Doble/Triple Arranque ODROID-C2 • BuildRoot • Homebridge • Juegos Linux

ODROID Magazine

Año Cuatro
Núm. #38
Feb 2017

Muevete con VU8-C

Llevate simpre tu ODROID-C0, C1 o C2 de viaje contigo

- Convierte tu ODROID-XU4 en un polivalente y versátil NAS
- Controla el consumo de energia de tu ODROID con SmartPower2

Qué defendemos...

Nos esmeramos en presentar una tecnología punta, futura, joven, técnica y para la sociedad de hoy.

Nuestra filosofía se basa en los desarrolladores. Continuamente nos esforzamos por mantener estrechas relaciones con éstos en todo el mundo.

Por eso, siempre podrás confiar en la calidad y experiencia que representa la marca distintiva de nuestros productos.

Simple, moderno y único.
De modo que tienes a tu alcance lo mejor para hacer todo lo que imagines

HARDKERNEL

Pollin ELECTRONIC

Realizamos envios de ODROID-C2 and ODROID-XU4 a los países de la UE! Ven y visita nuestra tienda online!

Dirección: Max-Pollin-Straße 1
85104 Pförring Alemania

Teléfono & Fax
telf : +49 (0) 8403 / 920-920
email : service@pollin.de

Nuestros productos ODROID se pueden encontrar en: http://bit.ly/1tXPXwe

EDITORIAL

Recientemente hemos desarrollado un kit ODROID-VU8 (90$ en http://bit.ly/2k8bML5), ahora tenemos una gran tablet de 64 bits con pantalla táctil, Ubuntu 16.04 y conexión Gigabit Ethernet. La resolución de 1024 x 768 es perfecta tanto para Android como para Linux, y la carcasa permite montar un ODROID-C0, C1, C1+ o C2 en su interior permitiendo el acceso a dos puertos USB. Se puede utilizar como tablet de juegos, sistema de entretenimiento, estación de trabajo de programación portátil, kit de diagnóstico de red y mucho más. Hardkernel también ha lanzado una versión actualizada de la popular fuente de alimentación SmartPower (40$ http://bit.ly/2j3hhcv), que incorpora un servidor web.

Para que te sea más fácil usar tu ODROID-VU8, te presentamos dos métodos para configurar un doble o triple sistema de arranque. Adrian nos muestra como configurar un ODROID-XU4 como NAS (Network Attached Storage), @korinel nos presenta una guía para usar HomeBridge y así poder integrar todas las tecnologías de tu hogar, y @Brian.K analiza el uso de BuildRoot para compilar Linux y generar una imagen de disco. También revelamos una simple modificación para reducir aún más el consumo de energía del C2, y Tobias compara la velocidad del emulador PPSSPP en varios modelos ODROID.

ODROID Magazine, que se publica mensualmente en http://magazine.odroid.com/, es la fuente de todas las cosas ODROIDianas. • Hard Kernel, Ltd. • 704 Anyang K-Center, Gwanyang, Dongan, Anyang, Gyeonggi, South Korea, 431-815 • fabricantes de la familia ODROID de placas de desarrollo quad-core y la primera arquitectura ARM "big.LITTLE" del mundo basada en una única placa.
Para información sobre cómo enviar articulos, contacta con odroidmagazine@gmail.com, o visita http://bit.ly/1yplmXs.
Únete a la comunidad ODROID con miembros en más de 135 países en http://forum.odroid.com/ y explora las nuevas tecnologías que te ofrece Hardkernel en http://www.hardkernel.com/

HARDKERNEL

NUESTRO MARAVILLOSO PRESONAL ODROIDIAN:

Rob Roy, Editor Jefe

Soy un programador informático que vive y trabaja en San Francisco, CA, en el diseño y desarrollo de aplicaciones web para clients locales sobre mi cluster ODROID. Mis principales lenguajes son jQuery, angular JS y HTML5/CSS3. También desarrollo SO precompilados, Kernels personalizados y aplicaciones optimizadas para ODROID basadas en las versiones oficiales de Hardkernel, por los cuales he ganado varios Premios. Utilizo mi ODROIDs para diversos fines, como centro multimedia, servidor web, desarrollo de aplicaciones, estación de trabajo y como plataforma de juegos. Puedes echar un vistazo a mi colección de 100 GB de software ODROID, kernel precompilados e imágenes en `http://bit.ly/1fsaXQs`.

Bruno Doiche, Editor Artístico Senior

Divertidas curiosidades: Bruno y Rob estaban hablando de las imágenes del VU8 para la portada. Bruno quería conseguir dos ángulos para presentar el VU8 como una plataforma versátil, pero todo el material del que disponíamos era de baja resolución. Rob tomó unas fotos de su nuevo VU8 y se las envió a Bruno. Debido a la diferencia horaria, Bruno las recibió a las 2AM y no se dio cuenta de que no tenían el enfoque perfecto, de modo que tuvo que trabajar mucho para conseguir que el dispositivo tuviera buen aspecto en portada. ¡Pero valió la pena!

Manuel Adamuz, Editor Español

Tengo 31 años y vivo en Sevilla, España, aunque nací en Granada. Estoy casado con una mujer maravillosa y tengo un hijo. Hace unos años trabajé como técnico informático y programador, pero mi trabajo actual está relacionado con la gestión de calidad y las tecnologías de la información: ISO 9001, ISO 27001, ISO 20000 Soy un apasionado de la informática, especialmente de los microordenadores como el ODROID, Raspberry Pi, etc. Me encanta experimentar con estos equipos y traducir ODROID Magazine. Mi esposa dice que estoy loco porque sólo pienso en ODROID. Mi otra afición es la bicicleta de montaña, a veces participo en competiciones semiprofesionales.

Nicole Scott, Editor Artístico

Soy una experta en Producción Transmedia y Estrategia Digital especializa en la optimización online y estrategias de marketing, administración de medios sociales y producción multimedia impresa, web, vídeo y cine. Gestiono múltiples cuentas con agencias y productores de cine, desde Analytics y Adwords a la edición de vídeo y maquetación DVD. Tengo un ODROID-U3 que utilizo para ejecutar un servidor web sandbox. Vivo en el área de la Bahía de California, y disfruta haciendo senderismo, acampada y tocando música. Visita mi web http://www.nicolecscott.com.

