

Logo Inicio ODROID • emuELEC • Análisis y Rendimiento N2 • MAME

# ODROID

Año Seis  
Num. #66  
Jun 2019

Magazine

## ZONEMINDER

MONTA TU PROPIO SISTEMA DE VIGILANCIA

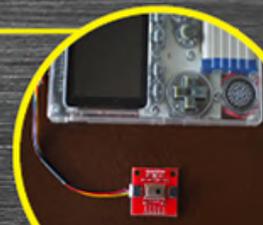


**GOOGLE  
STADIA:**

EL SERVICIO DE JUEGOS POR  
STREAMING QUE INFUNDIRA UNA  
NUEVA VIDA A TU ODROID-XU4

**JUEGOS RETRO MONKU:**  
AÑADE MANDOS Y BOTONES PERSONALIZADOS

**FAN DE GO:**  
VISION IR PARA TU ODROID-GO





## La Versión Alpha de emuELEC ahora disponible para el ODROID-N2

© June 1, 2019

EmuELEC es una imagen de emulación que te permite ejecutar juegos retro: Atari, NES, SNES, TG16, PSP, PSX, N64, Genesis / Megadrive y muchos más.



## Logotipo de Inicio Personalizado: Creando una Imagen de Inicio a Medida para el ODROID-C2

© June 1, 2019

Puedes crear un logotipo de inicio personalizado para el ODROID-C2 siguiendo las sencillas instrucciones que se detallan a continuación. El logotipo aparecerá en los primeros segundos del arranque mientras se carga el sistema operativo. El formato de imagen básico del logotipo de arranque del ODROID-C2 se describe en la siguiente [▶](#)



## ZoneMinder en ODROID-XU4: Monta tu Propio Sistema de Vigilancia

© June 1, 2019

ZoneMinder (ZM) es un conjunto integrado de aplicaciones que proporciona una completa solución de vigilancia que le permite al usuario capturar, analizar, registrar y monitorizar una zona a través de cualquier CCTV o cámara de seguridad.



## Cómo Crear una Consola de Juegos Retro Monku - Parte 2: Añadir Mandos y Botones Personalizados

© June 1, 2019

Esta es una continuación del artículo de la consola de juegos Retro del número de mayo de 2019, en el que aprendimos cómo ensamblar los componentes básicos de una consola de juegos retro.



## Me Vuelvo a Qwiic: Añade Visión IR a tu ODROID-GO

© June 1, 2019

Ahora que estás al día con el artículo sobre el adaptador Qwiic, aquí tienes otro proyecto Qwiic que le proporcionara a tu ODROID-GO el don de la visión infrarroja (IR).



## El Punto G: Tu Destino para todas las Cuestiones Relacionadas con Juegos Android: Google Stadia

© June 1, 2019

Sería negligente si no mencionara al menos Google Stadia. Se trata de un servicio de juegos via streaming que promete infundir una nueva vida a tu experiencia de juego con el ODROID-XU4; incluso el ODROID-C1 de menor potencia debería, en teoría, beneficiarse de este servicio de Google. ¿Qué diablos es [▶](#)



## Logotipo de Inicio Personalizado: Creando una Imagen de Inicio a Medida para el ODROID-C2

© June 1, 2019

¿Alguna vez has querido crear un logotipo de inicio personalizado para el ODROID-N2? Siguiendo las simples instrucciones, el logotipo aparecerá en los primeros segundos del arranque mientras se carga el sistema operativo. ¡A Disfrutar!



## ODROID-N2: Análisis del Buen Rendimiento en Pruebas con Linux

© June 1, 2019

El ordenador de placa reducida más reciente de Hardkernel, el ODROID-N2 es nuestra máquina más novedosa. ¡Ven a verla!



## Kodi y Mame Avanzado en ODROID-XU4 - Parte 1

© June 1, 2019

Esta es una guía para configurar Kodi con Mame en el SBC ODROID-XU4, lo cual lo convierte en un buen centro multimedia y de juegos. Incluye todos los pasos para instalar el software necesario en Ubuntu Linux



## ODROID-N2: Pruebas de Rendimiento

© June 1, 2019

Pruebas de rendimiento de ODROID-N2 de Hardkernel, una nueva placa que reemplaza el ODROID-N1 cancelado



## Conociendo un ODROIDian: Ry (@lordhardware)

© June 1, 2019

Nuestra columna mensual en la que presentamos a un miembro de nuestra comunidad sobresaliente, ¡vente, tenemos galletas!

# La Versión Alpha de emuELEC ahora disponible para el ODROID-N2

© June 1, 2019 By @shanti Juegos, ODROID-N2



EmuELEC es una imagen de emulación que te permite ejecutar juegos retro: Atari, NES, SNES, TG16, PSP, PSX, N64, Genesis / Megadrive y muchos más. Para ver a emuELEC en acción, mira el video de <https://youtu.be/ZPq0lt1Xcl0>. Este es la primera versión de mi proyecto, el cual está basada en **CoreELEC**. CoreELEC es una distribución de Linux "lo suficientemente funcional" para ejecutar Kodi en el típico hardware de bajo coste, que en sí mismo es una bifurcación menor de **LibreELEC**. Sin embargo, emuELEC es un proyecto de emulación puro que no incluye Kodi.



**Figura 1 - Menú principal del sistema operativo de emulación emuELEC**

El WiFi debería ser fácil de configurar editando el archivo llamado wifi.txt en la raíz de la tarjeta sd (edítalo justo después de grabar la imagen en la tarjeta SD), o creando un archivo wifi.txt en /storage/.config/wifi.txt. Pon los detalles de tu Wifi dentro del archivo como se muestra a continuación (no ponga nada más):

```
TUSSID:TUPASSWORD
```



Figura 2 - Configuración del sistema operativo de emulación emuELEC en ODROID



Figura 3 - Configuración del sistema operativo de emulación emuELEC en ODROID

A continuación, puedes seleccionar el script wifi de la categoría "RetroPie" en Emulation Station. Si todo ha salido bien deberías tener wifi. Por ahora solo se conecta a wlan0, y para el N2 necesitarás un dongle wifi. Para añadir música de fondo, copia los archivos MP3 en /storage/roms/BGM y habilita BGM en Emulation Station.

Usuario SSH: root Contraseña SSH: emuelec

La imagen incluye:

- Emulationstation
- advancemame
- PPSSPP
- Reicast
- Retroarch con muchos núcleos incluidos.
- Soporte para música de fondo (lee mp3 en /storage/roms/BGM/)
- Fácil integración de roms desde memorias USB externas (simplemente coloca un archivo llamado emuelecroms dentro de tu carpeta /roms en el USB)
- Gestor de descargas de temas



Figura 4 - Emulador Arcade ejecutándose en la imagen emuELEC de ODROID

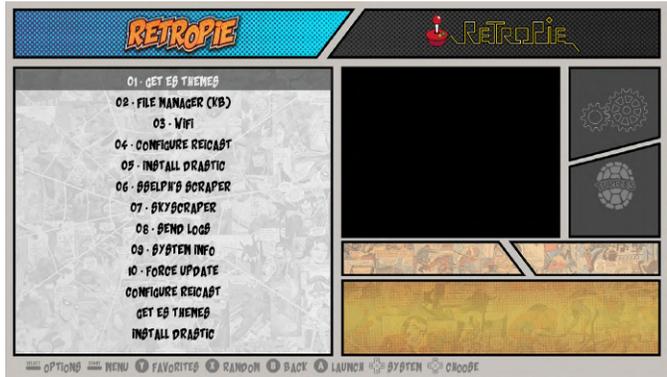


Figura 5 - RetroPie ejecutándose en la imagen emuELEC de ODROID



Figura 6 - TurboGrafx ejecutándose en la imagen emuELEC de ODROID

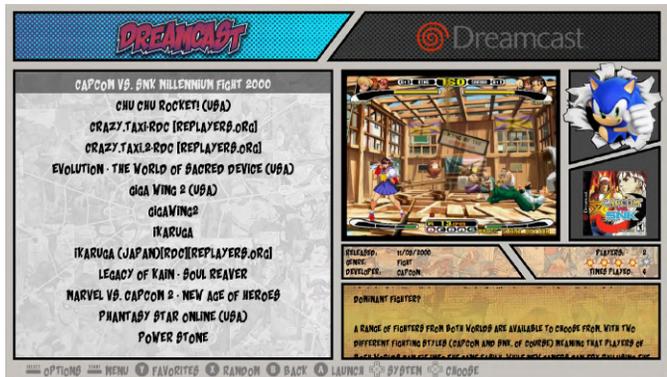
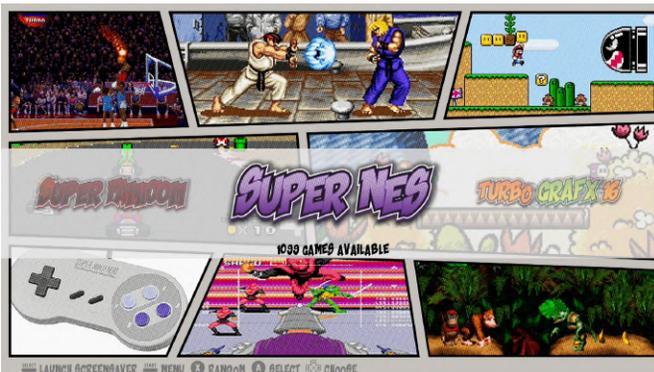


Figura 7: Emulador Dreamcast ejecutándose en la imagen emuELEC de ODROID



**Figura 8: Emulador SuperNES ejecutándose en la imagen emuELEC de ODROID**

#### Problemas conocidos:

- Cuando se sale de un emulador, a veces el búfer de imagen no se actualiza y la imagen permanece congelada, aunque puedes controlar el sistema y escuchar el sonido.
- Después de regresar a Emulation Station desde un emulador (si el primer problema no ocurre), en la pantalla aparecen muchos desgarros.
- Si utilizas las instantáneas de video, tras arrancar, tarda un poco en cargar los videos, pero una vez que se cargan, funcionan como es debido.
- A veces (bastante raro), puedes escuchar ruido de fondo durante una fracción de segundo
- Algunos de los scripts incluidos en RetroPie no funcionan en su totalidad
- Si tienes BGM habilitado y Emulationstation entra en modo salvapantallas, la música continuará reproduciéndose

- Para instalarlo, simplemente graba la imagen en una tarjeta microSD usando etcher (o tu programa favorito). Espero que les guste, y hacérmelo saber si encontráis otros problemas. Ten en cuenta que se trata de una versión alfa, y algunas cosas podrían no funcionar como deberían o quizás no lleguen ni a funcionar.

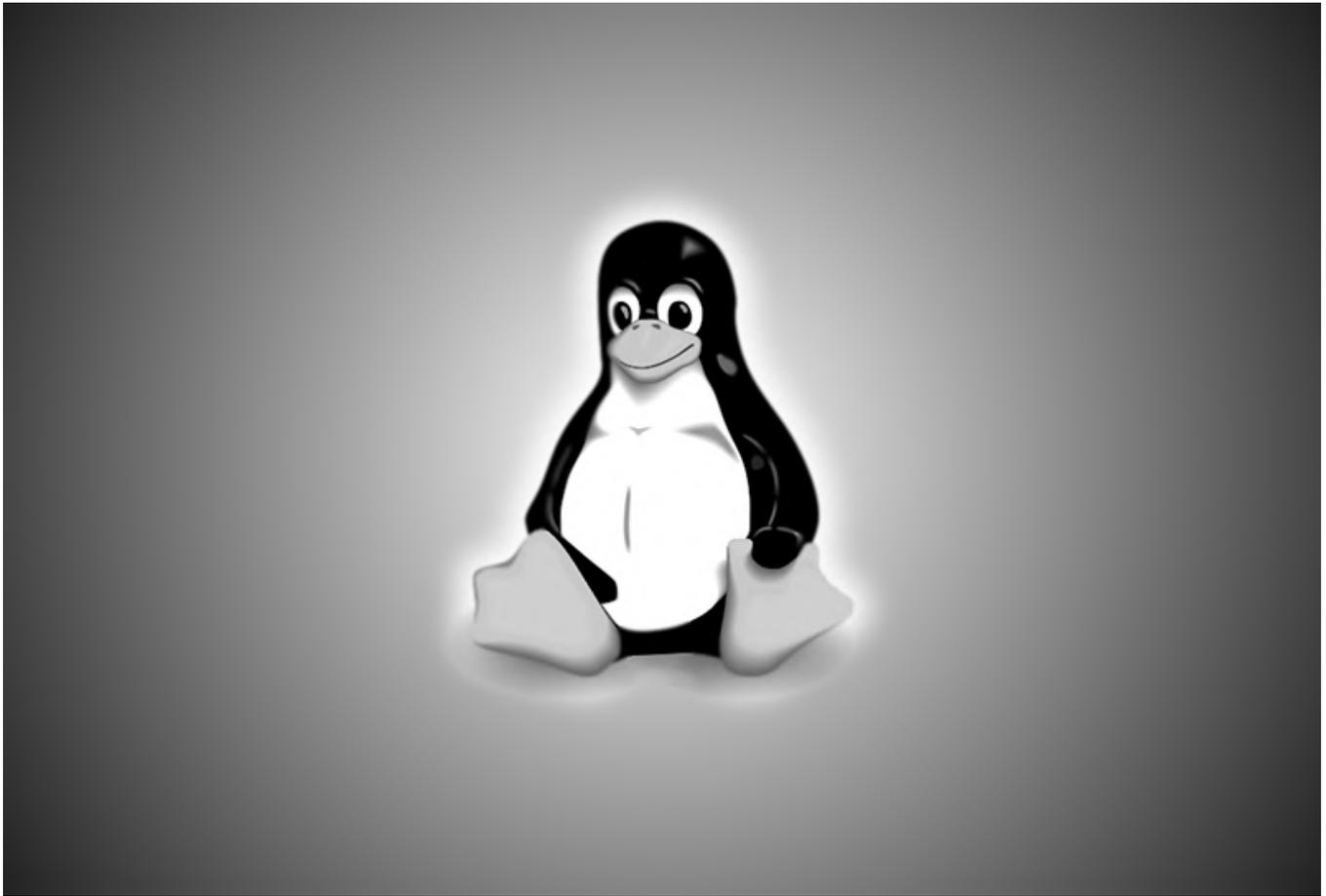


**Figura 9 - Street Fighter se ejecuta sin problemas en la imagen emuELEC de ODROID**

La última versión de emuELEC está disponible en <https://github.com/shantigilbert/EmuELEC/releases/tag/v2.5.2>. Para comentarios, preguntas y sugerencias, visita el post original en <https://forum.odroid.com/viewtopic.php?f=182&t=34355>, or Discord en <https://discord.gg/QqGYBzG>.

# Logotipo de Inicio Personalizado: Creando una Imagen de Inicio a Medida para el ODROID-C2

© June 1, 2019 By Justin Lee Android, Linux, ODROID-C2, Mecaniquero, Tutoriales



Puedes crear un logotipo de inicio personalizado para el ODROID-C2 siguiendo las sencillas instrucciones que se detallan a continuación. El logotipo aparecerá en los primeros segundos del arranque mientras se carga el sistema operativo. El formato de imagen básico del logotipo de arranque del ODROID-C2 se describe en la siguiente sección.

## Formato

```
Image Format : 24-bit Windows BMP image or 24-bit
Windows Gzipped BMP image (Without meta-data)
Image Size : 1280 by 720
Color Depth : 24bpp
File Name : 'boot-logo.bmp' or 'boot-logo.bmp.gz'
```

Puedes encontrar un archivo bmp de muestra en: [https://wiki.odroid.com/\\_media/en/boot-logo.bmp.gz](https://wiki.odroid.com/_media/en/boot-logo.bmp.gz) Recomendamos utilizar GIMP o KolourPaint como programas de edición. Aquí tienes las opciones avanzadas:

## GIMP

- Exportar como BMP de Windows - Opciones de compatibilidad: no escribir información sobre el espacio de color - Opciones avanzadas: color de 24 bits. - Nombre: "boot-logo.bmp"

## KolourPaint

- Guardar imagen como - Filtro: imagen de Windows BMP - Convertir a: Color de 24 bits

## Limitación del tamaño

Debes mantener el tamaño de tu archivo bmp por debajo de los 2 MB, ya que la partición del logotipo está limitada a 2 MB. El formato BMP Gzip es compatible, así que, si el tamaño supera los 2 MB, puedes usar el archivo bmp.gz.

```
$ gzip boot-logo.bmp
$ ls -l boot-logo.bmp.gz
```

## Opción de escalado automático

El uboot del ODROID-C2 admite el escalado de imagen en el logotipo de inicio, de modo que el logotipo que se muestre quedará redimensionado automáticamente por el modo output descrito en boot.ini. Por ejemplo, si utilizamos el modo "1024x600p60hz", el logotipo de inicio se mostrará a 1024x600, aunque el tamaño real del archivo bmp sea de 1280x720.

## Reemplazo del logo de inicio

El ODROID-C2 analiza la existencia de los siguientes componentes por orden numérico. boot-logo.bmp en la partición VFAT boot-logo.bmp.gz en la partición VFAT datos del logo en la partición LOGO de Android

## Android

En Android, puedes reemplazar el logotipo de inicio por una imagen personalizada. Hay dos métodos para cambiar la imagen del logotipo de arranque. Añadir una imagen a la partición VFAT. Reescribir los datos de la imagen en la partición LOGO de Android usando fastboot.

Método 1: VFAT Copia el nuevo boot-logo.bmp, o boot-logo.bmp.gz, a la partición VFAT.

Método 2: Partición logo de Android Si deseas reemplazar los datos del logotipo en la partición del logotipo, sigue estos pasos:

En primer lugar, debes acceder a la línea de comandos de U-Boot mientras presionas la tecla 'ENTER' cuando tu ODROID-C2 se esté encendido y ejecutar el comando fastboot desde U-Boot y conectarte a tu ordenador de escritorio mediante un cable micro USB.

```
odroidc2# fastboot
```

A continuación, ejecuta el comando fastboot desde tu ordenador de escritorio.

## PC HOST

```
$ fastboot flash logo boot-logo.bmp.gz
```

O

```
$ fastboot flash logo boot-logo.bmp
```

## Aviso

Si vas a utilizar los datos bmp en la partición del logotipo, asegúrate de que NO haya un archivo boot-logo.bmp.gz en tu área VFAT, ya que U-Boot verifica primero si hay boot-logo.bmp/boot-logo.bmp.gz en el área VFAT y luego chequea la partición del logotipo.

## Ubuntu

Con Ubuntu, la opción de mostrar el logotipo NO está incluida por defecto. Por lo tanto, debes agregar la imagen del logotipo de inicio en la partición VFAT. El método que usa la partición LOGO no está disponible en Ubuntu.

## Cómo añadir el comando showlogo en boot.ini

### 1080p60hz

En U-Boot, la visualización por defecto del logotipo funciona a una resolución de pantalla de 1080p60hz. Por lo tanto, no es necesario que agregues/modifiques ningún comando relacionado, pero asegúrate de que el archivo del logotipo de inicio se encuentre en las ubicaciones que hemos mencionado anteriormente. Si utilizas otra resolución que no sea 1080p60hz, tienes que añadir comandos a tu boot.ini antes de ejecutar bootcmd. En primer lugar, comprueba si el comando 'showlogo' se encuentra en tu boot.ini, Si no es así, ten en cuenta lo siguiente:

## Android

```
showlogo ${hdmimode}
```

## Ubuntu

```
# Boot Arguments
if test "${display_autodetect}" = "true"; then usb
pwren; hdmix edid; fi
if test "${m}" = "custombuilt"; then setenv cmode
"modeline=${modeline}"; fi
```

```
### You should add the following lines after
**hdmix edid** command.
```

```
showlogo ${m}
setenv logopt "osd1,loaded,0x3f800000,${m}"
```

```
# Boot Arguments - Add logo args on the existing
```

```
bootargs parameter
setenv bootargs "root=UUID=e139ce78-9841-40fe-
8823-96a304a09859 rootwait ro ${condev}
no_console_suspend hdmimode=${m} ${cmode}
m_bpp=${m_bpp} vout=${vout} fsck.repair=yes
net.ifnames=0 elevator=noop disablehpd=${hpd}
max_freq=${max_freq} maxcpus=${maxcpus}
monitor_onoff=${monitor_onoff}
disableuhs=${disableuhs}
mmc_removable=${mmc_removable}
usbmulticam=${usbmulticam} ${hid_quirks}
logo=${logoopt}"
```

## Resolución nativa de imagen personalizada

Si quieres utilizar una resolución nativa para una imagen bmp como 1920×1080, 1024×600 (para VU7+) o 800×480 (para VU7), configura arg[2]/arg[3] en el comando showlogo de la siguiente forma:

```
# help showlogo
showlogo - Displaying BMP logo file to HDMI screen
with the specified resolution

Usage:
showlogo [ ]
           resolution - screen resoluion on HDMI
screen
           '1080p60hz' will be used by
default if missing
           bmp_width (optional) - width of logo bmp
file
           '1280' will be used by default if
missing
           bmp_height (optional) - height of logo bmp
file
           '720' will be used by default if
missing
```

Reemplaza la imagen del logotipo de inicio por la tuya tal y como hemos descrito anteriormente y luego modifica el comando 'showlogo' en boot.ini. Aquí tienes unos ejemplos:

### Tamaño de la imagen del logotipo de ancho 1920 y alto 1080

Si la resolución de tu monitor es 1920×1080 y deseas configurar un archivo bmp en 1920×1080, configura el comando en boot.ini con:

```
setenv hdmimode "1080p60hz"
showlogo ${hdmimode} 1920 1080
```

### Tamaño de la imagen del logotipo de ancho 1024 y alto 600

```
setenv hdmimode "1024x600p60hz"
showlogo ${hdmimode} 1024 600
```

### Tamaño de la imagen del logotipo de ancho 800 y alto 480

```
setenv hdmimode "800x480p60hz"
showlogo ${hdmimode} 800 480
```

## Truco para el problema del Logo con VU7+

En algunos casos muy concretos, cuando VU7+ utiliza una fuente de alimentación adicional, pueden aparecer colores extraños y la pantalla puede parpadear durante la inicialización en la etapa u-boot. Para resolver este problema, recurre a la siguiente solución:

### Actualizar u-boot

Haz clic en el siguiente enlace para descargar el cargador de arranque para ajustar las pantallas a 1024x600p60hz, modo DVI: [http://dn.odroid.com/S905/BootLoader/ODROID-C2/c2\\_vu7plus\\_splash\\_20180720](http://dn.odroid.com/S905/BootLoader/ODROID-C2/c2_vu7plus_splash_20180720) Una vez descargado, copia el archivo tar.gz a /media/boot y arranca el sistema C2

```
# cd /media/boot
# tar xvfz c2_vu7plus_splash_20180720.gz
# cd ./sd_fuse
# ./sd_fusing.sh /dev/mmcblk0
```

(reboot)

### Configurar boot.ini

Algunos puntos dentro de boot.ini tiene que ser modificados.

```
### set "display_autodetect" as "false"
setenv display_autodetect "false"

### block a default "m" and change "m" as
"1024x600p60hz"
setenv m "1024x600p60hz"
```

```

### HDMI DVI/VGA modes
### set "vout" as "dvi"
setenv vout "dvi"

### turn on USB power
usb pwren

### add "logoopt"
setenv logoopt "osd1,loaded,0x3f80000,${m}"

### add "logo=${logoopt}" in "bootargs"
setenv bootargs "root=UUID=e139ce78-9841-40fe-8823-96a304a09859 rootwait ro ${condev}
no_console_suspend logo=${logoopt} hdmimode=${m}
${cmode} m_bpp=${m_bpp} vout=${vout}
fsck.repair=yes net.ifnames=0 elevator=noop
disablehpd=${hpd} max_freq=${max_freq}
maxcpus=${maxcpus} monitor_onoff=${monitor_onoff}
disableuhs=${disableuhs}
mmc_removable=${mmc_removable}
usbmulticam=${usbmulticam} ${hid_quirks}"

```

## Referencias

Puedes consultar el historial de este problema en estas páginas de los Foros ODROID: <https://forum.odroid.com/viewtopic.php?f=141&t=29262#p209113><https://forum.odroid.com/viewtopic.php?f=141&t=31590#p229069>

## Código fuente de muestra

Esta sección describe el historial de cambios del código U-boot. En el caso de 1024x600p60hz, no necesitas seguir esta sección, aunque consúltala y usa el `U-boot` pre-compilado. [https://wiki.odroid.com/odroid-c2/application\\_note/software/bootlogo#update\\_u-boot](https://wiki.odroid.com/odroid-c2/application_note/software/bootlogo#update_u-boot)

u-boot/board/hardkernel/odroidc2/odroidc2.c : en la función `board_late_init` <https://github.com/hardkernel/u-boot/blob/odroidc2-v2015.01/board/hardkernel/odroidc2/odroidc2.c#L468>

```

#ifdef CONFIG_DISPLAY_LOGO
    /* run_command("showlogo 1080p60hz", 0); */
    run_command("showlogo 1024x600p60hz", 0);
#endif

u-boot/common/cmd_showlogo.c

```

```

https://github.com/hardkernel/u-boot/blob/odroidc2-v2015.01/common/cmd_showlogo.c#L119
/*
if (NULL == getenv("vout_mode"))
    setenv("vout_mode", "hdmi");
*/
setenv("vout_mode", "dvi");

```

Esta guía está disponible en la wiki de ODROID en el siguiente enlace: [https://wiki.odroid.com/odroid-c2/application\\_note/software/bootlogo](https://wiki.odroid.com/odroid-c2/application_note/software/bootlogo)

## Animación de arranque en Android

En el caso de Android, puedes usar el método `bootanimation.zip` para mostrar tu logotipo personalizado usando animación. Por favor dirígete a este sitio: <https://android.googlesource.com/platform/frameworks/base/+master/cmds/bootanimation/FORMAT.md>

El sistema selecciona un archivo zip de inicio con animación de las siguientes ubicaciones.

```

/system/media/bootanimation.zip
/oem/media/bootanimation.zip

```

Antes de proceder a la copia, debes cambiar los permisos del sistema de archivos raíz a `r/w` y copiar tu `bootanimation.zip` en la carpeta `/system/media/`.

```

shell@odroidc2:/ $ su
root@odroidc2:/ #
root@odroidc2:/ # mount -o rw,remount /
[ 357.892532@2] EXT4-fs (mmcblk0p2): re-mounted.
Opts: (null)

```

## Muestras de animación de inicio

Aquí tienes ejemplos de videos con Android Marshmallow en ODROID-C2. Para mostrar estos ejemplos de animaciones de inicio, se han utilizado los siguientes archivos de referencia: [Ten en cuenta las fuentes de los archivos]. <https://forum.xda-developers.com/android/themes/bootanimation-android-marshmallow-t3180984>



**HARDKERNEL**

Ejemplo 1 de animación de inicio

### Distribución del directorio

desc.txt part0 part1 part2 part3 part4 desc.txt

```
814 214 60
c 1 30 part0
c 1 0 part1
c 0 0 part2
c 1 64 part3
c 1 15 part4
```

Tamaño de los archivos png es 814 por 214.



