta y vuelve a conectar la tarjeta SD para que vuelva a montarse limpiamente. Ahora puede copiar las ROM en el directorio Game de la tarjeta SD que acaba de crear.

Introduce la tarjeta SD en tu dispositivo y enciéndelo. Selecciona la opción Tarjeta SD del sistema de menú integrado: se encuentra en la última pantalla del menú. Una vez que la hayas seleccionado, deberías ver el mensaje "Searching For Games...".



Figura 7: Selecciona la tarjeta SD y espera mientras el sistema busca juegos

Ahora puede reproducir cualquier ROM Sega Genesis en tu dispositivo portátil Atgames Sega Genesis gracias a tu ODROID. ¡Goza y Disfruta!

# Juegos Linux: PC-Engine/TurboGrafx - Parte 6

© September 1, 2019 By Tobias Schaaf ↗ Juegos



Con este artículo, llegamos al final de la serie PC-Engine / TurboGrafx, que admito que ha durado bastante más de lo que esperaba. También ha sido mucho más interesante de lo que inicialmente pensaba. Veamos qué trae esta última entrega, si continua la serie de los famosos shooters, o si aún quedan algunas sorpresas por descubrir.

## Juegos que me han gustado

### Valis Series



Figura 1 - Valis I de la parte 4 de esta serie



**Figura 2: Buenos gráficos y un sistema de juego muy agradable en general**

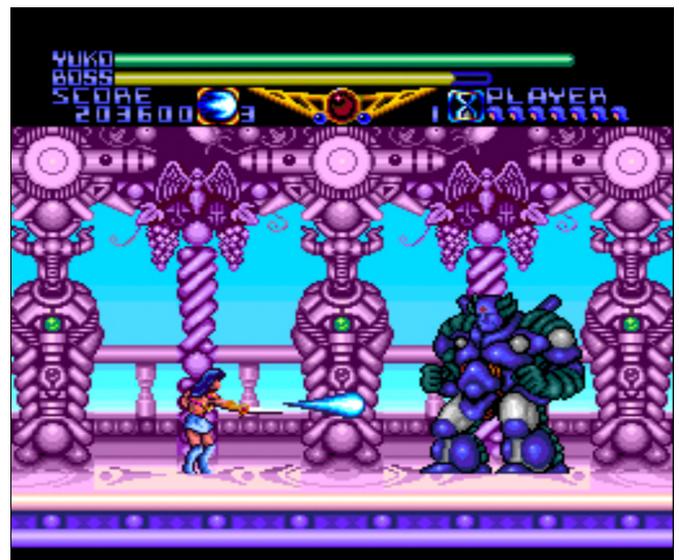
Valis es una serie de plataformas de acción donde juegas con una joven colegiala que empuña una espada mágica y lucha contra demonios. Ya hablé de la primera parte en la parte 4 de esta serie bajo el nombre de "Mugen Senshi Valis - Legend of a Fantasm Soldier". Esta es una nueva versión del juego original de Valis que salió primero para MSX y NES. La versión PC Engine fue una nueva versión con gráficos y escenas de anime bastante mejorados. Aunque es el primero de la serie, es el último que se lanzó para PC Engine en 1992. El primero que se lanzó para PC Engine fue Valis II en 1989.

Valis II es muy similar en jugabilidad a Mugen Senshi Valis - Legend of a Fantasm Soldier, pero con algunas pequeñas diferencias. No puedes deslizarte como lo haces en el Valis original, y las armas que recoges se muestran como iconos. Tu magia se puede activar simplemente presionando la tecla "arriba" que puede llevarte a activarla accidentalmente con el tiempo. No existe un medidor de puntos mágicos (MP) para tu magia que te indique con qué frecuencia puedes usar tu magia y qué puedes rellenar. En cambio, puedes recolectar elementos que te permiten un cierto número de usos de esa habilidad en particular. Aparte de esto, continúa siendo básicamente cortar y trocear, así como saltar y correr. Los gráficos no son tan buenos, no hay desplazamiento de paralaje en el juego, y los videos tienen animaciones muy limitadas. También tienen un borde verde un tanto desagradable que se eliminó en entregas posteriores

de la serie. El juego fue completamente doblado al inglés, lo cual hace que suene bastante a cliché, aunque te ayuda a entender la historia general.



**Figura 3: El borde verde en las secuencias de video se eliminó en posteriores lanzamientos de la serie**



**Figura 4 - Los jefes son bastante fáciles**

Valis III salió en 1990 para PC Engine (y 1992 para Turbo Graphics). Mejora con respecto a Valis II en muchos aspectos. Atrás quedó ese borde tan poco agradable que aparecían alrededor de las secuencias de video, y éstas mejoran bastante al incorporar muchas más animaciones. Los gráficos en conjunto también mejoraron.



Figura 5 - Gráficos mejorados en general en Valis III



Figura 6 - Los jefes son un poco más difíciles, pero todavía no representan un auténtico desafío.

Está de vuelta la barra de MP que se puede rellenar recogiendo elementos. Incrementa la fuerza de tus ataques al aumentar la longitud del medidor de espada: cuanto más se rellene el medidor antes de hacer un ataque, más daño causará. Ya no tienes diferentes estilos de espada, sino que puedes cambiar entre diferentes personajes. Éstos tienen diferentes ataques y estilos de ataque. Puedes cambiar de personaje en cualquier momento dentro del juego, a excepción de las peleas de jefes en las que debes mantenerte con el que inicias el combate.

También vuelve el deslizamiento ahora con el botón "START" mientras presionas la tecla hacia abajo. La magia se puede lanzar presionando la tecla hacia arriba y el botón de ataque en lugar de solo la tecla hacia arriba. En general, el juego es mucho mejor que la

segunda parte en muchos sentidos. Disminuye el desplazamiento de paralaje de la versión Mega Drive del juego, pero ofrece música con calidad de CD.

Valis IV mejora algo Valis III, ya que realmente tienes un poco de desplazamiento de paralaje aquí y allá. Lanzado en 1991 (un año antes de la nueva versión de Valis I), solo estaba disponible en japonés. Sin embargo, esta vez juegas con diferentes personajes y puedes cambiar entre tres personajes diferentes dentro del juego (similar a Valis III). Esta vez, sin embargo, esto es necesario para resolver algunos acertijos, ya que uno de los personajes puede hacer un doble salto para llegar a lugares más altos o más alejados.



Figura 7: Más de lo mismo en Valis IV, el sistema de juegos se parece mucho al de Valis III



Figura 8: Los jefes son divertidos y algo más desafiantes en esta ocasión

Esta vez, la magia ha desaparecido por completo y ha sido reemplazada por un ataque especial de tres fases para cada personaje. Cada fase requiere una cierta cantidad de energía de tu barra de WP que se rellena automáticamente siempre que no la uses o seas golpeado. Cada fase tiene un ataque diferente, y la tercera fase es un ataque de cubre la pantalla y causa mucho daño. Usarlo hará que disminuya tu medidor de WP general al nivel 2, y debes recoger un elemento para volver nuevamente al nivel 3. Sin embargo, no pierdes nada si usas los niveles 1 o 2.

Todos los juegos de la serie son divertidos y te los recomiendo si te gustan los juegos de plataformas de acción. Con el segundo y tercer juegos en inglés, es fácil seguir la historia por un tiempo y disfrutar de las secuencias de vídeo animadas.

### Wonderboy 3 - Monster Lair

Esta es una versión del clásico juego de arcade para PC Engine. No hay mucho que decir si ya conoces el juego arcade. Basicamente es una copia muy buena con sonidos de CD mejorados y eso es todo.



Figura 9: Excelentes colores brillantes y gráficos cómicos



Figura 10 - Segunda fase de cada nivel volando y luchando contra jefes

Si no conoces este juego, es un juego de plataformas de acción arcade diseñado para comer monedas. La pantalla se desplaza constantemente hacia la derecha y debes avanzar para evitar que te arrastren. La barra de vida verde también es una barra de tiempo que se va reduciendo lentamente con el tiempo, pero que se puede reponer recolectando frutas y artículos. Puedes recoger un par de armas diferentes que, por un corto tiempo, te ayudarán en tu búsqueda. El juego es divertido, la música esta bien, es una versión bastante buena del juego.

### Zero Wing



Figura 11 - Zero Wing para PC Engine



**Figura 12: Los jefes pueden llegar a ser enormes y causan que la pantalla parpadee.**

Este shooter no impresiona demasiado. En general un ambiente con buenos gráficos (pero nada sorprendentes), buena jugabilidad y buena música. El juego es bueno, no perfecto, pero divertido. Los gráficos carecen de un buen desplazamiento de paralaje, pero te enfrentas a muchos enemigos grandes y a situaciones desafiantes sin que lleguen a ser injustas. Incluso hay algunas situaciones en las que tienes que guiar tu nave espacial a través de espacios muy reducidos, que funcionan perfectamente bien, y no he tenido problemas como en otros juegos. Los jefes son inmensos, y es divertido descubrir los patrones para vencerlos.

Puedes recolectar tres tipos diferentes de armas representadas por los colores rojo, azul o verde. El primero es tu disparador de guisantes estándar (cañón vulcano) en rojo, que se despliega con el tiempo si lo actualizas y cubre gran parte de la pantalla. El azul es un arma tipo rayo láser, que va directo y causa mucho daño. Probablemente sea la mejor opción para matar a grandes enemigos y jefes, ya que causa mucho daño con el tiempo. El último es verde, que son disparos que apuntan y son dirigidos automáticamente a un enemigo de la pantalla. Cada una de las armas se puede actualizar si recolectas objetos del mismo color. Si cambias de color, cambias de arma, pero no aumentas ni disminuyes el poder de las armas. También hay otros potenciadores que te permiten volar más rápido, por ejemplo, o te dan una bomba que puedes arrojar a los enemigos, la cual

causa un daño masivo en un gran radio con el paso del tiempo.

También tienes la capacidad de "capturar" enemigos y usarlos como escudo o lanzarlos a otros enemigos, lo cual es divertido por un tiempo, pero lo dejé de hacer al final. Por supuesto que hay mejores shooters para PC Engine, pero realmente me gusta la simplicidad del estilo y la jugabilidad.

## Juegos que están bien

### Tenchi wo Kurau (aka Dynasty Wars)

Este juego de lucha con elementos RPG está bastante bien. Recolectas ciertos artículos para subir de nivel a tu personaje y obtener más HP, y puedes recolectar algunas armas de los enemigos que has eliminado para aumentar un poco tu fuerza. Puedes cargar tu ataque para hacer daño extra y luchas contra un montón de jefes al final del nivel. En general es un juego bastante bueno. Gráficamente, no es tan bueno como la versión arcade, pero está bastante bien gratificante y no tiene mala jugabilidad.

### Tengai Makyo - Kabuki Itouryodan

En este juego de estilo Street Fighter puedes elegir entre 8 personajes diferentes. En realidad, es bastante bueno para PC Engine y admite joysticks de 6 botones. Sin embargo, está completamente en japonés, e incluso los menús son difíciles de entender. También tiene un tiempo de carga bastante largo entre niveles en comparación con otros juegos. Aún así, si te gustan los juegos de Street Fighter, definitivamente deberías probarlo, ya que es uno de los mejores de este género para PC Engine.

### Tengai Makyou - Deden no Den

Básicamente es Bomberman con un conjunto de gráficos diferente, no hay nada particularmente bueno o malo en lo que destaque.

### The TV Show

No estoy seguro de si me gusta o no este juego. Puedes elegir entre 8 personajes diferentes, cada uno con una historia de fondo absurda, que los muestra desordenadas, y luego vas a algo llamado "The TV Show". Es una mezcla de rompecabezas y juego de lucha. Puedes configurar diferentes tipos de bombas,

y algunas matan o destruyen objetos cercanos, mientras que otras destruyen pisos enteros. Tu objetivo es romper ciertos objetos (antes de que caigan al pozo sin fin que hay debajo) para conseguir lo que tienen dentro. Las recompensas son en su mayoría diamantes para aumentar tu riqueza que necesitas para "continuar" cuando mueres o te equivocas. Es interesante, aunque solo sea por un rato.

### **Uchuu Senkan Yamato (aka Space Battleship Yamato)**

Este juego se reproduce como una película de animación interactiva. Eres el comandante del acorazado espacial Yamato. Controlas diferentes estaciones a las que le asignas diferentes tareas como disparar a los enemigos, enviar combatientes/bombarderos tras ellos, determinar dónde quieres concentrar tus reparaciones y fijar la dirección en la que quieres volar y cómo de rápido. Cosas como esta, combinadas con buenos gráficos y sonidos/música, hacen que este juego sea excelente, sin embargo, el juego está completamente en japonés y cuesta bastante entender lo que se supone que debes hacer. Es bastante fácil descifrar los controles, y los objetivos de la misión son un tema aparte. Es una pena, ya que me encantaría jugar a este juego en un idioma que entendiese. Es un concepto de juego único y disfruto bastante jugándolo.

### **Vasteel**

Este juego es un poco cajón desastre. La animación es larga, con una introducción y peleas totalmente en inglés, lo cual es bastante agradable, con muy buenos detalles. Sin embargo, el sistema de juego en sí es un poco diferente. Es un juego táctico similar a Advanced Wars para Gameboy Advance, History Line 1914-1918 o Battle Isle. Controlas unidades individuales de un ejército, creas otras nuevas (siempre que tenga los fondos) y las envías hacia ciudades y unidades enemigas. Si destruyes la planta de energía de una ciudad, y la tomas, obtienes a cambio más fondos. El objetivo es eliminar el principal edificio del enemigo y así ganar el mapa.

Esto suena fácil, pero puede llevarte muchas horas ganar un solo mapa, ya que constantemente intentas

capturar y mantener las mismas ciudades. Un poco diferente es el hecho de que puedes controlar tus unidades durante un combate. La mayoría de las veces, tus unidades tienen una pistola rápida muy poco eficaz, una pistola más grande lleva mucho tiempo disparar, o es lenta. Puedes intentar evitar los disparos de los enemigos e intentar golpearlos tú mismo. No está mal, pero como he comentado, un solo nivel puede llevarte muchas, muchas horas y se vuelve muy repetitivo. Sin embargo, está completamente en inglés, así que, si te gustan este tipo de juegos, no debería haber nada que te impida probarlo.

### **Vasteel 2**

Aunque está completamente en japonés, observé que este juego es más fácil que su predecesor. Además, usa la tarjeta Arcade, lo que significa memoria extra para las animaciones, que puedes ver muy bien en la introducción. Parece ser que sigues una historia, a diferencia del juego anterior donde tenías una historia y luego solo un mapa. Esta vez tienes objetivos de misión. No necesitas destruir un edificio principal del enemigo, pero a menudo tienes que destruir a todos los enemigos de la pantalla. Es divertido, aunque te sigue llevando bastante tiempo, aunque no tanto como el primer juego.

### **Where in the World is Carmen Sandiego**

Este juego de rompecabezas no tiene nada de especial. Viaja por todo el mundo para recoger pistas sobre el sospechoso de un crimen siguiendo indicios de personas que lo han visto. Creas una orden de arresto cuando tengas suficientes detalles sobre la persona, luego viajas hasta encontrarla. Es divertido descubrir las diferentes pistas, pero el texto se desplaza bastante lento, y el tema gráfico ni siquiera vale la pena mencionarlo. En general es muy normalito; No es malo ni tampoco bueno.

### **World Heroes II**

Este juego similar a Street Fighter 2 me sorprendió bastante. ¡La calidad de los gráficos y las animaciones son excelentes! Incluso puedes ver animaciones en el fondo del escenario como en las "grandes consolas", me sorprendió mucho ver ese nivel de animación y detalle. Sin embargo, la jugabilidad es

extremadamente difícil en mi opinión (o es que yo soy realmente muy malo). Ya era extremadamente difícil vencer al primer enemigo, a pesar de que el juego fue configurado en modo "fácil". Es una pena, ya que disfruté especialmente del juego y de los gráficos, pero el juego es demasiado difícil para mí como para disfrutarlo al completo.

### **Xak III - The Eternal Recurrence**

Este juego de rol fue traducido al completo por algunos entusiastas y, por lo tanto, se puede jugar en inglés, lo cual está muy bien. El juego en sí no es malo, con un estilo RPG de acción similar a los juegos Ys (descritos a continuación). Caminas y puedes atacar a los enemigos y saltar para superar obstáculos. A veces hay interruptores ocultos detrás de los ataúdes o una estantería, que se pueden mover. Las peleas de jefes suelen ser largas y bastante difíciles. Mientras que en cualquier momento del juego estar parado te cura, en una pelea de jefes esto no es así. El juego y la historia no están mal, y la música es buena, pero los gráficos están muy desactualizados. Puedes cargar tu ataque y hacer un ataque a distancia mágico (una bola de fuego) o crear una barra más fuerte para un daño extra. Lamentablemente, los controles no son tan buenos a la hora de hacer esto, y a menudo terminas mirando en una dirección diferente a la que pretendías.

### **Ys Series**

Aunque este es uno de los pocos juegos buenos y de los pocos juegos de rol en inglés para el sistema, prefiero jugarlo en otras consolas como la PSP, donde los gráficos son mucho mejores que en el PC Engine. Los Ys Book 1, 2 y 3 están disponibles completamente en inglés, y el tercero es el único en el que puedes saltar y atacar con un botón. La 4 entrega vuelve a las raíces y empiezas a toparte con todas las cosas nuevamente. Únicamente está en japonés, pero existe una traducción que al menos traduce los textos, aunque no el audio. En general, pienso que esta serie mejoró con el tiempo y con las diferentes versiones para otros sistemas, y como he comentado, prefiero jugarlo usando un emulador de PSP en ODRROID.

### **Yu Yu Hakusho - Anshoubu!! Ankoku Bujutsukai**

Los gráficos en este juego son magníficos tanto en la introducción animada como en el propio juego, aunque más bien se percibe como un shooter arcade donde apuntas con tu arma y disparas a una escena animada, y eso es básicamente lo que es. Apuntas a la cabeza del enemigo u a otros puntos débiles, como cuando corre a la izquierda y salta a la derecha, e intentas golpearlo repetidamente antes de que te golpee con demasiada frecuencia. Los gráficos impresionan y la animación es realmente buena, pero la jugabilidad se vuelve repetitiva en poco tiempo.

### **Juegos que no me han gustado**

#### **Travel Epuru**

Este juego puede estar bien al principio, pero realmente no es tan bueno. Hay dos niños pequeños que luchan entre sí con bombas, granadas, palos e incluso magia. El objetivo es derribar al otro niño. Todo gira en torno a un niño y una niña que pelean con disfraces de pingüinos absurdos. No es un escenario "amigable" ni la jugabilidad para nada es interesante.

#### **Conclusión**

Estoy un poco decepcionado con la última parte de la serie PC Engine/Turbografx, ya que supongo que estaba acostumbrado a muchos juegos impresionantes mientras escribía las otras entregas, y me llevó bastante tiempo hasta llegar a la última parte de la serie que pone de manifiesto algunas joyas.

Ha pasado mucho tiempo desde que empecé esta serie, y tengo que admitir que no esperaba que me llevara tanto tiempo, ni que la disfrutaría tanto. El PC Engine, especialmente la versión en CD, cuenta con algunos juegos realmente increíbles, y teniendo en cuenta el hardware que había detrás del Genesis/Sega CD y la SNES, supo mantenerse a flote en comparación con estas consolas, si también tienes en cuenta que la consola fue la primera en utilizar el formato CD años antes que el Sega CD. Era bastante asombroso lo que podías llegar a hacer con esta consola. He descubierto muchos juegos con lo que he disfrutado inmensamente el sistema, y con la reciente versión de PC Engine sobre ODRROID Go de @pelle7

tengo juegos aún más increíbles a los que puedo jugar sobre la marcha y que llegan a impresionarme mucho más que, por ejemplo, algunos simples juegos de NES.