James LeFevour, Editor Artístico

Soy un especialista en medios digitales que disfruta trabajando como freelance en marketing de redes sociales y administración de sitios web. Cuanto más aprendo sobre las posibilidades de ODROID más me ilusiona probar cosas nuevas con él. Me traslade a San Diego desde el Medio Oeste de los EE.UU. Continuo muy enamorado de muchos de los aspectos que la mayoría de la gente de la Costa Oeste ya da por sentado. Vivo con mi encantadora esposa y nuestro adorable conejo mascota; el cual mantiene mis libros y material informático en constante peligro.

Andrew Ruggeri, Editor Adjunto

Soy un ingeniero de sistemas Biomédicos anclado en Nueva Inglaterra que actualmente trabaja en la industria aeroespacial. Un microcontrolador 68HC11 de 8 bits y el código ensamblador son todo lo que me interesa de los sistemas embebidos. Hoy en día, la mayoría de los proyectos en los que trabajo están en lenguajes C y C ++, o en lenguajes de alto nivel como C# y Java. Para muchos proyectos, utilizo placas ODROID, pero aún sigo intentando utilizar los controladores de 8 bits cada vez que puedo (soy un fan de ATMEL). Aparte de la electrónica, soy un amante de la fotografía analógica y desarrollo la película friki con la que disfruto intentando hablar en idiomas extranjeros.

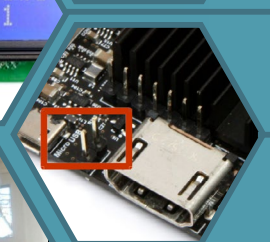
Venkat Bommakanti, Editor Adjunto

Soy un apasionado de los ordenadores desde la bahía de San Francisco en California. Procuro incorporar muchos de mis intereses en proyectos con ordenadores de placa reducida, tales como pequeños modificaciones de hardware, carpintería, reutilización de materiales, desarrollo de software y creación de grabaciones musicales de aficionados. Me encanta aprender continuamente cosas nuevas, y trato de compartir mi alegría y entusiasmo con la comunidad.

Josh Sherman, Editor Adjunto

Soy de la zona de Nueva York, y ofrezco mi tiempo como escritor y editor para ODROID Magazine. Suelo experimentar con los ordenadores de todas las formas y tamaños: haciendo trizas las tablets, convirtiendo Raspberry Pi en PlayStations y experimentado con los ODROIDs y otros SoCs. Me encanta trabajar con los elementos básicos y así poder aprender más, y disfrutar enseñando a otros escribiendo historias y guías sobre Linux, ARM y otros proyectos experimentales divertidos.

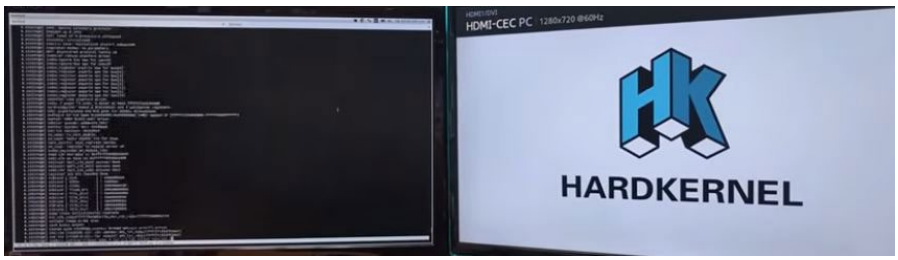
INDICE



DOBLE Y TRIPLE ARRANQUE PARA EL ODROID-C2

CAMBIA FACILMENTE ENTRE SISTEMAS OPERATIVOS EN UN UNICO ODROID

por @Molorius y @loboris



Existen un par de métodos para configurar el ODROID-C2 con un sistema de doble arranque, dependiendo de su experiencia técnica. Ambos métodos te permitirán ejecutar cualquier combinación de sistemas operativos, tales como Android y Ubuntu, OpenELEC y Lakka, o Kodibuntu y Debian. Si quieres utilizar un menú de arranque para seleccionar el sistema operativo o tener un sistema con triple arranque, recurre al método automatizado creado por @loboris. Para tener un control total sobre los scripts de intercambio, o para hacer recurrir a la programación, utiliza el método manual descrito por @Molorius, que se puede adaptar para tener disponibles tantos sistemas operativos como quieras.

Metodo manual

Primero, crea una imagen de arranque de Android en una SD o eMMC, descargándotela desde http://bit.ly/2be993R y copiándola a los soportes de almacenamiento utilizando Win32 Disk Imager o un método similar. Luego, haz lo mismo con una imagen de Ubuntu de http://bit.ly/2b58GEe.

1. Graba la imagen de Ubuntu en el Módulo eMMC. No insertes el módulo en tu Odroid (por ahora). Necesitamos editar el archivo /etc/fstab, el cual requiere acceso root. En mi ordenador, yo escribí en el terminal:

```
$ sudo gedit /media/username/rootfs/etc/fstab
```

Cambia “LABEL=boot” por “LABEL=VFAT”. Guarda y Salte.

2. Guárdalo todo en la partición de arranque en un directorio aparte en tu ordenador.

3. Para el siguiente paso, yo utilicé un programa llamado Disks, que viene con Ubuntu. Lo que estamos haciendo es crear un .img de la partición rootfs. Con Disks, navegué hasta la partición rootfs y seleccioné “Create Partition Image”. Guarda esto en algún lugar de tu ordenador. Anota cuanto ocupa el archivo, necesitarás esta información más adelante. Sólo tienes que hacer los pasos 1-3 una vez para obtener los datos. Si necesita volver a grabar Ubuntu, empieza desde el paso 4.

4. Graba Android en la eMMC. Sí, simplemente hazlo. No conozco otro modo de hacerlo. Inserta el módulo en tu Odroid y dejarlo que se instale. Abre Terminal Emulator y escribe:

```
$ vi storage/internal/boot.ini
```

Cambia “root =/dev/mmcblk0p2” por “root =/dev/mmcblk1p2”. Guarda y Salte.