**HARDKERNEL**

Figura 2 - Ejemplo 2 de animación de inicio

### Distribución del Directorio

desc.txt Part0 Part1 desc.txt

```
800 1280 24
p 1 0 Part0
p 0 0 Part1
```

El tamaño de los archivos png es de 800 por 1280.

La siguiente guía puedes encontrarla en la wiki de ODROID en [https://wiki.odroid.com/odroid-c2/application\\_note/software/bootlogo](https://wiki.odroid.com/odroid-c2/application_note/software/bootlogo).

# ZoneMinder en ODROID-XU4: Monta tu Propio Sistema de Vigilancia

June 1, 2019 By Michele Maticchione ODROID-XU4, Tutoriales



ZoneMinder (ZM) es un conjunto integrado de aplicaciones que proporciona una completa solución de vigilancia que le permite al usuario capturar, analizar, registrar y monitorizar una zona a través de cualquier CCTV o cámara de seguridad.

## Principales características

- Monitorización desde cualquier lugar: ZoneMinder cuenta una interfaz basada totalmente en la web a la que puede acceder desde cualquier dispositivo con acceso a Internet
- Utiliza cualquier cámara: ZoneMinder te permite usar cualquier cámara analógica o por IP.
- Tus datos bajo control: ZoneMinder está completamente en local; Te permite tener tus datos a buen recaudo y controlar a dónde van.
- Se puede ejecutar en pequeños y grandes entornos: adecuado para uso doméstico y de pequeñas empresas, así como para implementar soluciones empresariales multiservidor. Es compatible con

muchas plataformas, incluida la tecnología ARM (ODROID está basado en una plataforma ARM)

- Mantente al tanto de todo lo que sucede: ZoneMinder te permite buscar información de una forma muy intuitiva. Profundiza en lo que estás interesado en cuestión de segundos.
- Actualizado y sin coste alguno: un equipo comprometido con el código abierto mantiene muy activo ZoneMinder

Recientemente, he trasladado la aplicación ZoneMinder de mi viejo Radxa Rock Pro (una placa ARM) al ODROID-XU4 que es más potente. La mejor y más sencilla instalación que encontré fue la del ODROID-XU4 - Ubuntu 16.04.3 LTS - ZoneMinder 1.29\*.

NAME	FUNCTION	SOURCE	EVENTS	HOURS	DAY	WEEK	MONTH	ARCHIVED	INDEXES	ORDER	MARK
Outdoor - Ingresso	Modect	192.168.1.200	56	0	56	56	56	0	1	A,Y	
Outdoor - Dehor	Modect	192.168.1.18	16	0	16	16	16	0	1	A,Y	
Outdoor - Retro	Modect	192.168.1.202	3	0	2	2	2	0	1	A,Y	
Indoor - Zona game	Monitor	192.168.1.203	0	0	0	0	0	0	1	A,Y	
Indoor - Camera Dadi	Monitor	192.168.1.204	0	0	0	0	0	0	1	A,Y	
Indoor - Camera Ale	Monitor	192.168.1.205	0	0	0	0	0	0	1	A,Y	
Indoor - Remosa	Modect	192.168.1.206	0	0	0	0	0	0	1	A,Y	
			74	0	74	74	74	0	7		

Figura 1 - Consola ZoneMinder, configurada con 7 cámaras.

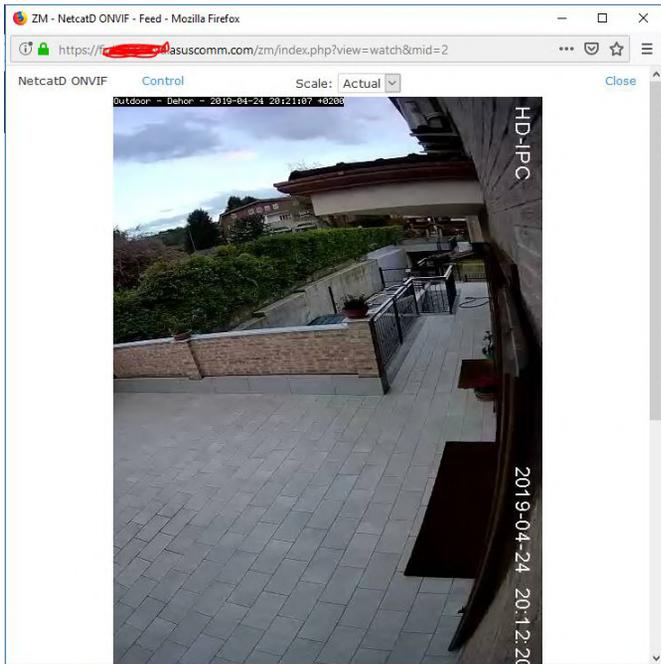


Figura 2 - ZoneMinder mostrando la cámara #2

## Installation

Vamos a instalar ZoneMinder en nuestra placa ODROID-XU4. En tu tarjeta SD, instala la imagen de Ubuntu 16.04.3 LTS (la versión 4.14.y) proporcionada por Hardkernel en la dirección: [https://wiki.odroid.com/odroid-xu4/os\\_images/linux/ubuntu\\_4.14/20171213](https://wiki.odroid.com/odroid-xu4/os_images/linux/ubuntu_4.14/20171213)

Luego actualiza el sistema:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt dist-upgrade
$ sudo apt install linux-image-xu3
$ sudo apt autoremove
$ sudo reboot
```

Ahora instala LAMP (Linux, Apache, MySQL, PHP) en la placa:

```
$ sudo apt install apache2
$ sudo apt install mysql-server
$ sudo apt install php libapache2-mod-php php-mysql
```

Ahora instala ZoneMinder 1.29; \*\*

```
$ sudo -i
```

Modifica la configuración de MySQL (no es necesaria para ZM 1.32 o superior):

```
$ rm /etc/mysql/my.cnf (this removes the current symbolic link)
$ cp /etc/mysql/mysql.conf.d/mysqld.cnf /etc/mysql/my.cnf
$ nano /etc/mysql/my.cnf
```

En la sección [mysqld] añade lo siguiente:

```
$ sql_mode = NO_ENGINE_SUBSTITUTION
```

A continuación, reinicia MySQL:

```
$ systemctl restart mysql
```

Ahora instala ZoneMinder:

```
$ apt-get install zoneminder
```

Crea una base de datos ZM (no necesario para ZM 1.32 o superior) :

```
$ mysql -uroot -p < /usr/share/zoneminder/db/zm_create.sql
$ mysql -uroot -p -e "grant lock tables,alter,drop,select,insert,update,delete,create,index,alter routine,create routine,trigger,execute on zm.* to 'zmuser'@localhost identified by 'zmpass';"
```

Añade los permisos:

```
$ chmod 740 /etc/zm/zm.conf
$ chown root:www-data /etc/zm/zm.conf
$ chown -R www-data:www-data /usr/share/zoneminder/
```

Activa los módulos y la configuración de ZoneMinder:

```
$ a2enmod cgi
$ a2enmod rewrite
$ a2enconf zoneminder

$ a2enmod expires
$ a2enmod headers
```

Activa ZoneMinder al inicio del sistema:

```
$ systemctl enable zoneminder
$ systemctl start zoneminder
```

Configura php.ini con la zona horaria correcta:

```
$ nano /etc/php/7.0/apache2/php.ini
```

Inserta tu zona horaria:

```
$ [Date]
$ ; Defines the default timezone used by the date
functions
$ ; http://php.net/date.timezone
$ date.timezone = America/New_York
```

Reinicia Apache:

```
$ systemctl reload apache2
```

¡Hecho!

Ahora, puedes localizar la página web de ZM en `http://IP_de_tu_placa/zm` y añadir cámaras. \*\*\* El mismo procedimiento es válido para ZM 1.32 (la última versión estable de ZM), evitando así tener que usar comandos innecesarios. En este caso, antes de instalar ZM, deberás añadir el repositorio de ZoneMinder de Isaac Connor (responsable del mantenimiento de ZM en Ubuntu):

```
$ add-apt-repository ppa:iconnor/zoneminder-1.32
```

Ten en cuenta que ZM 1.32 también está disponible para Ubuntu 18.04. Ten en cuenta que ffmpeg en la imagen 18.04 de Hardkernel no es compatible y debe adaptarse

(consulta <https://forum.odroid.com/viewtopic.php?f=95&t=33745#p247653>)

## Notas

- Hay otras posibles combinaciones, que discutiremos más adelante;
- No hay necesidad de instalar un repositorio dedicado de ZoneMinder puesto que ZM 1.29 ya está disponible en los repositorios de Ubuntu 16.04 (xenial)
- Puede funcionar cualquier tipo de cámara: cámaras ffmpeg y mjpeg conectadas mediante Wi-Fi, cable Ethernet o USB

## Otros ajustes y trucos

Tras la instalación, verifica si ZM se encuentra funcionando bien en tu LAN. Consulta las publicaciones anteriores para la instalación de ZM y la modificación de FFMPEG, si estas utilizando la distribución HK 18.04.

Abre un navegador web (se recomienda Firefox): `http://ipservidorLAN/zm` (por ejemplo, `http://192.168.1.200/zm`). Si aparece la página de ZoneMinder, todo está bien.

Configura el acceso ZM con el modo autenticado: dirígete a Opciones/Sistema. Marca `OPT_USE_AUTH`. Cierre el navegador, luego ábrelo de nuevo y se te pedirá que introduzcas `Userid/password`. Utiliza `admin/admin *`.

Modificar el `userid/password`: dirígete a Opciones/Usuarios. Cambia el nombre de usuario `admin` por lo que sea e introduce una nueva contraseña. Cierra el navegador, luego ábrelo de nuevo y se te pedirá que introduzcas el `Userid/password`. Usa los que has creado recientemente.

Comprueba `AUTH_TYPE = builtin`. Si no, configúralo: verifica `AUTH_RELAY = hashed` y nuevamente si no, configúralo. Cambia `AUTH_HASH_SECRET` por lo que sea \*

Habilitar https para ZoneMinder: esta es la forma fácil. Existe otro método que utiliza certificados de `letencrypt`. Entra en el terminal y añade SSL a Apache2 creando primero un certificado autofirmado:

```
$ make-ssl-cert generate-default-snakeoil --force-overwrite
```

Esto creará los siguientes archivos:

```
/etc/ssl/private/ssl-cert-snakeoil.key
/etc/ssl/certs/ssl-cert-snakeoil.pem
```

Activa el módulo SSL de Apache:

```
$ a2enmod ssl
```

Activa el host virtual ssl por defecto de Apache:

```
$ a2ensite default-ssl
```

Reinicia Apache:

```
$ service apache2 restart
```

Ahora deberías poder acceder al servidor web mediante `https://ipservidorLAN/zm` (por ejemplo, `https://192.168.1.200/zm`).

Habilita el reenvío de puertos del puerto 443 en tu router. En un modelo ASUS, dirígete a WAN/Port Forwarding:

Nombre del servicio: HTTPS\_Port\_Forwarding Puerto: 443 IP Local: LANserverip (ej.: 192.168.1.200) Puerto lcoal: 443 Protocolo: TCP

Ahora debería poder acceder al servidor web mediante `https://ipservidorWAN/zm` (por ejemplo, `https://5.157.104.224/zm`).

Puedes vincular dinámicamente tu ipservidorWAN a un nombre de host estático (por ejemplo, `https://nombredelhost/zm`) a través de las funciones de DNS dinámico de tu router (por ejemplo, en ASUS y DLink) o mediante servicios como NOIP. De esta forma, puedes exponer de forma segura el servicio ZoneMinder en WAN: acceso autenticado y acceso por https cifrados, de modo que toda la información en WAN quedara cifrada a través de una dirección web estática.

Si se te solicita que reinicies ZoneMinder, dirígete al Terminal e introduce el siguiente comando:

```
$ sudo service zoneminder restart
```

## Cómo configura la cámara

Hay cuatro formas de hacerlo: tres fáciles, la cuarta manual:

1. Usar PRESETS 2. Utilizar la detección automática ONVIF. 3. Tirar de Google: buscar por "inserting {modelo de la camara} ZoneMinder" 4. Configuración manual

Para que te venga la inspiración, aquí tienes la configuración de mis tres tipos de cámaras:

A: FOSCAM FI8910W (antiguo modelo PTZ MJPEG usado para interiores)

General Name: whatever Server: None Source Type: Remote Function: Monitor (sólo para ver)

Source Remote Protocol: HTTP Remote Method: Simple Remote Host: 192.168.1.203 Remote Host Port: 200 Remote Host Path: /videostream.cgi?

user=user-on-camera&pwd=pwd-on-camera&rate=3 Target Colorspace: 24 bit color Capture Width: 640 Capture Height: 480

Deja el resto de valores por defecto, veremos el control sobre los comandos PTZ más adelante.

B: FOSCAM FI8918W (modelo PTZ FFMPEG utilizado para interiores)

General Name: whatever Server: None Source Type: Ffmpeg Function: Modect Para ver y grabar cuando se detecte movimiento)

Source Source Path: rtsp://user-on-camera:pwd-on-camera@192.168.1.206:200/videoSub Remote Method: TCP Options: blank Target Colorspace: 24 bit color Capture Width: 640 Capture Height: 480

Deja el resto de valores por defecto, el control de los comandos PTZ lo veremos más adelante.

C: Los modelos en forma de cúpula Jidetch POE PTZ FFMPEG y LEFTEK POE PTZ FFMPEG que se utiliza para exteriores

General Name: whatever Server: None Source Type: Ffmpeg Function: Modect (Para ver y grabar cuando se detecte movimiento)

Source Source Path: rtsp://user-on-camera:pwd-on-camera@192.168.1.200:554/1/h264minor Remote Method: TCP Options: blank Target Colorspace: 24 bit color Capture Width: 640 Capture Height: 480

Deja el resto de valores por defecto, el control de los comandos PTZ lo veremos más adelante.

## Cómo grabar en SSD

- Instala cifs-utils

```
$ sudo apt-get install cifs-utils
```

- Crea un directorio para montar el SSD

```
$ mkdir /home/odroid/Documents/STORAGE
```

-Edita /etc/fstab

```
$ sudo nano /etc/fstab
```

Inserta las siguientes líneas:

Nota: el UUID correcto del SSD se puede obtener con `ls -l /dev/disk/by-uuid /*`

```
#SSD
UUID=BE28A67028A626FD
/home/odroid/Documents/STORAGE auto
nosuid,nodev,nofail 0 0
```

Presiona CTRL+O, luego CTRL+X

```
$ sudo mount -a
```

Reinicia, si fuera necesario.

```
$ mkdir /home/odroid/Documents/STORAGE/ZM-IMAGES
$ mkdir /home/odroid/Documents/STORAGE/ZM-IMAGES/images
$ mkdir /home/odroid/Documents/STORAGE/ZM-IMAGES/events
$ mkdir /home/odroid/Documents/STORAGE/ZM-IMAGES/temp
$ sudo nano /etc/fstab
```

Inserta las siguientes líneas para crear un enlace:

```
/home/odroid/Documents/STORAGE/ZM-IMAGES/images /var/cache/zoneminder/images none
defaults,bind 0 0
/home/odroid/Documents/STORAGE/ZM-IMAGES/events /var/cache/zoneminder/events none
defaults,bind 0 0
/home/odroid/Documents/STORAGE/ZM-IMAGES/temp /var/cache/zoneminder/temp none
defaults,bind 0 0
```

Presiona CTRL+O, luego CTRL+X