En mi opinión, PC Engine era realmente un sistema infravalorado. Era muy avanzado para su época, me hubiera gustado haberlo jugado "en el pasado". Sin embargo, también tiene algunos inconvenientes. Los juegos se centraron en el mercado japonés y carece de buenos juegos de rol que me gusten, también están escasamente representados en este sistema otros generos. Puede que no lo haya mencionado anteriormente, pero también he jugado a muchos otros juegos japoneses, y tengo que admitir que es una autentica pena no llegar a entender nada, ya que algunos de ellos son simplemente increíbles y van mucho más allá de lo que otros juegos podría ofrecer a la plataforma.

Un ejemplo característico de esto es el juego "Private Eye Dol", que es un juego de aventuras de mucho caminar, buscar, hablar con la gente, interactuar con objetos, etc. El juego es simplemente increíble en cuanto a calidad, ya que utilizó la memoria extra de la Arcade Card para el sistema. Hay muchas animaciones, muchas secuencias de video que están

totalmente interpretadas, y en general los gráficos y la presentación son simplemente increíbles y se parecen más a un juego de SNES que a un juego de PC Engine. Solo deseaba poder jugar a este juego en un idioma que pudiera entender.

Otro ejemplo es "Seiya Monogatari - Aearth Fantasy Stories", que es un juego de rol que se parece más al SNES o una máquina más potente. La introducción es muy detallada, y los gráficos del juego parecen ser de una consola más potente, combinada con una banda sonora increíble. Estos son los juegos que realmente me estoy perdiendo debido a la barrera del idioma.

Hay muchos más juegos impresionantes como Snatcher, que afortunadamente también salieron para Sega CD dos años después de su lanzamiento para PC Engine, pero con una pequeña censura en comparación con la versión de PC Engine. Incluso hay más juegos que realmente valen la pena, pero que solo están disponibles en japonés. Muestran perfectamente hasta qué punto el hardware podría haber sido exprimido y lo que era capaz de hacer. En general, me impresionó mucho el sistema, especialmente teniendo en cuenta la antigüedad y las cosas que ahora damos por sentado en los sistemas más modernos.

# ¡GO-Tank, Go! - Controla un Robot Tanque con tu ODRROID-GO

© September 1, 2019 By Dave Prochnow Linux, ODRROID-GO, Tutoriales



Usar un sistema de juegos portátil ODRROID-GO para controlar un par de pequeños motores DC ubicado dentro de un robot tanque es más fácil de lo que piensas. Esta tarea es aún más llamativa cuando te das cuenta que solo es necesario utilizar dos de los pines de entrada/salida de propósito general (GPIO) del GO.

En términos generales, la "típica" interfaz de control para un motor DC consta de cinco líneas separadas e independientes: las líneas de alimentación/control obvias, como es 5V o GND, y tres entradas lógicas para controlar el robot (por ejemplo, una entrada para avanzar, una entrada para retroceder y una línea lógica final para controlar la velocidad del motor). Sin embargo, los lectores más perspicaces se percatarán de que únicamente se conectan realmente al motor dos líneas. Las otras tres líneas configuran un controlador de motor para "manejar" estas dos líneas de alimentación conectadas (es decir, cambian la polaridad y controlan una serie de pulsos al motor a través de una señal de modulación de ancho de pulso (PWM)). ¿Cómo es esto posible?



**Figura 1.** Echa a andar a tu ODRROID-GO por la calle con este robot tanque.

El "ingrediente secreto" para que todo funcione es un pequeño circuito que reside entre el controlador del robot y el motor DC. Este circuito puede denominarse de varias formas como son un "controlador de motor", "Puente H" o "regulador de motor", aunque

su función es manejar el motor de acuerdo con los comandos enviados desde el controlador del robot.

En nuestro caso, el ODROID-GO es el controlador del robot. Una plataforma móvil en forma de tanque DFRobot es nuestro robot tanque. Un pequeño circuito transistor es nuestro controlador de motor. Éste, sin embargo, NO es un circuito H-Bridge, sino un simple interruptor de alimentación. Por otro lado, solo usaremos dos pines GPIO para controlar nuestro robot tanque: GPIO #4 y GPIO #15. Sin embargo, también necesitaremos usar el pin GND #1 y el pin #6 3.3V de la interfaz del cabezal GPIO del ODROID-GO. Estos pines de alimentación NO controlan el robot. Sino más bien se utilizan para ayudar al ODROID-GO a "comunicarse" con el circuito del controlador de motor basado en un transistor.



Figura 2. Desliza hacia la izquierda - gira a la derecha.



Figura 3. Desliza hacia la derecha - gira a la izquierda.



Figura 4. Ahora cha-cha-chá por todos lados, conduce hacia adelante.

Y lo mejor de GO-Tank es que tu dispositivo inteligente con WiFi será utilizado para controlar el robot tanque.

## Componentes

- Plataforma móvil DFRobot Devastator Tank #ROB0128 - 84.90\$ ([mouser.com](http://mouser.com))
- Hardkernel Level Shifter: 5V a 3.3V - 1\$ ([hardkernel.com](http://hardkernel.com) o [ameridroid.com](http://ameridroid.com))
- DC-DC Hardkernel Step Up - 1.50\$ ([hardkernel.com](http://hardkernel.com) o [ameridroid.com](http://ameridroid.com))
- 2 Transistores 2N3904 NPN - 1.28\$ ([bgmicro.com](http://bgmicro.com))
- 2 Diode 1N4001 - 0.04\$ ([bgmicro.com](http://bgmicro.com))
- 2 resistencias de 330 ohmios - 5.23\$ ([bgmicro.com](http://bgmicro.com))
- Baterías, placas de prueba, cable

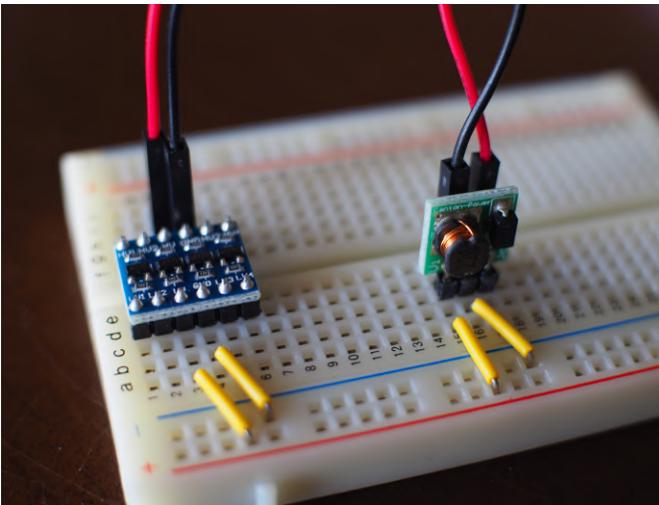
## Paso a paso

1. De serie, el tanque Devastator no parecía el apropiado. De modo que fabrique un de par paneles laterales para la carcasa inferior. Estos nuevos paneles me permitieron mover las ruedas dentadas y los motores a la parte trasera del robot.



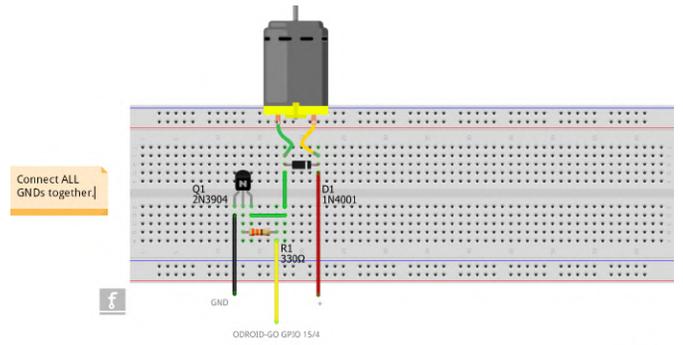
**Figura 5.** Este es el aspecto que debería tener el tanque adecuado - movimiento de la rueda trasera.

2. Sigue el diseño del circuito de la wiki ODROID Ultrasonic Distance Meter. Esta disposición permitirá que los pines GPIO de 3.3V en el ODROID-GO dirijan los controles del motor del interruptor transistor con alimentación de 5V. En este caso, el Step Up es utilizado para alimentar el Level Shifter y NO para alimentar los motores.

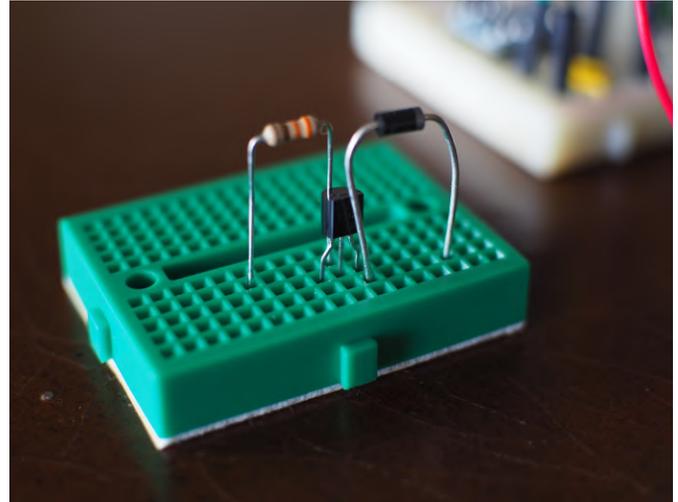


**Figura 6.** El circuito del el level shifting para manejar los interruptores de potencia del motor y NO los motores del robot.

3. Conecta dos controladores de motor del interruptor de potencia siguiendo el diseño de este diagrama de placa de pruebas (Figura 7) y la fotografía (Figura 7a):

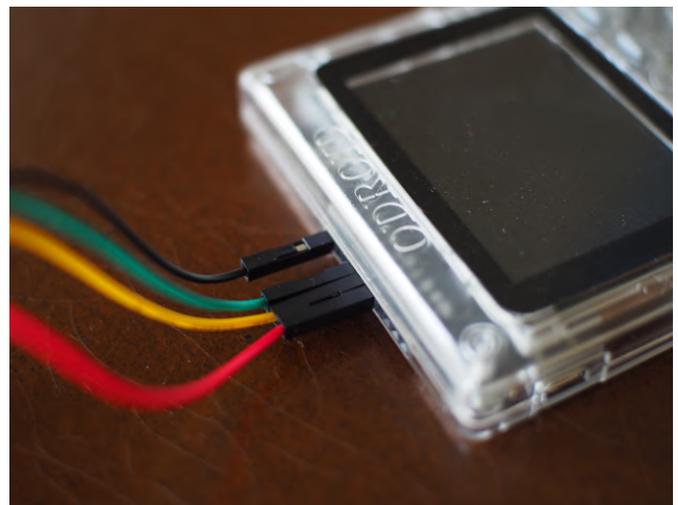


**Figura 7.** Utilizar una placa de pruebas es una excelente manera de crear los circuitos necesarios para controlar este robot.

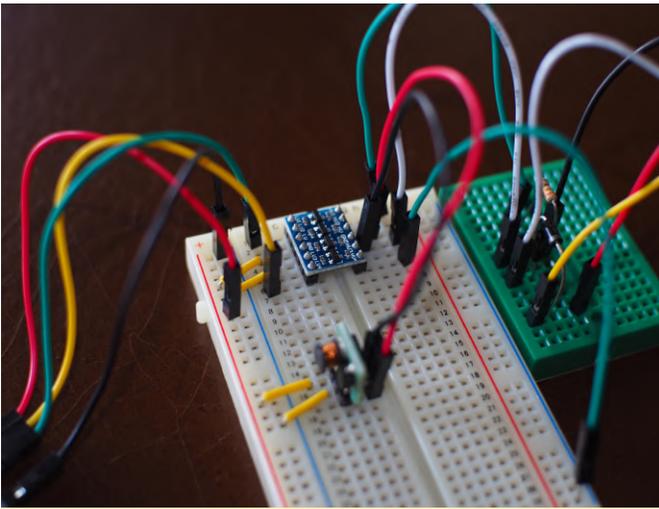


**Figura 7a.** Esta es una solución elegante para añadir un simple control de ENCENDIDO/APAGADO a un motor DC.

4. Conecta una de las salidas del controlador del motor a uno de los terminales de alimentación de los motores DC. Haz lo mismo para el otro controlador de motor y conéctalo al segundo motor DC. Conecta el cable de la puesta a tierra (GND; negro) al otro terminal no conectado en cada motor.



**Figura 8.** Las conexiones GPIO del ODROID-GO.



**Figura 9. Sigue las conexiones GPIO hasta el controlador del motor. Solo se muestra uno de los dos controladores de motor en esta imagen.**

5. Conecta el paquete de baterías del Devastator a ambos controladores del motor (como se muestra en la Figura 8 con los cables GND rojo y negro). Cuando se envía un comando por uno de los pines GPIO ODR0ID-GO, los transistores en estos controladores de motor conmutarán la energía de la batería directamente en cada motor. ¡Interesante!

6. Programa el ODR0ID-GO con un código similar al siguiente:

```
#include
#include

#define PIN_MOTOR_RIGHT 15
#define PIN_MOTOR_LEFT 4

const char *apSSID = "ODROID_GO_AP";
const char *apPWD = "12345678";

WiFiServer server(80);

ILI9341 lcd = ILI9341();

void setup()
{
  IPAddress gateway(192, 168, 4, 1);
  IPAddress subnet(255, 255, 255, 0);

  --snip--
```

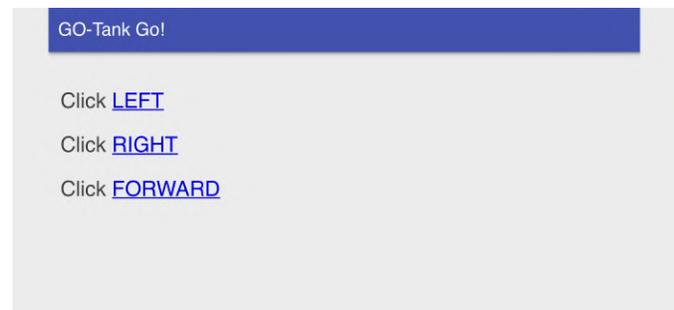
Inserta los parámetros del Punto de acceso WiFi, la configuración de la LCD y configura la interfaz web

Luego añade las funciones para definir movimientos hacia la izquierda, derecha y hacia adelante:

```
void goFORWARD() {
  lcd.setTextFont(4);
  lcd.setTextSize(2);
  lcd.setCharCursor(2, 5);
  lcd.setTextColor(PURPLE, BLACK);
  lcd.println("Come Back,");
  lcd.setTextSize(3);
  lcd.setCharCursor(2, 7);
  lcd.println("Shane!");
  digitalWrite(PIN_MOTOR_RIGHT, HIGH);
  digitalWrite(PIN_MOTOR_LEFT, HIGH);
  delay(2000);
  digitalWrite(PIN_MOTOR_RIGHT, LOW);
  digitalWrite(PIN_MOTOR_LEFT, LOW);
}
```

7. Coloca el robot Devastator encima de una pequeña caja para probar su funcionamiento.

Después de encender el ODR0ID-GO, conecta tu dispositivo inteligente al punto de acceso WiFi ODR0ID-GO creado dentro del código del programa y usa tu navegador para cargar la página web generada por el código en: <http://192.168.4.1>. Al presionar "LEFT" en esta página web, debería ver una pantalla roja en el ODR0ID-GO y la banda de rodadura izquierda debería moverse, esta acción girará el tanque a la derecha. Al presionar "RIGHT", la pantalla se vuelve verde y mueve el tanque hacia la izquierda, mientras que al presionar en "FORWARD" se ejecutan ambas huellas y el tanque se desplazará hacia adelante.



**Figura 10. La interfaz de control del robot en tu dispositivo inteligente.**

Ahora sal y disfruta de tu recién descubierto Tanque ODR0ID-GO móvil.

## Notas

1. *TODAS las líneas de puesta a tierra (GND) deben conectarse entre sí, ¡independientemente de los voltajes!* y espera al gemido del pequeño Joey por el pistolero que se marcha.

2. ¿Por qué Shane? Lee la clásica novela del oeste de Jack Schaefer de 1949, o ve la adaptación cinematográfica de 1953 protagonizada por Alan Ladd

# CoreELEC para el ODROID-N2

© September 1, 2019 👤 By @Cl6N0z ➔ ODROID-N2

---



Kodi ha sido durante años uno de los reproductores multimedia más populares y con bastante frecuencia es el que se suele recomendar cuando la gente pregunta por alguna solución multimedia, ya que está disponible para múltiples sistemas operativos y una amplia gama de dispositivos. La mayoría de los dispositivos de bajo coste que se suelen comprar con fines de reproducción multimedia están equipados con un sistema operativo Android donde Kodi a menudo está preinstalado o se puede instalar con facilidad. Puesto que Android y Kodi son ampliamente utilizados, podría pensarse que combinarlos sería la mejor experiencia multimedia. En realidad, no es el caso y la experiencia puede volverse un tanto mediocre cuando la reproducción se entrecorta o aparecen otros problemas. Afortunadamente, existen múltiples alternativas a Android y una de las mejores la presentamos en este artículo: CoreELEC.

CoreELEC es una distribución Linux basada en la tecnología Kodi con el principal objetivo de ser lo más ligera posible, ideal para dispositivos de baja potencia. Se separó del conocido LibreELEC en 2018 y se diferencia de éste por una filosofía que busca garantizar que los usuarios obtengan la mejor experiencia posible con el hardware y el software que hay actualmente disponible. Con esto en mente, CoreELEC en ODROID-N2 ha demostrado ser un fuerte competidor dentro de las mejores soluciones multimedia de bajo costo que existen hasta la fecha. Y con el soporte del administrador de paquetes Enware, tiene capacidad para ser un buen servidor de bajo coste además de una solución multimedia como vamos a ver en este artículo.

**¿Qué es CoreELEC?**



**Figura 1: Pantalla de bienvenida**

## Configuración de hardware

A continuación, se muestra una lista del hardware utilizado con algunas recomendaciones e información adicional:

Recomendado:

- ODRROID-N2 con 2/4 GB de RAM
- Fuente de alimentación 12V/2A
- Cable de alta velocidad 4K HDMI 2.0 (tipo A-A)
- Carcasas ODRROID-N2
- Mando a distancia IR
- Tarjeta Micro SD

Optional:

- Módulo eMMC N2 de 8/16/32/64/128 GB
- Módulo Bluetooth 2
- Pila de reserva RTC
- Módulo WiFi

Como yo quería usar ODRROID-N2 con CoreELEC como servidor doméstico usando Entware además de utilizarlo como reproductor multimedia, elegí 4 GB de RAM. No obstante, hay que tener presente que con 2 GB es suficiente para la mayoría de los usuarios. Lo mismo podemos decir del módulo eMMC, donde 8/16 GB es más que suficiente para usar Kodi con normalidad. Para el resto de las opciones, depende de lo que necesites y todas funcionan como cabría esperar con CoreELEC.

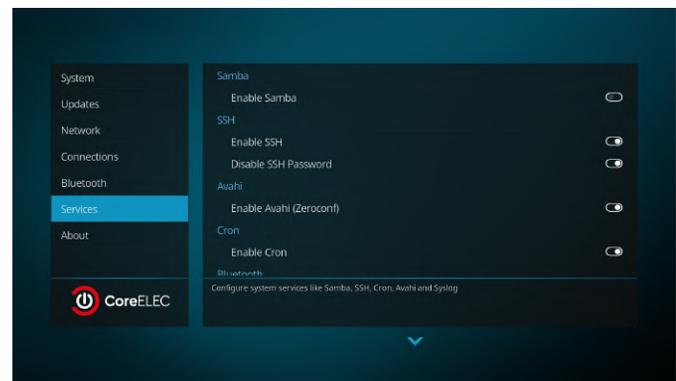
## Instalación

CoreELEC se instala fácilmente siguiendo la guía de instalación de <https://coreelec.org/>. Después de insertar la microSD grabada y desconectar el módulo eMMC antes de encender el ODRROID-N2 para

asegurarnos de que CoreELEC se inicie desde la tarjeta microSD, en menos de 2 minutos, CoreELEC estaba listo para usarse. Para aumentar el rendimiento, CoreELEC ha sido instalado en el módulo eMMC ejecutando el comando "installtoemmc" dos veces.