5. Vuelve a colocar el módulo en tu ordenador. Con Disks (o lo que estés usando), elimina las particiones cache y rootsystem. Si quieres, puedes poner en su lugar otra partición. Elimina también la última partición. Crea una nueva partición con el mismo tamaño que el archivo .img que creaste anteriormente (ver paso 3). En Disks, seleccioné la opción “Restore Partition Image” para restaurar la partición que acababa de crear. Seccciona el .img que creaste en el paso 3. Esto sólo debería reemplazar la última partición, no todo el módulo eMMC.

6. En la partición VFAT, cambia el nombre a “boot.ini” por “boot.ini.android”. Copia los archivos que guardaste en el paso 2 dentro de esta partición. NUNCA MUEVAS LA PARTICION VFAT. Sí lo sé, es raro tener tanto espacio. Pero no logré hacerlo de otro modo. Incluso moviéndola y luego colocándola de nuevo, hacía que Android no arrancara. Ubuntu carga sin importar dónde se encuentre. Si alguien encuentra otra forma de hacerlo, por favor que nos lo diga. Sería fantástico no tener que grabar Android cada vez.

Graba la imagen de Android a tu tarjeta SD. Insertala en tu Odroid. Abre Terminal Emulator y escribe:

```
$ su
$ mount -o rw,remount /
```

```
$ vi fstab.odroidc2
```

Cambia "/dev/block/mmcblk0p4" por /dev/block/mmcblk1p4 ". Cambia "/dev/block/mmcblk0p3" por /dev/block/mmcblk1p3 ". Apaga tu Odroid. Coloca la tarjeta en tu ordenador y elimina la partición VFAT. Puedes utilizar esto más adelante para obtener espacio adicional. NO MUEVAS NINGUNA DE LAS OTRAS PARTICIONES. ¡Ahora podrás cambiar de sistema operativo! Para ello, sólo tiene que cambiar el archivo "boot.ini" por el que quieras iniciar. Aquí tienes una forma de hacerlo rápidamente en Ubuntu y Android.

En Ubuntu, escribe lo siguiente en una ventana Terminal:

```
$ pluma boot_android.sh
```

Añade las siguientes líneas al archivo, guardalo y salte:

```
#!/bin/bash
mv /media/boot/boot.ini /media/boot/boot.ini.ubuntu
mv /media/boot/boot.ini.android /media/boot/boot.ini
reboot
```

Haz el script ejecutable:

```
$ sudo chmod 777 boot_android
```

Ejecuta el archivo escribiendo el siguiente comando en una ventana Terminal. Te sugiero que ponga un acceso directo en el escritorio para este comando:

```
$ ./boot_android.sh
```

En Android, escribe el siguiente comando en la app Terminal:

```
$ su
$ mount -o rw,remount /
$ vi /bin/boot_ubuntu.sh
```

Después, crea el script:

```
#!/bin/sh
mv /storage/internal/boot.ini /storage/internal/boot.
ini.android
mv /storage/internal/boot.ini.ubuntu /storage/inter-
nal/boot.ini
reboot
```

Hazlo ejecutable:

```
$ chmod 777 /bin/boot_ubuntu.sh
```

Ejecuta el archivo usando este comando en la app Terminal:

```
$ boot_ubuntu.sh
```

Te sugiero usar el widget Term Shortcut para poder ejecutar este script desde la pantalla principal. Para comentarios y sugerencias, visita el post original en http://bit.ly/2iGKvzb.

Método automatizado

Para aquellos que quieran crear un doble sistema de arranque usando un script automatizada, deben descargar los scripts ODROID Installer de http://bit.ly/2jtlVyX. Puedes utilizar las imágenes precompiladas descomprimiendo el archivo multiboot_install_images.zip y ejecutar el siguiente comando:

```
$ sudo dd if=multiboot_install_[c2|xu4].img \
  of=/dev/sdX bs=1MB oflag=direct
```

También puede instalar la imagen tú mismo compilando la imagen del instalador:

```
$ sudo ./prepare_selfinst <destination_
card>|<destination_image_name> c2|xu4
```

Una vez creado el archivo de imagen, puede escribir la imagen en la tarejta SD usando el comando dd.

A continuación, copia las fuentes de la instalación (imagen de actualización de Android "update.zip", el archivo .img de Linux y/o el archivo .tar de OpenELEC) en la primera partición de tu unidad USB. Cambia el nombre de la imagen de instalación de Linux por "linux.img", el nombre de la instalación de OpenELEC por oelec.tar y el nombre del archivo de instalación de Android por update.tar.gz o update.zip. Conecta tu unidad USB al ODROID, inserta la SD que contiene la imagen del instalador y enciende el C2. Selecciona "Install" en el menú y sigue las instrucciones para seleccionar los tamaños de partición y destino de la instalación (tarjeta SD o módulo eMMC). Una vez finalizada la instalación, retira la tarjeta SD del instalador e inicia el dispositivo desde el soporte principal.

Consejos prácticos

Puedes cambiar la resolución de Android editando manualmente el archivo boot.ini.android, ya que ODROID Utility de Hardkernel no funciona con esta configuración. También hay una opción para describir los sistemas operativos en el menú de arranque añadiendo la siguiente línea a los archivos boot.ini.*:

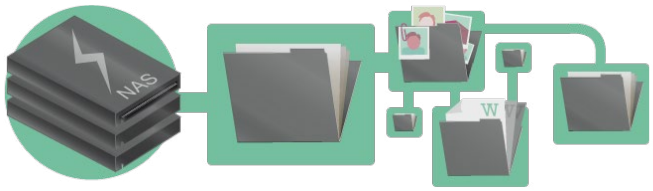
```
#DESCRIPTION "<description goes here>"
```

Para comentarios y sugerencia visita http://bit.ly/2jtlVyX.