```
$ sudo mount -a

$ sudo chown -R www-data:www-data
/home/odroid/Documents/STORAGE/ZM-IMAGES
```

Se podría seguir un procedimiento similar para montar un USB HD o un disco de almacenamiento en red.

### Cómo grabar cuando se detecte movimiento

En la consola, haga clic en las zonas de la columna. En la cámara, quieres MOTION DETECTION. Haga clic de nuevo en la imagen. Fija el área que desea utilizar para la detección. Deja sin cambios si quieres chequear toda el área. Ahora configura lo siguiente:

Name: whatever Type: Active Unit: Pixels Alarm Colour: whatever Alarm check method: Blobs Min-Max Pix Thresh: 40 - 0 Filter: 5 -5 Min/max Alarmed area: 200 - 0 Min/max Filtered area: 20 - 0 Min/max Blob area: 20 - 0 Min/max Blob: 2 - 0 Overload Frame: 2

Guarda. Pon en la consola y la cámara en modo MODECT (función de columna).

### Cómo implementar comandos PTZ

Habilitar los comandos PTZ: dirígete a Opciones/Sistema: marca en OPT\_CONTROL. En la consola, haga clic en la fuente de la columna, para la cámara que quieras activar

PTZ:

Control Controllable: flag Control Type: Selecciona tu modelo de cámara Control device: según tu modelo (en mi caso blanco) Control address: camera\_ip:camera\_port (en mi caso 192.168.1.200:8999)

# Cómo Crear una Consola de Juegos Retro Monku - Parte 2: Añadir Mandos y Botones Personalizados

© June 1, 2019 By Brian Ree Juegos, ODROID-C1+, ODROID-C2



Esta es una continuación del artículo de la consola de juegos retro del número de mayo de 2019, en el que aprendimos cómo ensamblar los componentes básicos de una consola de juegos retro. Esta entrega te ayudará a personalizar los mandos y los botones para el proyecto.

## Herramientas necesarias

- Un pequeño juego de destornilladores que contiene varios destornilladores phillips de reducido tamaño
- Un kit de soldadura que incluya soldador de punta fina, estaño con fundente y control de temperatura, si es posible.
- Cinta de carroceros o cinta de pintor.
- Una superficie de trabajo limpia y libre de energía estática.
- Taladro y una buena selección de brocas.
- Monitor o TV con soporte HDMI para probar el dispositivo.

Yo he utilizado este kit de soldadura (<https://amzn.to/2jyy85h>). Cuesta alrededor de 19.00\$ y, aunque es barato, incluye todo lo que necesitas para este proyecto y el soldador viene con varias puntas, incluida una punta fina y un control de temperatura que es maravilloso. No había soldado nada desde el instituto y fui capaz de hacerlo sin fastidiarla. Tampoco es que tuviera un buen pulso, me había tomados 3 cafés antes de ponerme manos a la obra. Así que, si yo pude hacerlo, tú también puedes.

## 1: Componentes necesarios

- Un ODROID-C1+/ODROID-C2: 35.00\$/46.00\$ (<https://www.hardkernel.com/shop/odroid-c1/>)
- Una carcasa: 4.50\$ (<https://www.hardkernel.com/shop/odroid-c2-c1-case-black/>)
- Un juego de botones: 7.99\$ (<https://amzn.to/2HNRekl>)

- Un juego de cables de placa de pruebas: 7.99\$ (<https://amzn.to/2wgeXo8>)
- Dos Tarjetas Micro SD de 64GB: 16.99\$ (<https://amzn.to/2JyNjLZ>)
- Un Cable HDMI: 1.00\$ (<https://www.hardkernel.com/shop/hdmi-1-4-cable-type-a-a/>)
- Un cable micro USB: 1.50\$ (<https://www.hardkernel.com/shop/micro-usb-cable/>)
- Tres adaptadores de alimentación USB 5V/2A: 11.99\$ (<https://amzn.to/2VXlfY0>)
- Un mando GameSir conectado por cable: 17.00\$ (<https://bit.ly/2JxEu4V>)

El coste total del proyecto tal y como hemos descrito descrito anteriormente y sin incluir los gastos de envío o las herramientas necesarias es de alrededor de unos 104\$. Sin embargo, si excluyes los botones de control personalizados o puedes comprar las piezas individualmente, puedes llegar a ahorrar en torno a unos 16\$. Tampoco necesitas dos tarjetas micro SD. Me gusta contar con una de repuesto por si alguna de ellas falla, no obstante, el conjunto que he mencionado antes tiene un precio bastante bueno. Además, la tarjeta está muy bien valorada y, según mi experiencia personal, solo he tenido un error inesperado en las 12 que he usado regularmente durante la configuración y desarrollo de estos dispositivos. Si dispones de un cable HDMI, un cable micro USB y un cargador USB de 5V/2A, puedes ahorrarte aún más en el proyecto. Si bien el precio estimado es de alrededor de unos 104\$, probablemente puedas reducir el coste a unos 80\$. No está mal, una vez que veas de lo que son capaces estos dispositivos.

## 2: Introducción y Objetivos

Este artículo te mostrará en detalle cómo montar una consola de videojuegos Monku Retro 1 (ODROID-C1 +) o Monku Retro 2 (ODROID-C2) desde cero. En primer lugar, algunos conceptos básicos de hardware: el botón de hardware permite un reseteo completo del dispositivo. El botón de software se conecta a los pines GPIO y está configurado para mantener un registro del tiempo que se mantiene pulsado y ejecuta un determinado comando en función del tiempo registrado. Así, por ejemplo, mantener

pulsado el botón 2 segundos reinicia el software, mantenerlo pulsado 4 segundos apaga el software, etc. Si estas interesado en añadir botones de control personalizados, aquí lo vamos a tratar en profundidad. Todo, desde los componentes hasta los pins, y cómo soldarlo todo. Esta característica es opcional. Tu dispositivo funcionará bien sin ellos, pero en el caso de que aparezca un fallo, tendrás que apagar y encender la consola manualmente y esta acción podría dañar el sistema de archivos de Linux. El botón de reinicio del hardware facilita esta tarea, aunque el botón de control del software para el apagado del sistema operativo y el reinicio es bastante más seguro. Siempre puedes volver atrás y añadirlos, no hay problema. También nos detendremos en la configuración del software, incluida la instalación y configuración de Ubuntu, retroarch y antimicro en las partes 2 y 3 de esta serie de tutoriales.

### Características R1/C1+

- Calidad ODROID
- Botón de control de software personalizado
- Reseteo del hardware a medida
- Soporte para Atari 2600, Atari 7800, ColecoVision, MSX-1, MSX-2, NES, GameBoy, GameBoy Color, Sega SG-1000, Sega Mark III y Sega Master System configuradas y listas para usar
- Retroarch con XBM, scripts personalizados para monitorizar el botón de software, iniciar retroarch, mantener antimicro
- Configurado para un consumo bajo de memoria y para usarse con el mando incluido
- Todas las ROM testeadas para ver si se cargan y se asocian correctamente con su emulador
- Completo entorno de escritorio Linux cuando no está activo el modo kiosco del juego a través de antimicro

### Funciones del botón del software R1/C1+:

- 2 segundos pulsado: se reinicia software
- 4 segundos pulsado: se apaga el software
- 6 segundos pulsado: se desactiva el modo kiosco del juego
- 8 segundos pulsado: la resolución se cambia a 1024x768x32bpp y se reinicia

- 10 segundos retenidos: la resolución se cambia a 720p x 32bpp y se reinicia.

### Características del R2/C2:

- Calidad ODROID
- Botón de control de software personalizado
- Reseteo del hardware a medida
- Soporte para Atari 2600, Atari 7800, Atari Lynx, ColecoVision, MSX-1, MSX-2, NES, GameBoy, GameBoy Color, Virtual Boy, SNES, GameBoy Advance, WonderSwan Pocket/Color, NEO
- GEO Pocket/Color, Sega SG-1000, Sega Mark 3, Sega Master System, Sega Genesis, Sega GameGear, NEC Turbo Graphics 16 y NEC Super Graphics configurados y listos para usar
- Retroarch con XBM, scripts personalizados para monitorizar el botón de software, iniciar retroarch, mantener antimicro
- Configurado para un consumo bajo de memoria y para usarse con el mando incluido
- Control total del entorno de escritorio de Linux por gamepad cuando no está activado el modo kiosco de juego mediante antimicro

### Funciones del botón de software R2/C2:

- 2 segundos pulsado: se reinicia el software
- 4 segundos pulsado: se apaga el software
- 6 segundos pulsado: se desactiva el modo kiosco del juego
- 8 segundos pulsado: configura el video como automático para el modo VGA, posiblemente modifique retroarch.cfg para el audio USB si está presente
- 10 segundos pulsado: configura el modo video a 720p, cambia retroarch.cfg a audio HDMI

### 3: La configuración

Lo primero es lo primero: examinemos las herramientas y los componentes, colócalos y prepárete para montarlos. Nosotros tenemos un juego de destornilladores electrónicos. Si has montado un ODROID-GO, el mismo juego de destornilladores debería valerte. Ten en cuenta que tenemos nuestro dispositivo, un ODROID-C2 que es el que se muestra a continuación, este tutorial se aplica igualmente al ODROID-C1+ o la versión ODROID-C2

de este dispositivo. El ODROID-C1+ que uso para juegos retro de 8 bits es el ODROID-C2 que me gusta usar para todos los sistemas de 8 bits y 16 bits más los dispositivos portátiles: los ejecuta de maravilla. Probablemente pueda ejecutarlos en un C1+ pero me gusta más el C2 para los juegos de 16 bits, por alguna razón. Tenemos nuestros botones de control personalizados, conectores, carcasas y herramientas listas para usar.



Tenemos nuestro soldador, observa el soporte y la esponja para limpiar el metal y el control de temperatura del metal. El control de la temperatura es crucial, aunque, es posible que no necesites uno, especialmente si eres bueno en la soldadura. El control de temperatura te permite aplicar calor elevado durante una pequeña cantidad de tiempo. He leído algunos tutoriales y he visto algunos videos y me he dado cuenta que aplicar calor durante mucho tiempo no es tan bueno como aplicar mucho calor durante un período de tiempo más corto. Creo que el resultado final es más limpio, no me veo sujetando el metal sobre un conector esperando a que la gota de soldadura se derrita por completo. Otra cosa que necesitas es una punta fina sobre el metal.

Nuevamente, si eres habil, tal vez puedas arreglártelas con una punta diferente, pero yo no, una punta de soldadura fina fue mi salvación. Truco: si no dispones de un dispositivo de sujeción para la placa de circuito que te ayude con la soldadura, he descubierto que la cinta de carroceros funciona muy bien. Entonces se derretirá, aunque no debería calentarse. Te mostraré cómo usar la cinta para sujetar lo que estés soldando y así conseguir una unión de soldadura limpia. Funciona igual de bien que los dispositivos de sujeción de placa de circuitos, es súper barata y tiene más flexibilidad. Si realmente quisiera, podrías soldar al revés, colgando del techo si usaras suficiente cinta.

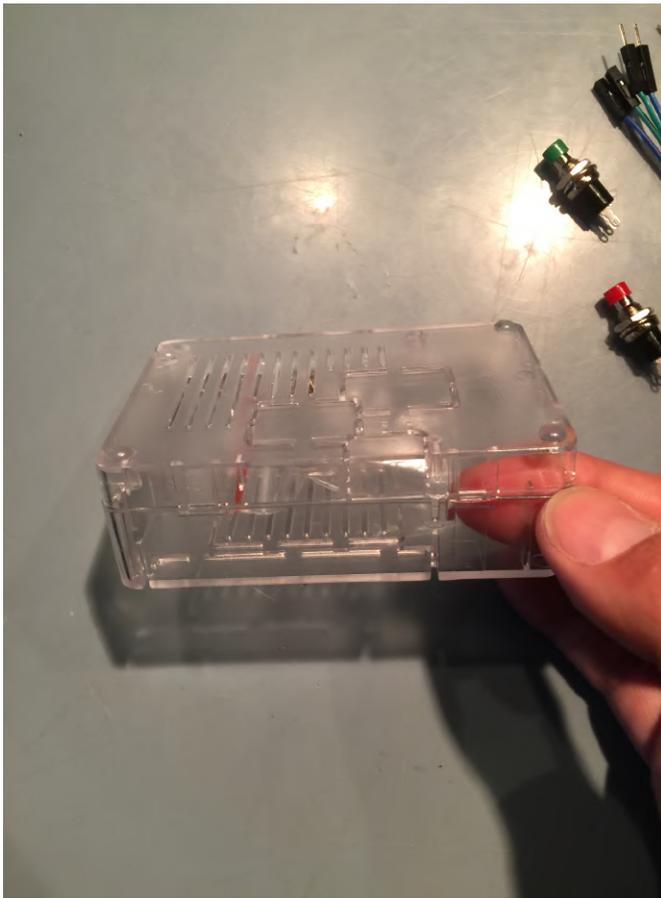


La carcasa puede ser un tanto jodida de abrir. Deberías necesitar poco o ningún esfuerzo, pero se necesitan uno o dos chasquidos para hacerla abrir de esta forma. Podrías forzarla, pero a mí me fastidia un montón tener los clips de sujeción rotos, así que fui bastante cuidadoso a la hora de abrirla. Lo primero que querrás hacer es desmontar la parte posterior de la carcasa. Esto se puede hacer aplicando una ligera presión en la parte inferior de la carcasa empujándola hacia la derecha, mientras empujas la parte superior

de la carcasa hacia la izquierda con un ligero empujón, tal y como se muestra a continuación. Aplicas una pequeña fuerza en este punto, no se abrirá, pero se separará.

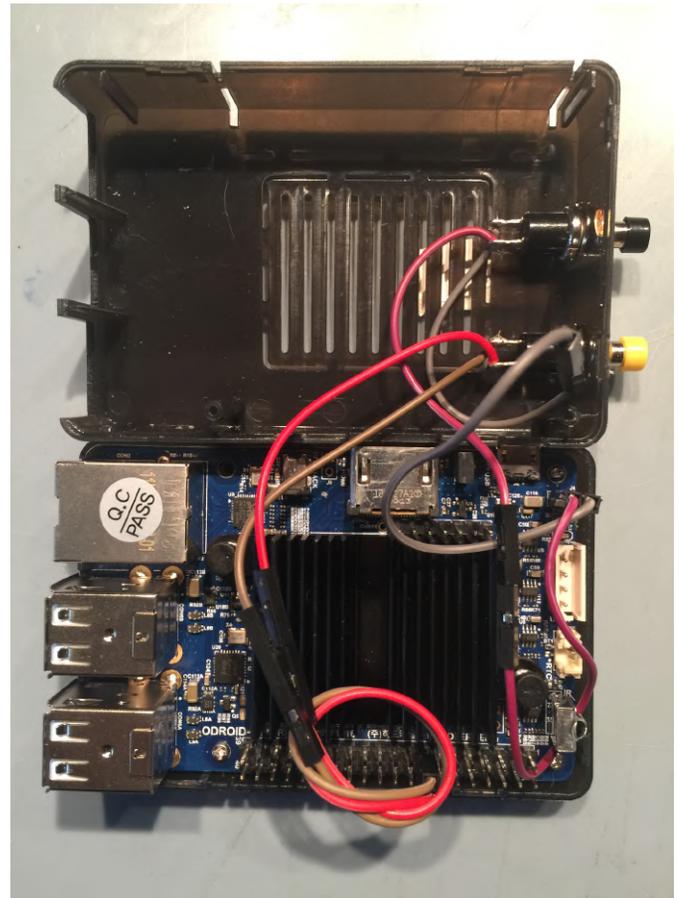


Una vez que se haya separado el clip posterior, podemos movernos hacia el lado que tiene dos clips. Consejo: estos dos clips laterales son un poco complicados. Puedes abrir el frontal colocando el dedo dentro de la carcasa y aplicando la misma fuerza a la que recurrimos para abrir el clip posterior. Para abrir el último clip de este lateral, deberá usar la uña. Desliza la uña dentro de la junta de la carcasa junto al clip posterior, el primero que separamos, arrastra la uña hacia el lado que tiene los dos clips y se separará el clip restante. Se tarda un poco en hacerlo, pero no te preocupes, tómate tu tiempo. Sin darse cuenta, puede que se vuelva a encajar un clip; es normal, simplemente empieza de nuevo. Lo importante en este punto es contar con una carcasa sin fisuras que albergará nuestra magnífica consola.



Ahora que tenemos la carcasa abierta, puedes acceder al premio que hay en su interior, una pequeña bolsa con tornillos. Sugerencia: si te sobran los tornillos del ODROID-GO, estos son un poco más grandes que los tornillos incluidos y creo que funcionan mejor y son menos propensos a soltarse. Ahora que la carcasa está abierta, tenemos que tomar una decisión sobre el hueco para introducir la tarjeta SD. Normalmente, la carcasa no tendrá acceso a la tarjeta SD a menos que revientes el hueco de entrada de la tarjeta SD. Por lo tanto, si la consola está completamente cerrada y tu tarjeta SD muere, tendrá que cortar con mucho cuidado la entrada de la tarjeta SD o abrir la carcasa. A mí me gusta abrir el hueco de la tarjeta SD en los modelos de desarrollo y así poder abrir y sacar fácilmente las tarjetas SD.

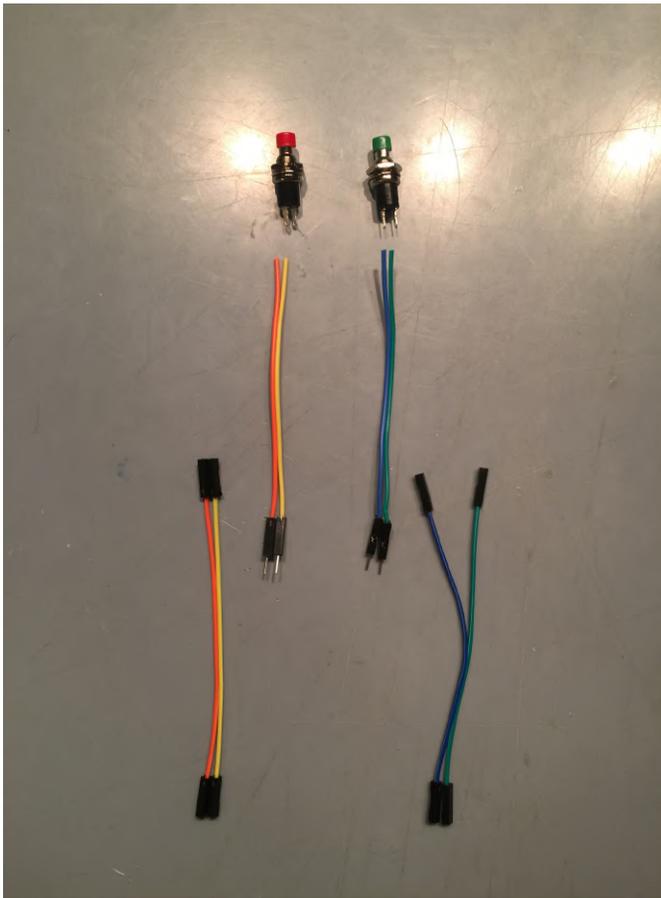
Para cortar el hueco de la entrada de la tarjeta SD, necesitarás una cuchilla de afeitar o un pequeño cuchillo afilado. No te llevará demasiado. Necesitarás unos minutos, no hay una forma súper rápida de hacerlo sin riesgo de dañar la carcasa. Raspa el punto de conexión con la cuchilla de afeitar. Arrástrala a través de los puntos, rayando y cortando el pequeño plástico de conexión. Consejo: puedes colocar la parte inferior de la carcasa de pie y de lado, de esta manera puedes ejercer más presión hacia abajo en dos de las conexiones de plástico del hueco de entrada de la tarjeta SD sin riesgo de dañar la carcasa. A continuación, se muestra el hueco abierto de la tarjeta SD en una Monku Retro 1 (ODROID-C1 +).



A continuación, se muestra una foto de una Monku Retro 1 (ODROID-C1 +) acabada con botones de control personalizados y varias conexiones. Me gusta usar 4 conectores por botón, sé que puedo arreglármelas sólo con dos, pero me gusta tener conexiones adicionales. Para esta opción, necesitarás 4 conectores hembra-hembra y 4 conectores macho-cualquier cosa. Existen otras combinaciones, simplemente asegúrate de colocar los conectores y observar su uso. Asegúrese de que todas las conexiones coincidan. Tendrás que cortar un lado de los conectores de los cables que están soldados a los botones.

#### 4: La soldadura

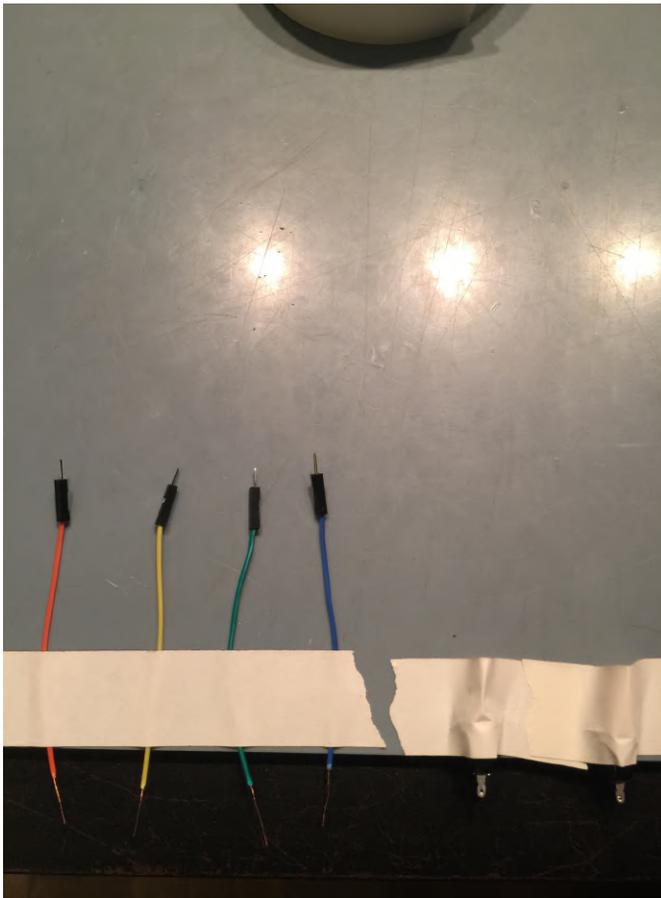
Ahora nos prepararemos para soldar la conexión que necesitamos para resetear el hardware, control directo de la energía de la placa. Ahora sería un buen momento para coger el soldador. Conéctalo y deja que se caliente. Usaremos unos 400 grados, si tienes control de temperatura, para los cables y los botones. Vamos a diseñar los conectores y botones para visualizarlos. Normalmente uso pins de la ranura de expansión del ODROID-GO, pero soy consciente de que no todo el mundo los tiene, así que usé conectores en este montaje. En esta fotografía se observa que utilicé colores erróneos, oops.



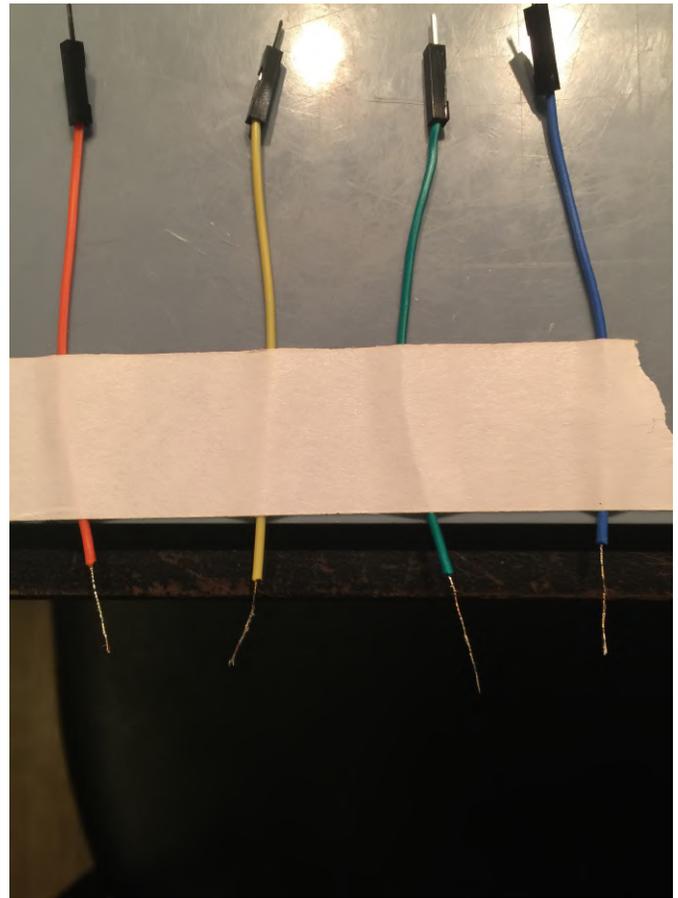
Cogemos los conectores que están más cerca de los botones como se muestra arriba. Luego los cortamos, por un lado. Ahora usa unas tijeras o la uña de tu mano para retirar la funda del cable. Ejerce un poco de presión con las tijeras sobre el cable y gira hasta que observes una rotura en la funda. Es lo mismo que usar una uña. Una vez que la funda se vuelva un poco blanca y parezca que está a punto de romperse, generalmente puedes pellizcarla y tirar ella. Una vez hecho esto, coge la punta de los cables de cobre y gíralos para que queden bien retorcidos. Haz lo mismo con los otros cables de conector.



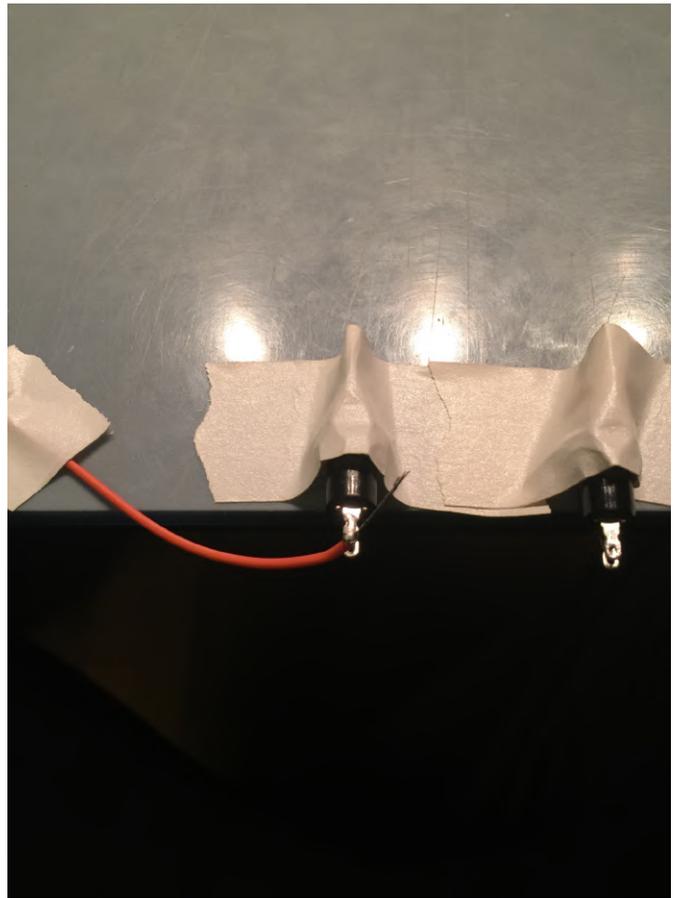
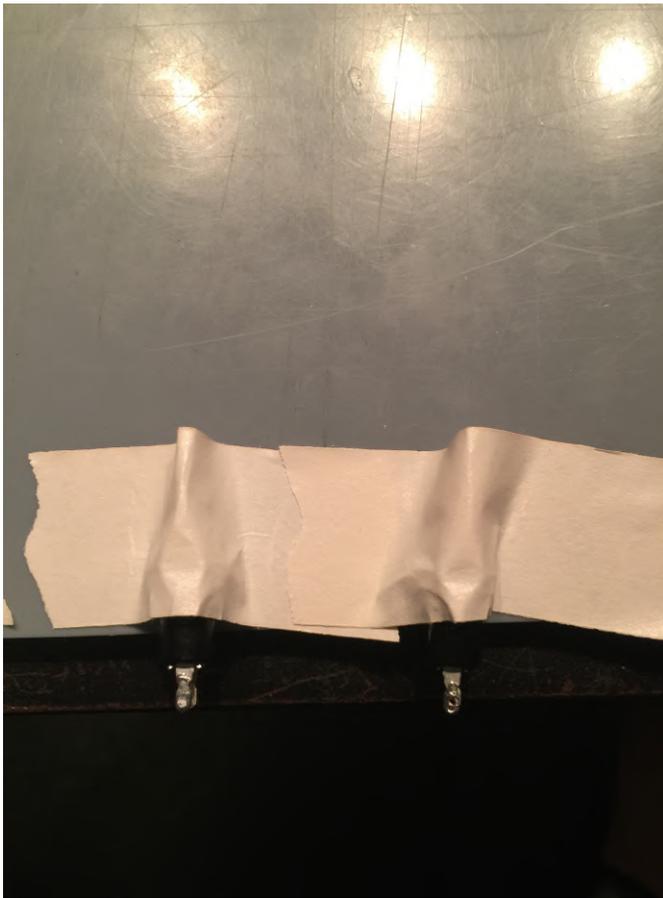
Asegúrate de verificar el estado de tu soldador. Accidentalmente empecé a fundir la carcasa de plástico de mi juego de destornilladores durante el montaje. Coloca los cuatro cables y los dos botones. Prepara el metal, limpia la punta con una esponja húmeda y cubre de estaño el metal con un poco de soldadura y fundente. Asumo que tienes soldadura con el fundente en su interior en este punto. Vamos a continuar con todas las conexiones antes de conectarlas. La soldadura se une casi instantáneamente a las conexiones con estaño. Aunque, no cubro de estaño las placas, me detengo "tan poco tiempo como sea posible" en la placa en sí. Observa que estamos usando la cinta de carroceros para asegurar cada componente.



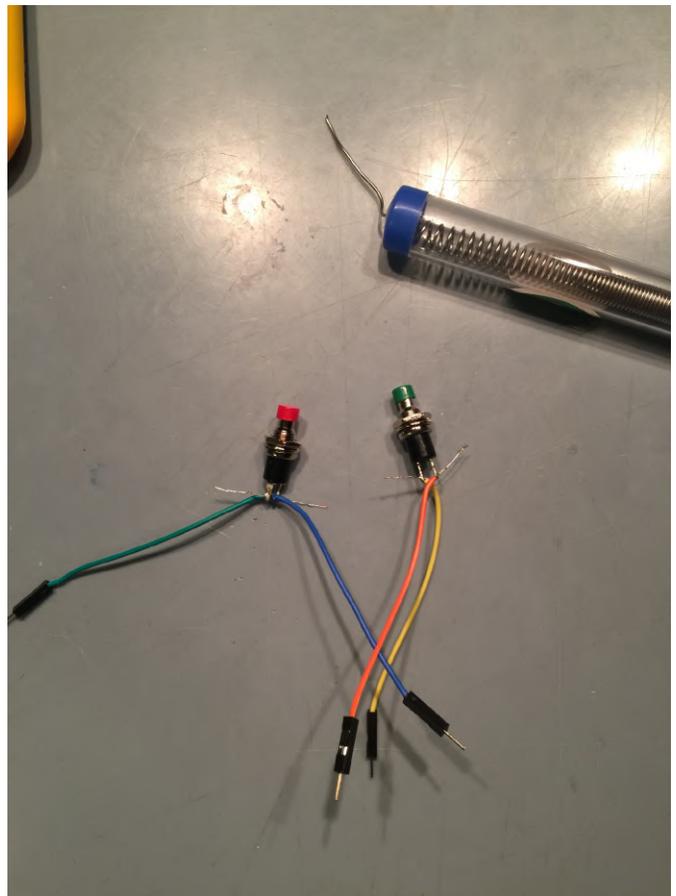
Situa el soldador debajo del cable y simplemente frota la soldadura, la trenza de cobre absorberá la soldadura y podrás extenderla moviendo el metal por el cable. No necesitas mucho, solo un poco en cada uno. Haz lo mismo con los cuatro cables.



Ahora necesitas cubrir de estaño los puntos de contacto del botón. No te preocupe si rellenas uno de los agujeros en el punto de contacto, no será un problema.

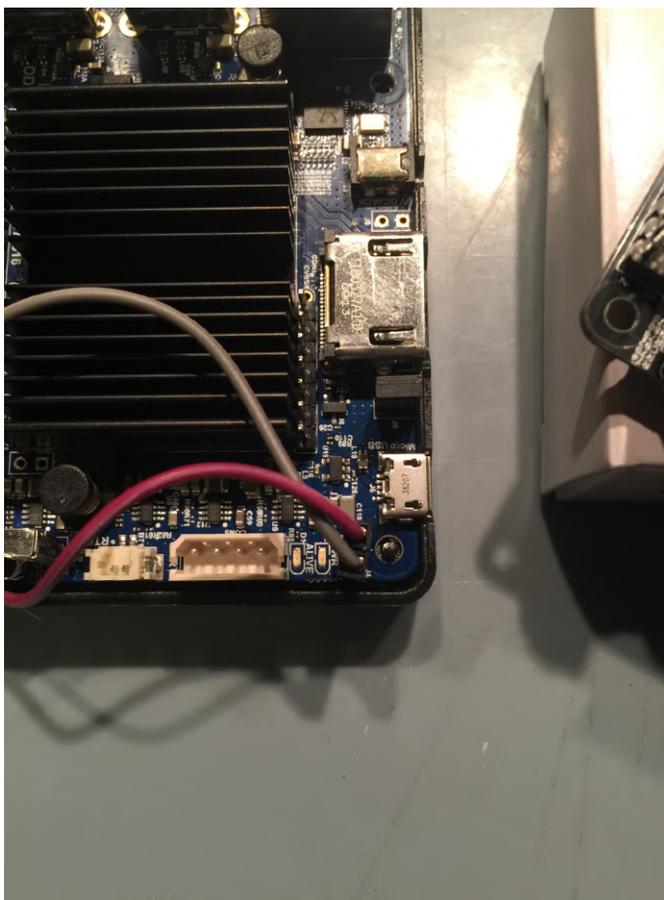


Coloca uno de los cables que pelamos y cubrimos de estaño, manteniendo los colores que elijas durante todo el montaje, dentro del orificio de contacto del botón. Es posible que tenga que forzar ligeramente el metal para lograr que el cable pase. Dobla y asegura el cable al lateral del botón con cinta adhesiva. Ahora suelda el punto de contacto. Haz lo mismo para ambas conexiones en ambos botones. Usa unas tijeras para cortar el exceso de cable. Deberías tener algo como lo que se muestra a continuación.

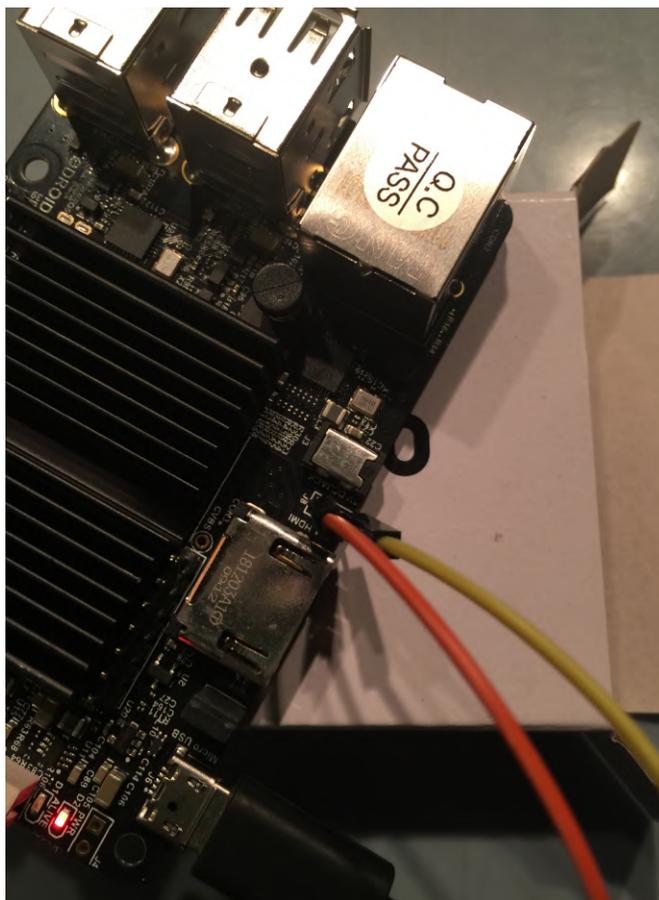


ALERTA: Existe una diferencia entre el C1+ y el C2, de modo que debes detenerte en este punto y comprobar bien cual es la placa que tienes y, en el

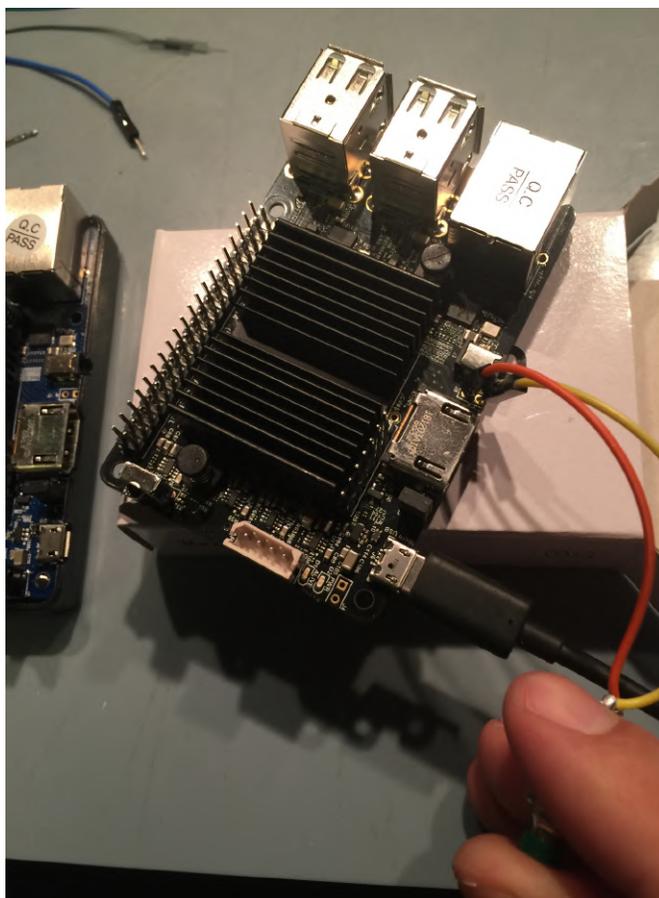
caso de ser un ODROID-C2, qué versión de esta placa. A continuación, se muestra una foto del C1+ con el botón de encendido soldado en la placa con el texto "pwr". La placa C2 tiene la misma marca, pero puede que no sea la ubicación real del botón de encendido. Puedes usar los botones que acabas de montar para averiguar la ubicación del botón de encendido. Coloca los cables del conector macho dentro de los orificios en una de las posiciones disponibles, J8 o pwr, apriétalos ligeramente con el dedo para que hagan contacto. Con la alimentación encendida, pero sin la tarjeta SD ni nada más, presione el botón. Si el LED rojo se apaga, has encontrado el botón de encendido. Si no sucede nada, es posible que tenga un botón defectuoso. Prueba el otro que montaste. Si aún no pasa nada, puede intentar hacer un mal contacto y provocar un cortocircuito en los dos pequeños orificios de las ubicaciones disponibles hasta que identifiques cuál es el que controla la potencia.

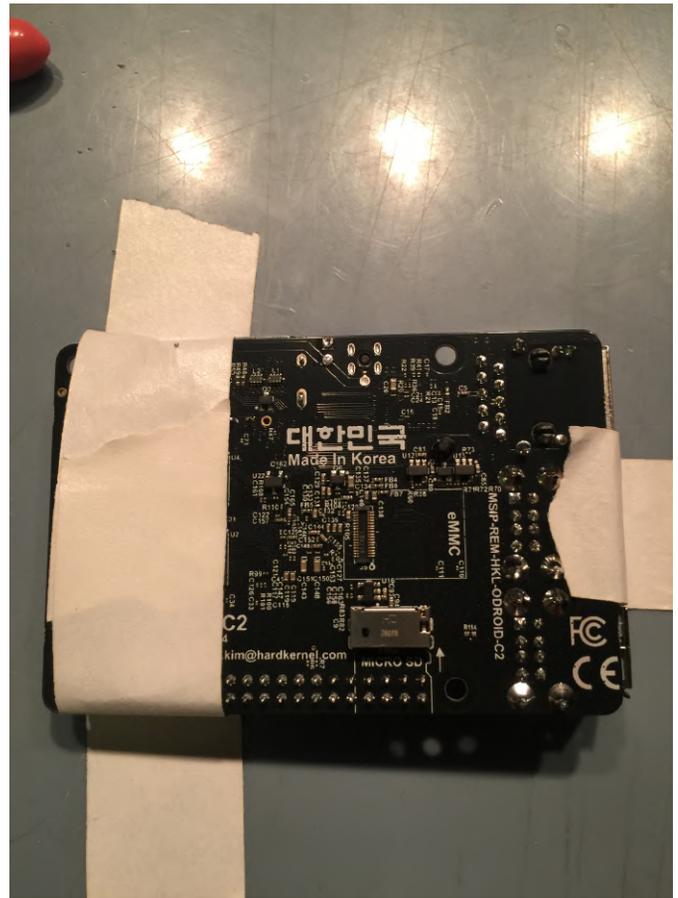
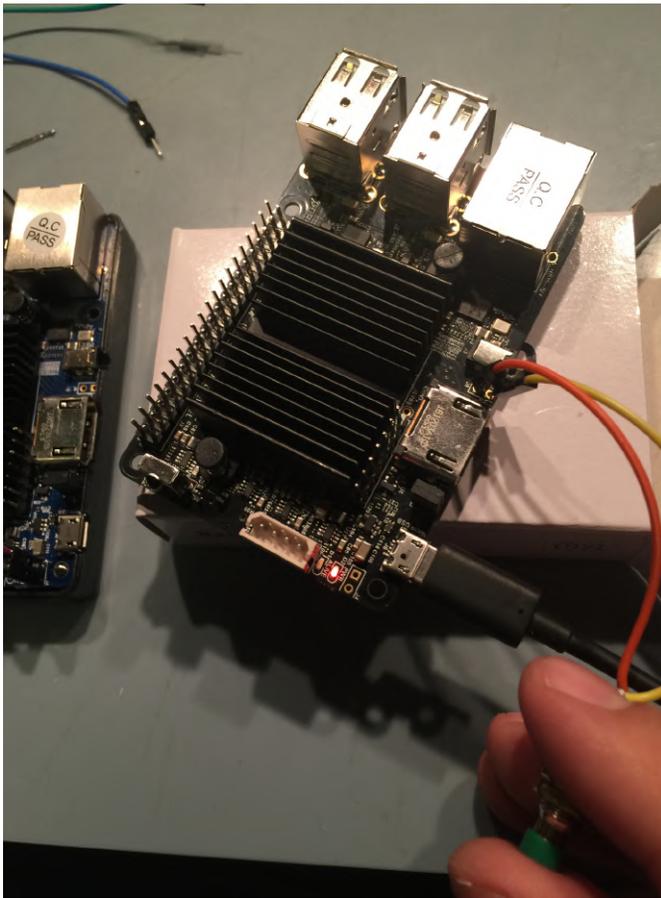


NOTA: Si estás trabajando con un ODROID-C1+ este es el lugar exacto de la placa donde debes añadir los conectores para el interruptor de alimentación. Asegúrate de haberlo siguiendo el proceso descrito anteriormente.



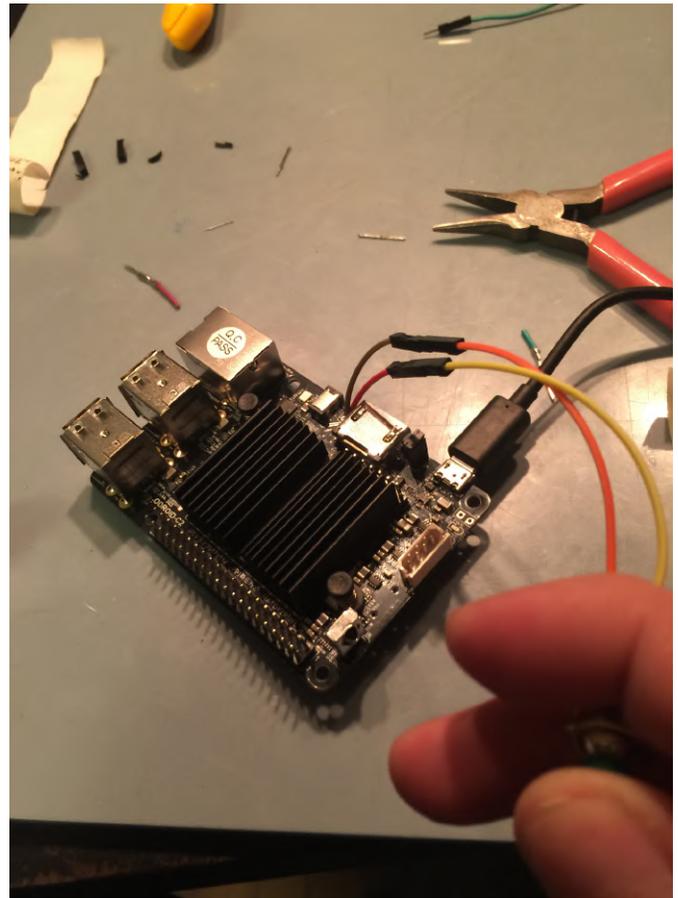
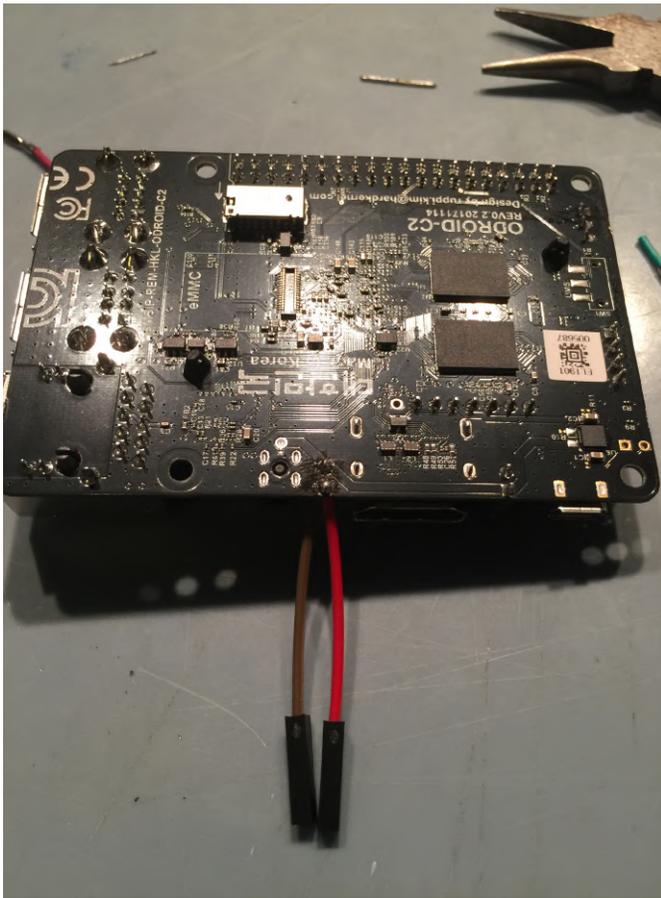
Las siguientes fotos muestran pruebas del botón en acción.



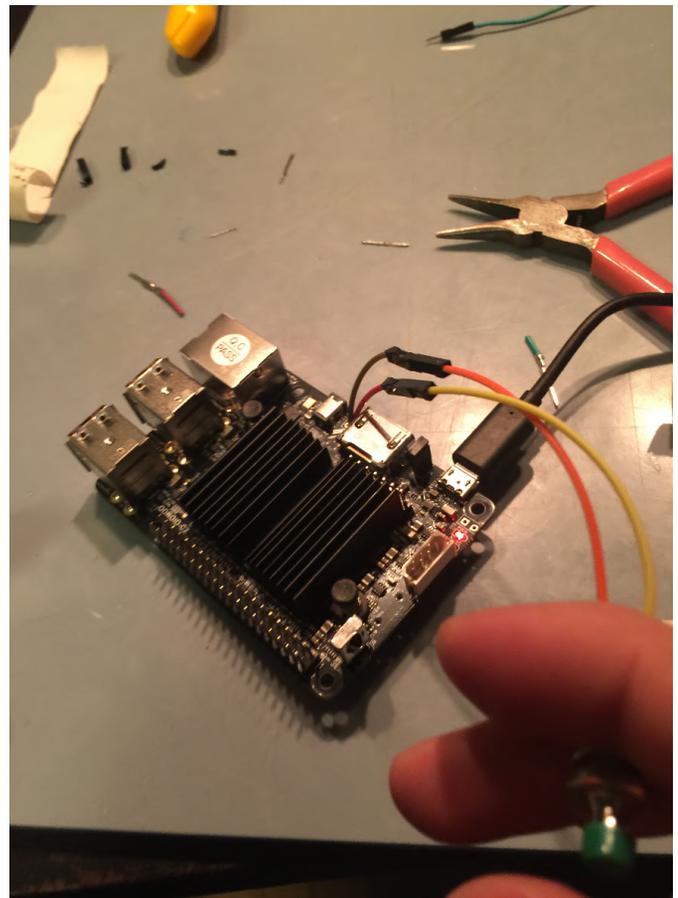


El siguiente paso es soldar la placa. Situa la temperatura de tu soldador en torno a los 450 mas o menos. Asegura la placa a tu superficie de trabajo, asegúrate de tenerla firme para que no se mueva cuando el soldador la toque. Normalmente, usaría pines del clip de expansión del ODROID-GO, pero no todos tenemos uno a nuestro alcance, así que, en lugar de fijar los conectores a los pines, soldaremos el conector a la placa y lo conectaremos en el otro lado.

Pela y retuerce los cables como lo hicimos para los botones. Empieza por el hueco más alejado de la placa, empuja el cable y suéldalo, un poco de soldadura, un poco de tiempo. Repite el proceso para el hueco que está más cerca del borde. Llegados a este punto, deberías tener algo como la siguiente imagen. Para el C1+ esta conexión - cerca de la esquina de la placa - esta un poco ajustada, simplemente ten cuidado, tomate tu tiempo.



Enhorabuena, hemos terminado de soldar. Vamos a probarlo. Conecta uno de tus botones funcionales. Enciende la unidad sin una tarjeta SD. Deberías poder hacer que el LED rojo se apague y se encienda como antes.



Si tienes una tarjeta SD C1+/C2 funcional, es un buen momento para realizar una prueba y ver si todo funciona bien. Si no tiene una tarjeta SD, no te

preocupes, pronto estaremos con el tema del software. Lo único que te queda con el tema que estamos tratando son los agujeros de la carcasa y montar los botones.

En la próxima entrega de este tutorial, veremos cómo crear una carcasa para la consola.

Referencias

[http://middlemind.com/tutorials/odroid\\_go/mr1\\_build.html](http://middlemind.com/tutorials/odroid_go/mr1_build.html)



# Me Vuelvo a Qwiic: Añade Visión IR a tu ODROID-GO

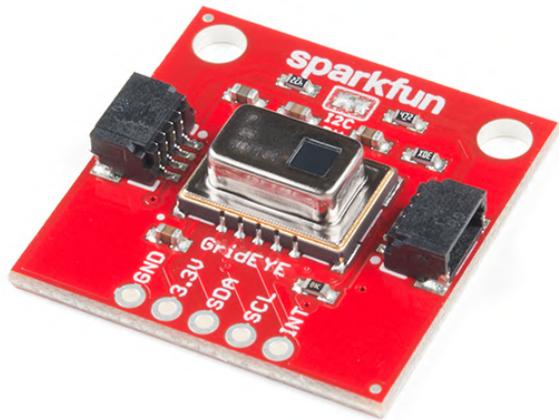
© June 1, 2019 By Dave Prochnow Desarrollo, ODROID-GO



¿Recuerda el proyecto del adaptador Qwiic I2C de SparkFun Electronics (SFE) que montamos en la [número de Mayo de 2019 de ODROID Magazine](#)? ¿Recuerda el proyecto del adaptador Qwiic I2C de SparkFun Electronics (SFE) que montamos en la edición de mayo de 2019 de ODROID Magazine? Oh, ¿No recuerdas haber leído este artículo? Bien, esperaré a que refresques tu memoria. Ahora que estás al día con el artículo sobre el adaptador Qwiic, aquí tienes otro proyecto Qwiic que le proporcionara a tu ODROID-GO el don de la visión infrarroja (IR).

Este tipo particular de visión IR es proporcionada por un sensor matriz IR Panasonic AMG8833 (consulta la Figura 1). Dentro de este sensor hay una matriz en forma de "red" termopila de 8x8. ¿Qué es exactamente una termopila? Bueno, es una elegante palabra para referirse a una pila de termos. No, en serio, una termopila es un dispositivo que aloja una serie de termopares, cada uno de los cuales mide la temperatura a través de un par de cables metálicos

diferentes. A su vez, estos cables miden la diferencia de potencial que se crea en la unión de los cables. ¡Uf! ¿Sabes qué? Me gusta más la definición de "pila de termos".



**Figura 1 - La placa del sensor Qwiic de SparkFun Grid\_EYE. Fotografía cortesía de SparkFun Electronics.**

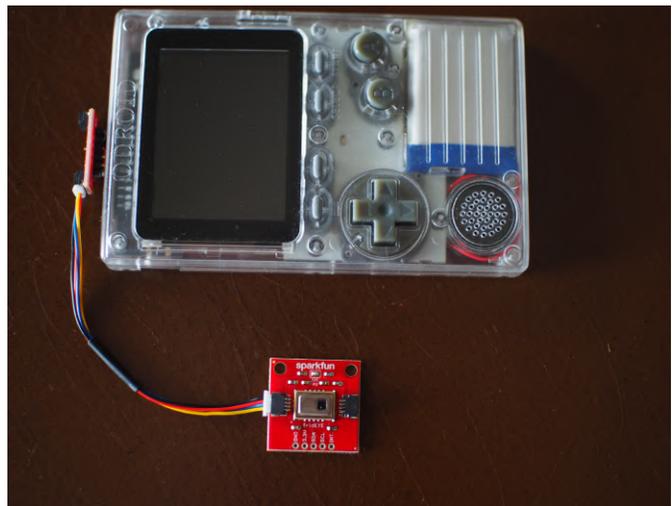
Por supuesto, una matriz visual de 8x8 es de muy baja resolución, pero los objetos colocados a 200-300 mm frente al sensor de Panasonic pueden proporcionar una delicia visual muy colorida en la pantalla LCD de ODROID-GO. Al agregar el Grid-EYE al sistema Qwiic, SFE ha hecho de la visión por infrarrojos un proyecto muy simple “plug and go” para el ODROID-GO.

### Paso a paso

1. Inserta el adaptador Qwiic para la PCB ODROID-GO en el cabezal hembra de entrada/salida de uso general (GPIO) ubicado a lo largo del borde superior del ODROID-GO. Si aún no dispones de este adaptador de PCB, puede aprender a hacer uno recurriendo a la edición de mayo de 2019 de ODROID Magazine.

2. Conecta un cable Qwiic (por ejemplo, 100 mm; SFE PRT-14427) a la placa adaptadora Qwiic.

3. Conecta la matriz de infrarrojos SparkFun Grid-EYE array - AMG8833 (Qwiic) (SFE SEN-14607) en el otro extremo del cable. Un ODROID-GO está listo para escanear en busca de fuentes de calor tal y como se puede apreciar en la Figura 2.



**Figura 2 - Todo conectado y listo para buscar cosas interesantes**

4. Introduce este breve esquema de Arduino (consulte la Figura 3 - Figura 3c) en el Entorno de desarrollo integrado (IDE) de Arduino y compila/carga el esquema en tu ODROID-GO. Recuerda crear tu propio subconjunto de librería Display.h (tal y como se describe en el proyecto del Adaptador Qwiic) desde la librería principal de ODROID-GO. De lo contrario, el esquema no se compilará dentro del IDE de Arduino.