## Configuración de Kodi

La configuración por defecto tras la instalación posiblemente satisfaga las necesidades de la mayoría de los usuarios, son obstante aún hay algunas configuraciones a las que deberíamos prestar cierta atención. Empecemos con CoreELEC.



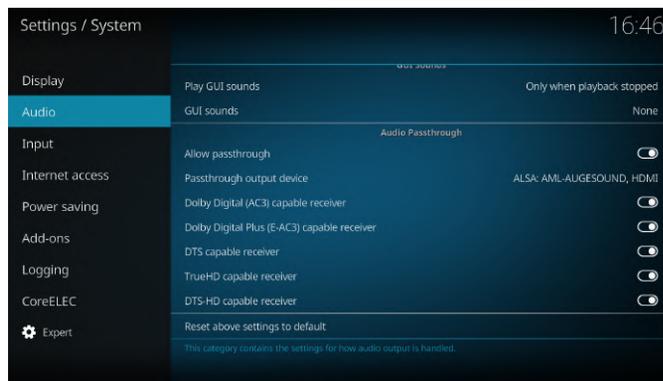
**Figura 2: Pantalla de configuración CoreELEC**

Dos características que se suelen utilizar bastante son Samba y SSH. Si habilitas alguna de ellas, recuerda configurar o cambiar las credenciales. Para Samba, hemos activado "Use Samba Password Authentication" con una contraseña segura. Para SSH, puedes configurar una nueva contraseña o simplemente insertar una clave pública. Las claves SSH son la solución más segura y más rápida, de modo que lo hice copiando mi clave pública en `.ssh/certified_keys`. Por último, deshabilitamos "SSH Password" puesto que ya no era necesario.

Siguiendo con el tema de la configuración no encontramos con System, donde vale la pena detenerse en los dos aspectos Display y Audio. Si tu televisor tiene problemas con determinadas resoluciones o si deseas cambiar a una resolución más baja que tu GUI de Kodi, selecciona las resoluciones compatibles con tu televisor en la Whitelist. Algunas personas pueden encontrar tentador cambiar la resolución de la GUI a 4K en lugar de 1080p. El ODRROID-N2 tiene más que suficiente potencia para ejecutar una GUI a 4K, pero yo prefiero

una experiencia más sensible y limpia, así que dejé la GUI a 1080p. Esto no afecta a la resolución de los soportes reproducidos y, por lo tanto, únicamente debes tenerlo en cuenta si quieres imágenes/posters de mayor calidad.

Por último, me dirigí a Audio y como voy a usar un AVR (receptor de audio/video), quiero que éste decodifique el audio para tener mejor calidad. Mi configuración está basada en la guía de inicio rápido de Kodi ([https://kodi.wiki/view/Audio\\_quickstart\\_guide](https://kodi.wiki/view/Audio_quickstart_guide)). ([https://kodi.wiki/view/Audio\\_quickstart\\_guide](https://kodi.wiki/view/Audio_quickstart_guide)).



## Añadir soportes multimedia

Como usuario de Plex, conectar todos mis dispositivos Kodi a mi servidor Plex fue cuestión de minutos con el complemento PlexKodiConnect (<https://github.com/croneter/PlexKodiConnect/>). Para los usuarios que no usan Plex, pueden crear y configurar la biblioteca de videos siguiendo la guía de Kodi ([https://kodi.wiki/view/HOW-TO:Create\\_Video\\_Library](https://kodi.wiki/view/HOW-TO:Create_Video_Library)).

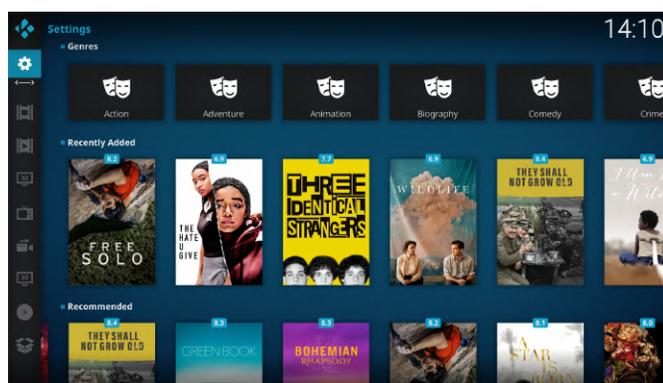


Figura 4: Biblioteca multimedia con el skin Estuary Mod V2

## Complementos

La selección de complementos ha sido durante años una de las mayores ventajas de Kodi y la gente los usa

con frecuencia, incluido yo. Hay algunos complementos que yo uso bastante:



Figura 5: Complementos

## Netflix

Visita el siguiente enlace (<https://forum.kodi.tv/showthread.php?tid=329767>) para ver las instrucciones de instalación, Netflix está listo para usarse. ODRROID-N2 es lo suficientemente potente como para ver Netflix en 1080p con sonido DD+ 5.1 con subtítulos.

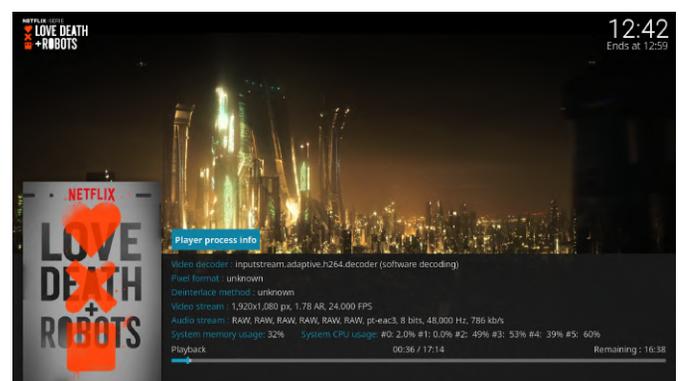
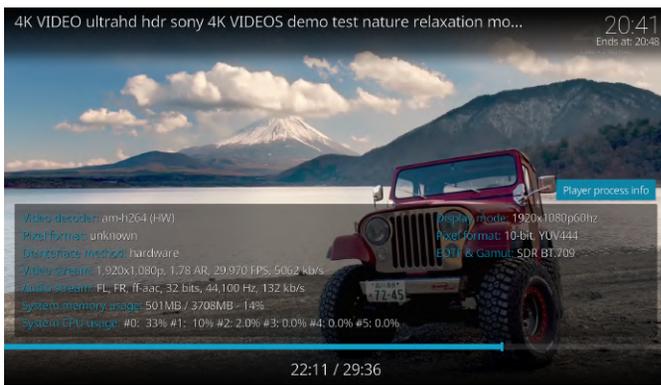


Figura 6: Netflix con 1080p y DD+ 5.1

## Youtube

Youtube se puede instalar desde el repositorio de complementos de Kodi y con el complemento InputStream Adaptive, que se encuentra en CoreELEC Add-ons, se puede configurar Youtube con reproducción 4K.



**Figura 7: Youtube en 1080p sin problemas. Lo mismo puede decirse del 4K.**

## Pruebas

Usando los videos de muestra de <https://kodi.wiki/view/Samples>, uno puede probar la mayoría de los formatos de audio y video. Resultados a julio de 2019 con passthrough habilitado y ODROID-N2 conectado a un AVR.

Audio	Estado	Video	Estado
AC-3 (DD)	Funcionando	720p and lower 23-60 Hz	Funcionando
E-AC-3 (DD+)	Funcionando	1080p 23-60 Hz	Funcionando
Dolby TrueHD	Funcionando	1080p 3D HSBS/HT AB	Funcionando
DTS	Funcionando	1080p 3D-MVC	No funciona
DTS-HD	Funcionando	2160p 23-60 Hz	Funcionando
DTS-HD MA	Funcionando	2160p 23-60 Hz HDR10	Funcionando
LPCM	No funciona, Una solución es seleccionar canales 2.0 y habilitar la transcodi	2160p 23-60 Hz HLG	Funcionando

	función AC-3			
FLAC	Funciona parcialmente, ver LPCM		2160p 23-60 Hz Dolby Vision	No funciona

Resultado de las pruebas de audio y vídeo

En resumen, ODROID-N2 es capaz de reproducir Blu-Ray Ultra HD con HDR10 y Dolby Atmos/DTS-X. Incluso los videos de prueba con 3 veces la tasa de bits de las especificaciones del Blu-Rays UHD (triple capa - 128 Mbit) se reproducen sin problemas desde un NAS.

## Usar ODROID-N2 con CoreELEC como servidor

Entware, un administrador de paquetes con más de 2000 paquetes, se puede instalar fácilmente ejecutando el script `installentware` a través de CoreELEC. Los paquetes de Entware fueron diseñados inicialmente para ejecutarse en dispositivos de baja potencia como routers y NAS y, por lo tanto, son especialmente livianos. Los paquetes se instalan con `opkg install *Nombre del paquete*` y se puede buscar paquetes con `opkg search NOMBRE` u `opkg find NOMBRE`. Como estaba interesado ejecutar mi ODROID-N2 como servidor, ejecuté los siguientes comandos:

```
$ opkg install lighttpd mc youtube-dl netdata
```

Algunos paquetes destacados que son instalados:

- Lighttpd - un servidor web de código abierto optimizado para entornos de velocidad crítica
- GNU Midnight Commander (mc) - Un administrador de archivos visual
- Youtube-dl - un programa para descargar videos de YouTube y de muchos más sitios
- Netdata - un demonio que permite monitorizar el rendimiento en tiempo real de sistemas Linux

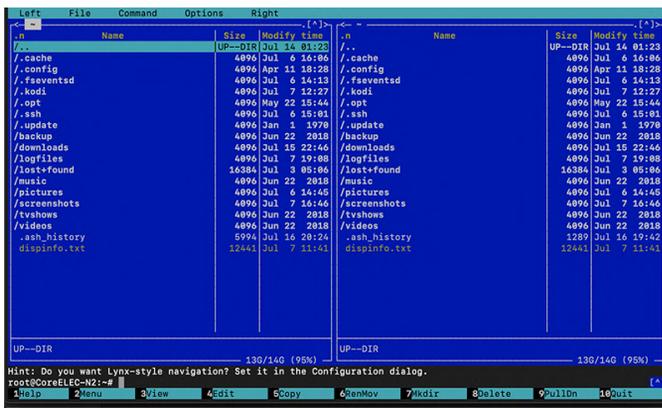


Figura 8: Midnight Commander

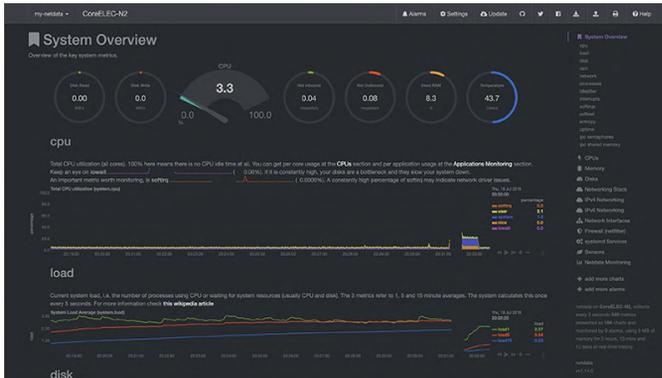


Figura 9: Netdata

## Support

Uno de los aspectos más subestimados a la hora de comprar un ordenador de placa reducida (SBC) es el soporte, el comprador no suele tener en cuenta el soporte de la SBC cuando mira el precio. El soporte de un SBC es extremadamente importante para conseguir la mejor experiencia de usuario posible y este es uno de los aspectos en el que destaca el ODROID-N2. Con un gran soporte en <https://forum.odroid.com/> y <https://discourse.coreelec.org>, se puede hacer preguntas para obtener ayuda y a menudo se le responde en cuestión de minutos o en unas pocas horas. Las personas en los foros están ansiosas por ayudar y contribuir, lo cual es importante para que la comunidad crezca y la sensibilización sobre los SBC siga aumentando.

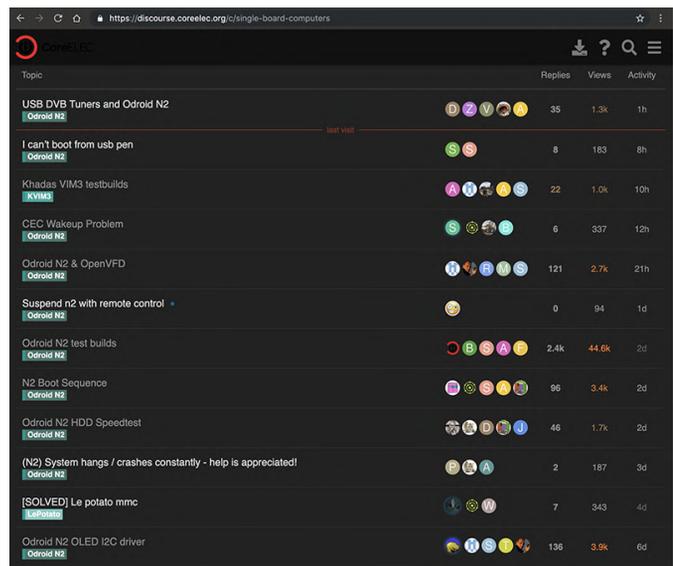


Figura 10: Foros de CoreELEC

Para obtener más información sobre cómo usar CoreELEC en el ODROID-N2, consulta la Centro multimedia CoreELEC de ODROID-N2 en <https://www.cnx-software.com/2019/08/26/odroid-n2-coreelec-edition-media-center/>.

## Nota del editor

Personalmente, tengo un ODROID-N2 con CoreELEC y he descubierto que es un reproductor multimedia rápido y sensible. Después de leer este artículo, quedará claro por qué el ODROID-N2 es un excelente dispositivo multimedia. Además, Hardkernel incluso ha creado un paquete con todo lo que necesitas para poner en marcha CoreELEC en un N2, de modo que puedes ver de primera mano lo increíble que es este dispositivo en <https://www.hardkernel.com/shop/odroid-n2-2gb-coreelec-edition/> o, para quien viva en los EE. UU., en <https://ameridroid.com/collections/new/product/s/odroid-n2-coreelec-edition>.

# Cómo Crear una Consola de Juegos Retro Monku - Parte 3: Dando las Últimas Pinceladas

© September 1, 2019 By Brian Ree Juegos, Linux, ODROID-C2



Esta es la continuación del artículo sobre la consola de juegos retro del mes pasado, donde aprendimos cómo configurar el software de una consola de juegos retro. Esta entrega te ayudará a finalizar el proyecto enseñándote, en detalle, cómo pulir tu consola de videojuegos Monku Retro 1 (ODROID-C1 +) o Monku Retro 2 (ODROID-C2). Optimizaremos el entorno de escritorio MATE Linux, conectaremos nuestro botón de control personalizado, pondremos RetroArch en modo quiosco y configuraremos boot.ini.

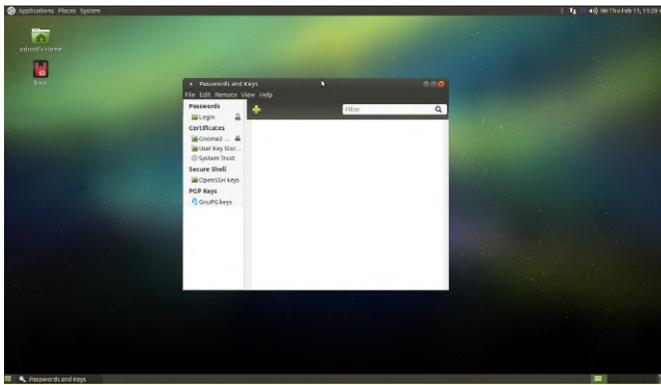
## Herramientas y Componentes

Este tutorial no requiere nuevas herramientas y componentes. Configuraremos la consola que ya has creado, ajustando las cosas para que realmente deslumbren.

## Introducción y Objetivos

### Finalizando MATE ... Casi

En esta sección, lo primero que vamos a hacer es deshacernos de esa molesta solicitud de autenticación que aparece cuando abres un navegador por primera vez después de iniciar sesión. AVISO: No estamos ante una configuración de alta seguridad: deliberadamente vamos a reducir el nivel de seguridad para facilitar su uso como consola de juegos y decodificador navegador web. Dirígete a la siguiente ubicación del menú: Applications -> Accessories -> Passwords and Keys, debería ver una ventana similar a la que se muestra a continuación. AVISO: si en algún momento se te solicita iniciar sesión cuando trabajes con el terminal, utiliza la contraseña, odroid.



**Figura 1 - Configurando los parámetros para eliminar la solicitud de autenticación**

Localiza la entrada de inicio de sesión de la lista en el lado izquierdo de la ventana. Si la ventana de autenticación que he mencionado hace referencia a un Password keychain diferente, busca esa entrada en la lista del lado izquierdo de la ventana. Sigue los siguientes pasos para desbloquear el keychain.

1. Haz clic derecho en la entrada de destino y selecciona Change Password.
2. Te pedirá que introduzcas la contraseña anterior: escribe odroid y haz clic en Continúe.
3. Ahora te pedirá que introduzcas una nueva contraseña: deja ambos campos en blanco y haz clic en Continúe.
4. Aparecerá otro cuadro de diálogo y te preguntará si estás de acuerdo en permitir que se desbloquee el keychain: haz clic en Continúe.
5. Cierra todos los cuadros de diálogo y cierra la ventana Password and Keys; estamos listos.

Lo siguiente que haremos con el entorno MATE es configurar un poco los paneles y widgets. Tú puedes hacer lo que quieras aquí. Yo simplemente te mostraré cómo configurar las cosas y por qué. En primer lugar, esperamos interactuar con este sistema, al menos parte del tiempo, con un gamepad. Funciona muy bien, pero no es un mouse. Realmente no vamos a hacer ninguna tarea de programación rigurosa en Linux, por lo que hay algunas cosas que no necesitamos. La segunda ventaja de esta configuración es que reducimos un poco la sobrecarga de memoria.

Sigue estas instrucciones para suprimir el panel inferior. Agregaremos controles al panel superior para compensar parte de la pérdida de funcionalidad, pero no agregaremos el widget de selección de

escritorio. Puede resultar un poco exagerado para nuestras necesidades. No obstante, si lo quieres conservar, no perjudica en nada. Sigue estas instrucciones para limpiar los paneles y widgets.

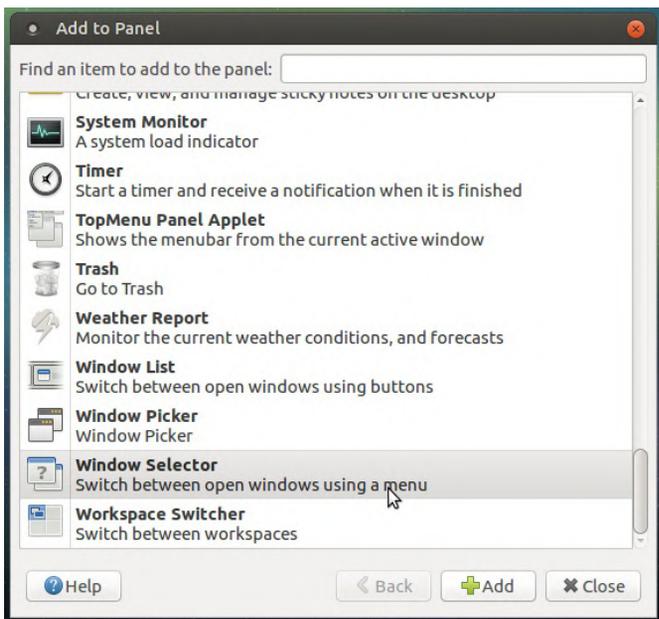
1. Dirígete al panel inferior y haz clic derecho, selecciona Delete This Panel, luego haz clic en Delete nuevamente cuando se te solicite.
2. Dirígete a la parte superior derecha de la pantalla y haz clic con el botón derecho en el botón de encendido, selecciona Remove From Panel

Lo que vamos a hacer es volver a añadir algunos widgets, haciendo que el panel superior sea un punto de control más centralizado. Esto hará que utilizar un gamepad para controlar las cosas resulte mucho más fácil. Tu escritorio debería verse como la siguiente captura de pantalla.