CONVIERTE TU XU4 EN UN POLIVALENTE Y VERSATIL DISPOSITIVO DE ALMACENAMIENTO EN RED (NAS)

TU SERVIDOR DOMESTICO AL ESTILO NAVAJA SUIZA

por Adrian Popa (@mad_ady)



Compré mi ODROID-XU4 con la intención de convertirlo en un NAS. Sin embargo, no quería conformarme con una distribución NAS especifica como OpenMediaVault porque quería que mi ODROID-XU4 hiciera mucho más de lo que puede llegar a hacer un simple NAS. Por ejemplo, tenía pensado transcodificar programas de TV grabados desde mi TV al estándar H264, usando el codificador de hardware del ODROID-XU4 (tal y como se describe en http://bit.ly/2jnv4Za), y también dar uso a los pines GPIO más adelante. Un problema añadido que tunia con OpenMediaVault es que se ejecuta sobre Debian, y yo quería seguir usando Ubuntu para poder beneficiarme de los paquetes más recientes.

Estaría perdiendo gran parte de la comodidad que supone usar una distrobución especializada, de modo que tenía que encontrar formas alternativas de hacer las cosas de un modo simple y sencillo. Esto representaría una gran oportunidad para adquirir nuevos conocimientos.

Estos son los pasos que debemos seguir:

Instalar el kernel ver. 4.9 estandar (opcional)
Instalar Webmin (http://bit.ly/J5Wtfl) para gestion del sistema
Montar los discos
Configurar los recursos compartidos en red (Samba / NFS)
Instalar Owncloud
Asegurar y optimizar el sistema operativo

Para poder seguir estas instrucciones, se supone que cuentas con un nivel medio/alto de experiencia con el sistema.

Instalar el kernel estándar (4.9)

El ODROID-XU4 tiene la ventaja de ser compatible con el kernel estándar, que necesitaba específicamente para la transcodificación de video. El kernel estándar cuenta con los drivers más recientes, además de tener un mejor soporte. Sin embargo, presenta algunos problemas: soporte HMP con errores, USB3 inestable y no hay sonido a través del HDMI, por nombrar algunos. Esto te lleva a tener que sopesar los pros y los contras, y tomar una decisión de acuerdo a tus necesidades.

Para instalar el kernel estándar, necesitarás seguir las instrucciones oficiales (Hardkernel está trabajando en un núcleo oficial 4.9 para ODROID-XU4 y lo más probable es que lo publique como un paquete deb) o sigue las instrucciones generales en http: /bit.ly/2jnv4Za. Recuerda desenchufar completamente tu ODROID-XU4 de todas las fuentes de energía (alimentación, HDMI, USB), de lo contrario perderás tu red cuando arranques el sistema por primera vez.

Webmin

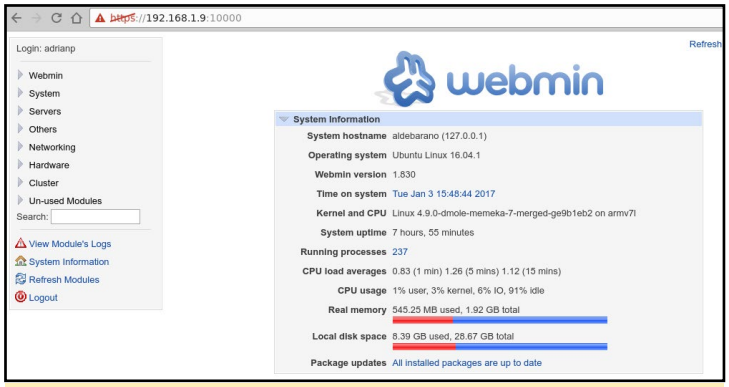
Todo NAS necesita una interfaz web agradable. Desafortunadamente, la interfaz de usuario de Open-MediaVault no es una opción, y después de buscar durante mucho tiempo alternativas, decidí usar Webmin. Webmin existe desde 1997 y cuenta con buen soporte para las tareas de mantenimiento generales de servidores. Tiene la ventaja de que incluso los usuarios inexpertos pueden desenvolverse y con el sistema de ayuda integrada pueden llegar a configurar y administrar todo tipo de servidores como Apache, MySQL, Mail, DNS y muchos más. Cuenta con un soporte muy robusto para la gestión de RAID y LVM, y además permite la posibilidad de compartir archivos por Samba y NFS. Lamentablemente, carece de soporte para servicios como Transmission o Owncloud, pero éstos siempre los podemos configurar manualmente.

Para instalarlo, sigue estos pasos:

```
$ echo "deb http://download.webmin.com/download/re-
pository \
```

```
  sarge contrib" | sudo tee /etc/apt/sources.list.d/
webmin.list
$ wget http://www.webmin.com/jcameron-key.asc
$ sudo apt-key add jcameron-key.asc
$ rm jcameron-key.asc
$ sudo apt-get update
$ sudo apt-get install libapt-pkg-perl \
  libnet-ssleay-perl libauthen-pam-perl \
  libio-pty-perl apt-show-versions \
  apt-transport-https
$ sudo apt-get install webmin
$ sudo systemctl enable webmin
$ sudo systemctl start webmin
```

Puedes iniciar sesión con la IP de tu dispositivo en el puerto 10000 para usar la interfaz web: https://ip-odroid:10000. Sin embargo, tras iniciar sesión (con cualquier usuario del sistema con acceso sudo) posiblemente te impresione muy poco la interfaz por defecto. Parece sacada de la década de los 90.



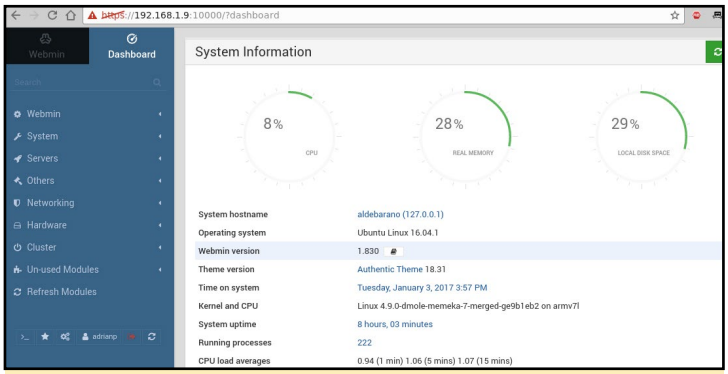
Típica interfaz de Webmin

Lo primero que debemos hacer es mejorar su aspecto con ayuda de un tema. El tema más atractivo se llama "Authentic Theme", que cuenta con un montón de funcionalidades, incluyendo la vista para móviles. Puedes obtener la versión más reciente en http://bit.ly/2jf468e e instalarla con el comando:

```
$ wget http://bit.ly/2jRfecC
```

Navega dentro de Webmin hasta "Webmin Configuration -> Webmin Themes -> Install themes -> From uploaded file" y selecciona el tema que acabas de descargar. Tras una breve espera y actualización posterior, aparecerá una página como la que se muestra en la imagen de arriba a la derecha.