```

1  /*
2  * Using the Panasonic Grid-EYE Sensor
3  * By: Nick Poole
4  * SparkFun Electronics
5  * Date: January 12th, 2018
6  */
7  // MIT License: Permission is hereby granted, free of charge, to any person obtaining a copy of this
8  // software and associated documentation files (the "Software"), to deal in the Software without
9  // restriction, including without limitation the rights to use, copy, modify, merge, publish,
10 // distribute, sublicense, and/or sell copies of the Software;
11 // THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
12 // BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
13 // NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
14 // DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
15 // OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
16
17 // Feel like supporting our work? Buy a board from SparkFun!
18 // https://www.sparkfun.com/products/14568
19
20 // Hardware Connections:
21 // Attach the Qwiic Grid-EYE to the Qwiic Adapter board mounted on your ODROID-GO
22 // Display on the ODROID-GO @ 320x240
23 //
24
25 #include <SparkFun_GridEYE_Arduino_Library.h>
26 #include <Wire.h>
27 #include <Display.h>
28
29 // Use these values (in degrees C) to adjust the contrast
30 #define HOT 40
31 #define COLD 20
32
33 // This table can be of type int because we map the pixel
34 // temperature to 0-3. Temperatures are reported by the
35 // library as floats

```

**Figura 3: Incluye estas tres librerías al principio de tu esquema**