**Figura 2: Tu nuevo panel superior más centralizado**

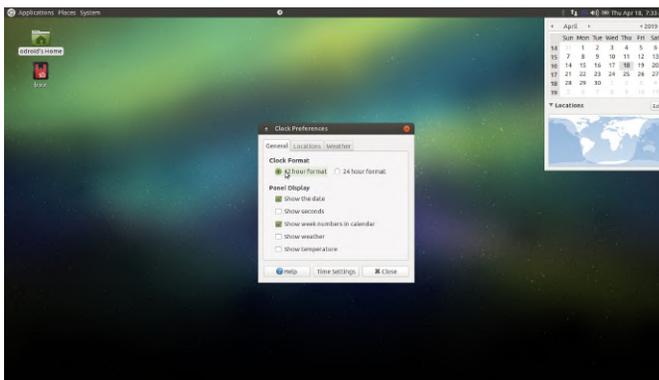
Como eliminamos los botones de selección de ventana abierta cuando suprimimos el panel inferior, vamos a añadir un nuevo widget al panel superior que sea más apropiado para un gamepad y que requiera menos movimiento de cursor para utilizarlo. Haz clic derecho en el panel superior y selecciona Add to Panel. Desplázate hacia abajo en la lista de opciones hasta que veas la entrada que se muestra a continuación, luego haz clic en Add.



**Figura 3 - Lista de opciones del panel**

Tenemos que realizar algunos pasos más con respecto a las aplicaciones de la bandeja y la configuración de fecha y hora, pero casi hemos terminado con este punto.

A continuación, haz clic en la fecha y hora en la esquina superior derecha. Debería aparecer un menú desplegable con el calendario. Expande la sección de Ubucaciones y haz clic en el botón Add. Configura la pestaña General como se muestra a continuación o como mejor te parezca



**Figura 4: Una posible opción de configuración**

Agreguemos información de la ubicación para que la hora sea la correcta cuando tengamos una conexión a Internet y la sincronicemos con el protocolo de hora en red (NTP). Haz clic en la pestaña Locations y luego haz clic en el botón Add.

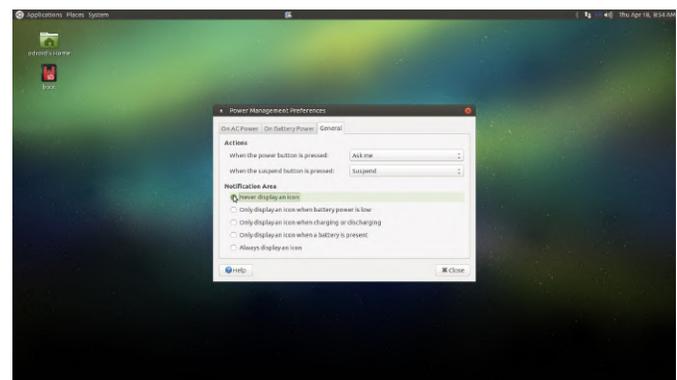
Empieza escribiendo la ciudad principal que tienes más cerca en el cuadro de texto Name. Si esto no funciona, prueba con otra ciudad importante o representativa de tu zona horaria (por ejemplo, en mi

caso es Nueva York). Selecciona una ubicación de la lista que aparece. Haz clic en Ok una vez que hayas encontrado la adecuada. Ahora verás una entrada de ubicación en la lista de ubicaciones tal y como se muestra a continuación.

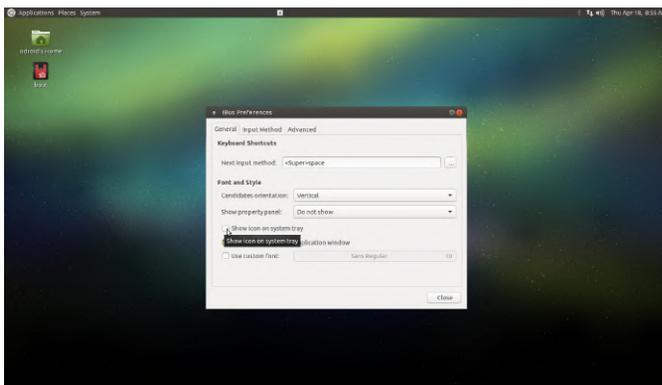


**Figura 5: Tu ubicación añadida a la lista de ubicaciones**

Solo quedan unas cuantas cosillas más por hacer aquí. Haz clic derecho en el icono de la batería en la bandeja superior derecha. Selecciona la opción Preferences. Haz clic en la pestaña General y desmarca Never Display An Icon. Si tiene una cadena EN o UK en la bandeja del sistema, haz clic derecho sobre ella y selecciona Preferencias. En la pestaña General, desmarca Show Icon On System Tray. Realmente no nos preocupa el tener que cambiar el idioma del teclado. Si lo necesitas, puede volver a activar el icono de la bandeja del sistema utilizando la opción de menú System -> Control Center y haciendo clic en Power Management and iBus Preferences respectivamente. Las siguientes capturas de pantalla muestran las cuestiones que acabamos de tratar



**Figura 6 y 7: Selecciona tus preferencias**



**Figura 6 y 7: Selecciona tus preferencias**

Solo nos quedan dos cositas en esta sección, luego pasaremos al botón de control personalizado y a los scripts. Mueve el ratón al panel superior y haz clic derecho. Selecciona Add to Panel, luego desplázate hacia abajo hasta que veas la opción Show Desktop como se muestra a continuación. Haz lo mismo para la opción Trash



**Figura 8 - Mostrar escritorio**

Usa el botón central del ratón, o la rueda del ratón, para mover los íconos del widget del panel superior. Arrastramos estos dos nuevos widgets un poco más cerca del menú Systems, y sepáralos un poco. Ahora estamos listos para empezar a añadir scripts personalizados. Estos scripts iniciarán automáticamente RetroArch durante el arranque e iniciarán AntiMicro cuando RetroArch se cierre, devolviendo el control del ratón al gamepad. También haremos algo de magia con el botón de control personalizado.

### Scripts y Botón de Control Personalizado

En esta sección, vamos a configurar algunos scripts personalizados para controlar el software que hemos instalado y configurado. Esto hará que pasemos de una experiencia de escritorio de Linux a una experiencia más de consola de juegos. Descárgate el

paquete del script a tu dispositivo y luego cópialo y pégalo en la carpeta install\_zips que creaste en el segundo tutorial. Se encuentra en el directorio de inicio del usuario ODRROID.

- [Scripts Monku R1 / C1+](#)
- [Scripts Monku R2 / C2](#)

Una vez que haya descargado y copiado el archivo zip en la carpeta install\_zips, haz clic derecho y selecciona Extract Here. Once archivos deberían aparecer en un subdirectorio. Ábrelo y cópialos todos en el directorio de inicio del usuario ODRROID. El directorio de inicio del usuario ODRROID es la ubicación por defecto del administrador de archivos. También puede acceder a él desde un acceso directo que hay e el escritorio, o desde el panel izquierdo del administrador de archivos donde se enumeran los accesos directos a carpetas.

Nos aseguraremos de que estos scripts tengan los permisos correctos y se puedan ejecutar. Abre un terminal, Applications -> System Tools -> MATE Terminal, y ejecuta los siguientes comandos.

```
$ sudo chmod 755 power_btn power_btn_test
pwrbuttonsvc.service restart_now shutdown_now
start_am start_antimicro start_auto start_pwr
start_ra stop_auto
```

```
$ sudo chmod +x power_btn power_btn_test
pwrbuttonsvc.service restart_now shutdown_now
start_am start_antimicro start_auto start_pwr
start_ra stop_auto
```

Ahora cierra el terminal y dirígete a System -> Control Center en los menús. Busca y selecciona la opción Startup Applications. Debería ver algo similar a lo que se muestra a continuación.



**Figura 9 - Panel de control del sistema**

Haz clic en el botón Add y completa el formulario que se muestra a continuación. Aquí tienes el texto exacto también.

Name: Start RetroArch Path: /home/odroid/start\_ra  
Description: Launches RetroArch on startup.

Ahora, también queremos iniciar un script especial AntiMicro. Así que, hacemos lo mismo para este script, que también se muestra a continuación. Aquí tienes los valores utilizados.

Name: Start AntiMicro Path: /home/odroid/start\_am  
Description: Launches AntiMicro on startup.

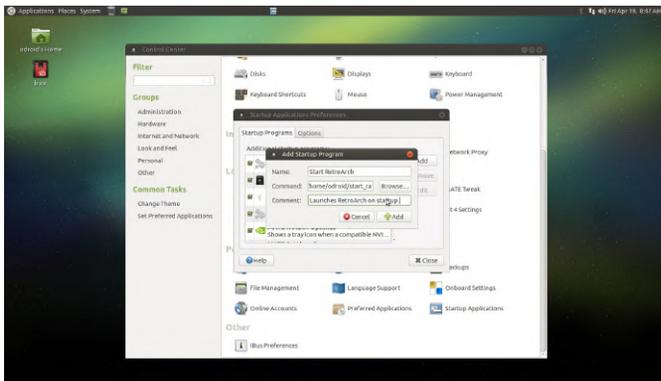


Figura 10 - Iniciar RetroArch

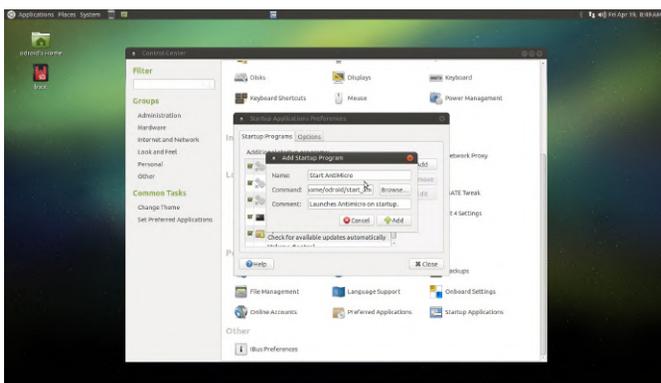


Figura 11 - Iniciar AntiMicro

Apaga el dispositivo: System -> Shut Down. Usa el botón de reinicio de hardware para volver a encenderlo. Deberías observar que RetroArch se inicia automáticamente. Ahora, si cierras RetroArch y esperas unos 5 segundos, deberías ver la ventana de AntiMicro en la bandeja del sistema y que se reanuda el control del dispositivo por gamepad.

Las cosas se están desarrollando bien. Nuestro magnífico ODROID se parece cada vez más a una gran consola de juegos retro. Finalmente añadiremos soporte para el botón de control personalizado. Antes de conectarlo al sistema, debemos probarlo. Dirígete al terminal: Applications -> System Tools -> MATE

Terminal y escribe sudo ./power\_btn\_test y luego presiona INTRO. Puede que recibas un error de exportación de PIN GPIO si lo ejecuta dos veces seguidas. Puedes ignorar este error; La prueba seguirá funcionando. Debería ver una serie de 1 desplazándose a lo largo del terminal si está utilizando un C2, ceros si es un C1+. Mantén presionado el botón de control personalizado y observa qué sucede. Asegúrate de no presionar el botón de reinicio de hardware por accidente o de lo contrario tendrá que esperar a un reinicio. Deberías ver algo como lo que se muestra a continuación.



Figura 12 - Probando el botón de encendido

Para el C1+, los 0 y 1 están invertidos, tal y como lo configuré por causalidad. La imagen de arriba muestra la prueba del C2. Si no logras el mismo resultado, apaga el dispositivo, retira la carcasa - recuerda, decidimos no cerrarla, y comprueba las posiciones del conector en el cabezal GPIO. Si funciona, ¡genial! Intenta mantener presionado el botón durante diferentes períodos de tiempo. En el terminal aparecerá el script que se ejecutará durante esa duración. A continuación, tienes una lista de funciones que configuraremos.

Funciones del botón de software: - 02 Segundos presionado: reinicio de software. - 04 04 Segundos presionado: apagado del software. - 06 Segundos presionado: Desactivar el modo kiosco del juego. - 08 Segundos presionado: cambiar a una resolución de 1024x768x32bpp y reiniciar. - 10 segundos presionado: cambiar a una resolución de 720px32bpp y reiniciar.

Ahora que hemos probado la funcionalidad del botón de control personalizado, vamos a añadirlo al sistema como un servicio. A veces, dependiendo de la tarjeta SD y algunas otras cosas, el dispositivo puede llegar a

funcionar mal. Este botón de control de software es realmente útil para obligar al sistema a que se reinicie limpiamente de forma segura y controlada. Abre un terminal y ejecuta los siguientes comandos. Estos registrarán, iniciarán y marcarán nuestro servicio para que se inicie automáticamente durante el arranque. Abre un terminal - Applications -> System Tools -> MATE Terminal - y ejecuta los siguientes comandos. Resiste el impulso de intentar probar las funciones de 6, 8 y 10 segundos. No vamos a experimentar todavía con el boot.ini.

```
$ sudo cp pwrbuttonsvc.service /etc/systemd/system
$ sudo cp ./power_btn /usr/bin/power_btn_test
$ sudo systemctl start pwrbuttonsvc
$ sudo systemctl enable pwrbuttonsvc
```

Si alguna vez necesitas detener el servicio, puedes usar este comando, aunque no lo ejecutes ahora.

```
$ sudo systemctl stop pwrbuttonsvc
```

¿Estás listo? Mantén presionado el botón de control personalizado durante 4 segundos y un poquito más. Puedes practicar con el script de prueba que ejecutamos antes. El sistema debería apagarse muy rápido y, si has contado correctamente, no debería arrancar. Si lo has mantenido durante 2 segundos menos, hará lo mismo, pero luego volverá a arrancar.

## Finalizando MATE ... Esta vez sí

Espero que hasta ahora estés realmente disfrutando de todo. De verás que es genial finalmente conectar el botón y hacer que, literalmente, sea parte del sistema operativo. Tenemos una cosa más que hacer con MATE, pero es algo rápida, lo prometo. Haz clic con el botón derecho en el panel superior y selecciona Add to Panel, luego desplázate hacia abajo por la lista de opciones hasta encontrar la entrada Shut Down. Haz clic en Add y usa el botón central del ratón o la rueda del ratón para coger el nuevo widget y colocarlo de manera que esté aproximadamente a una pulgada del widget de programas abiertos. Al mantener todos los controles en un grupo reducido, mejoramos en gran medida la experiencia del usuario a la hora de controlar las cosas con el gamepad.

A continuación, añadiremos dos botones personalizados al panel superior. Haz clic con el botón derecho en el panel y selecciona Add to Panel. Elige la primera opción, Custom Application Launcher. Agregaremos primero el botón de parada y luego el botón de inicio. A continuación, tienes los valores de los campos de formulario. Usa las capturas de pantalla para navegar al icono adecuado. Puedes ver la ruta en la captura de pantalla cerca de la parte superior de la ventana.

Valores del botón de parada:

```
Type: Application
Name: Stop RetroArch
Command: /home/odroid/stop_auto
Comment: Stops RetroArch if running windowed.
(Really just stops RetroArch and resets AntiMicro)
```

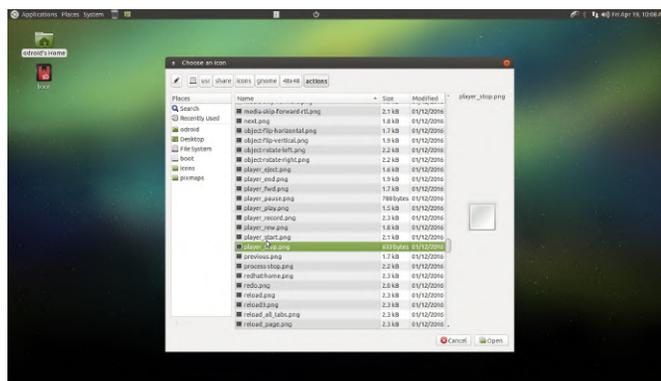
Valores del botón de inicio:

```
Type: Application
Name: Start RetroArch
Command: /home/odroid/start_auto
Comment: Starts RetroArch and AntiMicro scripts.
```

Las siguientes capturas de pantalla describen este paso. Úsalas para ayudar a encontrar el icono correcto si fuera necesario.



Figura 13, 14 y 15: configuración de aplicaciones personalizadas





Probemos los nuevos controles. Cierra RetroArch y cualquier otra ventana que tengas abierta. Haz clic en el botón de inicio y debería ver aparecer RetroArch. Haz clic en el botón de parada y RetroArch se cerrará. Espera unos 5 segundos, y debería ver AntiMicro en la bandeja del sistema que nos devuelve el control total del gamepad.

¡Caray!, es realmente genial. Hemos personalizado completamente nuestro hardware y software ODRROID-GO para crear una consola de juegos retro con el modo de quiosco RetroArch y un completo entorno Linux, si fuera necesario. Lo siguiente que vamos a hacer es ejecutar RetroArch en modo de pantalla completa y ajustar algunos parametros de video. No me detendré en ninguna configuración avanzada. Este tutorial tiene la duración que me gusta para hacer las cosas, dedicare más adelante un pequeño tutorial de seguimiento a las erratas del emulador y a la configuración avanzada de RetroArch.

¡Inicia RetroArch, y podrás usar el pequeño widget que acaba de crear! Desplázate hacia la derecha hasta la sección de Drivers y localiza la entrada Video



**Figura 16: Desplázate hacia la derecha hasta la sección Drivers y encuentra la entrada Video**

Aplica las configuraciones que se enumeran a continuación en el orden en que aparecen. La

aplicación se puede cerrar y volver a abrir para algunas de las configuraciones; es normal.

Windowed Fullscreen Mode: Off  
 Show Window Decorations: Off  
 Threaded Video: On  
 Bilinear Filtering: Off  
 Start in Fullscreen Mode: On

Usa la tecla ESC para cerrar RetroArch cuando esté en modo de pantalla completa o utiliza el teclado/ratón para navegar hasta la sección del Menú principal y seleccionar Quit RetroArch.

Mantén presionado el botón de control personalizado durante 2 segundos y un poco más y el sistema se reiniciará. Cuando vuelva iniciarse, deberías ver una pantalla completa de RetroArch como se muestra a continuación. Desplázate hasta tus ROM con el gamepad y arranca uno. ¡juego activado!



**Figura 17 - Pantalla completa de RetroArch**

### Actualizando boot.ini

En este apartado, lo primero que haremos es hacer una copia de seguridad del archivo boot.ini. Abre el icono de inicio en el escritorio y copia boot.ini en boot.ini.orig. También vamos a hacer dos copias más: una que llamaremos boot.ini.1024x768p32bppVga y otra que llamaremos boot.ini.1280x720p32bppHdmi. Vamos a editarlos en breve. Deberías tener algo similar a lo que se muestra a continuación.



**Figura 18 - Copia boot.ini en boot.ini.orig**

Bajo mi experiencia, estas configuraciones de video funcionan bastante bien. La resolución VGA de 1024x768 es compatible con la mayoría, si no todas, de las pantallas recientes de ordenador, y la resolución de 720p es compatible con la mayoría, si no todos, los televisores recientes. Aunque el ODRROID-C2 tiene notablemente más potencia, realmente no necesitamos más de 720p para mostrar juegos de 8 bits y 16 bits. Por supuesto, puedes seleccionar la que mejor te parezca. La idea es que, si conectamos nuestro dispositivo a un televisor, podamos mantener presionado el botón de control personalizado durante el tiempo necesario para que el dispositivo se reinicie con la resolución HDMI adecuada y eficiente en memoria. Si nos llevamos el dispositivo al trabajo y queremos jugar algunos juegos durante el almuerzo, podamos mantener presionado el botón de control personalizado durante el tiempo necesario para que el dispositivo se reinicie con la resolución VGA adecuada.