Puede explorar las caracteristicas de Webmin utilizando la herramienta de búsqueda de la interfaz. Ten en cuenta que puede instalar módulos de terceros disponibles en http://bit.ly/2jf6KLd. Hay una cosa que debes cambiar cuanto antes: por defecto webmin tiene un proceso en segundo plano que monitoriza la temperatura del disco duro, esto hacia que mis discos se activaran cada 5 minutos. Tras una laboriosa búsqueda



NAS - Webmin con Authentic theme

encontré una solución en http://bit.ly/2kyzXBv. Simplemente añade la línea "collect_notemp=1" a /etc/webmin/system-status/config y reinicia el proceso webmin.

Montar los discos

En primer lugar, tendrás que decidir si va a utilizar RAID o LVM con tus discos, y qué sistema de archivos vas a utilizar. No voy a entrar en detalles sobre la configuración de RAID/LVM porque este tema ya ha sido tratado en artículos anteriores. Sin embargo, incluso sin tener mucha experiencia, puedes utilizar Webmin para que haga el trabajo pesado y utilizar la ayuda integrada para profundizar en el tema. Webmin te pedirá que instales cualquier dependencia que falte. Una vez que hayas preparado tus particiones, puedes empezar a montarlas.

El método convencional para montar las particiones es usar /etc/fstab (y Webmin también tiene un completo módulo para gestionar esto), pero puede tener problemas si inicias tu sistema sin el disco conectado (a systemd le gusta esperar el disco). Yo prefiero usar autofs, que permite montar discos (local o basado en red) a demanda y los desmonta cuando no están en uso. Por desgracia, webmin no lo gestiona, así que tendrá que utilizar el intérprete de comandos:

```
$ sudo apt-get install autofs
```

Tendrás que editar /etc/auto.master y añadir una entrada para montar tu disco, especificando el directorio base y su archivo de configuración:

```
$ sudo vi /etc/auto.master
# add at the end your mountpoint
/media/yourdisk /etc/auto.yourdisk --timeout 20
```

En el comando anterior, reemplaza tu disco con la ruta que quieres usar. Luego, edita este archivo de configuración y añade tus particiones y tus parámetros de montaje, usando el comando "blkid" para encontrar el UUID correcto para el disco:

```
$ sudo blkid
$ sudo vi /etc/auto.yourdisk
```

```
xfs-partition    -fstype=xfs,dev,exec,suid,noatime
:UUID=9d2d675d-cb08-45b2-b222-c981a8d00c06
```

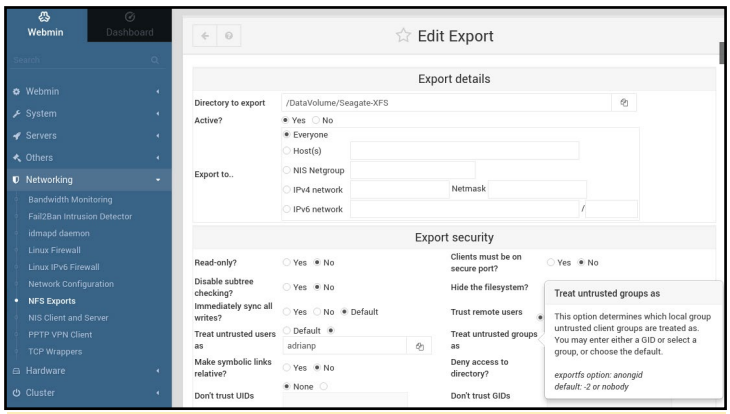
Reinicia autofs y cuando accedas a /media/yourdisk/xfspartition, tu partición se montará automáticamente:

```
$ sudo service autofs restart
```

Deberás tener cuidado con los parámetros de montaje porque cada sistema de archivos tiene sus propios parámetros y puede afectar al rendimiento. Por ejemplo, sino activas el parámetro big_writes en NTFS, tendrás un rendimiento muy bajo. Si tiene dudas, puede engañar y usar Webmin para crear entradas en /etc/fstab, probarlas para ver que los parámetros son los correctos y migrarlas al diseño de autofs más tarde (eso es lo que yo hice). Para forzar a que los discos automontados sean desmontados, simplemente puedes reiniciar el servicio autofs.

Configurar los recursos compartidos de archivos

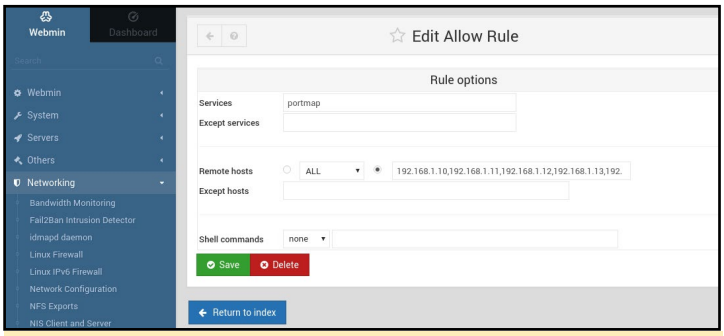
Para configurar los recursos Samba (y también instalar Transmission para descargar torrent), puede seguir la guía "Dis-



Crear recursos compartidos NFS

eñando tu propio Seedbox" que aparece en Odroid Magazine http://bit.ly/2j3xpaK. También puede experimentar con la interfaz de Webmin y crear fácilmente recursos compartidos y usuarios con unos cuantos clics. Por ejemplo, la Figura 3 muestra el cuadro de diálogo "Create NFS share". Al hacer clic en los elementos del formulario se muestra un menú de ayuda contextual que explica muy bien lo que hace cada elemento. Esto puede ayudarle con cosas que tal vez desconozcas.