```

35 // Library as floats
36 int pixelTable[64];
37
38 GridEYE grideye;
39
40 ILI9341 lcd = ILI9341();
41
42
43 void setup() {
44   // Start your preferred I2C object
45   Wire.begin();
46   // Library assumes "Wire" for I2C but you can pass something else with begin() if you like
47   grideye.begin();
48   // Four a bowl of serial
49   Serial.begin(115200);
50   //
51   // Setup LCD
52   lcd.begin();
53   lcd.setRotation(1);
54   lcd.fillScreen(BLACK);
55   lcd.setBrightness(255);
56   lcd.setFont(1);
57   lcd.setTextSize(2);
58   lcd.setCursor(16, 3);
59   lcd.setTextColor(LIGHTGREY);
60   lcd.println("ODROID-GO");
61   lcd.setCursor(17, 6);
62   lcd.println("Grid-EYE");
63   lcd.setTextSize(3);
64
65 }
66
67 void loop() {
68

```

**Figura 3a: Configura la pantalla LCD de ODROID-GO para mostrar tus datos IR**

```

67 void loop() {
68   // Initiate the text cursor position
69   lcd.setCursor(0, 0);
70
71   // Loop through all 64 pixels on the device and map each float value to a number
72   // between 0 and 3 using the HOT and COLD values we set at the top of the sketch
73   for(unsigned char i = 0; i < 64; i++) {
74     pixelTable[i] = map(grideye.getPixelTemperature(i), COLD, HOT, 0, 3);
75   }
76
77   // Loop through the table of mapped values and print a character corresponding to each
78   // pixel's temperature. Add a space between each. Start a new line every 8 in order to
79   // create an 8x8 grid
80   for(unsigned char i = 0; i < 64; i++) {
81     if(pixelTable[i] == 0) {
82       Serial.print(" ");
83       lcd.setTextColor(BLACK, BLACK);
84       lcd.print(" ");
85     } else if(pixelTable[i] == 1) {
86       Serial.print("0");
87       lcd.setTextColor(BLUE, BLACK);
88       lcd.print("0");
89     } else if(pixelTable[i] == 2) {
90       Serial.print("0");
91       lcd.setTextColor(ORANGE, BLACK);
92       lcd.print("0");
93     } else if(pixelTable[i] == 3) {
94       Serial.print("0");
95       lcd.setTextColor(RED, BLACK);
96       lcd.print("0");
97     }
98     Serial.print(" ");
99     if((i+1)%8 == 0) {
100      Serial.println();

```

**Figura 3b: Asignación de colores térmicos a la pantalla LCD y al puerto serie**

```

100 Serial.println();
101 lcd.println();
102
103 //
104
105 // In between updates, throw a few linefeeds to visually separate the grids. If you're using
106 // a serial terminal outside the Arduino IDE, you can replace these linefeeds with a clearscreen
107 // command
108 Serial.println();
109 Serial.println();
110
111 // Toss in a delay because we don't need to run all out
112 delay(100);
113
114 }
115

```

**Figura 3c - Finalmente, añade un breve retardo en las actualizaciones del puerto serie y la pantalla**

Desconecta el ODROID-GO, enciéndelo y ve a buscar algunas cosas interesantes para medir. Una taza de

café caliente es un sujeto ideal (consulte la Figura 4). Tu nueva visión IR en ODROID-GO ahora está lista para mostrarte las cosas de tu vida que están demasiado calientes para cogerlas (consulta la Figura 5).



**Figura 4: Esta es una configuración útil para usar tu ODROID-GO y medir el calor IR**



**Figura 5: Pantalla de la misma configuración del ODROID-GO que aparece en la Figura 4**

# El Punto G: Tu Destino para todas las Cuestiones Relacionadas con Juegos Android: Google Stadia

© June 1, 2019 By Dave Prochnow Android, Juegos, ODROID-C0, ODROID-C1+, ODROID-XU4



Sería negligente si no mencionara al menos Google Stadia. Se trata de un servicio de juegos via streaming que promete infundir una nueva vida a tu experiencia de juego con el ODROID-XU4; incluso el ODROID-C1 de menor potencia debería, en teoría, beneficiarse de este servicio de Google.

¿Qué diablos es Google Stadia? En pocas palabras, tal y como se anunció en la Conferencia de desarrolladores de Google 2019, Google Stadia es un sistema de juegos via streaming basado en la nube que permitirá a cualquier persona jugar a cualquier juego en cualquier dispositivo en cualquier momento. Del mismo modo, cuando este sistema se utiliza junto con la red de YouTube de Google, podrás pasar de ver el tráiler del juego a jugar a ese juego en cuestión de minutos.

Dirigido a los juegos AAA de gran presupuesto, Google Stadia promete revolucionar la industria de los juegos de hoy en día. De hecho, durante una

conferencia telefónica sobre la rentabilidad financiera de Alphabet de abril, el jefe de Google, Sundar Pichai, declaró que los productores de juegos de renombre "... están muy ilusionados, porque creo que ven la oportunidad de un cambio, un punto de inflexión, aunque son consciente del reto técnico que supone sacar algo como esto. Pero una vez que se consigue la tecnología y luego ven la experiencia, creo que realmente quien gana es la gente".

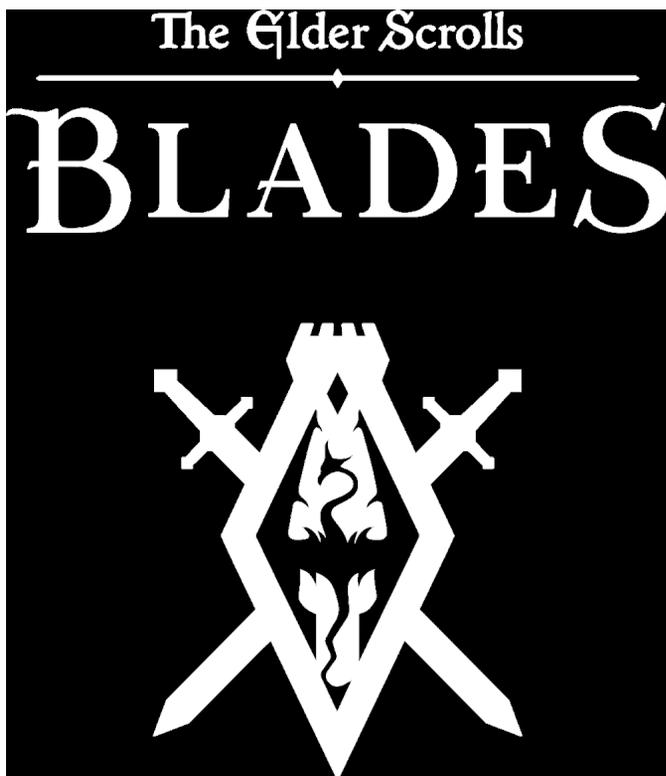


Figura 1 - Logotipo de Google Stadia

Sobre el papel, la promesa de Stadia suena casi demasiado buena para ser verdad: juego donde quieras, cuando quieras; juega al instante, sin actualizaciones de software, controladores o instaladores; juego en 4K HDR a 60 frames por segundo (fps); y hay una hoja de ruta de actualizaciones integradas que será mantenida por Google. Sin embargo, hay un enorme inconveniente (sabías que habría uno, ¿verdad?). Para utilizar Stadia en tu (s) dispositivo (s), necesitará una conexión a Internet sólida con un gran ancho de banda. Y para muchos usuarios, será una cuestión prácticamente de suerte el que vean este tipo de rendimiento durante el juego.

Puede ver el infomercial de Google Stadia, apropiadamente llamado "El futuro de los juegos de Google", aquí:

Stadia Official Reveal Trailer: The Future of Gaming f...



Durante esta conferencia financiera de Alphabet, no se revelaron ni el coste ni la fecha de inicio. Según el Sr. Pichai, "Con Stadia, podrás jugar a juegos AAA avanzados en cualquier tipo de pantalla al instante, sin necesidad de descargar el juego o instalar actualizaciones. El recibimiento de los jugadores dentro de la industria ha sido increíble y esperamos compartir más cuando se lance a finales de este año".

El vicepresidente de Google, Phil Harrison, abordó esta vaguedad de manera más directa, e indicó que Google programaría "dos eventos más" para proporcionar toda la información de los "elementos fundamentales" de cara a los jugadores antes del lanzamiento de Stadia.

### ¡Eso es todo (o No) amigos!

¿Eres un fan de Cartoon Network? Entonces seguro que el juego gratuito Cartoon Network Arcade te tiene enganchado. Abarrotado de juegos que están poblados con tus personajes favoritos de Cartoon Network, también hay una conexión de video entre el juego y la red de TV para capturar personajes de Cartoon Network. Es un beneficio mutuo para satisfacer tu hábito de dibujos animados 24/7.



**Figura 2: Juega con tus personajes favoritos en Cartoon Network Arcade**

<https://youtu.be/l6w6P0-Sdo4>

### **Actualización: Elder Scrolls: Blades**

Como mencionamos el mes pasado, el lanzamiento oficial de Elder Scrolls: Blades from Bethesda se está acercando y está a punto de hacerse realidad. Este

mes hemos recibimos un estupendo regalo: un lanzamiento Beta de acceso al que ahora puede jugar cualquiera. Solo aparece el tema de registrarse para el acceso, y luego serás catapultado a la batalla. Simplemente siga este enlace para obtener más información sobre este emblemático juego:

<https://elderscrolls.bethesda.net/en/blades>

### **Going Downhill Fast**

Hola jugadores de Linux; ¿Recordáis a Tux Racer, un juego que muestra a ese adorable pingüino esquiando cuesta abajo en busca de peces? Bueno, los jugadores de Android ahora pueden compartir la emoción de atravesar montañas cubiertas de nieve con el hábilmente llamado Grand Mountain Adventure. Lanzado por Toppluva AB a fines de marzo, Grand Mountain Adventure combina divertidos desafíos cuesta abajo junto con el esquí de montaña. Un juego perfecto para jugar sentado junto a la piscina.

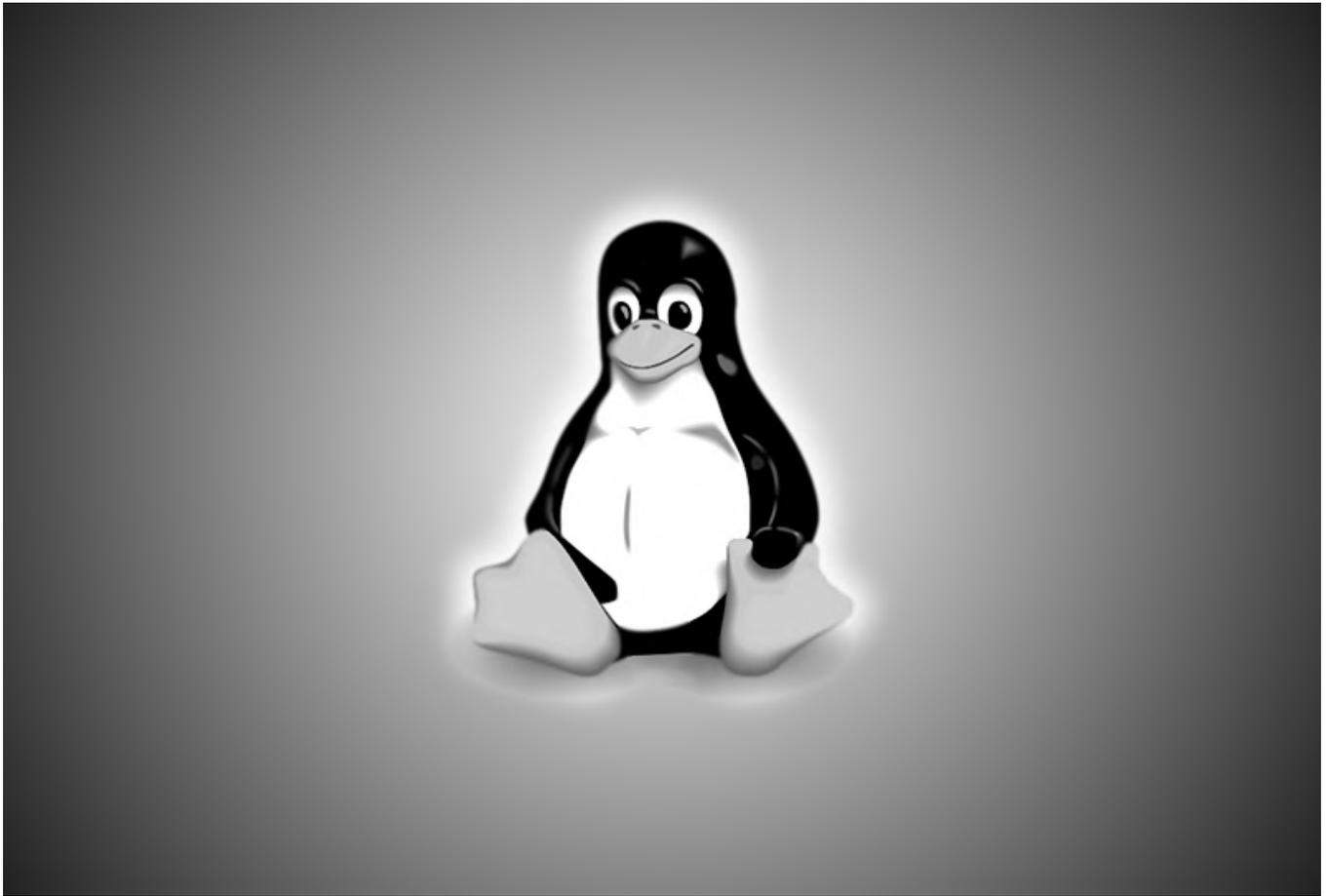
Échale un vistazo a este trailer:

<https://youtu.be/n-mr4Z5oGWc>

¿Quieres jugar a un juego de Android? Top Ten de ODRROID Magazine 10. The War of Mine-14.99\$/pack de expansión para desarrolladores 1.99\$ 9. Asphalt 9: Legends-GRATUITO 8. Minecraft-6.99\$ 7. Crashlands-4.99\$ 6. Fire Emblem Heroes-GRATUITO 5. Riptide GTP Series-2.99\$ 4. Shadowgun Legends-GRATUITO 3. Elder Scrolls: Blades-GRATUITO; esta versión beta empieza ascender a la cima 2. Fortnite-GRATUITO 1. PUBG Mobile-GRATUITO

# Logotipo de Inicio Personalizado: Creando una Imagen de Inicio a Medida para el ODROID-C2

© June 1, 2019 👤 By Justin Lee ➦ ODROID-N2, Mecaniquero, Tutoriales



Puedes crear un logotipo de inicio personalizado para el ODROID-C2 siguiendo las sencillas instrucciones que se detallan a continuación. El logotipo aparecerá en los primeros segundos del arranque mientras se carga el sistema operativo. El formato de imagen básico del logotipo de arranque del ODROID-C2 se describe en el siguiente apartado.

## Formato

```
Image Format : 24-bit Windows BMP image or 24-bit  
Windows Gzipped BMP image (without meta-data)  
Image Size : 1280 by 720  
Color Depth : 24bpp Color  
File Name : 'boot-logo.bmp' or 'boot-logo.bmp.gz'
```

El nombre del archivo debe ser "boot-logo.bmp" o "boot-logo.bmp.gz". Puedes encontrar un archivo bmp de muestra en: [https://wiki.odroid.com/\\_media/en/boot-](https://wiki.odroid.com/_media/en/boot-)

[logo.bmp.gz](https://www.logo.bmp.gz). Recomendamos utilizar GIMP o KolourPaint.

GIMP - Exportar como BMP de Windows - Opciones de compatibilidad: no escribe información sobre el espacio de color - Opciones avanzadas: color de 24 bits. - Nombre: "boot-logo.bmp"

KolourPaint - Guardar imagen como - Filtro: imagen de Windows BMP - Conversión a: Color de 24 bits

## Limitación del tamaño

Debes mantener el tamaño de tu archivo bmp por debajo de los 2 MB, ya que la partición del logotipo está limitada a 2 MB. El formato BMP Gzip es compatible, así que, si el tamaño supera los 2 MB, puedes usar el archivo bmp.gz.

```
$ gzip boot-logo.bmp  
$ ls boot-logo.bmp.gz
```

## Opción de escalado automático

El uboot del ODROID-C2 admite el escalado de imagen en el logotipo de inicio, de modo que el logotipo que se muestre quedará redimensionado automáticamente por el modo output descrito en boot.ini. Por ejemplo, si utilizamos el modo "1024x600p60hz", el logotipo de inicio se mostrará a 1024x600, aunque el tamaño real del archivo bmp sea de 1280x720.

## Reemplazo del logo de inicio

El ODROID-C2 analiza la existencia de los siguientes componentes por orden numérico. boot-logo.bmp en la partición VFAT boot-logo.bmp.gz en la partición VFAT datos del logo en la partición LOGO de Android

## Android

En Android, puedes reemplazar el logotipo de inicio por una imagen personalizada. Hay dos métodos para cambiar la imagen del logotipo de arranque. Añadir una imagen a la partición VFAT. Reescribir los datos de la imagen en la partición LOGO de Android usando fastboot.

Método 1: VFAT Copia el nuevo boot-logo.bmp, o boot-logo.bmp.gz, a la partición VFAT.

Método 2: Partición logo de Android En primer lugar, debes acceder a la línea de comandos de U-Boot mientras presionas la tecla 'ENTER' cuando tu ODROID-C2 se esté encendido y ejecutar el comando fastboot desde U-Boot y conectarte a tu ordenador de escritorio mediante un cable micro USB.

```
odroidn2# fastboot
```

A continuación, ejecuta el comando fastboot desde tu ordenador de escritorio.

## HOST PC

```
$ fastboot flash logo boot-logo.bmp.gz
```

o

```
$ fastboot flash logo boot-logo.bmp
```

Si vas a utilizar los datos bmp en la partición del logotipo, asegúrate de que NO haya un archivo boot-logo.bmp.gz en tu área VFAT, ya que U-Boot verifica

primero si hay boot-logo.bmp/boot-logo.bmp.gz en el área VFAT y luego chequea la partición del logotipo.

## Ubuntu

Con Ubuntu, la opción de mostrar el logotipo NO está incluida por defecto. Por lo tanto, debes agregar la imagen del logotipo de inicio en la partición VFAT. El método que usa la partición LOGO no está disponible en Ubuntu.

## Añadiendo el comando showlogo: caso 1080p60hz

En U-Boot, la visualización por defecto del logotipo funciona a una resolución de pantalla de 1080p60hz. Por lo tanto, no es necesario que agregues o modifiques ningún comando relacionado, pero asegúrate de que el archivo del logotipo de inicio se encuentre en las ubicaciones que hemos mencionado anteriormente.

## Resolución distinta a 1080p60hz

Si utilizas otra resolución que no sea 1080p60hz, tienes que añadir comandos a tu boot.ini antes de ejecutar bootcmd. En primer lugar, comprueba si el comando 'showlogo' se encuentra en tu boot.ini, Si no es así, ten en cuenta lo siguiente:

```
### Boot Arguments
if test "${display_autodetect}" = "true"; then
hdmix edid; fi
if test "${hdmimode}" = "custombuilt"; then setenv
cmode "modeline=${modeline}"; fi

### Add showlogo with ${hdmimode} size
hdmix mode ${voutmode}
showlogo ${hdmimode}
```

## Cómo usar una imagen personalizada con Resolución Nativa

Si quieres utilizar una resolución nativa para una imagen bmp como 1920x1080, 1024x600 (para VU7+) o 800x480 (para VU7), configura arg[2]/arg[3] en el comando showlogo de la siguiente forma:

```
odroidn2 # help showlogo
showlogo - Displaying BMP logo file to HDMI screen
with the specified resolution
```

Usage:

```
showlogo [ ]
    resolution - screen resolution on HDMI
screen
    '1080p60hz' will be used by
default if missing
    bmp_width (optional) - width of logo bmp
file
    '1280' will be used by default if
missing
    bmp_height (optional) - height of logo bmp
file
    '720' will be used by default if
missing
```

Reemplaza la imagen del logotipo de inicio por la tuya tal y como hemos descrito anteriormente y luego modifica el comando 'showlogo' en boot.ini. Aquí tienes unos ejemplos:

### Tamaño de la imagen del logotipo de ancho 1920 y alto 1080

Si la resolución de tu monitor es 1920×1080 y deseas configurar un archivo bmp en 1920×1080, configura el comando en boot.ini con:

```
setenv hdmimode "1080p60hz"
showlogo ${hdmimode} 1920 1080
```

### Tamaño de la imagen del logotipo de ancho 1024 y alto 600

```
setenv hdmimode "1024x600p60hz"
showlogo ${hdmimode} 1024 600
```

### Tamaño de la imagen del logotipo de ancho 800 y alto 480

```
setenv hdmimode "800x480p60hz"
showlogo ${hdmimode} 800 480
```

## Animación de arranque en Android

En el caso de Android, puedes usar el método bootanimation.zip para mostrar tu logotipo personalizado usando animación. Por favor dirígete a este sitio:

<https://android.googlesource.com/platform/frameworks/base/+master/cmds/bootanimation/FORMAT.md>

El sistema selecciona un archivo zip de inicio con animación de las siguientes ubicaciones.

```
/system/media/bootanimation.zip
/oem/media/bootanimation.zip
```

Antes de proceder a la copia, debes cambiar los permisos del sistema de archivos raíz a r/w y copiar tu bootanimation.zip en la carpeta /system/media/.