Arranquemos nuestro dispositivo ODRROID y cerremos RetroArch presionando la tecla escape en el teclado o navegando hasta la opción de salida usando el mando. Applications -> System Tools -> MATE Terminal, y escribe el siguiente comando.

```
$ nano power_btn
```

Desplázate hasta la parte inferior del archivo y edita el texto de modo que coincida con la imagen que se muestra a continuación. Una vez que hayas modificado el archivo, presiona Ctrl + O para sobrescribir el archivo y Ctrl + X para cerrar el editor nano. Este es el texto que debemos introducir:

```
$ elif [ $count -ge 8 ] && [ $count -lt 10 ] && [ $GPIO_VALUE -eq 1 ]; then
```

```
count=0
$ sudo cp /media/boot/boot.ini.1024x768p32bppVga
/media/boot/boot.ini
$ sudo shutdown -r now
```

```
#killall retroarch
#cp /home/odroid/Scripts/retroarch.cfg.usb
/home/odroid/.config/retroarch/
#/home/odroid/start_ra
```

```
$ elif [ $count -ge 10 ] && [ $GPIO_VALUE -eq 1 ];
then
```

```
$ count=0
$ sudo cp /media/boot/boot.ini.1280x720p32bppHdmi
/media/boot/boot.ini
$ sudo shutdown -r now
```

```
#killall retroarch
#cp /home/odroid/Scripts/retroarch.cfg.hdmi
/home/odroid/.config/retroarch/
#/home/odroid/start_ra
```

```
$ fi
```

Ten en cuenta que para el C1+ el texto anterior será ligeramente diferente. Éste utiliza comparaciones [ $\$GPIO\_VALUE -eq 0$ ], aunque en realidad solo nos interesan los comandos de copia de archivos y los comandos comentados. No debes cambiar nada más. A continuación, ejecuta el siguiente comando en el terminal para activar el script.

```
$ sudo cp ./power_btn /usr/bin/power_btn_test
```

Una vez hecho esto, ajustaremos el archivo boot.ini copiado que creaste para dejar constancia del resultado y la resolución correctos.

Básicamente, copiaremos el archivo boot.ini con una versión preconfigurada que fija una determinada resolución de pantalla, luego reiniciaremos el dispositivo. Detallo los cambios que necesitamos hacer en cada archivo para el C1+ y el C2, aunque también proporcionaré una descarga para que las cosas te resulten un poco más fáciles. Centrémonos en el C2 primero.

Abre un terminal y escribe los siguientes comandos. Primero haremos el modo VGA 1024x768.

```
$ cd /media/boot/
$ nano boot.ini.1024x768p32bppVga
```

Queremos configurar la salida de video para que sea 1024x768 VGA. Comenta las líneas que aparece a continuación. Asegúrate de que no haya modos de salida de video no comentados, excepto el que queremos.

```
# setenv display_autodetect "true"
# setenv m "1080p60hz" # Progressive 60Hz
```

Asegúrate de que las siguientes líneas no estén comentadas. Si cometes un error, simplemente restaura la copia boot.ini.orig que realizaste anteriormente. Con cualquier ordenador con Windows o Mac podrás acceder a la partición de arranque de la tarjeta SD de tu ODRROID porque es una partición Fat32. Puedes recurrir a esto para reemplazar tu boot.ini si el dispositivo no se inicia correctamente.

```
$ setenv m "1024x768p60hz"
$ setenv vout "vga"
```

A continuación, fijaremos el modo HDMI 720p. Abre un terminal y escribe los siguientes comandos:

```
$ cd /media/boot/
$ nano boot.ini.1280x720p32bppHdmi
```

Queremos configurar la salida de video a 720p HDMI. Comenta las líneas que se enumeran a continuación. Asegúrate de que no haya modos de salida de video no comentados, excepto el que queremos.

```
# setenv display_autodetect "true"
# setenv m "1080p60hz" # Progressive 60Hz
```

Asegurate de que la siguiente línea no esté comentada.

```
$ setenv m "720p60hz" # 60Hz
```

AVISO: Muchas pantallas de ordenador no admiten 720p. Si este es tu caso, es probable que termines arrancando en una pantalla en blanco. Simplemente espera un minuto o dos y luego usa el botón de control personalizado para cambiar el modo de video a 1024x768. Se necesita un poco de tiempo para habituarse, pero una vez que te acostumbras, puedes cambiar de modo con bastante facilidad. A continuación, se muestra una captura de pantalla de un Monku Retro 2 (ODROID-C2) que se ejecuta en

modo VGA en una pantalla de ordenador después de cerrar RetroArch.



Figura 19 - Monku R1 Build 36

Y aquí está el R2 ejecutándose a 720p en un televisor.



Figura 20 - El R2 se ejecuta a 720p en un televisor

Como cualquier buen programa de cocina de televisión, ya tenemos un plato preparado para mostrar. Para el C1+, sigue las mismas pautas anteriores junto con los archivos que se proporcionan a continuación. El proceso es similar, los archivos proporcionados indican los cambios que debes realizar. O puedes simplemente copiarlos y pegarlos en tu directorio /media/boot, realmente depende de ti.

- Monku R1 / C1+ boot.ini - Monku R2 / C2 boot.ini

### Terminando

Cerrar la carcasa es bastante simple, aunque lo vamos a repasar de todos modos. No es muy divertido terminarlo todo, cerrar la caja y apretar los tornillos exteriores y acordarse de no haber colocado

los tornillos interiores, confía en mí. Lo primero que querrás hacer es organizar tus conectores. Pueden enredarse un poco, así que generalmente yo los rizo y los doblo.

Recomiendo encarecidamente recortar la ranura de la tarjeta SD porque nos aporta una gran ventaja a la hora de poder cambiar o retirar fácilmente las tarjetas SD, etc. Yo utilicé una cuchilla de afeitar para cortar todas las pequeñas pestañas de plástico que la sujetaban. Dos están cerca del borde de la carcasa, son fáciles de acceder y es seguro cortarlas con un poco de presión. Después de un rato, lograrás recortar las dos pestañas que están cerca del borde de la caja. Luego dobla la puerta hacia arriba para torcer las pestañas restantes. Consejo: debilita las pestañas tanto como pueda rascándolas con la cuchilla de afeitar.

Monta la placa en la carcasa, coloca y aprieta los dos tornillos que están en un lateral de la carcasa. Los tornillos restantes se colocan desde el exterior de la carcasa cerrada. Antes de cerrarla, intente ver si combina fácilmente. Puede que tenga que reajustar algunos conectores para que ajuste bien la carcasa. Encaja la carcasa, coloca y aprieta los tornillos exteriores, y ya está todo listo.



**Figura 21 - El resultado final**

Espero que te hayas divertido con todo este proceso de montar y configurar tu propia consola de juegos retro desde cero. Yo lo hice. Hablare de más cosas en un tutorial de seguimiento en el que tocaré algunos temas avanzados relacionados con RetroArch y la configuración del emulador. Hasta entonces, ¡disfruta jugando!

Para comentarios, preguntas y sugerencias, visita el artículo original en [http://middlemind.com/tutorials/odroid\\_go/mr1\\_build\\_fn.html](http://middlemind.com/tutorials/odroid_go/mr1_build_fn.html).

# Conexión de Video Compuesto ODROID-N2: Usa tu Antiguo Monitor con el Ordenador más Potente de Hardkernel

© September 1, 2019 By Dongjin Kim ODROID-N2, Mecaniquero



El video compuesto es un sistema de transmisión de vídeo analógico que fue muy popular antes de la era de la televisión digital, con resoluciones compatibles 480i y 576i. El ODROID-N2 admite esta función de vídeo y está diseñado para conectarse a un televisor a través de su conector de audio.

## Conexión de hardware

La señal de video Composite Video Blanking and Sync (CVBS) está asignada al conector de audio (CON6) y la TV se puede conectar con un cable específico que tiene disponible Hardkernel en <https://www.hardkernel.com/shop/3-5mm-male-plug-to-3-rca-female-audio-video-cable/>.

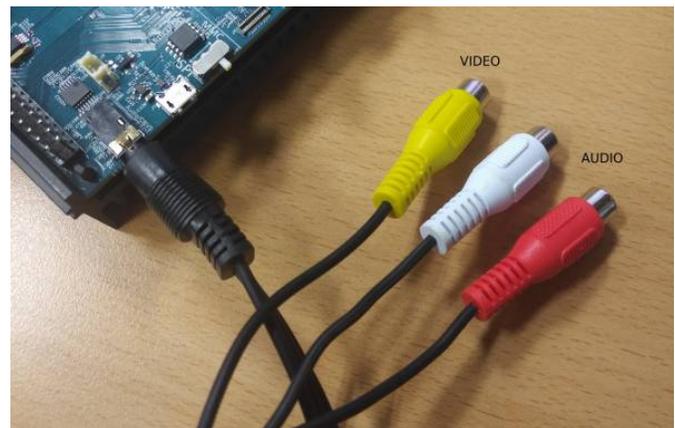


Figura 1 - Cable de video compuesto conectado al ODROID-N2

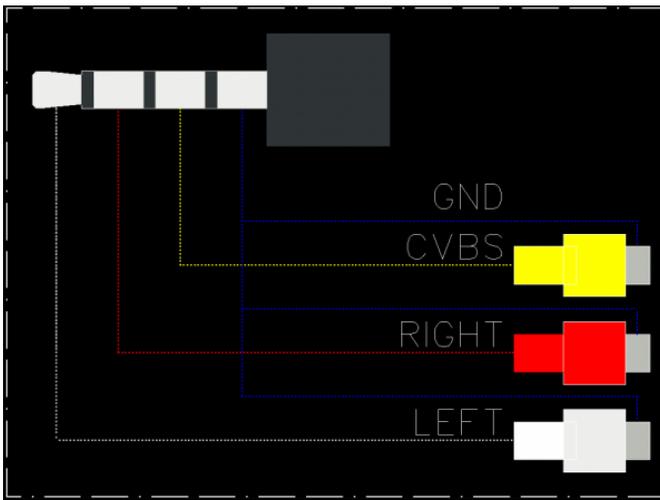


Figura 2 - Esquema de la conexión del cable de video compuesto



Figura 3: Conexión de TV con cables RCA macho a macho



Figura 4: el video compuesto con un monitor antiguo te permite disfrutar de los juegos retro tal y como era originalmente

## Configuración del software

Para seleccionar CVBS como salida de video en lugar de HDMI, la línea de comando del kernel de Linux debe configurarse correctamente y el cable HDMI no debe estar conectado. El formato de imagen CVBS se puede configurar en el archivo boot.ini:

```
# NTSC
setenv cvbsmode "480cvbs"
# PAL
#setenv cvbsmode "576cvbs"
```

```
setenv bootargs "${bootargs} cvbsmode=${cvbsmode}
cvbscable=${cvbscable}"
```

Por ejemplo, si el televisor admite el formato de imagen 480CVBS y tu cable está conectado, la línea de comando del kernel de Linux debería ser la siguiente:

```
$ cat /proc/cmdline
console=ttyS0,115200n8 ... cvbsmode=480cvbs
cvbscable=1 ...
```

## Overscan (Android)

El sistema operativo Android soporta un método de zoom acercamiento/alejamiento para administrar el overscan. Puedes ajustarlo a través de la aplicación de Android ODROID Settings, o editando directamente el archivo boot.ini. El valor del zoom está entre el 80% y el 100%:

```
# Display Zoom Rate
setenv zoom_rate "100"

# Shrink the picture as 10%
setenv overscan=10

setenv bootargs "${bootargs} overscan=${overscan}"
```

Para comentarios, preguntas y sugerencias, visita el artículo original de Wiki en [https://wiki.odroid.com/odroid-n2/application\\_note/cvbs](https://wiki.odroid.com/odroid-n2/application_note/cvbs). Para ver el ODROID-N2 usando una conexión de video compuesto en acción, visita <https://www.youtube.com/watch?v=Uk8T1s6ufnM>.

# El Punto G: Tu Destino para Todas las Cuestiones Relacionadas con Juegos Android

September 1, 2019 By Dave Prochnow Android, Juegos, ODROID-N2, ODROID-XU4



La noticia más destacada del mundo de los juegos de agosto viene de la mano del lanzamiento del 29 de julio de Gigantic X para Android. Como recordarán, este shooter de ciencia ficción de arriba hacia abajo prometió su lanzamiento hace varios meses. Pues bien, la espera ha terminado. Un gran festín de verano para todos los fanáticos de los shooter



**Figura 1 - Un auténtico existo de verano: Gigantic X**

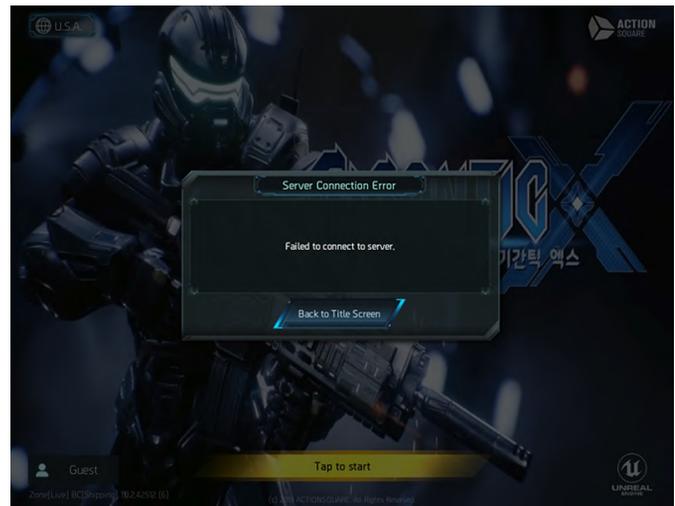
Durante el juego, lucharás contra parásitos alienígenas a lo largo de toda la galaxia mientras buscas todo el botín que puedes desbloquear para actualizar tus armas. Como bono especial, existe un modo de jugador cooperativo que te permite que tus amigos te ayuden a eliminar todos estos molestos

parásitos. Como ventaja añadida, los desarrolladores han agregado una serie de recompensas especiales para el día del lanzamiento y que los jugadores pueden ganar. Así que prepárate para Google Play.

GLOBAL RELEASE		
Goal	Reward 1	Reward 2
Squad Creation	 100	-
Squad Level 10	 100	Revive Coin x5
Squad Level 20	 100	1 Day Megacorp Doubler
Squad Level 30	 100	1 Day Frontline Doubler

**Figura 2: algunas de las recompensas especiales del día de lanzamiento que puedes ganar. Imagen cortesía de Action Square.**

Ten en cuenta que, si quieres jugar a Gigantic X en una tablet Android sin conexión WiFi, te sentirá decepcionado a menos que puedas conectarte a la red durante el juego. Desafortunadamente, el desarrollador del juego, Action Square, requiere la descarga/instalación de un parche de la aplicación de unos 40-100MB antes de empezar el juego, ¡cada vez, Arrgh! ¿Por qué, Action Square, por qué?



**Figura 3: sin conexión al servidor, ¡no hay juego!**

<https://www.facebook.com/GiganticX/>



¡Free Africa! Bueno, así que los shoot-em-ups no son tu estilo. Si la estrategia adictiva va más contigo, entonces te encantará lo que el desarrollador Łukasz Jakowski ha hecho con Age of Civilisations: Africa.

Este tradicional juego de estrategia por turnos sigue teniendo el mismo objetivo: la total dominación del continente de África. En tu intento de conquistar África, debes hacer frente a 235 civilizaciones junto con 436 provincias que están envueltas en cinco escenarios de juego. La sorpresa para este verano es que Age of Civilisations: Africa ahora es GRATIS. De acuerdo, este título ha sido una descarga gratuita durante varios meses, pero ¿qué mejor manera de combatir el calor del verano que sentarse al lado del aire acondicionado, tratando de conquistar uno de los continentes más cálidos del mundo?

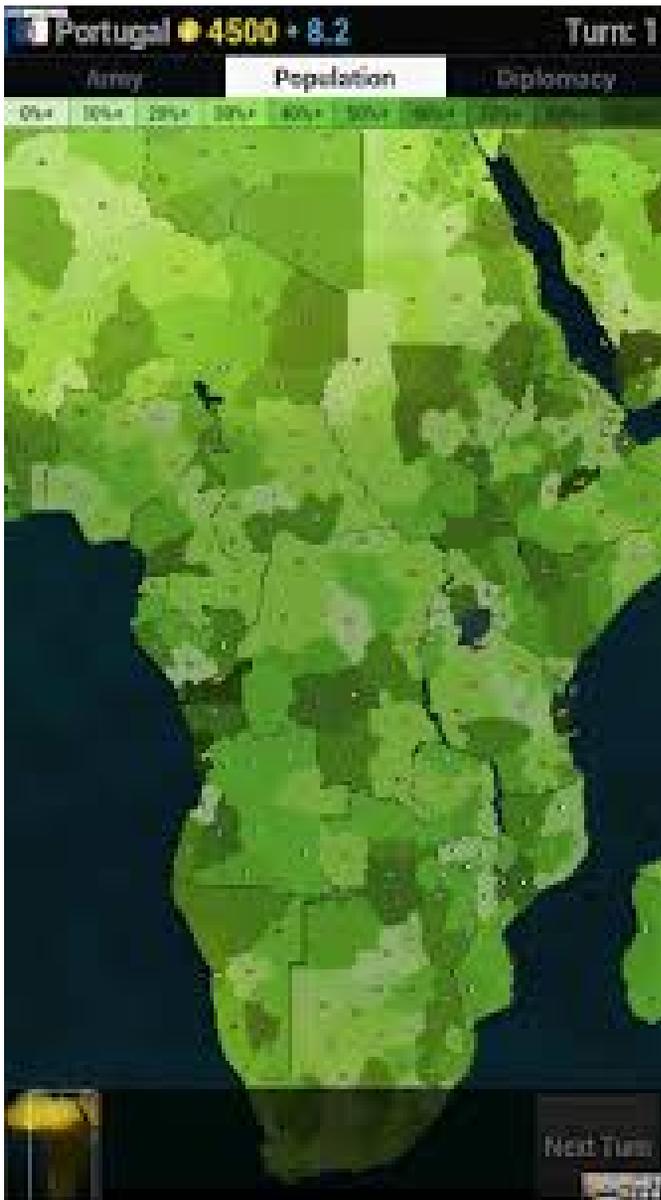


Figura 4: Puedes apropiarte de muchos bienes inmuebles en este juego

Save 5\$ Otro popular desarrollador de juegos que te pude dar algo para combatir el calor es Light Wave Games, el mismo equipo que te trajo Strike Team Hydra, con su título Demon's Rise 2. Esta aplicación de Android de 5.99\$ ahora está a la venta por solo 0.99\$. Antes de gastar tus USD, recuerda que este es un juego con un toque diferente. En este juego de rol eres un comandante de las fuerzas oscuras de las bestias, los bárbaros y, por supuesto, los demonios. Tu habilidad para manejar el fuego del infierno te ayudará a calcinar todos los mundos civilizados que habitan en este juego. En efecto, No más Mister Nice Guy.

Demon's Rise 2 - Lords Of Chaos - iOS ...