Cuando crees recurso compartidos Samba/NFS, ten presente desde un principio la seguridad. Samba se autentifica por ID de usuario y contraseña, pero NFS autentifica a los usuarios sólo por IP. Si conoces qué hosts de tu red pueden tener acceso a los recursos compartidos, especifícalo en la configuración. Por ejemplo, a un recurso compartido NFS se le puede dar visibilidad a "Todo el mundo", pero el acceso puede limitarse con iptables o /etc/hosts.allow y /etc/hosts.deny (que son utilizados



Configuración /etc/hosts.allow limitando el acceso de algunos hosts

por el módulo de TCP Wrappers de Webmin).

Utilizandolo con tu configuración por defecto, Samba te dará un rendimiento aceptable, pero con los siguientes ajustes, extraídos de los foros de ODROID, deberías conseguir menos "pausas" en la transferencia de grandes archivos. Añade las siguientes líneas a la sección [global] de /etc/samba/smb.conf:

```
write cache size = 524288
getwd cache = yes
use sendfile = yes
min receivefile size = 16384
```

Instalar Owncloud

Owncloud es un servicio personal "en la nube" que te permite compartir archivos con personas a través de Internet. No voy a entrar en los detalles de su instalación, porque está tratada en un artículo anterior de la revista http://bit.ly/2kgVZpn, pero hay algunas cuestiones que me gustaría señalar.

La instalación es bastante simple en Ubuntu 16.04. Yo utilicé la guía de http://do.co/2bzxhxG y en 10 minutos lo tenía funcionando. Si tiene un nombre DNS (por ejemplo, DNS dinámico para tu hogar), deberías conseguir un certificado SSL válido desde Let's Encrypt (http://bit.ly/1qmIXIY) usando los pasos descritos en http://do.co/2bQpv4M.

Básicamente necesitas instalar los siguientes requisitos previos antes de instalar OwnCloud:

```
$ sudo apt-get install php \
  libapache2-mod-php php-mcrypt \
  php-mysql php-bz2 php-curl \
  php-gd php-imagick php-intl \
  php-mbstring php-xml php-zip \
  mysql-server apache2
```

A continuación, instala el repositorio OwnCloud para Ubuntu y actualiza los paquetes disponibles:

```
$ sudo curl https://download.owncloud.org/download/\
repositories/stable/Ubuntu_16.04/Release.key \
| sudo apt-key add -
```

```
$ echo 'deb https://download.owncloud.org/download/\
repositories/stable/Ubuntu_16.04/ /' \
| sudo tee /etc/apt/sources.list.d/owncloud.list
$ sudo apt-get update
```

Finalmente, puedes instalar OwnCloud:

```
$ sudo apt-get install owncloud
$ sudo systemctl reload apache2
```

Necesitarás crear un usuario de base de datos OwnCloud:

```
$ sudo mysql -u root
> CREATE DATABASE owncloud;
> GRANT ALL ON owncloud.* to 'owncloud'@'localhost'
IDENTIFIED BY 'databasePassword';
> FLUSH PRIVILEGES;
> exit
```

Después de todo este trabajo, puedes iniciar sesión a través de la interfaz web en https://<ip-odroid>/owncloud y finalizar la instalación. Dado que OwnCloud es accesible desde Internet, debes dedicar algún tiempo a pulir tu instalación, tal como se describe en http://bit.ly/2jOTe1F. En mi caso, quería ejecutar el servicio OwnCloud en un puerto diferente (para que los usuarios externos no tuvieran acceso a mis sitios internos), fijar reglas iptables para permitir el acceso sólo desde mi país (basado en datos geo-ip), y configurar fail2ban para protegerme del cálculo de contraseñas automatizadas.

Para ejecutar el host virtual de OwnCloud en un puerto diferente, necesitas hacer algunos ajustes en apache:

```
$ sudo cp /etc/apache2/sites-available/default-ssl.
conf \
  /etc/apache2/sites-available/owncloud.conf
$ cd /etc/apache2/sites-available
$ sudo ln -s ../sites-available/owncloud.conf \
  020-owncloud.conf
```

A continuación, edita /etc/apache2/sites-available/owncloud.conf y realiza los siguientes cambios:

Añade "`Listen 8443`" como primera fila
Cambia la definición de VirtualHost para utilizar el puerto 8443 en lugar de 443 (<`VirtualHost _default_:8443`>)
Cambia DocumentRoot para que apunte a tu instalación de owncloud "`DocumentRoot /var/www/owncloud`"

Cuando hayas terminado, puede reiniciar el demonio de Apache y debería poder acceder sólo a tu OwnCloud con `https://<odroid-ip>:8443/`.

Para empezar con las reglas GeoIP del firewall, necesitarás tener las fuentes del kernel (o las cabeceras del kernel) disponibles. Después, podrás instalar los módulos iptables adicionales con el siguiente comando:

```
$ sudo apt-get install \
  xtables-addons-dkms \
  xtables-addons-common \
  xtables-addons-source
```

El paquete dkms no se instalará correctamente porque algunos de los módulos fallan al compilarse con el kernel 4.9. Puede desactivar los módulos que han fallado y volver a compilar los ajustando los siguientes valores con "n" en lugar de "m" en el archivo /var/lib/dkms/xtables-addons/2.10/build/mconfig:

```
$ sudo vi /var/lib/dkms/xtables-addons/2.10/build/
mconfig
build_ACCOUNT=n
build_LOGMARK=n
build_SYSRQ=n
build_pknock=n
build_psd=n
```

A continuación, necesitarás compilar manualmente el resto:

```
$ cd /var/lib/dkms/xtables-addons/2.10/build/
$ sudo autoconf
$ sudo ./configure
$ sudo make
$ sudo make install
```

Antes de utilizar el módulo geoip, necesitarás activar la base de datos geoip (el prefijo para asignar el país). Es posible que tenga que repetir este paso de vez en cuando para sacar provecho de los datos más recientes:

```
$ sudo apt-get install libtext-csv-xs-perl
$ sudo mkdir /usr/share/xt_geoip
$ sudo /usr/lib/xtables-addons/xt_geoip_dl
$ sudo /usr/lib/xtables-addons/xt_geoip_build -D /
usr/share/xt_geoip /root/GeoIPCountryWhois.csv
```