```
console:/ $ su
console:/ # mount -o rw,remount /system
[ 173.674067@2] EXT4-fs (mmcblk0p11): re-mounted.
Opts:
block_validity,delalloc,barrier,user_xattr,acl,inode_readahead_blks=8
```

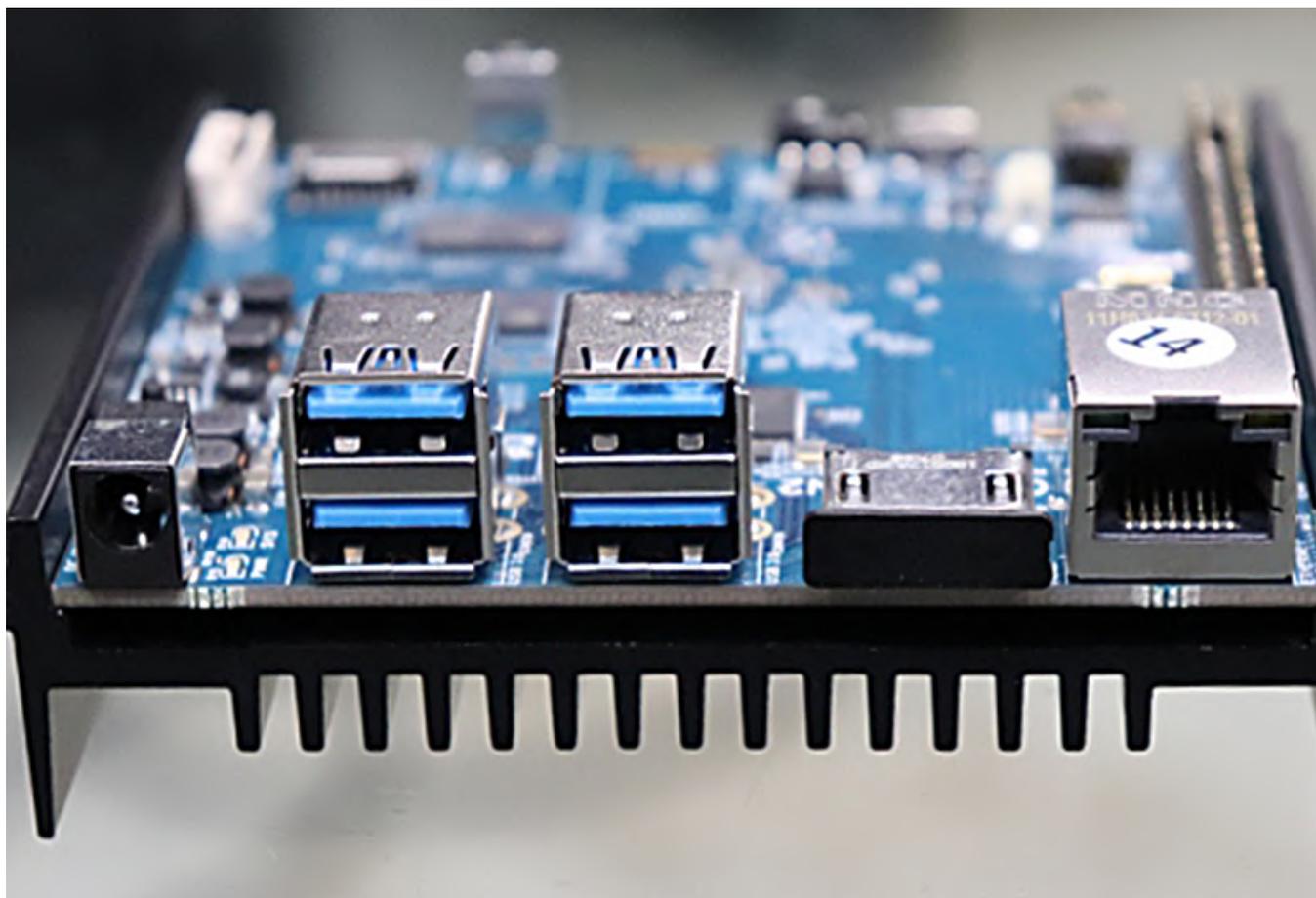
Por favor, asegúrese de los permisos correctos para bootanimation.zip:

```
console:/ # chmod 755
/system/media/bootanimation.zip
console:/ # ls -al /system/media/bootanimation.zip
-rwxr-xr-x 1 root root 5030729 2019-03-27 00:41
/system/media/bootanimation.zip
```

Esta guía está disponible en la wiki de ODROID en [https://wiki.odroid.com/odroid-n2/application\\_note/bootlogo\\_n2](https://wiki.odroid.com/odroid-n2/application_note/bootlogo_n2).

# ODROID-N2: Análisis del Buen Rendimiento en Pruebas con Linux

© June 1, 2019 By Michael Larabel from Phoronix.com ➤ ODROID-N2



El ordenador de placa reducida más reviente de Hardkernel es el ODROID-N2, que me enviaron hace unas semanas para realizar pruebas de rendimiento. El ODROID-N2 está basado en el SoC Amlogic S922X y presenta cuatro núcleos Cortex-A73 y dos núcleos Cortex-A53, opciones de 2GB o 4GB de memoria del sistema DDR4, conectividad eMMC, Ethernet Gigabit y cuatro puertos USB 3.0, cuyo coste está simplemente por encima de los 60\$.



Figura 1 - vista superior de ODROID-N2

El uso por parte de ODROID-N2 de un diseño grande.LITTLE de Amlogic S922X permite una interesante distribución de los cuatro núcleos Cortex-A73 que alcanzan los 1.8GHz y los dos núcleos Cortex-A53 capaces de alcanzar los 1.9GHz. Este SoC utiliza la GPU Bifrost G52 de Mali, que con el tiempo debería

ver un buen soporte de drivers a través de la pila del controlador de gráficos de código abierto Panfrost.



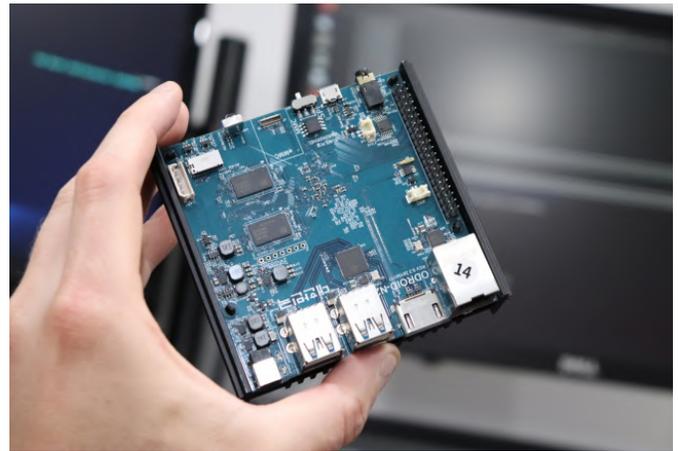
**Figura 2 - Vista lateral del ODROID-N2**

El ODROID-N2 emplea memoria DDR4 que ofrece un potencial de rendimiento mucho mayor que el ODROID-N1 y otros que dependen de memoria del sistema DDR3. El disipador térmico de aluminio en la parte inferior del SBC está diseñado para garantizar la suficiente potencia de enfriamiento para el SoC y la RAM. El SoC que se coloca en la parte inferior de la PCB es una característica habitual del diseño en comparación con la mayoría de los SBC ARM de similar coste que vemos, aunque funciona bastante bien con el disipador de metal como base.



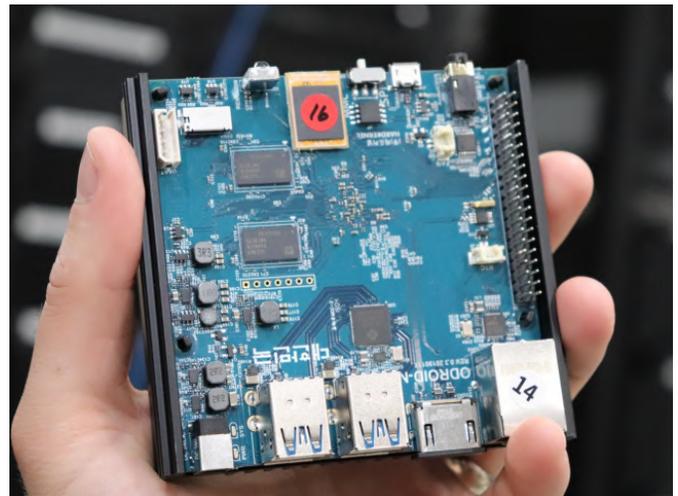
**Figura 3 - Vista del fondo y del disipador de calor.**

La conectividad del ODROID-N2 incluye HDMI 2.1, Ethernet Gigabit, cuatro puertos USB 3.0 host, puerto micro-USB 2.0 OTG, cabezal GPIO de 40 pines y un conector de alimentación DC.



**Figura 4 - vista superior de la placa**

El ODROID-N2 es oficialmente compatible con los BSP de Ubuntu 18.04 LTS y Android 9 Pie en estos momentos, aunque es probable que aparezcan más distribuciones de Linux compatibles con el N2 a medida que avance el año. La imagen de Ubuntu 18.04 LTS AArch64 para el ODROID-N2 utiliza el kernel de Linux 4.9.



**Figura 5 - vista superior desde arriba con la tarjeta emm insertada**

El ODROID-N2 con 2GB de RAM tiene un precio de alrededor de unos 65\$, mientras que la versión de 4GB está por encima de los 80\$. Hardkernel amablemente nos envió el ODROID-N2 para que realizáramos pruebas de rendimiento sobre Phoronix.

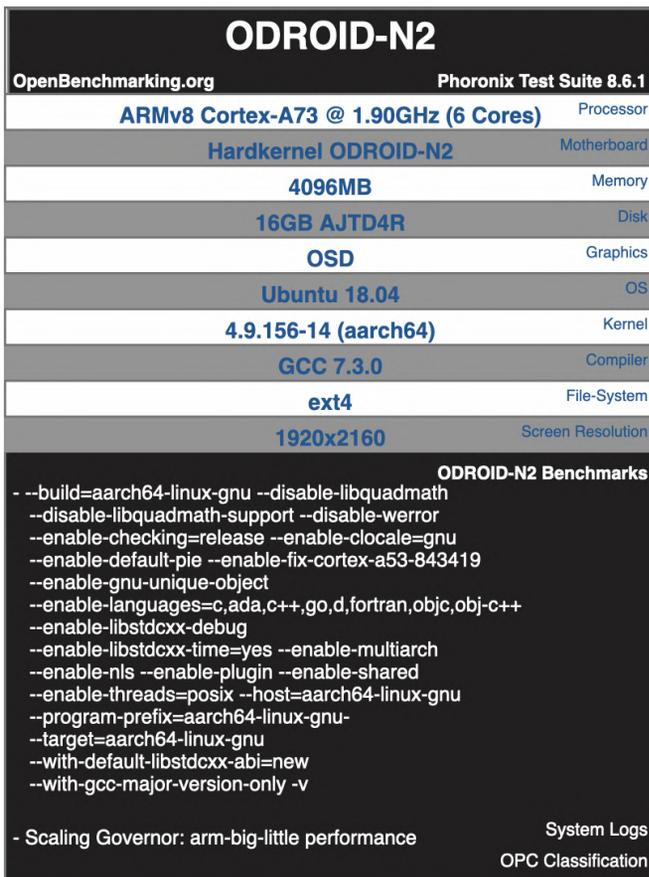


Figura 6 - descripción general de las especificaciones

Para que te hagan una idea del potencial de rendimiento del ODROID-N2 4GB, aquí tienes algunas pruebas de rendimiento iniciales en la que lo comparamos con el NVIDIA Jetson TX1, Jetson TX2, Jetson AGX Xavier, Jetson Nano, Raspberry Pi 3 Modelo B +, ASUS TinkerBoard, ODROID-C2 y ODROID XU-4.

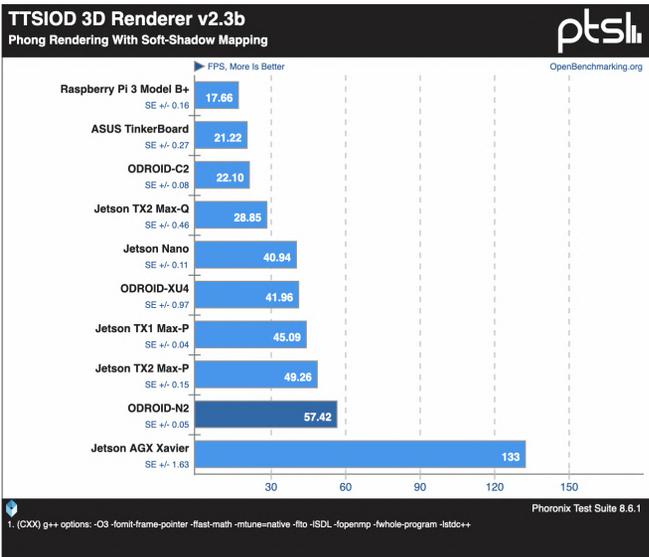


Figura 7 - Prueba de rendimiento TTSIOD 3D Renderer

Desde el principio, el ODROID-N2 ha tenido un rendimiento de CPU muy aceptable en las pruebas de

rendimiento TTSIOD 3D Renderer de múltiples subprocesos. El ODROID-N2 tiene mejor rendimiento de CPU que el Jetson Nano de 99\$ y el Jetson TX2, mientras que obviamente no supera a la Jetson AGX Xavier premium. Pero de todas las placas probadas, el ODROID-N2 es la más competitiva, especialmente por su precio de 65\$/82\$.

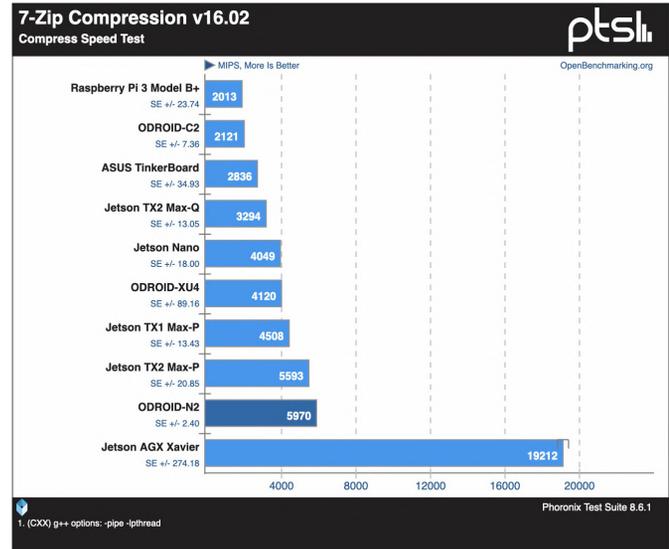


Figura 8 - Pruebas de compresión 7-Zip

El puesto en el ranking fue similar con las pruebas de compresión 7-Zip, donde el AGX Xavier solo superó a ODROID-N2, mientras que éste último supuso una gran mejora con respecto a la Raspberry Pi 3.

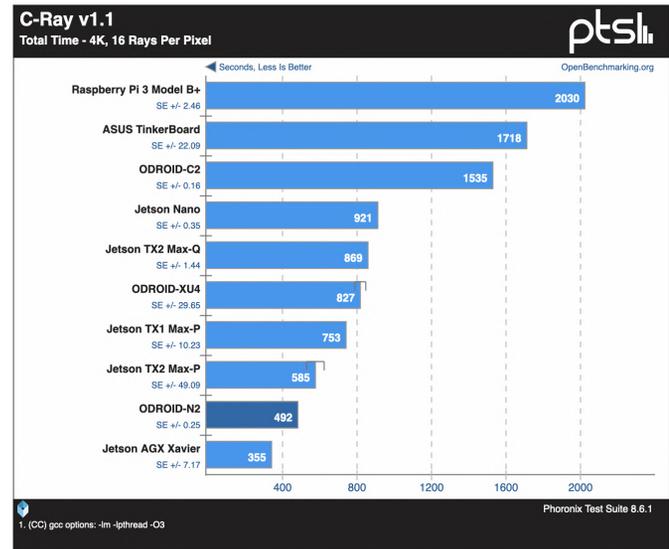


Figura 9 - Prueba de rendimiento C-Ray

El ODROID-N2 de seis núcleos también funciona muy bien con el trazador de rayos multi-proceso C-Ray.

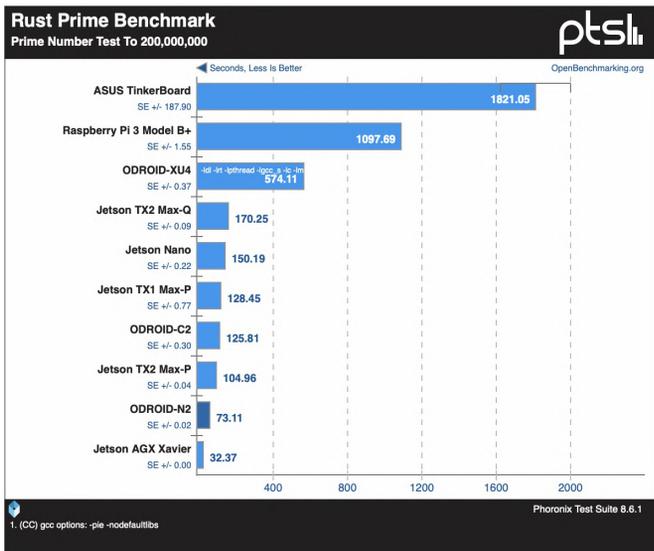


Figura 10 - Prueba de rendimiento Rust Prime

El ODROID-N2 también tuvo un buen rendimiento con Rust.

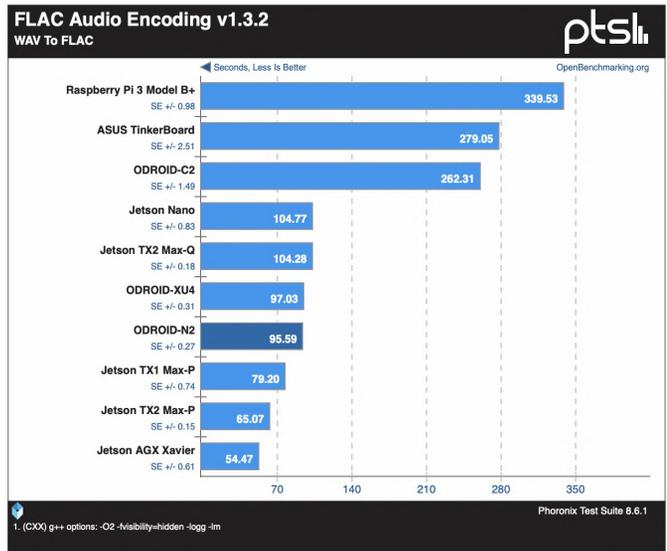


Figura 12 - Prueba de rendimiento de audio FLAC

En las pruebas de rendimiento de codificación de audio FLAC de un único subproceso, el ODROID-N2 presentó un rendimiento similar al del ODROID-XU4.

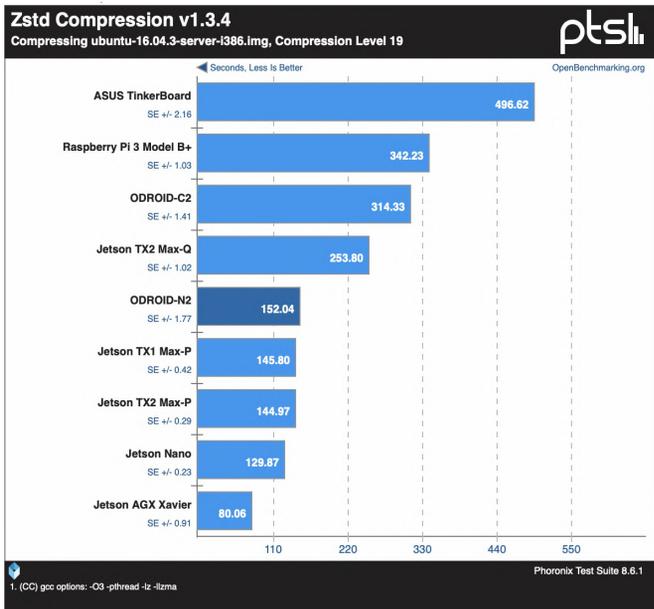


Figura 11 - Prueba de compresión Zstd

Con la compresión Zstd de múltiples subprocesos, el Jetson Nano y el TX1 / TX2 sacaron una ligera ventaja, aunque por lo general, el ODROID-N2 continúa teniendo un gran rendimiento teniendo en cuenta su precio, si lo comparamos con la Raspberry Pi 3 Modelo B +, ASUS TinkerBoard y otras placas.

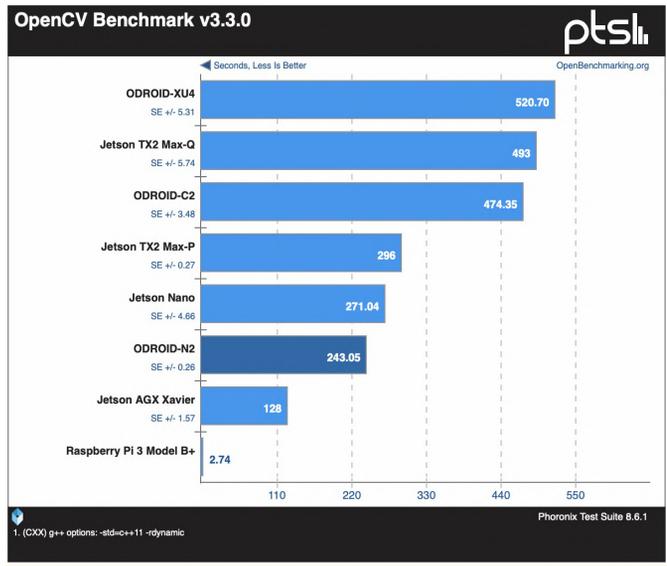


Figura 13 - Prueba de rendimiento OpenCV

El rendimiento del OpenCV de las CPU también fue bueno. (no se tiene en cuenta la ejecución de la Raspberry Pi ya que fue abortada).

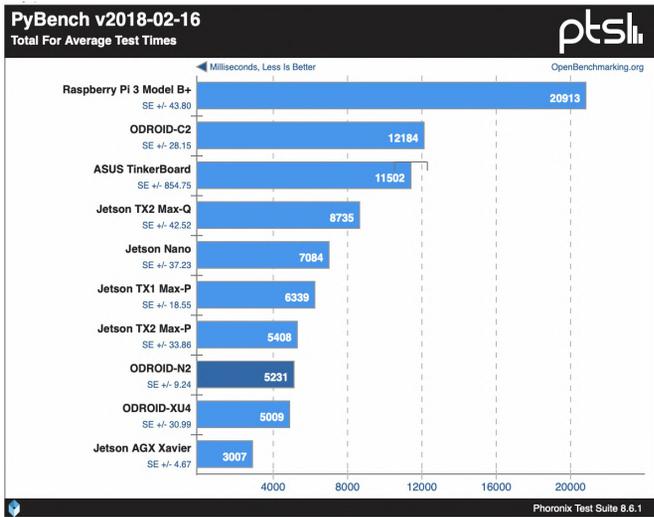


Figura 14 - Prueba de rendimiento PyBench

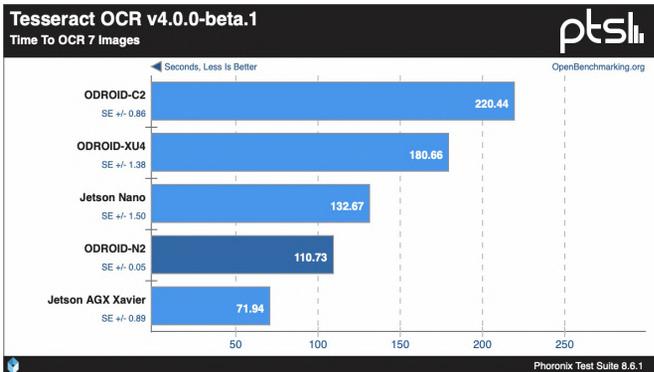


Figura 15 - OCR Tesseract

En general, el ODROID-N2 ofrece muy buen rendimiento para el precio que tiene y que se situa por debajo de los 100\$. Estamos muy contentos con el rendimiento del ODROID-N2 de Hardkernel. El rendimiento con sus cuatro núcleos Cortex-A73 y dos Cortex-A53 ha sido muy competitivo, especialmente teniendo en cuenta que la placa cuesta poco más de 80\$ en la versión de 4 GB de RAM (o 65\$ si sólo necesitas 2 GB de RAM). El rendimiento es muy bueno, de hecho, el ODROID-N2 es capaz de cubrir los múltiples usos de los aficionados e incluso funcionar como un escritorio ARM Linux muy liviano o en otros entornos. Lo que también nos gusta es: - Memoria del sistema DDR4 de 4GB - 4 puertos USB 3.0 - Ethernet Gigabit - Los gráficos de Mali Bifrost están en proceso de ser liberados gracias al esfuerzo de Panfrost DRM/Gallium3D Para aquellos que quieran comparar sus propias placas Linux con el rendimiento del ODROID-N2, he subido resultados adicionales a un archivo en OpenBenchmarking.org: <https://openbenchmarking.org/result/1904211-HV-ODROIDN2942>

Tras instalar Phoronix Test Suite, disponible en <https://www.phoronix-test-suite.com/>, simplemente ejecuta `phoronix-test-suite benchmark 1904211-HV-ODROIDN2942` para realizar tu propia comparativa automática de principio a fin.

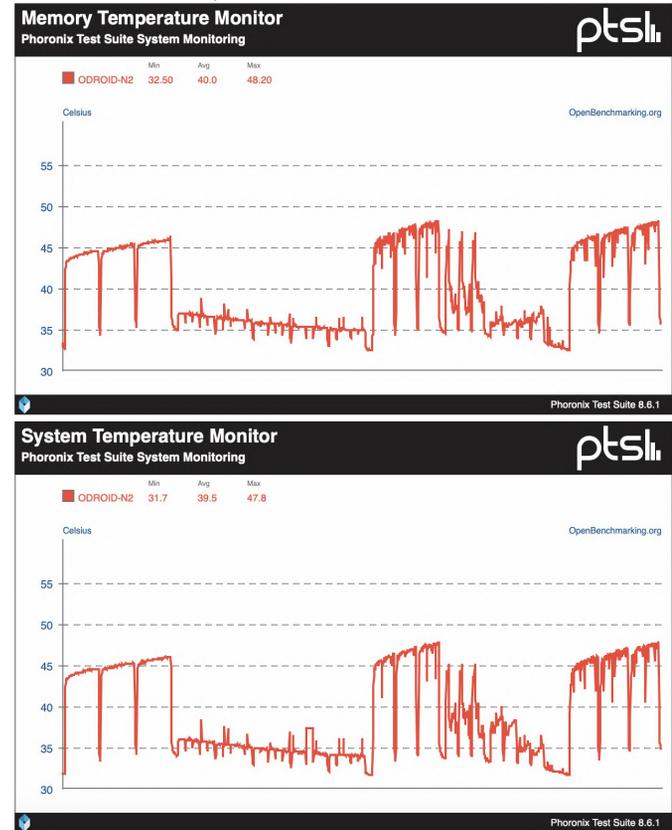


Figura 16 - Valores del monitor de temperatura

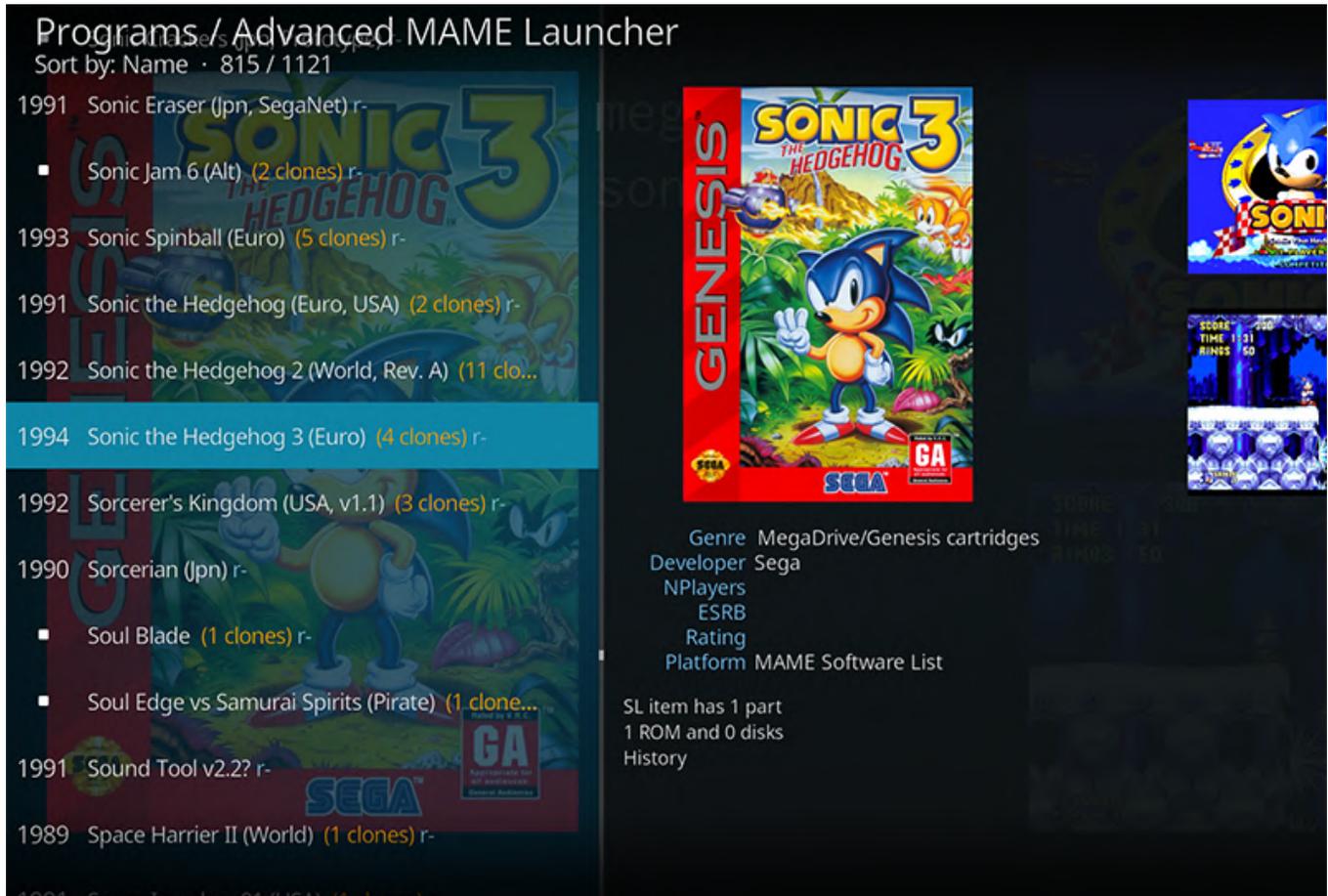
Para aquellos que se estén preguntando como es el rendimiento térmico del ODROID-N2, gracias al gran disipador térmico de aluminio, no hemos percibido ninguna limitación térmica que sea notable o cualquier otro problema. Aquí tienes varias pruebas en las que se ha medido la temperatura del SoC y de DDR con diferentes cargas de trabajo. Con varias cargas de trabajo de múltiples subprocesos, el SoC y la memoria DDR4 nunca se rompió la barrera de los 50 grados Celsius y la temperatura media fue de unos 40 grados gracias a este gran disipador de calor pasivo. La temperatura mínima durante el reposo y en el inicio de las pruebas de rendimiento fue de tan solo 31 grados. Tienes más datos en el archivo de resultados de OpenBenchmarking.org: [https://openbenchmarking.org/result/1904255-HV-ODROIDN2T59&obr\\_nbp=1](https://openbenchmarking.org/result/1904255-HV-ODROIDN2T59&obr_nbp=1)

Gracias a Hardkernel por enviarnos el ODROID-N2 para realizar pruebas y aquellos que quieran

aprender más sobre este SBC ARM pueden recurrir a El artículo anterior de Michael Larabel, ha sido publicado en su sitio web y lo puedes encontrar en <http://www.phoronix.com/vr.php?view=27780>.

# Kodi y Mame Avanzado en ODRROID-XU4 - Parte 1

© June 1, 2019 By Juegos, Linux, ODRROID-XU4, Tutoriales



Programs / Advanced MAME Launcher  
Sort by: Name · 815 / 1121

- 1991 Sonic Eraser (Jpn, SegaNet) r-
- Sonic Jam 6 (Alt) (2 clones) r-
- 1993 Sonic Spinball (Euro) (5 clones) r-
- 1991 Sonic the Hedgehog (Euro, USA) (2 clones) r-
- 1992 Sonic the Hedgehog 2 (World, Rev. A) (11 clones) r-
- 1994 Sonic the Hedgehog 3 (Euro) (4 clones) r-
- 1992 Sorcerer's Kingdom (USA, v1.1) (3 clones) r-
- 1990 Sorcerian (Jpn) r-
- Soul Blade (1 clones) r-
- Soul Edge vs Samurai Spirits (Pirate) (1 clone...) r-
- 1991 Sound Tool v2.2? r-
- 1989 Space Harrier II (World) (1 clones) r-

Genre MegaDrive/Genesis cartridges  
Developer Sega  
NPlayers  
ESRB Rating  
Platform MAME Software List

SL item has 1 part  
1 ROM and 0 disks  
History

Esta es una guía para configurar Kodi con Mame en el SBC ODRROID-XU4, lo cual lo convierte en un buen centro multimedia y de juegos. Incluye todos los pasos para instalar el software necesario en Ubuntu Linux. La instalación de Mame es opcional, así como la configuración de la unidad USB, el joystick, etc. Puede detenerte en Kodi o ir más allá y montar una máquina de video, música y juegos completamente funcional. No todos los pasos son sencillos. Algunos pueden requerir consultar información online. He hecho toda la investigación por ti y te proporciono enlaces a los recursos, de modo que contarás con toda la información para entender los conceptos necesarios. La primera parte está centrada en la instalación y configuración de Kodi que es independiente. La parte de Mame requerirá recursos adicionales externos. El resto es igualmente independiente.

## Instalar Linux y Kodi

Para instalar todo esto, sigue los siguientes pasos:

1. El primer paso ha de realizarse en otra máquina Linux. Si aún no tiene una, es bueno migrar a Linux lo antes posible. Descárgate una imagen de: [https://wiki.odroid.com/ODROID-XU4/os\\_images/linux/ubuntu\\_4.14/20181203](https://wiki.odroid.com/ODROID-XU4/os_images/linux/ubuntu_4.14/20181203)

2. Asegúrate de que la imagen sea válida con md5sum y compara el resultado con el correspondiente archivo md5sum de [https://odroid.in/ubuntu\\_18.04lts/XU3\\_XU4\\_MC1\\_HC1\\_HC2/](https://odroid.in/ubuntu_18.04lts/XU3_XU4_MC1_HC1_HC2/).