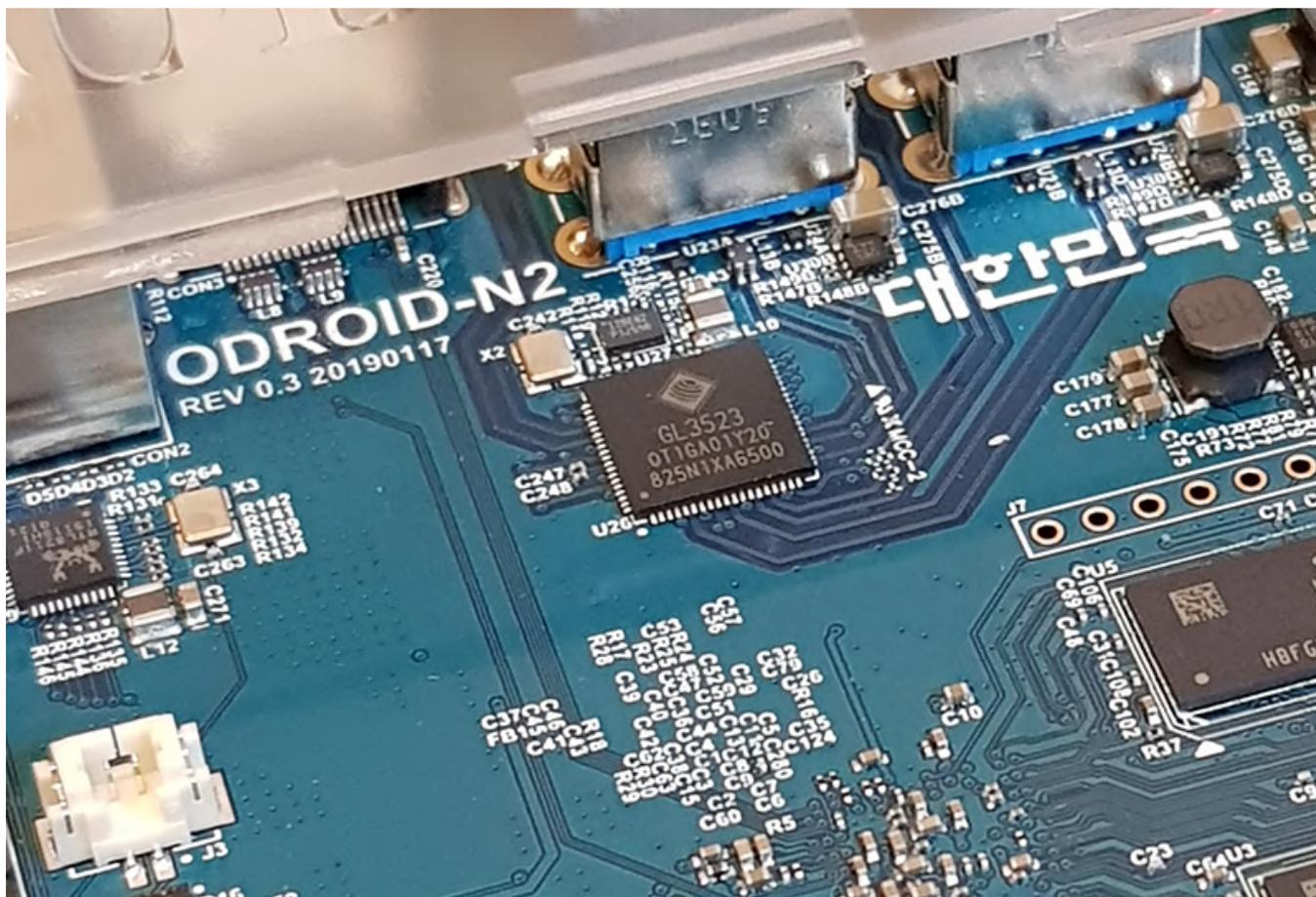


Grandes juegos para paliar un caluroso verano

Majotori (Ahora GRATIS) Venus Eleven Traitors Empire Card DOOM & DOOM II (Ahora cada uno 4.99\$ en Google Play.)

# Análisis del ODROID-N2

© September 1, 2019 By Carlos Eduardo de Paula (carlosedp.com) ⇨ ODROID-N2



Hace poco he recibido un SBC ODROID-N2 de Hardkernel, que es una nueva placa que reemplaza el anterior ODROID-N, el cual fue cancelado. Compré la placa ODROID-N2, la fuente de alimentación, una carcasa transparente, un adaptador USB WiFi y una tarjeta eMMC de 32GB, ya que los módulos eMMC son mucho más rápidos que las tarjetas SD.



Figura 01 - El ODROID-N2

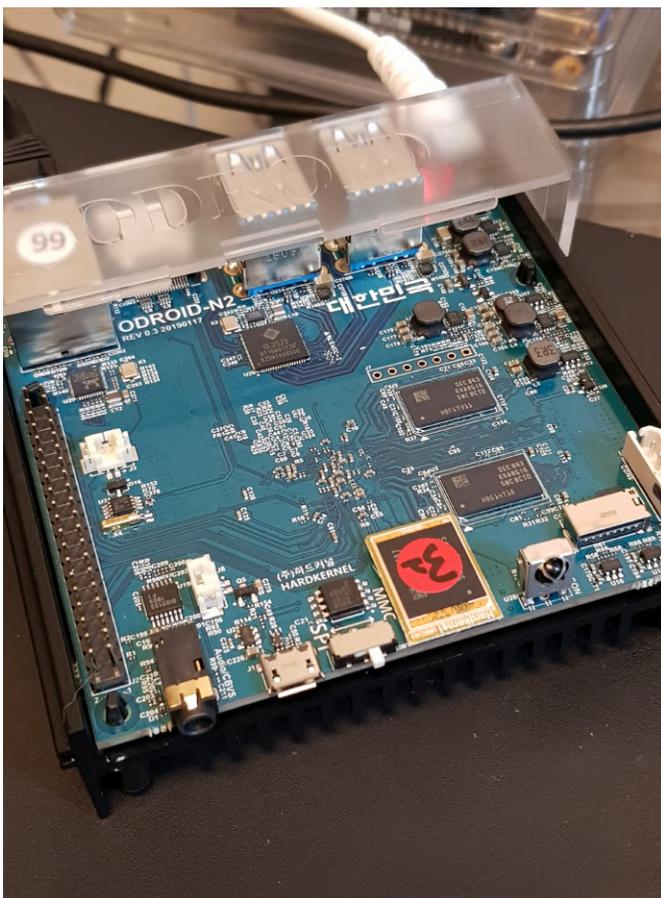


Figura 02

Lo bueno del ODROID-N2 es que utiliza un SOC diferente, un Amlogic S922X, que ofrece una nueva perspectiva en comparación con la mayoría de las placas RK3399 de gama alta que vemos en la actualidad.

Algunas especificaciones de la placa:

- CPU Amlogic S922X Hexa-core con cuatro núcleos ARM Cortex-A73 y dos núcleos Cortex-A53
- 4 GB de RAM DDR4
- Ethernet de 1Gbps
- 4 puertos USB 3.0 (Hub USB 3.0 tras un único puerto USB 3.0 del SOC)

Puedes encontrar más detalles sobre las especificaciones

en <https://www.hardkernel.com/shop/odroid-n2-with-4gbyte-ram/>. Mis pruebas siempre han estado centradas en cargas de trabajo a nivel de servidor y de consola. Hay muchas pruebas de rendimiento en Youtube y otros blogs donde se ejecutan juegos sobre escritorios de Android o Linux.

Ahora con las últimas características de Docker, que te permite compilar imágenes ARM tan fáciles como para x86 (tal y como describí en <https://bit.ly/321J80b>), prácticamente no hay inconvenientes para usar un SBC ARM para cubrir necesidades a nivel de servidor. Espero que Hardkernel y Amlogic envíen los parches para dar soporte a esta placa. Deberás usar el propio Kernel de Hardkernel

en <https://github.com/hardkernel/linux>. Tienes más detalles en su wiki en <https://wiki.odroid.com/odroid-n2/odroid-n2>.

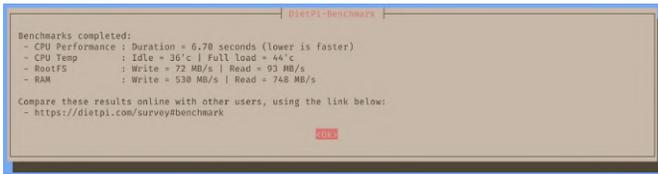
Lo primero que hice fue instalar DietPi, una distribución ligera de Linux basada en Debian. Ya tiene una imagen para el ODROID-N2. Simplemente la descargue, desempaquete el archivo y lo grabé en el módulo eMMC usando Balena Etcher (<https://www.balena.io/etcher/>). Recuerda comprar el lector ODROID eMMC-USB (<https://bit.ly/2ZjxC2L>) - te facilitará bastante la vida.

## Prueba de Rendimiento CPU/Memoria

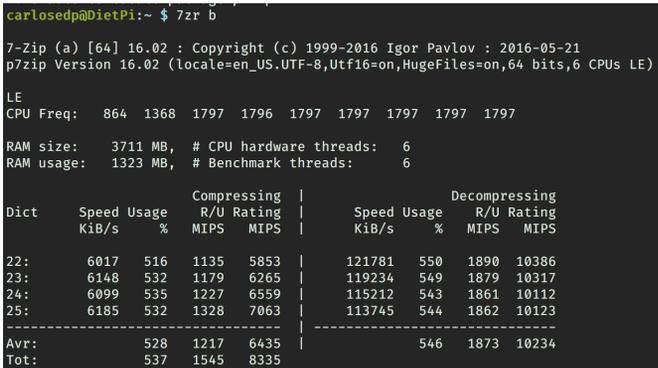
Primero, comparé parámetros artificiales (indicadores de referencia DietPi y 7zip). Estas pruebas ofrecen

una visión general del rendimiento del ODROID-N2 y un SBC equipado con RK3399.

## ODROID-N2-4GB



[ Fig. 03 ]

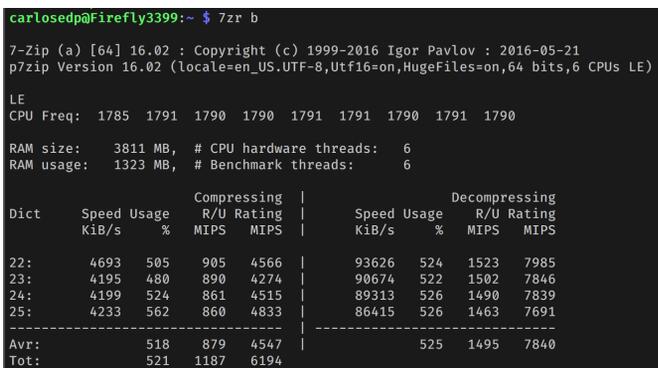


[ Fig. 04 ]

## Firefly RK3399-4GB



[ Fig. 05 ]



[ Fig. 06 ]

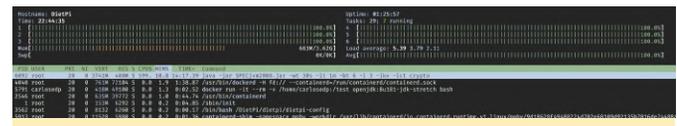
De media, el ODROID-N2 es 30-35% más rápido que Firefly RK3399, mi placa por defecto. También tiene un rendimiento de memoria mucho mejor, hasta un 40% más rápido. También comparé otras placas RK3399 en el pasado y todas tienen unos resultados similares a la Firefly.

## Prueba de rendimiento Java

A continuación, lleve a cabo algunas pruebas de rendimiento de Java en consonancia con mi publicación anterior comparando los resultados en SPECjvm2008. Estas pruebas, también las ejecuté en el Firefly RK3399, de modo que los resultados se ajustan al resto de pruebas que se muestran en este artículo.

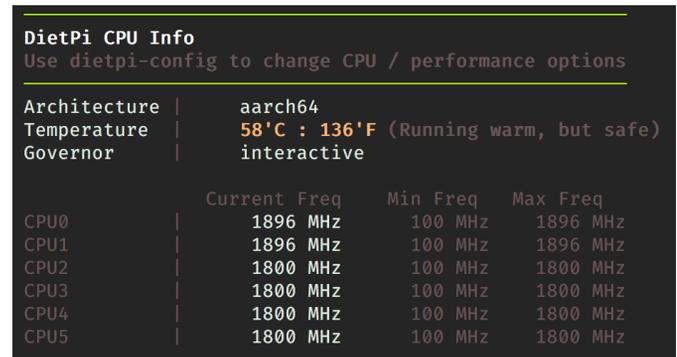
Las pruebas de rendimiento se ejecutaron en un contenedor Docker con los parámetros:

```
$ docker run -it --rm -v $(pwd):/test
openjdk:8u181-jdk-stretch bash
    Java -jar SPECjvm2008.jar -wt 30s -it 1m -bt
    6 -i 3 -ikv -ict
    [benchmark]
```



[ Fig. 07 ]

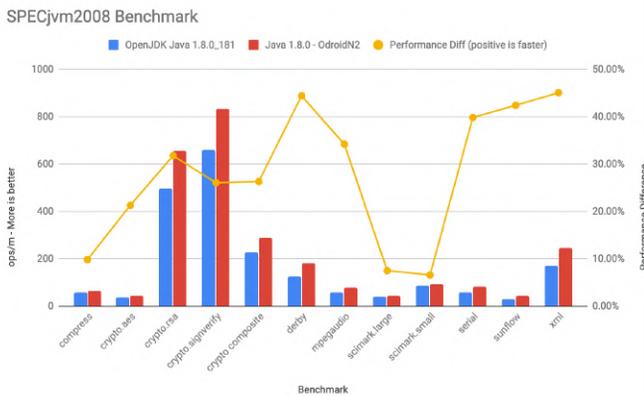
Aquí tienen las velocidades del núcleo y una evaluación de la temperatura durante las pruebas de rendimiento (100% en todos los núcleos)



[ Fig. 08 ]

Benchmark	OpenJDK Java 1.8.0_181	Java 1.8.0 - OdroidN2	Performance Diff (positive is faster)
compress	58.72	64.5	9.84%
crypto.aes	35.85	43.49	21.31%
crypto.rsa	497.01	654.92	31.77%
crypto.signverify	659.5	831.44	26.07%
crypto.composite	227.35	287.16	26.31%
derby	126.33	182.43	44.41%
mpegaudio	57.82	77.58	34.18%
scimark.large	41.06	44.14	7.50%
scimark.small	86.01	91.68	6.59%
serial	58.15	81.32	39.85%
sunflow	30.42	43.32	42.41%
xml	169.83	246.37	45.07%
Average:			27.94%

[ Fig. 09 ]



[ Fig. 10 ]

Como puedes ver, el rendimiento aumenta alrededor de un 30% en comparación con el RK3399.

## Red

Luego probé la red usando iperf3. Testeo tanto TX como RX usando la Ethernet de 1 Gbps conectado al mismo switch que otro ordenador. Como servidor, utilicé mi Macbook Pro conectado con un adaptador Ethernet de 1 Gbps. La Figura 11 muestra los resultados del ODROID-N2:

```
carlosedpa@DietPi:~$ iperf3 -c 192.168.15.141
Connecting to host 192.168.15.141, port 5201
[ 4] local 192.168.15.15 port 42276 connected to 192.168.15.141 port 5201
[ ID] Interval           Transfer     Bandwidth   Retr  Cwnd
[ 4] 0.00-1.00 sec      103 MBytes  859 Mbits/sec  0    3.98 MBytes
[ 4] 1.00-2.00 sec      101 MBytes  851 Mbits/sec  0    4.01 MBytes
[ 4] 2.00-3.00 sec      100 MBytes  837 Mbits/sec  8    2.00 MBytes
[ 4] 3.00-4.05 sec      98.8 MBytes  792 Mbits/sec  8    1.02 MBytes
[ 4] 4.05-5.01 sec      95.0 MBytes  830 Mbits/sec  21   235 KBytes
[ 4] 5.01-6.04 sec      98.8 MBytes  801 Mbits/sec  64   212 KBytes
[ 4] 6.04-7.00 sec      100 MBytes  876 Mbits/sec  0    382 KBytes
[ 4] 7.00-8.00 sec      101 MBytes  847 Mbits/sec  0    430 KBytes
[ 4] 8.00-9.00 sec      99.6 MBytes  837 Mbits/sec  0    458 KBytes
[ 4] 9.00-10.00 sec     97.5 MBytes  817 Mbits/sec  23   358 KBytes

[ ID] Interval           Transfer     Bandwidth   Retr
[ 4] 0.00-10.00 sec     995 MBytes  834 Mbits/sec  124
[ 4] 0.00-10.00 sec     991 MBytes  831 Mbits/sec

sender
receiver
```

[ Fig. 11 ]

El modo tráfico inverso tiene resultados más bajos, aunque observé resultados similares cuando probé la placa RK3399.

```
carlosedpa@DietPi:~$ iperf3 -R -c 192.168.15.141
Connecting to host 192.168.15.141, port 5201
Reverse mode, remote host 192.168.15.141 is sending
[ 4] local 192.168.15.15 port 42280 connected to 192.168.15.141 port 5201
[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-1.00 sec      29.8 MBytes  250 Mbits/sec
[ 4] 1.00-2.00 sec      57.2 MBytes  480 Mbits/sec
[ 4] 2.00-3.00 sec      78.5 MBytes  659 Mbits/sec
[ 4] 3.00-4.00 sec      49.5 MBytes  415 Mbits/sec
[ 4] 4.00-5.00 sec      60.0 MBytes  502 Mbits/sec
[ 4] 5.00-6.00 sec      60.7 MBytes  510 Mbits/sec
[ 4] 6.00-7.00 sec      63.0 MBytes  528 Mbits/sec
[ 4] 7.00-8.00 sec      72.6 MBytes  608 Mbits/sec
[ 4] 8.00-9.00 sec      63.5 MBytes  533 Mbits/sec
[ 4] 9.00-10.00 sec     85.6 MBytes  719 Mbits/sec

[ ID] Interval           Transfer     Bandwidth
[ 4] 0.00-10.00 sec     621 MBytes  521 Mbits/sec
[ 4] 0.00-10.00 sec     621 MBytes  521 Mbits/sec

sender
receiver
```

[ Fig. 12 ]

Intenté desactivar la descarga de la suma de comprobación de la red (un problema conocido en los SOC de Rockchip) pero los resultados de rendimiento fueron los mismos.

## Conclusión

La placa tiene un enorme potencial y es el SBC ARM más potente que he visto. Se adapta a muchos posibles usos, desde un servidor doméstico/mini hasta un centro multimedia con todas las funciones o un ordenador de escritorio que puede ejecutar casi cualquier carga de trabajo instalada o en contenedores. Es fantásticamente adecuado para un clúster Kubernetes con múltiples nodos. Además, su consumo de energía es sorprendente y puede estar siempre ENCENDIDO con solo 2.8 W en reposo y 6.5 W mientras se realiza pruebas de rendimiento con los 6 núcleos al 100%.

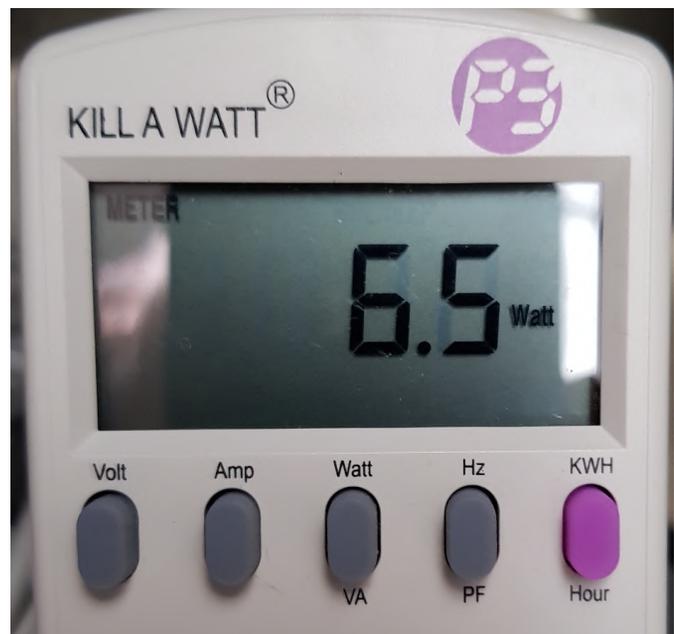


Figura 13 - Consumo estando encendido y trabajando al 100%

Es fácil grabar nuevas imágenes (usando eMMC) y la conectividad es suficiente para la mayoría de los posibles usos. Me encantaría ver una ranura PCI-E o un conector M.2. para unidades NVME. En realidad, encontré un documento que dice que el S922XSOC contiene un PCI-E de 1 carril que, en el caso ODROID-N2, fue usado para el puerto USB 3.0. Otras empresas podrían proporcionar esta conexión PCI-E como un conector M.2 o ranura PCI-E.