Todo lo que queda por hacer es crear y probar las reglas iptables para permitir sólo el tráfico que quieras que llegue a tu configuración de cloudcloud. Un ejemplo de regla podría ser:

```
$ sudo iptables -N geo-owncloud
$ sudo iptables -A INPUT -p tcp -m tcp --dport 8443
-j geo-owncloud
$ sudo iptables -A geo-owncloud -s 192.168.1.0/24 -j
ACCEPT
```

```
$ sudo iptables -A geo-owncloud -m geoip ! --src-cc
RO -j DROP
```

No te olvides de guardar tus reglas y aplicarlas al inicio (ya sea con iptables-save o con webmin). Puedes encontrar más detalles sobre geoip en http://bit.ly/2jnwUJD.

```
adrianp@aldebarano:~> sudo tail -f /var/log/fail2ban.log
2017-01-03 07:53:19,143 fail2ban.actions        [1466]: INFO    Set banTime = 600
2017-01-03 07:53:19,145 fail2ban.filter         [1466]: INFO    Set maxRetry = 5
2017-01-03 07:53:19,194 fail2ban.jail           [1466]: INFO    Jail 'sshd' started
2017-01-03 07:53:19,255 fail2ban.jail           [1466]: INFO    Jail 'owncloud' started
2017-01-04 14:38:45,757 fail2ban.filter         [1466]: INFO    [owncloud] Found 172.22.22.2
2017-01-04 14:38:55,097 fail2ban.filter         [1466]: INFO    [owncloud] Found 172.22.22.2
2017-01-04 14:39:02,457 fail2ban.filter         [1466]: INFO    [owncloud] Found 172.22.22.2
2017-01-04 14:39:07,463 fail2ban.filter         [1466]: INFO    [owncloud] Found 172.22.22.2
2017-01-04 14:43:49,032 fail2ban.filter         [1466]: INFO    [owncloud] Found 172.22.22.2
2017-01-04 14:43:49,407 fail2ban.actions        [1466]: NOTICE  [owncloud] Ban 172.22.22.2
```

Fail2Ban haciendo su trabajo en los inicios de sesión fallidos

Configurar fail2ban no es muy complicado una vez que sigue el tutorial de http://bit.ly/2kipXxn. Recuerda instalar primero fail2ban (y probarlo con credenciales falsas):

```
$ sudo apt-get install fail2ban
```

Puesto que hemos añadido un puerto especial para owncloud, tendremos que ajustar la configuración de fail2ban para tenerlo en cuenta. Edita /etc/fail2ban/jail.local y añade "port 8443" a la línea de puerto y reinicia fail2ban:

```
$ sudo vi /etc/fail2ban/jail.local
port = http,https,8443
$ sudo service fail2ban restart
```

Para levantar manualmente la prohibición de una dirección IP en la lista negra, puedes ejecutar el siguiente comando:

```
$ sudo fail2ban-client set owncloud unbanip
172.22.22.2
```

Emular HMP

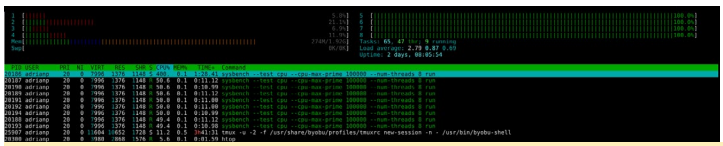
HMP, que significa Heterogeneous Multi-Processing Device, es una mejora opcional que te proporcionará una experiencia más fluida. El ODROID-XU4 viene con dos tipos de núcleos CPU: 4 núcleos pequeños de baja potencia que son los más idóneos para las tareas de fondo y 4 grandes núcleos que están destinados a tareas más pesadas. El kernel oficial 3.10 viene con un programador de tareas "mágico" de Samsung que conoce la potencia real del procesador, y puede trasladar las tareas de los pequeños núcleos a los grandes núcleos cuando la carga de trabajo es alta. Sin embargo, los parches de Samsung han sido rechazados para el kernel estándar (http://bit.ly/2iTbUhr), lo que significa que el kernel asignará de forma aleatoria las tareas a los diferentes núcleos CPU obteniéndose así un rendimiento inconsistente. Hay planes para añadir esta funcionalidad a los sistemas grandes.Little (http://bit.ly/2kiveoJ), pero todavía no hay nada desarrollado. Necesitaremos emular la funcionalidad HMP de alguna forma. Podemos usar cgroups tal y como se indica en http://bit.ly/2jP6KlU.

"cgroups" es una funcionalidad de los modernos kernels que permite la asignación de recursos para varios procesos. En nuestro caso necesitaremos el cpoup "cpuset" para crear un grupo "littlecores" y otro grupo "bigcores". Cada grupo forzará a que los procesos se ejecuten en núcleos específicos configurando la afinidad. Así, los littlecores tendrán las cpus 0-3 y bigcores las 4-7. Afortunadamente, crear los cgroups es fácil:

```
# mkdir -p /sys/fs/cgroup/cpuset/littlecores \
  /sys/fs/cgroup/cpuset/bigcores
# echo "0-3" > /sys/fs/cgroup/cpuset/\
littlecores/cpuset.cpus
# echo "0"> /sys/fs/cgroup/cpuset/\
littlecores/cpuset.mems
# chmod -R 777 /sys/fs/cgroup/cpuset/\
littlecores
# echo "4-7"> /sys/fs/cgroup/cpuset/\
bigcores/cpuset.cpus
# echo "0"> /sys/fs/cgroup/cpuset/\
bigcores/cpuset.mems
# chmod -R 777 /sys/fs/cgroup/cpuset/\
bigcores
```

Desafortunadamente, los comandos solo se mantendrán hasta el siguiente reinicio. De modo que vamos a configurar un servicio para activarlo en el arranque:

```
$ sudo wget -O /etc/systemd/system/cpuset.service
https://raw.githubusercontent.com/mad-ady/\
odroid-ODROID-XU4-optimizations/master/cpuset.service
$ sudo systemctl enable cpuset
$ sudo systemctl start cpuset
```