```
$ md5sum ubuntu-18.04.1-4.14-mate-ODROID-XU4-20181203.img.xz
```

3. Suponiendo que estés haciendo todo esto en Linux, expande el archivo que has descargado:

```
$ xz -d ubuntu-18.04.1-4.14-mate-ODROID-XU4-20181203.img.xz
```

4. Conecta tu tarjeta SD en un puerto USB de tu máquina Linux.

5. Busca el dispositivo con lsblk.

```
$ lsblk
```

Nota: Supongamos que tu tarjeta SD está en "/dev/sdX". Ten mucho cuidado de elegir el dispositivo correcto o te arriesgarás a borrar el contenido de otro disco.

6. Escribe la imagen en la tarjeta SD.

```
$ sudo dd if=ubuntu-18.04.1-4.14-mate-ODROID-XU4-20181203.img.xz of=/dev/sdX bs=1M conv=fsync
$ sudo sync
```

7. Desmonta tu tarjeta SD desde el entorno de escritorio o desde la línea de comandos:

```
$ sudo umount /dev/sdX
```

8. Inserta tu tarjeta SD en tu ODROID-XU4 y reinícialo. Luego inicie sesión en el escritorio con las credenciales que se indican en:

<https://bit.ly/30syIGW>

Llegados a este punto, deberás tener un escritorio ODROID-XU4 completamente funcional. El siguiente paso será instalar Kodi y eliminar el acceso directo al escritorio para que tu dispositivo arranque directamente con la interfaz de Kodi, algo así como un televisor normal.

9. Abre un terminal (busca en el menú de la parte superior izquierda de la pantalla).

10. Actualiza los paquetes:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt dist-upgrade
```

Durante la actualización, la primera vez que inicies tu máquina, puede aparecer un error relacionado con un archivo lock con dpkg (" /var/lib/dpkg /lock "). Esto significa que uno de los procedimientos de actualización automáticos todavía se está ejecutando, lo cual es probable que sea importante para la nueva instalación. Simplemente déjalo ejecutarse hasta el final y vuelve a intentar los comandos. Si aparece un mensaje en el que se indica que boot.ini ha sido reemplazado, simplemente selecciona 'OK'. Si recibes una pregunta sobre "/etc/apt-fast.conf", responde Y (para Sí).

11. Instala Kodi:

```
$ sudo apt install kodi kodi-bin kodi-data libcec4 python-libcec openbox
```

12. El driver gráfico de Mali se instala por defecto y funciona bastante bien. En un terminal, ejecuta "glmark2-es2" con el que deberías ver una demo (una imagen de un caballo). Los FPS deberían estar alrededor de 300 imagen/segundo. Ejecuta `glmark2` y deberías ver la misma demo. El FPS debería estar alrededor de 30 cuadros/segundo - horrible. El motivo es que el driver de Mali solo es compatible con OpenGL ES (sistema integrado) y no con OpenGL simple. La solución es usar una librería llamada "gl4es" (<https://github.com/ptitSeb/gl4es>), que puedes encontrar en Github.

13. Instala "gl4es"

En un terminal, instala las siguientes herramientas de desarrollo:

```
$ sudo apt install git cmake xorg-dev gcc g++ build-essential
```

Consigue el código fuente de gl4es. Gl4es es una implementación de OpenGL utilizando OpenGL ES (en lugar de implementar la librería directamente). Por lo tanto, en sistemas como ODROID-XU4 con un driver que solo admite OpenGL ES, también es posible utilizar software basado en OpenGL con aceleración de hardware:

```
$ git clone https://github.com/ptitSeb/gl4es.git
$ cd gl4es
```

Compilado

```
$ mkdir build
$ cd build
$ cmake .. -DODROID=1
$ make
```

Se compilará y mostrará su progreso con una gran cantidad de mensajes. Al final, si la compilación sale bien, deberías ver lo siguiente:

```
...
[ 96%] Building C object
src/CMakeFiles/GL.dir/glx/utils.c.o
[ 98%] Linking C shared library ../lib/libGL.so.1
```

```
[100%] Built target
...
```

Guarda la versión Mesa de OpenGL (la que está instalada por defecto):

```
$ sudo mv /usr/lib/arm-linux-gnueabi/libGL.so.1
/usr/lib/arm-linux-gnueabi/save.libGL.so.1
$ sudo mv /usr/lib/arm-linux-
gnueabi/libGL.so.1.0.0 /usr/lib/arm-linux-
gnueabi/save.libGL.so.1.0.0
```

y copia tu nueva versión compilada:

```
$ sudo cp lib/libGL.so.1 /usr/lib/arm-linux-
gnueabi/
$ sudo ldconfig
```

Ahora comprueba que todo está bien ejecutando:

```
$ glxinfo | head -6
```

Si ves lo siguiente, estás listo para continuar (por supuesto, la fecha de la segunda línea es diferente a la tuya):

```
LIBGL: Initialising gl4es
LIBGL: v1.1.1 built on Mar 28 2019 06:22:26
LIBGL: Using GLES 2.0 backend
LIBGL: loaded: libGLESv2.so
LIBGL: loaded: libEGL.so
LIBGL: Using GLES 2.0 backend
```

Ahora puedes ejecutar glmark2 de nuevo:

```
$ glmark2
```

Esta vez deberías ver entre 300 y 600 FPS. En este momento, tu ODROID-XU4 también es compatible con OpenGL. Incluso puedes utilizar software como Blender (<https://www.blender.org/>).

14. Por razones de seguridad, Kodi y sus programas asociados serán ejecutados por un usuario con privilegios limitados, sin contraseña y con inicio de sesión automático. Llamaremos a este usuario "kodi":

```
$ sudo adduser --disabled-password --gecos "" kodi
```

El resultado será:

```
Adding user `kodi' ...
Adding new group `kodi' (1000) ...
Adding new user `kodi' (1000) with group `kodi'
...
```

```
Creating home directory `/home/kodi' ...
Copying files from `/etc/skel' ...
```

15. Asigna algunos privilegios a este usuario:

```
$ sudo usermod -a -G
cdrom,video,plugdev,users,dialout,dip,input,netdev
,audio,pulse,pulse-access,games kodi
```

16. Añade las opciones para que Kodi puede iniciar su propio servidor X:

```
$ sudo sed -ie
's/allowed_users=console/allowed_users=anybody/g'
/etc/X11/Xwrapper.config
$ sudo sed -ie "$aneeds_root_rights = yes"
/etc/X11/Xwrapper.config
```

17. Agrega el servicio Kodi a "systemd" (es mejor copiar y pegar lo siguiente en lugar de escribirlo):

```
$ cat << EOF | sudo tee
/etc/systemd/system/kodi.service
[Unit]
Description = Kodi Media Center
After = systemd-user-sessions.service
network.target sound.target mysql.service
dbus.service polkit.service upower.service
Wants = mysql.service
[Service]
User = kodi
Group = kodi
Type = simple
#PAMName = login # you might want to try this
one, did not work on all systems
ExecStart = /usr/bin/xinit /usr/bin/dbus-launch -
-exit-with-session openbox --startup
/usr/bin/kodi-standalone -- :0 -nolisten tcp vt7
Restart = on-abort
RestartSec = 5

[Install]
WantedBy = multi-user.target
EOF
```

Una nota rápida sobre la primera línea: el comando "sudo" funciona sobre "tee" para escribir con derechos de superusuario en el archivo, porque si lo hiciera con "cat" y siguiera el mismo patrón que el resto comandos de este documento, la redirección al archivo no funcionaría. De hecho, la redirección se realiza por shell (que pertenece al usuario `odroid`) y no por el comando. Así que "sudo cat" no funcionaría.

18. Activa Kodi en el momento del arranque:

```
$ sudo systemctl enable kodi
$ sudo systemctl set-default multi-user.target
```

19. En la Wiki de Hardkernel (<https://bit.ly/2Qjvcdf>), se dice que un paquete reciente de Canonical EGL bloqueó el acceso a la GPU Mali. Es cierto, la solución es la siguiente:

```
$ sudo apt install mali-x11 --reinstall
```

Cuando estaba escribiendo esta publicación, esto corrigió el error de acceso a la GPU Mali. Así que es muy recomendable aplicar esta solución. De hecho, si actualizas el sistema como dije antes, aparecerá un error y tendrás que solucionarlo con la reinstalación "mali-x11" como se ha indicado con anterioridad.

20. Ahora tienes un sistema Kodi completamente funcional. Si quieres reiniciar, tu ODROID-XU4 se iniciará directamente con Kodi. Si deseas añadir más funciones, como MAME (<https://www.mamedev.org>, <https://en.wikipedia.org/wiki/MAME>), una unidad USB externa, un joystick, un firewall, puede continuar con el resto del artículo.

Si quieres parar y disfrutar de tu ODROID-XU4 ahora, puedes reiniciarlo:

```
$ sudo reboot
```

Todo lo que aparece a continuación se puede hacer ya sea conectándote a tu ODROID-XU4 por "ssh" o directamente en la pantalla (necesitarás un teclado). Si desea conectarte directamente a la pantalla, después de reiniciar tu ODROID-XU4 en Kodi, puedes presionar "Ctrl + Alt + F1" para cambiar a un terminal (pantalla de texto). En cualquier momento, puedes volver a Kodi presionando "Alt + F7".

## Instalar una unidad externa USB

1. Cambia a un terminal o inicia sesión con "ssh".
2. Instala "usbmount" para montar las unidades USB:

```
$ sudo apt install usbmount
```

Existe un problema con la versión 0.0.22 de "usbmount" en la versión estándar de Ubuntu Linux al conectar 2 unidades USB al mismo tiempo. El error se ha solucionado con "usbmount 0.0.24" que aún no se

encuentra en el repositorio de Ubuntu. Puede actualizarlo manualmente, si quieres.

Si tu segunda unidad (o incluso la primera) no se monta correctamente al conectarla, puedes intentar actualizar "usbmount" de la siguiente manera:

```
$ git clone https://github.com/rbrito/usbmount.git
$ cd usbmount
$ sudo apt-get update && sudo apt-get install -y
debhelper build-essential fakeroot
$ dpkg-buildpackage -us -uc -b
$ cd ..
$ sudo dpkg -i usbmount_0.0.24_all.deb
```

3. Por defecto, usbmount montará la unidad USB externa con la opción "sync". "sync" significa que todo el acceso de escritura al disco se vaciará inmediatamente en todo el disco. Esto puede ralentizar radicalmente todas las operaciones que se hagan en el disco.

```
$ sudo sed -ie
's/^MOUNTOPTIONS=.*MOUNTOPTIONS="noexec,nodev,noa
time,nodiratime"/' /etc/usbmount/usbmount.conf
```

Supongo que es seguro usar la opción "sync" puesto que se supone que no desconectaremos este disco en ningún momento. Cuando hice esto, la velocidad de escritura de mi disco duro externo se multiplico por 10.

## Instalar MAME para ejecutar juegos arcade desde Kodi

Esta sección es algo extensa y requiere recursos externos si quieres disponer de una instalación MAME al completo con todas las funciones. De vez en cuando, haremos referencia a guías externas.

1. Instala MAME

```
$ sudo apt install mame mame-data mame-doc mame-
extra mame-tools libsd12-gfx-1.0-0 libsd12-image-
2.0-0 libsd12-mixer-2.0-0 libsd12-net-2.0-0
libsd12-net-2.0-0 python-pil
```

Crea un nuevo archivo de configuración con las rutas correctas, utilizando OpenGL ES 2, el driver de audio ALSA, adecuando la velocidad y utilizando todos los núcleos:

```

$ cat << EOF | sudo tee /etc/mame/mame.ini
inipath $HOME/.mame;/etc/mame
rompath /home/kodi/AML-ROMs/
samplepath /home/kodi/AML-assets/samples/
ctrlrpath /home/kodi/AML-assets/ctrlr
hashpath /usr/share/games/mame/hash
cfg_directory $HOME/.mame/cfg
nvram_directory $HOME/.mame/nvram
memcard_directory $HOME/.mame/memcard
input_directory $HOME/.mame/inp
state_directory $HOME/.mame/sta
snapshot_directory $HOME/.mame/snap
diff_directory $HOME/.mame/diff
comment_directory $HOME/.mame/comments
video auto
render opengles2
audioplayer alsaaudio
audiodriver alsaaudio
samplerate 32000
numprocessors 8
mouse 1
uimodekey INSERT
autoframeskip 1
EOF
$ sudo chmod ugo+r /etc/mame/mame.ini

```

A continuación, debemos eliminar el mensaje de inicio “gl4es” para poner contenido a MAME. En pocas palabras: un complemento de Kodi extraerá la base de datos de juegos `MAME.xml` con el típico resultado ejecutando “mame”. Cuando se inicie “mame”, “gl4es” se inicia y muestra un mensaje agradable, como el de arriba. Sin embargo, el complemento Kodi captura el resultado que se supone que es un archivo XML, excepto el mensaje de bienvenida de “gl4es”. Entonces, cuando Kodi intenta leer el archivo XML (de hecho, una librería Python también lo intenta), falla:

```

$ cat << EOF | sudo tee /usr/local/bin/mame
#!/bin/sh
export LIBGL_SILENTSTUB=1
export LIBGL_NOBANNER=1
/usr/games/mame "$@"
EOF
$ sudo chmod ugo+x /usr/local/bin/mame

```

Al configurar el complemento AML en Kodi, usaremos este nuevo comando mame que acabamos de crear.

2. Instala y configura el complemento Advanced Mame Launcher desde Kodi

Vamos a seguir la guía oficial del complemento de Kodi llamado Advanced Mame Launcher (AML) (<https://forum.kodi.tv/showthread.php?tid=304186>) y adaptaremos algunos pasos de acuerdo a nuestra configuración. Suponemos que tenemos una unidad USB externa conectada al ODROID-XU4. Será útil para almacenar datos para MAME.

Suponiendo que te encuentras en un terminal de texto (consulta más arriba cómo cambiar a un terminal de texto desde Kodi), el primer paso es crear los directorios para almacenar datos MAME. La unidad USB externa está montada en “/media/usb0” y por defecto en “/media/usb”, también. Esta es la configuración con la que seguiremos de aquí en adelante.

```

$ cd /media/usb
$ sudo mkdir mame
$ sudo chown kodi.kodi mame
$ sudo su - kodi
$ cd /media/usb/mame
$ mkdir -p AML-ROMs AML-CHDs AML-SL-ROMs AML-SL-CHDs AML-assets/samples/
$ cd AML-assets
$ mkdir artpreviews artwork cabinets clearlogos covers_SL cpanels fanarts fanarts_SL flyers manuals manuals_SL marquees PCBs snaps snaps_SL titles titles_SL videosnaps videosnaps_SL
$ cd
$ ln -s /media/usb/mame/* .
$ exit

```

Luego debemos completar los directorios con los últimos datos de Mame, tal como se describe en <https://forum.kodi.tv/showthread.php?tid=304186>. Usamos el siguiente script para descargar e instalarlo todo automáticamente:

```

$ sudo su - kodi
$ cd /media/usb/mame/AML-assets
# catlist.ini catver.ini genre.ini genre_OWS.ini mature.ini not_mature.ini
$ wget http://www.progettosnaps.net/catver/ -q -O - | grep 'download?tipo=catver' | sed "s#.*href=\"(.*.zip)\".*#wget -q 'http://www.progettosnaps.net/' -O file.zip#" | sh
$ unzip -jq file.zip '*.ini'
$ rm file.zip
# nplayers.ini
$ wget http://nplayers.arcadebelgium.be/ -q -O - |

```

```

grep -E 'nplayers[[ :digit: ]]{4}\.zip' | sed
"s#.*href="\(\(http://nplayers.*zip\)")>.*#wget -q
'□' -O file.zip#" | sh
$ unzip -jq file.zip nplayers.ini
$ rm file.zip

# bestgames.ini
$ wget http://www.progettosnaps.net/bestgames/ -q
-O - | grep 'download?tipo=bestgames' | sed
"s#.*href="\(\.*\.\.zip\)")>.*#wget -q
'http://www.progettosnaps.net□' -O file.zip#" | sh
$ unzip -jq file.zip '*.ini'
$ rm file.zip

# series.ini
$ wget http://www.progettosnaps.net/series/ -q -O
- | grep 'download?tipo=series' | sed "s#.*href="\
(\.*\.\.zip\)")>.*#wget -q
'http://www.progettosnaps.net□' -O file.zip#" | sh
$ unzip -jq file.zip '*.ini'
$ rm file.zip

# history.dat
$ wget https://www.arcade-history.com/?
page=download -q -O - | grep -o 'href="
[^\"]*\.\.zip"' | sed 's#href="\.\.\.\(\.*zip\)")>#wget
https://www.arcade-history.com□ -q -O
file.zip#' | sh
$ unzip -jq file.zip history.dat

# mameinfo.dat
$ wget 'http://mameinfo.mameworld.info' --
header="User-Agent: Firefox/70.0" -q -O - |grep -o
'href="[^\"]*Mameinfo.*\.\.zip"' | sort | tail -1 | sed
's#href="\(\.*zip\)")>#wget --header="User-Agent:
Firefox/70.0" □ -q -O file.zip#' | sh
$ unzip -qjp file.zip *.7z > mameinfo.7z
$ 7z e '-i!mameinfo.dat' mameinfo.7z > /dev/null
$ rm file.zip mameinfo.7z

# gameinit.dat
$ wget http://www.progettosnaps.net/gameinit/ -q -
O - | grep 'download?tipo=gameinit' | sed
"s#.*href="\(\.*\.\.zip\)")>.*#wget -q
'http://www.progettosnaps.net□' -O file.zip#" | sh
$ unzip -jq file.zip english/gameinit.dat
$ rm file.zip

# command.dat
$ wget http://www.progettosnaps.net/command/ -q -O
- | grep 'download?tipo=command' | sed
"s#.*href="\(\.*\.\.zip\)")>.*#wget -q
'http://www.progettosnaps.net□' -O file.zip#" | sh

```

```

$ unzip -jq file.zip Longhand/command.dat
$ rm file.zip
$ exit

```

3. Modifica "mame.ini" para indicar la nueva estructura del directorio:

```

$ sudo sed -i 's#^rompath \+.*$#rompath
/home/kodi/AML-ROMs/#' /etc/mame/mame.ini
$ sudo sed -i 's#^samplepath \+.*$#samplepath
/home/kodi/AML-assets/samples/#'
/etc/mame/mame.ini

```

4. Añade los recursos.

Vuelve a Kodi: escribe "Ctrl + D" en tu terminal, luego "Atl + F7" para volver a Kodi.