En general, recomiendo encarecidamente la placa y la pongo en el top de mi lista por sus características, rendimiento y precio razonable. Además, en un futuro próximo, creo que podría transformarlo en un escritorio ARM64 con una distribución de Linux con

todas las funciones como Ubuntu o Fedora, si logro exportarlo.

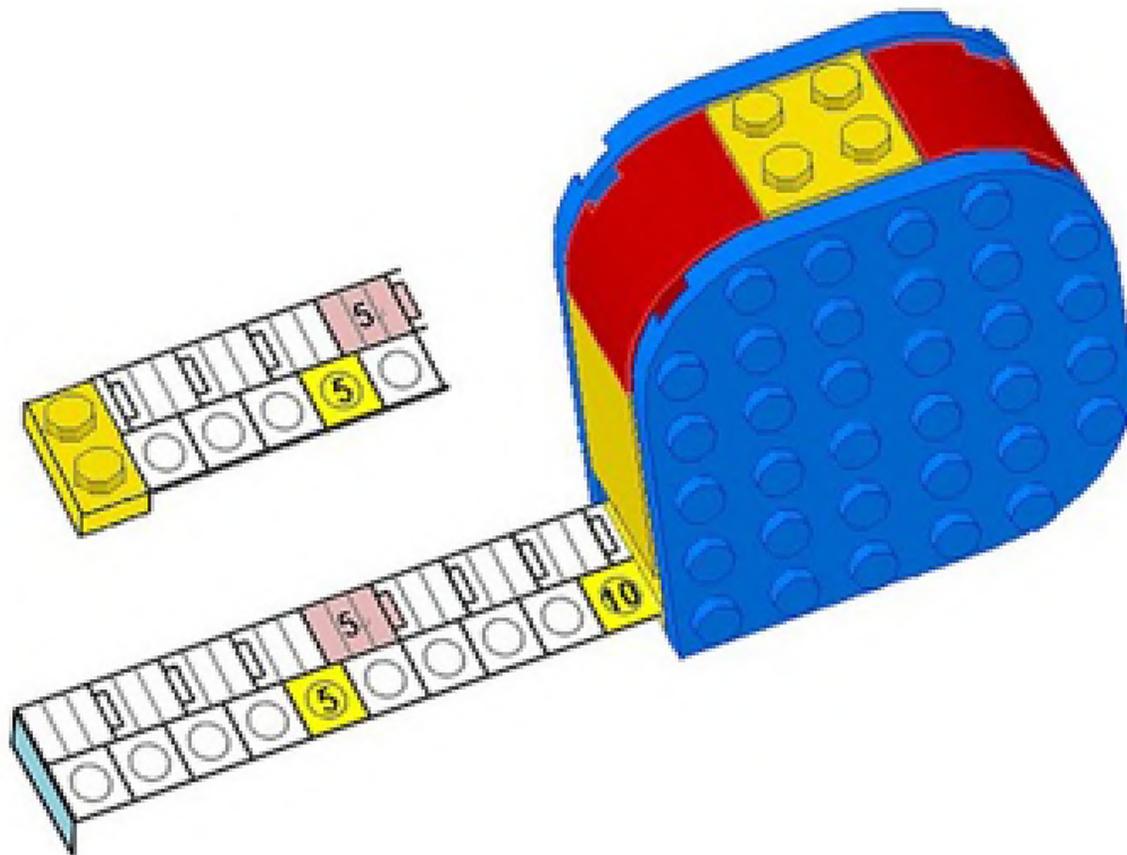
<https://bit.ly/2ziyPbX>

<https://github.com/MichaIng/DietPi/issues/2028>

## **Referencias**

# Cinco Minutos de Diversión con tu Monku R1: Redimensionar la Partición de la Tarjeta SD

© September 1, 2019 By Brian Ree Linux, Tutoriales



Este tutorial te mostrará cómo ajustar el tamaño de las las particiones en una tarjeta SD hecha a partir de una imagen de una tarjeta SD de menor tamaño. Por ejemplo, tiene una nueva tarjeta SD de 32 GB lista para tu Monku1000/ODROID-GO (<https://www.hardkernel.com/shop/odroid-go/>) y tiene una imagen backup del dispositivo de un amigo, pero ésta es de solo 16 GB. Después de escribir la imagen en la nueva tarjeta SD, observas que solo puedes usar alrededor de 16 GB de espacio. ¿Qué ha pasado con los otros 16 GB? No te preocupes, te voy a enseñar cómo cambiar el tamaño de las particiones usando tu dispositivo ODROID. Necesitarás un dispositivo ODROID con Ubuntu como el que usamos para compilar R1 o R2 ([http://middlemind.net/tutorials/odroid\\_go/mr1\\_build.html](http://middlemind.net/tutorials/odroid_go/mr1_build.html)), y R3 ([http://middlemind.net/tutorials/odroid\\_go/mr3\\_build.html](http://middlemind.net/tutorials/odroid_go/mr3_build.html)).

## Herramientas necesarias

- Un ODROID-C +, -C2 o -XU4
- Damos por sentado que estos dispositivos están configurados con Ubuntu MATE.
- Tarjeta microSD de 8GB o de mayor tamaño
- Adaptador MicroSD a USB

Enciende tu dispositivo ODROID y comencemos. Si has instalado gparted, estará disponible desde el menú del sistema en System -> Administration -> GParted. Si no lo tienes instalado, ejecuta los siguientes comandos. Abre el terminal MATE desde esta ubicación del menú Applications -> System Tools -> MATE Terminal. Luego, introduce:

```
$ sudo apt-get install gparted -y
```

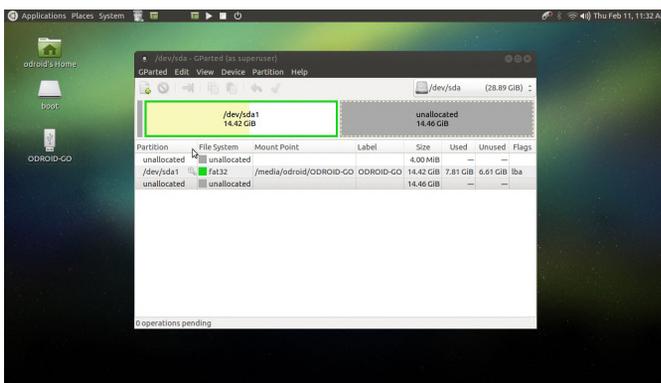
Después de la instalación, ejecuta gparted. Abre esta ubicación del menú System -> Administration ->

GParted. Debería aparecer una aplicación como la que se muestra a continuación. Si se te solicita una contraseña, introduce la contraseña root por defecto, odroid o la contraseña que estés utilizando.



**Figura 1 - Solicitud de contraseña al iniciar Gparted**

Asegúrate de seleccionar la unidad correcta en la lista desplegable de la esquina superior derecha de la pantalla. AVISO: Comprueba que has seleccionado la unidad correcta o podrías perder datos seleccionando la unidad incorrecta. ¡Observa que la información de la partición que aparece en gparted muestra 14 GB de espacio sin asignar! Queremos usar ese espacio extra, pero como grabamos esta tarjeta SD desde una imagen de una tarjeta SD de 16GB, ¡ignoramos el espacio de almacenamiento restante! Toma nota del sistema de archivos utilizado para la partición activa de la tarjeta SD, necesitaremos usar esto cuando creamos una nueva partición más grande. AVISO: anotar el sistema de archivos utilizado si es la primera vez que utilizas gparted, porque tendrás que restaurar este sistema de archivos más adelante.

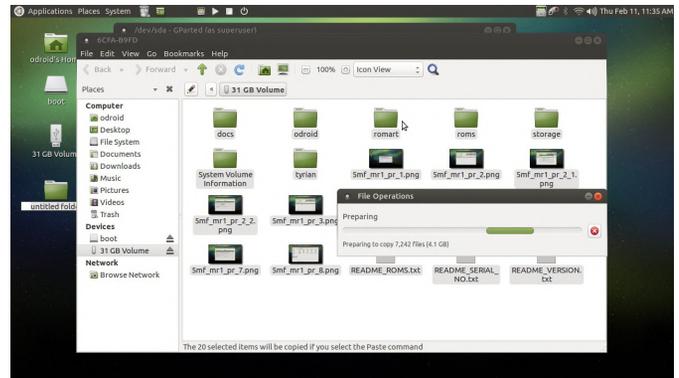


**Figura 2 - Información de partición con Gparted**

## Cambiar el tamaño de la partición

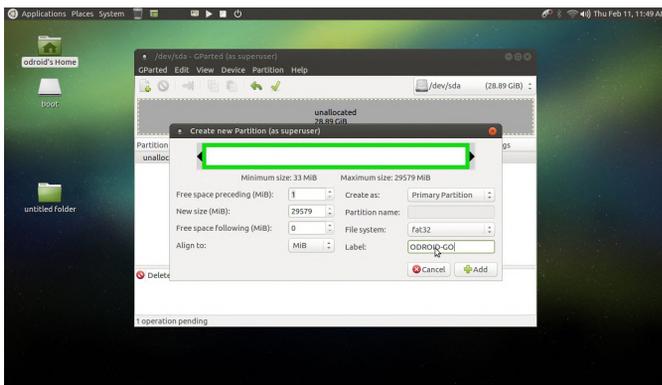
Hay diferentes formas de cambiar el tamaño de la partición. El que voy a describir a continuación lleva un poco más de tiempo, pero te asegurará contar con

una tarjeta SD que MacOS y Windows pueden leer perfectamente. Lo primero que tienes que hacer es hacer clic derecho en el escritorio y crear una nueva carpeta. Nómbrala como quieras, la usaremos temporalmente para guardar los archivos de la tarjeta SD original. Copia todos los archivos y carpetas de la tarjeta SD en la carpeta que acaba de crear. Deberías ver una barra de copia de archivos como la que se muestra a continuación.



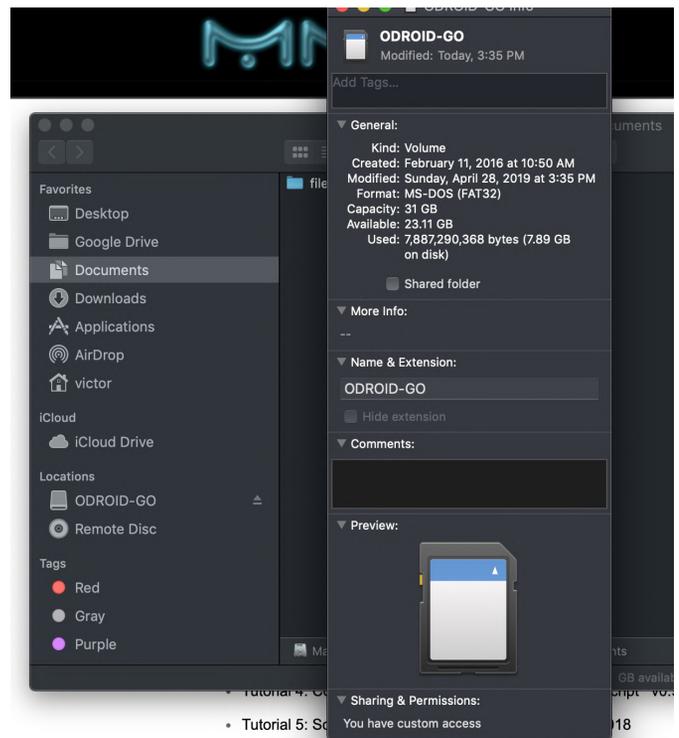
**Figura 3 - Copiando todos los archivos necesarios para la tarjeta SD**

Una vez completada la copia de seguridad del archivo, cierra y vuelve a abrir gparted desde esta ubicación del menú, System -> Administration -> GParted. Asegúrate de seleccionar la unidad adecuada en el menú desplegable en la parte superior derecha de la ventana de la aplicación. Haga clic derecho en las particiones activas enumeradas y desmóntalas, luego bórralas. Deberías terminar sin particiones y con una entrada que muestre el tamaño completo de la tarjeta SD sin asignar. Luego, haz clic derecho y selecciona la opción Add a New partition. Usa las opciones que se detallan a continuación. Básicamente, debes mantener el sistema de archivos de la tarjeta SD original, en este caso fat32, y asegúrate de que no haya 0 Mibs sin usar después de la partición; la partición individual es tan grande como lo puede ser. Recuerda el sistema de archivos que observamos cuando vimos por primera vez la tarjeta SD. Ahora haz clic en el botón Add y luego haz clic en el botón Apply All Operations. Aparecerá un botón de verificación verde cerca de la parte superior de la ventana de la aplicación y gparted aplicará los cambios que hayas seleccionado.



**Figura 4 - Crear una nueva partición en Gparted**

Cuando la partición esté lista, cierra gparted, desconecta y vuelve a conectar (montar) tu tarjeta SD y restaura los archivos originales de los que hiciste una copia de seguridad en tu dispositivo ODROID. Cuando termines de restaurar los archivos, puede probar la tarjeta SD comparándola con la original o utilizándola con el mismo dispositivo. Simplemente coloca la tarjeta microSD en su Monku1000 / ODROID-GO para ver si funciona. Tal y como se muestra a continuación, nuestro dispositivo se enciende bien y reconoce nuestras ROM. En segundo lugar, necesitamos verificar la tarjeta en una máquina MacOS o Windows para asegurarnos de que funcione bien con estos sistemas operativos. Podemos ver que MacOS reconoce que la tarjeta SD tiene una capacidad de 31 GB. AVISO: No verás la capacidad completa indicada por el fabricante en una tarjeta SD debido a las asignaciones de mantenimiento del sistema de archivos. Ahora puede acceder a los 14 GB restantes de espacio de la tarjeta microSD. ¡Increíble!



**Figura 5 - ODROID-GO con todos los archivos que necesitas**



**Figura 6 - Tarjeta SD montada en macOS**

Esperamos que este tutorial te haya proporcionado información detallada de cómo puedes ajustar el tamaño de las particiones en las tarjetas SD. Este artículo ha sido adaptado

de [http://middlemind.net/tutorials/odroid\\_go/5f\\_mr1\\_pr.html](http://middlemind.net/tutorials/odroid_go/5f_mr1_pr.html), donde tienes disponibles más tutoriales

# Gestiona el Tiempo de Ordenador de tu Hijo con mqttNanny

September 1, 2019 By Adrian Popa Linux, Tutoriales



En mi último artículo configuré mi ODRROID-H2 para que fuera el primer ordenador (ejecutando Linux) de mi hijo de 7 años. Como sabes, un gran poder conlleva una gran responsabilidad, esto significa que tengo que ser capaz de imponer algunos límites de tiempo de uso del ordenador. Particularmente dado que, mi hijo pasará las vacaciones de verano con sus abuelos. Es la excusa perfecta para crear un sistema de gestión y monitorización de tiempo en Linux e integrarlo en Home Assistant

## Cómo funciona

El software es esencialmente un script en Python 3 que se ejecuta como un demonio. Tiene dos modos de funcionamiento: local y remoto. Se inicia en modo local cuando no puede conectarse a un agente MQTT. En este modo, carga sus límites desde un archivo y es menos flexible. Cuando se ejecuta en modo remoto, el usuario actual obtiene un permiso desde un agente MQTT permitiendo que el padre lo controle (y siga lo que el usuario está haciendo) dinámicamente.

Básicamente, el modo remoto ofrece todas las características, mientras que el modo local es una alternativa en caso de problemas de conectividad.

El programa principal se ejecuta cada minuto en bucle y recopila (y envía) información, como:

- ¿Cuál es el TTY activo?
- ¿Qué usuario ha iniciado sesión con X11 en el TTY actual?
- ¿Se está ejecutando el protector de pantalla?
- ¿Cuál es el título de la aplicación activa?

Para el usuario conectado en ese momento, se activa un temporizador de cuenta atrás que va restando minutos mientras que el protector de pantalla no esté funcionando. Se envían algunas notificaciones al usuario (notificación-envío con sonido) cuando quedan 10/5/1 minutos. Cuando se agota el tiempo, el protector de pantalla se activa y la cuenta de usuario se deshabilita (el inicio de sesión con contraseña falla). Si el protector de pantalla no se

activa a la quinta 5 vez consecutiva, el sistema se apaga. Una vez que la cuota asignada sea mayor que cero, el demonio restaura la contraseña de esa cuenta.

MQTT te permite ver los datos recopilados, cambiar la asignación de tiempo para cada usuario y también solicitar capturas de pantalla de la sesión de escritorio del usuario. Para seguir adelante, necesitarás conocer y tener un agente MQTT y también ejecutar Home Assistant; estas cuestiones no las trataremos en este artículo, aunque ya han sido analizadas en anteriores artículos.

## Instalación y configuración

Puedes descargar e instalar el código desde mi página de GitHub:

```
$ sudo apt-get install git
$ cd /usr/local/share
$ sudo git clone https://github.com/mad-
ady/mqttNanny.git
$ cd mqttNanny
$ sudo cp mqttNanny.yaml /etc/mqttNanny.yaml
$ sudo cp mqttNanny.service
/etc/systemd/system/mqttNanny.service
```

Necesitarás instalar también algunas dependencias:

```
$ sudo apt-get install python3-pip python3-yaml
python3-notify2
    espeak xscreensaver xdotool imagemagick
$ sudo pip3 install paho-mqtt
```

Ten en cuenta que, actualmente, solo admite sistemas Linux, aunque el código está escrito para que también se pueda extender a otros sistemas operativos (las peticiones son bienvenidas). El sistema Linux necesita ejecutar Xorg (Xwayland probablemente necesita muchos cambios) y el programa del protector de pantalla debe ser Xscreensaver (no mate-screensaver, xfce-screensaver, etc.). No obstante, en el futuro se podría añadir soporte para otros protectores de pantalla (bienvenidas son las peticiones). El código ha sido probado en un ODROID-H2 (x86\_64), ODROID-XU4 (armhf) y un ODROID-N2 (arm64). Para pasar de mate-screensaver a xscreensaver, puede hacer lo siguiente:

```
$ sudo apt-get purge mate-screensaver
$ xscreensaver-demo
```

Mientras se ejecuta xscreensaver-demo, puedes seleccionar qué protectores de pantalla quieres usar y también activar la opción "Lock screen after" para forzarlo a solicitar una contraseña. Deberás hacer que el entorno de escritorio inicie el protector de pantalla automáticamente copiándolo en la carpeta autostart:

```
$ mkdir ~/.config/autostart
$ cp /usr/share/applications/xscreensaver-
properties.desktop
    ~/.config/autostart/
$ sed -i 's/xscreensaver-demo/xscreensaver/'
    ~/.config/autostart/xscreensaver-
properties.desktop
```

Además, asegúrate de que la hora del sistema esté configurada correctamente en el arranque (ya sea a través de un RTC, NTP o fake-hwclock), de lo contrario, el temporizador en modo local no funcionará correctamente.