En <https://forum.kodi.tv/showthread.php?tid=304186>, puedes seguir el párrafo llamado "Setting up MAME assets and Software List assets" para agregar más recursos y activos. Esta es la forma de obtener imágenes adicionales, logotipos, etc. Puede leer la página <http://forum.pleasuredome.org.uk/index.php?showtopic=30715> sobre los paquetes extra de MAME para conocer todos los tipos de recursos que puedes encontrar en la red para Mame.

En la página mencionada anteriormente, puedes encontrar enlaces a los recursos de Mame en Internet. Vienen en enormes paquetes, simplemente debes moverte a "/media/usb/mame", que hemos creado antes. Permíteme proporcionarte algunos ejemplos de estos paquetes, que pueden ser muy útiles durante el proceso.

Si encuentras un paquete llamado `MAME 0.xxx EXTRAS` donde `xxx` es un número de versión de Mame, ve al directorio donde lo has descargado. En ese directorio, encontrarás más directorios. Mueve su contenido al directorio correspondiente en "/media/usb/mame/AML-asset/"

Otro paquete similar podría ser algo así como "MAME 0.xxx Multimedia". Deberías proceder de la misma forma.

Sin embargo, también encontrarás archivos ".zip". Simplemente puedes descomprimirlos y moverlos al directorio final como hemos hecho anteriormente. Algunos directorios no tienen el mismo nombre, así

que manten los nombres que creamos con anterioridad. En este caso, transfere el contenido del directorio en lugar del directorio en sí.

Si encuentras paquetes con “bios-devices” en su nombre, no los necesitas si has descargado los paquetes más grandes con “ROMs” en su nombre. Son solo un subconjunto para una situación diferente. Puede leer más sobre todos estos paquetes en: <http://forum.pleasuredome.org.uk/index.php?showtopic=34705>. Estos paquetes se crean cuando quieres una versión mínima de las ROM; por ejemplo, si te estás quedando sin espacio en el disco.

También hay paquetes llamados Rollbacks Roms (<https://bit.ly/2HL6Bdt>). Solo los necesita si tiene un administrador de ROM para Mame que pueda gestionar varias versiones. No los necesitarás en este tutorial.

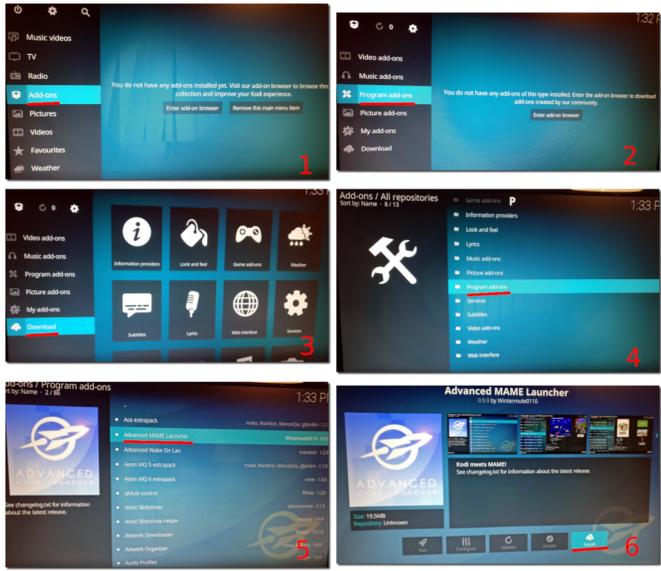
También están las listas de software ROM y CHDs. Obviamente, su contenido irá a los directorios “/media/usb/mame/AML-SL-ROMS” y “/media/usb/mame/AML-SL-CHD” que hemos creado anteriormente. “SL” significa “Software List”.

Finalmente, es posible que hayas cometido algún error con el permiso de los usuarios al descargar y mover los archivos. Así que haz las cosas bien asignando el usuario “kodi” al directorio “mame”:

```
$ sudo chown -R kodi.kodi /media/usb/mame
```

5. Instala y configura el complemento AML. Lo encontrarás en los complementos de programas. Se llama `_Advanced Mame Launcher_`.

En Kodi, ve a **Configuración**, **Configuración de complemento**, **Instalar desde el repositorio**. En **Complementos del programa**, busca **Advanced Mame Launcher** e instálalo. Mira estas imagenes:



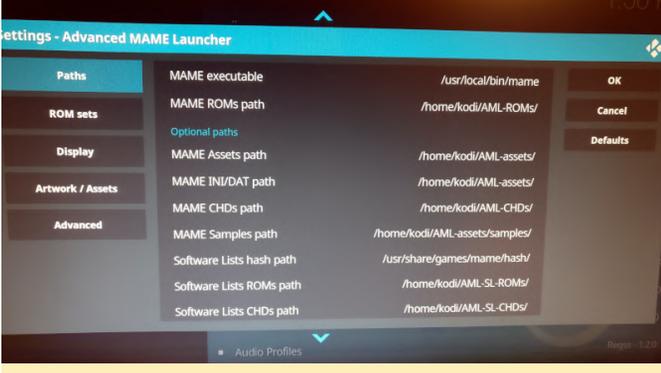
### Instalando Advanced Mame Launcher

Después de instalarlo, haga clic con el botón derecho en el logotipo del complemento y ve a `_Settings_`:



### Configuración de Advanced Mame Launcher.

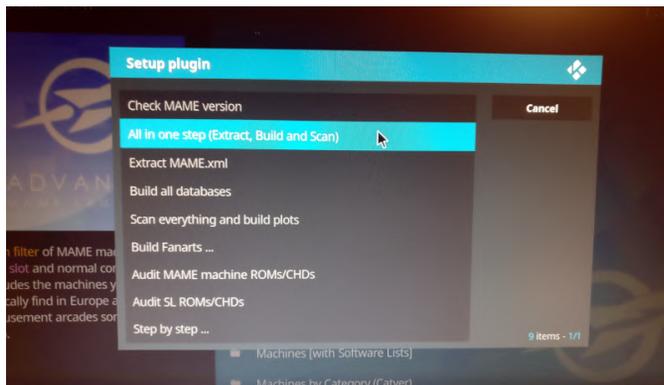
Cambia la ruta al archivo ejecutable y a los directorios de datos como se muestra en esta imagen:



### Actualizando las rutas en la configuración de Advanced Mame Launcher.

Vuelve a la pantalla inicial de Kodi y busca el complemento AML, en general **Complementos**, **Complementos del programa**, **Advanced Mame Launcher**. Selecciona cualquier fila, abre el menú contextual con un clic derecho y selecciona **Complemento de instalación**. Ejecute una

configuración completa y la configuración del complemento seleccionando las opciones \_All in one step\_ en el menú contextual del complemento.



**Elegiendo la opción All In One Step en el complemento Advanced Mame Launcher**

Este paso puede durar de varios minutos a una hora. Verás un montón de barras de progreso. Si hiciste todo bien antes, debería funcionar sin mensajes de error. El complemento ya está configurado y listo, y Mame ahora está listo para ser utilizado.

En la próxima entrega de este artículo, veremos cómo instalar un joystick. Para comentarios, preguntas y sugerencias, visita el artículo original en <https://forum.odroid.com/viewtopic.php?f=52&t=34760>.

# ODROID-N2: Pruebas de Rendimiento

© June 1, 2019 By Carlos Eduardo ↗ ODROID-N2



Hace poco recibí de Hardkernel un ODROID-N2, una nueva placa que reemplaza al ODROID-N1 cancelado. Dentro del paquete, encontré la placa ODROID-N2, la fuente de alimentación, la carcasa transparente, el adaptador USB WiFi y una tarjeta eMMC de 32 GB. El Modulo eMMC es mucho más rápido que las tarjetas SD.



Figura 01 - Vista superior de ODROID-N2

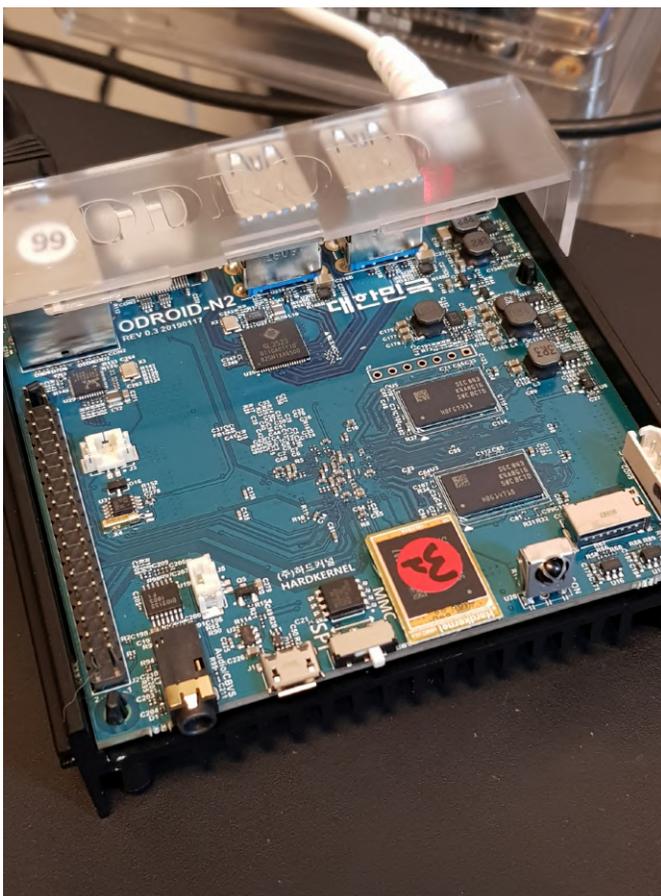


Figura 02 - Vista frontal del ODROID-N2

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

**Algunas especificaciones de la placa:** CPU Amlogic S922X con cuatro núcleos ARM Cortex-A73 y dos núcleos Cortex-A53 4 GB de RAM DDR4 Ethernet de 1 Gbps 4 puertos USB 3.0, hub USB 3.0 tras un único puerto USB 3.0 del SOC

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

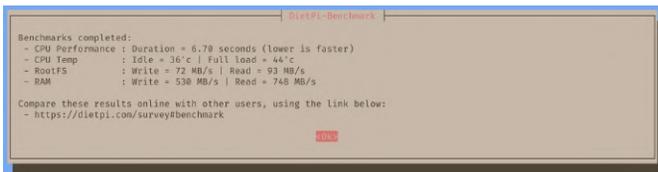
Con las últimas funcionalidades de Docker, con las que puedes crear imágenes ARM tan fácilmente como para x86, prácticamente no hay inconvenientes a la hora de usar un SBC ARM para que cubras las necesidades de tu servidor. Espero que Hardkernel y Amlogic incorporen los parches para soportar esta placa. Por ahora necesitas usar el propio árbol del Kernel de Hardkernel (<https://github.com/hardkernel/linux>) o recurrir a su wiki para más detalles (<https://wiki.odroid.com/odroid-n2/odroid-n2>).

Primero, instala DietPi, una distribución Linux muy liviana basada en Debian. Ya tienen una imagen para el ODROID-N2. Simplemente he descargado y desempaquetado el archivo y lo he guardado en la memoria del eMMC usando Balena Etcher. Recuerda comprar el lector eMMC-USB el cual te facilitará la vida. El lector eMMC-USB está disponible en: <https://www.hardkernel.com/shop/emmc-module-reader-board-for-os-upgrade/>

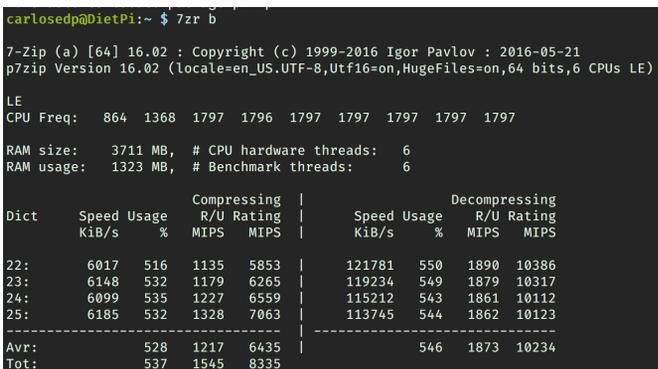
### Pruebas de rendimiento CPU/Memoria

Aquí he comparado parámetros artificiales (Prueba de rendimiento DietPi y 7zip). Estas pruebas nos ofrecen una visión general del rendimiento de las placas

### ODROID-N2 4GB



**Figura 03 - Prueba de rendimiento con Diet-pi en ODRROID-N2**

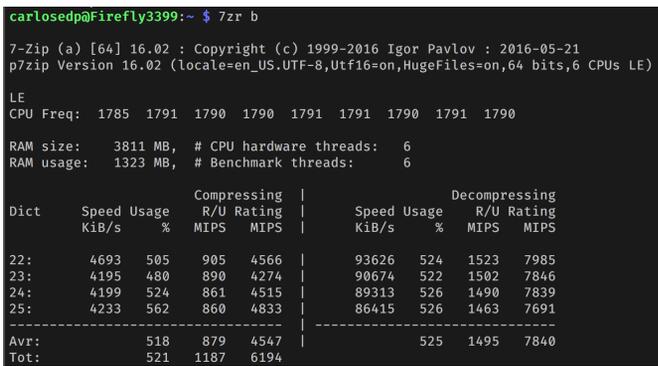


**Figura 04 - Pruebas de rendimiento con 7zr b en ODRROID-N2**

### Firefly RK3399-4GB



**Figura 05 - Pruebas de rendimiento con Diet-pi RK3399**



**Figura 06 - Pruebas de rendimiento con 7zr b RK3399**

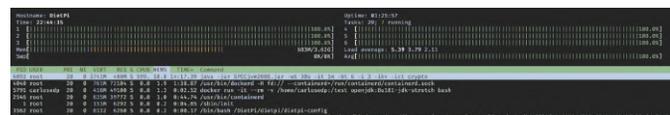
Como término medio, el ODRROID-N2 es 30-35% más rápido que el Firefly RK3399, que es mi placa por defecto. Además, tiene un rendimiento de memoria mucho mejor, hasta un 40% más rápida. También he realizado pruebas con otras placas RK3399 y todas proporcionaron resultados similares.

### Pruebas de rendimiento de Java

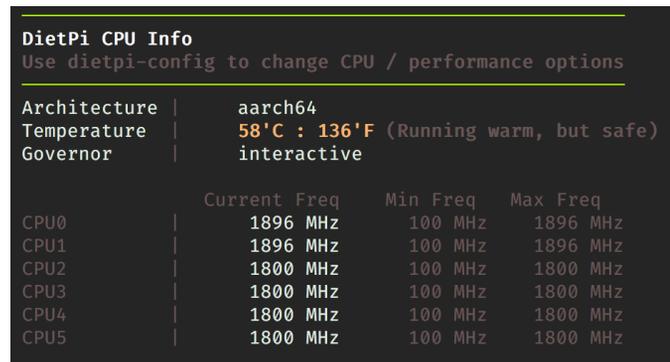
Ejecuté algunas pruebas de rendimiento de Java en consonancia con mi publicación anterior comparando resultados en SPECjvm2008. Estas pruebas, también las ejecuté en el Firefly RK3399 para que los

resultados pudieran ser comparados con las otras pruebas de rendimiento que ya había hecho con anterioridad. Las pruebas se ejecutaron en un contenedor Docker con los siguientes 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]
```



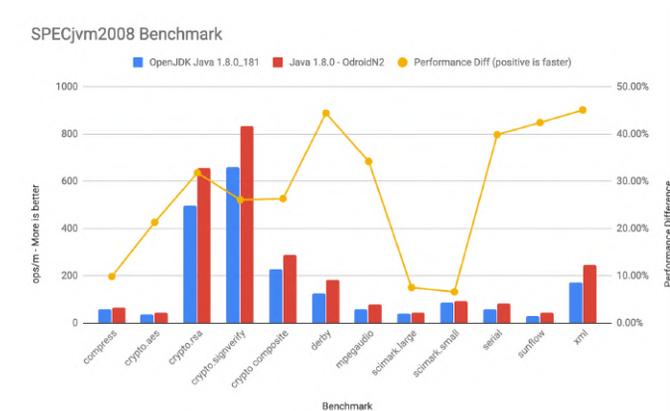
**Figura 07 - Aquí tienes las velocidades del núcleo y la temperatura durante las pruebas de rendimiento (100% en todos los núcleos)**



**Figura 08 - Información de la CPU**

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%
<b>Average:</b>			<b>27.94%</b>

**Figura 09 - Gráfica de resultados SPEC 2008**



**Figura 10 - Gráfica de resultados SPEC 2008**

Como puede verse, el incremento del rendimiento de alrededor de un 30% fue continuado en comparación

con RK3399.

## Red

En este caso testee la red usando iperf3. Probé tanto el TX como el RX utilizando el Ethernet de 1Gbps conectado al mismo switch del otro ordenador. Como servidor de pruebas, usé mi Macbook Pro conectado con un adaptador Ethernet de 1Gbps. Aquí puedes ver los resultados del ODROID-N2:

```
carlosedp@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
```

Figura 11 – Pruebas de rendimiento iperf3

El modo de tráfico inverso alcanza resultados más bajos, pero observé números similares al probar la placa RK3399.

```
carlosedp@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
```

Figura 12 – Pruebas de rendimiento iperf3 de tráfico inverso

Probé a deshabilitar la descarga de la suma de verificación de la red, un problema conocido en los SOC de Rockchip, pero el rendimiento no cambió.

## Conclusión

La placa ODROID-N2 tiene un gran potencial y es la SBC ARM más potente que he visto. Los posibles usos son prácticamente infinitos, desde un mini servidor doméstico a un centro multimedia con todas las funciones o un escritorio que ejecute casi cualquier carga de trabajo vía instalación o a través de contenedores. Además, es fantástico para montar un clúster Kubernetes con varios nodos.

Su consumo de energía es sorprendente pudiendo estar siempre encendido con tan sólo un consumo de 2.8W mientras esté inactivo y de 6.5W mientras haces una evaluación comparativa con los 6 núcleos al 100%. Además, resulta muy fácil grabar nuevas imágenes, usando eMMC, y la conectividad es suficiente en la mayoría de los casos. Me encantaría ver una ranura PCI-E o una M.2. Un conector para unidades NVMe. Encontré un documento que indicaba que el S922X SOC contiene 1 carril PCI-E que en el caso de ODROID-N2 ha sido usado para el puerto USB 3.0. Otras empresas podrían utilizar este carril PCI-E como conector M.2 o ranura PCI-E.

En conjunto, recomiendo la placa y la ubico en el primer puesto de mi lista por sus prestaciones y su precio razonable teniendo en cuenta su rendimiento y las características que ofrece. Creo que podría transformarlo pronto en un escritorio ARM64 con una distribución Linux totalmente funcional como Ubuntu o si incluso Fedora, si puedo exportarlo.

Este artículo ha sido escrito por Carlos Eduardo y está publicado en su página, que está disponible en <https://medium.com/@carlosedp/hardkernel-odroid-n2-review-and-benchmarks-b6996b002582>.

# Conociendo un ODROIDian: Ry (@lordhardware)

© June 1, 2019 By Rob Roy Conociendo un ODROIDian



*Por favor h́ablanos un poco sobre ti.* Soy de Australia, nací en la década de los 90 y crecí en un pequeño pueblo a pocas horas de la ciudad más cercana. No tuve acceso a mucha tecnología al crecer, pero logré comprar un Atari 2600 en un evento de intercambios de comunidad, que fue mi primer contacto con todo lo interactivo.

Desde entonces, la tecnología ha sido sin duda una de mis pasiones.

Trabajo como arquitecto de soluciones para una pequeña empresa de TI en Australia, ahora mucho más cerca de la ciudad que soy más viejo. Estudié programación en la universidad, pero dejé de estudiar cine. Pensé, si alguien va a criticar mis resultados creativos, ¡que lo hagan desde un punto de vista artístico! Tengo un socio con el que llevo una década trabajando en finanzas. No comparte mi afición por la electrónica, pero al menos la tolera.

*¿Cómo empezaste con los ordenadores?* Cuando me mudé a los barrios residenciales de niño, mi familia

no tenía dinero para comprarme un ordenador, pero solía recopilar los panfletos de precios de todas las tiendas de informática de la zona y preguntaba constantemente en el mercado local de todo lo relacionado con los PCs a los vendedores. A los 9 años, logré montar mi primer ordenador con los componentes que encontraba en la basura (despilfarros de ayuntamientos, una colección anual de todos los artículos rotos no deseados). Era una IBM con un disco duro de 20MB (solo el espacio suficiente para instalar Commander Keen) con una tarjeta Matrox que funcionaba al tercer arranque.

*¿Qué te atrajo a la plataforma ODROID?* Siempre me ha gustado el mecaniqueo y, honestamente, el precio frente al rendimiento es lo que más me atrae.

*¿Cómo usas tus ODROIDS?* Tengo un XU3, que actualmente es el cerebro de mi sistema de calefacción por conductos. Mi Odroid-W se está consumiendo en una caja y me culpo a mí mismo cada vez que lo miro. Sin embargo, mi Odroid-Go está

haciendo un gran trabajo cumpliendo el objetivo que tenía mi Odroid-W.



**Figura 1 - Ry tiene un buen taller de música.**



**Figura 2 - Ry tiene varios instrumentos que toca.**

*¿Cuál es tu ODROID favorito y por qué?* Tiene que ser el Odroid-Go. Cuando crecí y conseguí algo de dinero por mi cuenta, el Gameboy Pocket fue una de las primeras cosas que tuve en mis manos. El Go es un maravilloso viaje a la nostalgia sin el increíble coste que terminaban siendo las pilas AAA.



**Figura 3: El ODROID-Go le permite a Ry revivir sus nostálgicos recuerdos con la Gameboy Pocket**

*¿Qué innovaciones te gustaría ver en futuros productos de Hardkernel?* Me encantaría ver sistemas pequeños y eficiente desde el punto de vista energético. Siempre me ha fascinado el hecho de que la gente consiguiera ejecutar programas o funciones increíblemente complejas en un hardware que, sin discusión alguna, no estaba pensado para ello. Honestamente, creo que vivimos en una época dorada en la que las posibilidades de nuestro hardware superan con creces nuestras capacidades para utilizarlos de una manera eficiente, por lo que, de muchas maneras, nos hemos vuelto perezosos con nuestra programación. Ver las cosas que la gente hace con los sistemas integrados me da mucha esperanza.

*¿Qué aficiones e intereses tienes aparte de los ordenadores?* Me apasiona la música, monté un estudio en mi garaje y suelo quedarme allí durante las vacaciones para tocar la batería.



**Figura 4: Ry tiene un montón de proyectos sin terminar en su taller que incluye un ODROID-W**

*¿Qué consejo le darías a alguien que quiera aprender más sobre programación? Simplemente sigue así. La programación es una espada de doble filo. Se trata de aprender reglas lógicas y cómo aplicarlas, pero también requiere que pienses lateralmente y seas creativo para resolver los problemas que te vas encontrando. Por un lado, es como aprender un nuevo idioma, mientras que, por otro lado, se te pide que escribas poesía con el matiz expresivo del nuevo lenguaje en mente. Nunca te volverás bueno lamentando no haber hecho más de algo, aunque puedes estar increíblemente satisfecho incluso con la hazaña más pequeña.*