Necesitarás editar la configuración (/etc/mqttNanny.yaml) y fijar tus valores por defecto más relevantes. Asegúrate de que el sangrado sea el correcto en el archivo (puedes validarlo en <http://www.yamllint.com/>), de lo contrario, el programa no se iniciará. Las opciones disponibles son las siguientes:

- **mqttServer** - el nombre ip/dns de tu agente MQTT
- **mqttPort** - el puerto TCP en el que se ejecuta tu agente (por defecto 1883)
- **mqttUser/mqttPass** - tus credenciales MQTT. Si tu agente no usa autenticación, simplemente omite las líneas
- **baseTopic** - un prefijo para crear los temas utilizados para enviar/recibir mensajes. Yo personalmente uso ha//
- **mqttTimeTopicSuffix** - el tiempo restante para cada usuario se transmitirá en un tema creado desde //. En mi caso es algo así como ha/pc/odroid/timeRemaining
- **mqttScreenshot** - el tema donde deseas recibir imágenes de captura de pantalla (MQTT también puedes transmitir datos binarios)
- **mqttScreenshotCommand** - el tema donde puede solicitar capturas de pantalla o no. Esto se puede

asignar a un parámetro en Home Assistant para activar o desactivar las capturas de pantalla

- **mqttScreenshotDuration** - tiempo hasta que la función de captura de pantalla se apague. Si lo quieres siempre activado, configúralo en 0
- **mqttScreenshotInterval** - con qué frecuencia (en segundos) debería tomar capturas de pantalla
- **screenshotHeight** - cambia el tamaño de la captura de pantalla a esta altura (manteniendo la relación de aspecto), por razones de eficiencia
- **checkInterval** - con qué frecuencia se debe marcar el reloj interno del script. Un valor de 60 segundos significa que la asignación se verifica cada minuto. También recibes cambios de aplicación cada minuto..
- **externalNotify** - debe ser False o apuntar a un programa/script que toma una cadena como argumento y te envía el mensaje. Por ejemplo, yo lo configuré con `/usr/local/bin/telegram-send` y recibo notificaciones de eventos a través de un robot de telegram

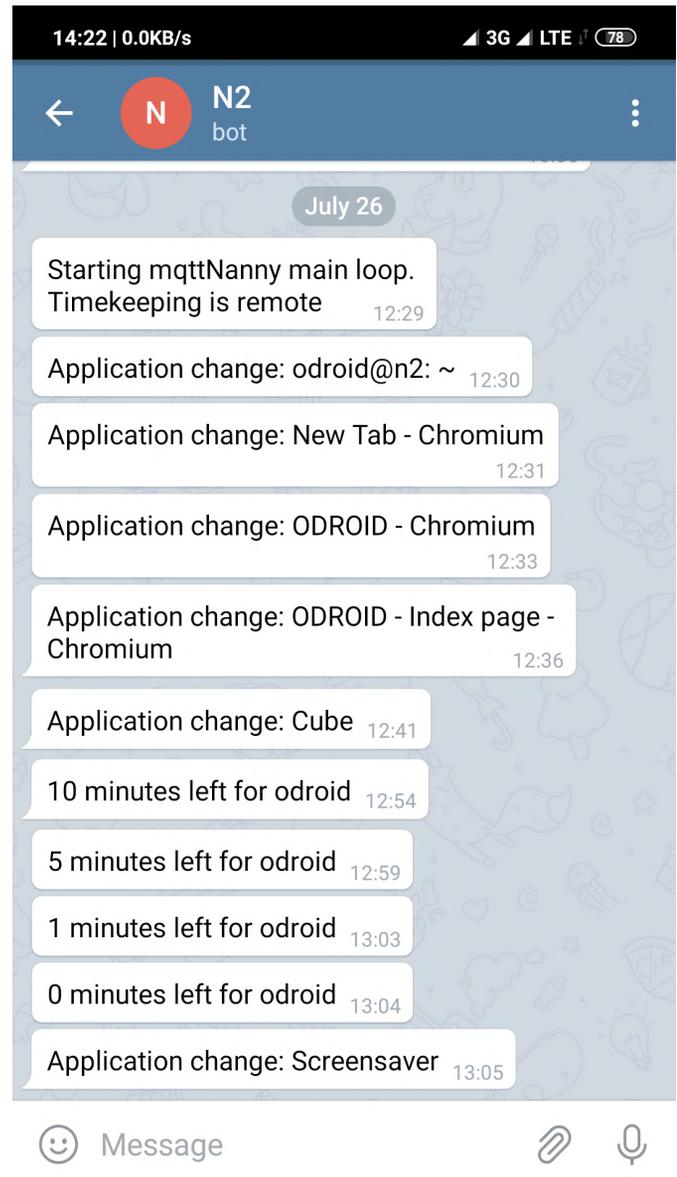


Figura 1. Notificaciones externas

- **no-signal** - debe apuntar a un archivo de imagen que se muestra cuando se desactivan las capturas de pantalla
- **users** - contiene una lista de usuarios para monitorizar. Si un usuario no está en la lista, su tiempo no se gestionará, pero los datos sobre su sesión y capturas de pantalla serán reportados a través de MQTT
- **defaultOfflineTime** - cuántos minutos tiene el usuario al inicio del programa en el caso de que el programa se ejecute en modo local, sin una conexión con el agente MQTT. Si hay un archivo con la asignación actual del usuario en `/usr/local/share/mqttNanny//`, ese valor será el que se cargue en su lugar. Este archivo se actualiza con cada cambio/disminución de la cuota asignada.



```
binary_sensor:
  - platform: mqtt
    state_topic: 'ha/pc/screensaver'
    name: N2 Screensaver
    payload_on: True
    payload_off: False
```

Una vez que se vuelva a cargar la configuración de Home Assistant, podrás agregar los nuevos elementos en la interfaz web de Lovelace. A continuación, se muestra una configuración de ejemplo basada en los anteriores valores.

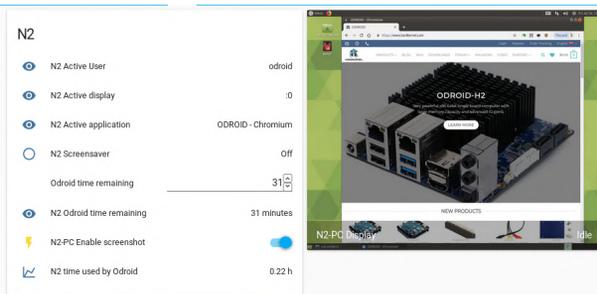
Añade un panel de entidades con esta configuración:

```
entities:
  - entity: sensor.n2_active_user
  - entity: sensor.n2_active_display
  - entity: sensor.n2_active_application
  - entity: binary_sensor.n2_screensaver
  - entity: input_number.pc_odroid_time_remaining
  - entity: sensor.n2_odroid_time_remaining
  - entity: switch.n2_pc_enable_screenshot
show_header_toggle: false
title: N2
type: entities
```

Añade un panel de entidad imagen con esta configuración:

```
camera-view: live
entity: camera.n2_pc_display
type: picture-entity
```

El resultado final debería ser como el de la siguiente imagen:



**Figura 4. Control de Home Assistant**

También necesitarás configurar algo de automatización para poder cambiar el tiempo permitido para un determinado usuario. Realiza tus cambios en automations.yaml y reinicia Home Assistant:

```
- id: '1557839383'
  alias: N2 change remaining time for Odroid
  trigger:
    - entity_id:
        input_number.pc_odroid_time_remaining
      platform: state
    action:
      - data:
          payload: '{{
states(input_number.pc_odroid_time_remaining) |
int }}'
        retain: true
        topic: ha/pc/odroid/timeRemaining
      service: mqtt.publish

- id: '1562675409185'
  alias: N2 get remaining time for Odroid
  trigger:
    - platform: mqtt
      topic: ha/pc/odroid/timeRemaining
      condition: []
    action:
      - data_template:
          entity_id:
            input_number.pc_odroid_time_remaining
          value: '{{ trigger.payload }}'
      service: input_number.set_value

- id: '1562675848675'
  alias: N2 set daily time allowance for user
  Odroid
  trigger:
    - at: 00:15:00
      platform: time
      condition: []
    action:
      - data:
          payload: 35
          retain: true
          topic: ha/pc/odroid/timeRemaining
      service: mqtt.publish
```

La última automatización fija el tiempo diario permitido para el usuario odroid en 35 minutos y se ejecuta a las 00:15. Esto es simplemente un ejemplo. Puede diseñar tus propias automatizaciones que definen cuándo el usuario puede usar el ordenador. Por ejemplo, si solo deseas dar acceso durante un intervalo de tiempo, puede ejecutar una automatización para agregar la asignación a la hora de inicio deseada y una automatización diferente

para eliminar la asignación antes de la hora de irse a dormir. Puede controlar la cantidad de tiempo que se asigna a cada usuario en función de determinadas cosas como los días de colegio (<https://www.home-assistant.io/components/workday/>) o tal vez si ha terminado o no sus deberes (me gustaría saber cómo podríamos medir esto de forma automática).

## Fallos y futuras mejoras

Ninguno de los códigos que escribo es perfecto y éste no es una excepción. Hay algunos problemas y formas de evitar mqttNanny que he identificado hasta ahora (parches/ideas son bienvenidos). El programa está destinado a bloquear a una persona con poca experiencia en Linux, pero no será muy eficaz con un administrador de sistemas experimentado. Cuando se ejecuta en modo local, no hay un mecanismo integrado que permite conceder más tiempo al usuario. Si el ordenador está encendido durante un período prolongado de tiempo, se debe reiniciar el servicio mqttNanny para añadir la correspondiente asignación diaria. Por lo tanto, podría no funcionar como cabría esperar, si se suspende el ordenador en lugar de apagarlo. Al cambiar entre los modos local/remoto, los datos recibidos del agente MQTT tienen mayor prioridad. Por ejemplo, el niño podría empezar sin conexión a la red, agotar su asignación predeterminada y luego reiniciar la conexión de red al agente. Recibiría su asignación remota como si nada hubiera pasado. Cuando se ejecuta en modo local, la protección es bastante pobre si el usuario puede cambiar la hora del sistema. Podría "pedir prestado" tiempo de otros días en el futuro/pasado. Solo es monitorizada/bloqueada la sesión X11. El usuario

puede iniciar sesión en un TTY antes del bloqueo, o mediante ssh con autenticación de clave y puede seguir usando el sistema (basado en terminal).. Si el usuario mata el proceso del protector de pantalla, no hay nada que bloquee la sesión. El proceso intenta bloquear el protector de pantalla 5 veces consecutivas y, si falla, se apagará el sistema. Pero el usuario podría usar el exploit en el intento número 4 y desbloquear el protector de pantalla desde la línea de comandos (podría ejecutar un script para desbloquearlo continuamente). Si el ordenador no se apaga correctamente (por ejemplo, se desenchufa directamente de la corriente), no tendrás el estado correcto en MQTT. El archivo `/etc/mqttNanny.yaml` utilizado por el código debe ser legible solo por root, ya que contiene sus credenciales MQTT. De lo contrario, el usuario podría conocerlas y usar `mosquitto_pub` para cambiar su tiempo asignado. Los archivos que almacenan la asignación diaria utilizada en `/usr/local/share/mqttNanny//` no se limpian automáticamente y eventualmente pueden llenar su disco en unos pocos siglos (1,5 MB / año). Una tarea cron para eliminar archivos antiguos sería de gran utilidad.

Espero que las versiones futuras del código agreguen soporte para otros protectores de pantalla para Linux, así como soporte para MacOS y Windows (el código específico del sistema operativo es modular y debería ser fácilmente ampliable). Esperamos nuevas versiones/ parches.

## Referencias

<https://www.home-assistant.io/>

# Una Potente Imagen Multiarranque para el ODROID-C2: Ejecuta Android, Ubuntu MATE y LibreELEC desde un Único Dispositivo

© September 1, 2019 By Alexander G (@alexgg) Android, Linux, ODROID-C2, Tutoriales



Soy desarrollador web y me han llamado bastante la atención los denominados entornos de multiarranque, especialmente en los dispositivos System On a Chip (SoC). Inicié este viaje con un dispositivo Raspberry Pi 2 y un gestor de arranque llamado: "Berryboot" (no soy su desarrollador) que en realidad todavía está activo y es mantenido por la comunidad Raspberry Pi.

Luego oí hablar de este increíble dispositivo llamado ODROID-C2 más potente que un dispositivo Raspberry Pi 2 y prácticamente por el mismo precio, así que más potencia más diversión, ¿verdad? Un par de semanas después ya contaba con un dispositivo ODROID-C2 en mis manos y empezaron las pruebas. Probé Ubuntu (Xenial en ese momento), Android, LibreELEC, Recalbox, Lakka y algunos sistemas operativos más. Más tarde observé que alguien (@loboris) en el Foro ODROID (<https://forum.odroid.com>) había desarrollado

scripts bash para crear imágenes multiarranque en dispositivos ODROID, incluido el ODROID-C2, tras un par de días haciendo algunas pruebas, logré que funcionara. Desde ese momento, he creado mis propias imágenes de arranque múltiple listas para usar y listas para arrancar en el ODROID-C2.

Puedes descargar más imágenes de multiarranque e imágenes independientes para ODROID-C2 desde aquí (<https://berryboot.alexgoldcheidt.com/odroid-c2/>). Gestiono este sitio web y me alegraría escuchar todas tus sugerencias, problemas descubiertos y solicitudes de imágenes. Recientemente, he creado una imagen multiarranque (Ver. 4) para ODROID-C2 con los siguientes SO preinstalados:

- Android 6.0.1 [2019-01-17]
- Ubuntu MATE 18.04 Bionic LTS [2018-06-21] (Linux in the boot menu)

- LibreELEC RR 9.1 [2019.04.24] (OpenELEC in the boot menu)

Quiero mostrarte cómo puedes crear tú mismo esta imagen de multiarranque, de modo que comparto estos archivos y pasos a modo de guía de referencia. Primero, ¿qué es LibreELEC RR? Bueno, esta versión de LibreELEC incluye EmulationStation, RetroArch y Pegasus. Tienes más información sobre esta compilación en <https://bit.ly/30Aifjw>. Con esta sorprendente versión puede evitar la instalación de imágenes adicionales como: RetroPie, Lakka o Recalbox. En mi caso funciona muy bien. Android y Ubuntu también funcionan bien como imágenes multiarranque.

## Que necesitarás

- Linux (yo he usado Debian con el entorno de escritorio Xfce). Puedes descargarlo desde (<https://www.debian.org/CD/http-ftp/#stable>), y simplemente seleccionar la arquitectura correcta para tu ODROID o puedes descargar cualquier versión activa (no requiere instalación) como Ubuntu MATE, que se puede descargar desde (<https://ubuntu-mate.org/download/>). Nuevamente, solo tienes que seleccionar la arquitectura correcta para tu ODROID.
- Conexión a Internet
- Dispositivo ODROID-C2

Primero, empecemos con la imagen base multiarranque. Yo he usado el repositorio personalizado de @sdip porque éste soluciona un par de problemas del repositorio original de @loboris. Puedes encontrar información adicional en: <https://bit.ly/2Hs1obf>. Puedes generar tú mismo la imagen base o descargar la imagen lista para arrancar desde: <https://bit.ly/2Znjwbl>. Una vez completada la descarga, descomprímela y guárdala en una tarjeta microSD o módulo eMMC. Recomiendo usar un módulo eMMC compatible.

**SO #1:** Android. Descarga el archivo Android Daily desde: <https://bit.ly/2Nxxczg>. Una vez que se complete la descarga, cámbiale el nombre: update.tar.gz y coloca este archivo en una unidad USB con un nuevo formato de partición EXT4.

**SO #2:** Descarga Ubuntu MATE 18.04 Bionic LTS desde: <https://bit.ly/2Nwt4zr>. Una vez que se complete la descarga, descomprímelo y cámbiale el nombre: linux.img y coloca este archivo en la misma unidad USB junto con el archivo de Android.

**SO #3:** LibreELEC RR. Descarga LibreELEC RR v9.1 desde: <https://bit.ly/30zlrLp>. Una vez que se complete la descarga, cámbiale el nombre: oelec.tar y coloca este archivo en la misma unidad USB con los archivos de Android y Ubuntu.

Ahora inserta la tarjeta microSD o el módulo eMMC con la imagen base y conecta la unidad USB (que contiene Android, Linux y LibreELEC) en tu dispositivo ODROID-C2 y enciéndelo. Una vez que veas la pantalla principal multiarranque, sigue estos pasos:

- Presiona "I" (Prepara la tarjeta para el Multiarranque, Instala el SO).
- Cambia el tamaño de las particiones a tu gusto.
- Una vez más, presiona "I" para iniciar la instalación.
- Cuando se complete este proceso, reinicia el dispositivo.

En caso de que no estés seguro del tamaño que vas a necesitar en las particiones, aquí tienes mis sugerencias: usa al menos 9GB o 10GB en la partición de Ubuntu, ya que este es el sistema operativo más grande de la imagen multiarranque: el tamaño de la imagen original de Ubuntu es de alrededor de 7GB. Además, asigna suficiente espacio en la partición LibreELEC si tienes pensado almacenar ROM de juegos.

Partition	File System	Label	Size	Used	Unused	Flags
unallocated	unallocated		32.00 MB	---	---	
/dev/sdc1	fat32	emuserdata	2.38 GiB	606.38 MB	1.78 GiB	lba
/dev/sdc2	ext4	emsystem	3.50 GiB	1.22 GiB	2.28 GiB	
/dev/sdc3	ext4	emcache	1.62 GiB	60.69 MB	1.57 GiB	
*/dev/sdc4	extended		21.59 GiB	---	---	
/dev/sdc5	ext4	emdata	4.12 GiB	748.92 MB	3.39 GiB	
/dev/sdc6	ext4	emoelec	4.88 GiB	926.80 MB	3.97 GiB	
/dev/sdc7	ext4	emlinux	12.56 GiB	6.48 GiB	6.08 GiB	
unallocated	unallocated		29.00 MB	---	---	

**Figura 1 - Este es mi diseño de particiones del módulo eMMC**

Ahora, llegados a este punto, podrás iniciar Android y Ubuntu sin hacer nada más, para iniciar LibreELEC RR debe seguir los pasos que aparecen en: <https://bit.ly/324hcji> de (y gracias a) @malomehi. A modo de referencia, compartiré mi directorio multiarranque actual de mi módulo eMMC por si quieres comparar o descargar archivos al final. Puedes ver estos archivos en: <https://bit.ly/30yAI04>. Ten en cuenta que si comparas mis archivos de arranque (desde el módulo eMMC) con la tarjeta microSD, debes cambiar las etiquetas de las particiones. Por ejemplo: emuserdata se usa en el módulos eMMC pero en la tarjeta microSD se usa sduserdata. Además, en mis archivos de arranque he configurado una resolución de 1360x768p60hz para Android y Ubuntu, ya que ésta es la resolución máxima de mis pantallas. Es posible que tengas cambiar esta resolución para que se ajuste a tu pantalla. Si has llegado hasta aquí, felicidades, ahora tiene una imagen de arranque múltiple que contiene los tres sistemas operativos más populares de este mundo. Si, por alguna razón, algo ha salido mal,

durante o después del proceso de instalación, avísame a través de: <https://bit.ly/2zic1jk>. Puedo echar un vistazo al problema e intentar solucionarlo.

## Problemas conocidos

Después de una actualización completa de Ubuntu

```
apt upgrade -y; apt dist-upgrade -y
```

Los archivos Image, meson64\_odroidc2.dtb y ulnitr se actualizan fuera del directorio de multiarranque. Todo lo que tiene que hacer es ir a la partición de arranque: emuserdata en los módulos eMMC o sduserdata en las tarjetas SD, añade .linux a los nombres de archivo (ejemplo: Image.linux) y muévelos al directorio multiarranque (puede sobrescribirlos).

Por último, quiero agradecer a @loboris, @odroid, @dimitris\_c, @barturblits, @luffyman, @sdip y @malomehi de los Foros ODROID por hacer esto posible. Es un honor formar parte de esta comunidad y hace todo lo posible para mejorarla.