8 Subprocesos sysbench son forzados a ejecutarse en 4 núcleos

Llegados a este punto, los cgroups estan creados, pero no son utilizados activamente por nadie. Para iniciar manualmente un proceso en un cgroup específico, puedes utilizar cgexec:

```
$ sudo apt-get install cgroup-tools
$ cgexec -g cpuset:bigcores sysbench --test=cpu \
--cpu-max-prime=100000 --num-threads=8 run
```

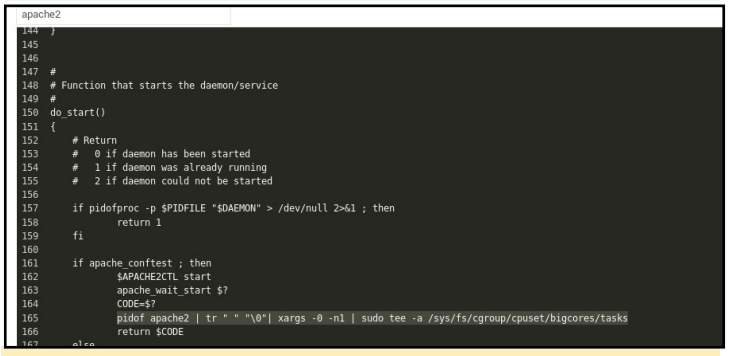
Estamos a mitad de camino. Necesitaremos indicar qué procesos específicos se ejecutarán en los pequeños núcleos y

cuales se ejecutarán en los grandes. Aquí es donde tienes que hacer una lista y decidir qué es lo que quieres. Empieza con una lista con los servicios activos de webmin (System -> Bootup and Shutdown) y desactiva todo lo que no vas a utilizar. En mi caso, desactivé los siguientes servicios: ModemManager, NetworkManager-wait-online, NetworkManager, accounts-daemon, alsa-restore, alsa-state, apport, apport-forward.socket, bluetooth, copas-exploradas, copas. Path, cups.service, cups.socket, lightdm, lxc-net, lxc, lxcfs, plymouth*, rsync, saned, speech-dispatcher y whoopsie.

Necesitarás editar los scripts de inicio para los servicios que quieras y hacer que éstos añadan su PID al cgroup correcto. Una vez que el proceso principal (y sus procesos hijos) formen parte del cgroup correcto, cualquier nuevo proceso hijo heredará el cgroup. Mi intención era añadir cosas como MySQL, Apache, Samba, NFS e incluso Webmin al grupo grande y cosas como SSH (y toda mi actividad con el intérprete de comandos), cron, Munin y Transmisión al grupo pequeño. Esto permite que los procesos que forman parte del NAS sean más rápidos, mientras que resto de tareas pueden funcionar perfectamente con los núcleos pequeños. Si también está utilizando la interfaz gráfica de usuario X11, puede añadir lightdm al grupo "bigcores".

Hay dos tipos de scripts de inicio: scripts systemd nativos y scripts sys-v heredados (/etc/init.d/). Al editar un script systemd (por ejemplo, nfs-mountd.service) necesitará añadir algo como esto a la sección [Service]:

```
ExecStartPost=-/bin/sh -c 'echo $MAINPID | tee -a /
sys/fs/cgroup/cpuset/bigcores/tasks'
```



Cambiando la configuración de inicio de apache

Cuando editamos un script sys-v, es más complicado. Necesitarás encontrar la función de inicio, extraer el PID del proceso iniciado y añadirlo a la lista de tareas. El siguiente es un ejemplo para cambiar el script de inicio de apache:

```
pidof apache2 | tr " " "\0"| xargs -0 -n1 | sudo tee
-a /sys/fs/cgroup/cpuset/bigcores/tasks
```

Debes reiniciar cada servicio después de cambiarlo y asegúrate de comprobar que el PID del proceso está en el archivo de tareas cpuset correcto. Realiza un reinicio completo del sistema y compruébale de nuevo tras reiniciar. Si esto te parece demasiado complicado y estás usando el núcleo 4.9, hay una forma de hacer trampas y ejecutar todas las tareas en los grandes núcleos. Simplemente puedes fijar la afinidad de systemd, y todos tus procesos hijos lo heredarán. La afinidad puede controlarse con el parámetro CPUAffinity en /etc/systemd/system.conf, pero ten en cuenta que desaprovecharás núcleos.

Duración del disco

Para prolongar la vida de tu(s) disco(s), es posible que quieras que dejen de girar tras un período de inactividad. Si utilizas discos SSDs, puede omitir esta sección porque sólo es de aplicación a los discos mecánicos. Los discos pueden recibir un comando "stop" para que se detengan desde un controlador interno, desde la conexion USBSATA o directamente desde el sistema operativo. Sin embargo, a veces los controladores no están bien ajustados y el comando de parada nunca llega. Esto provoca que el disco siga girando, lo cual genera mucho calor y puede causar que la unidad falle antes de lo normal.

Lo normal para gestionar esto es decirle al disco que deje de girar tras un período de inactividad, lo cual se puede hacer con hdparm:

```
$ sudo apt-get install sdparm hdparm
```

Para configurar manualmente el disco y éste se detenga tras 10 minutos de inactividad, ejecutar el siguiente comando:

```
$ sudo hdparm -S 120 /dev/sda
```

Si obtienes errores (como "bad/missing sense data"), hdparm podría no ayudarte con este disco.

Si obtienes errores (como "bad/missing sense data"), hdparm podría no ayudarte con este disco.

Para gestionar las paradas del disco, lo mejor es dejar que udev ejecute el comando una vez que hayas conectado el disco. Puesto que cada disco puede tener funciones diferentes, es posible que desee crear diferentes temporizadores de parada (por ejemplo, un disco para copias de seguridad debería deternerse pronto, otro que tiviera más actividad debería tardas más en pararse), decidí ajustar la regla UDEV basandome en el número de serie del disco. Puede obtener este número de serie mirando en dmesg cuando conectes un disco:

```
[1885221.800435] usb 4-1.3: Product: My Passport 0730
[1885221.800436] usb 4-1.3: Manufacturer: Western
Digital
[1885221.800437] usb 4-1.3: SerialNumber:
57584431414130563623937
```

Para configurar la regla, crea un archivo como este y vuelve a cargar udev:

```
$ sudo vi /etc/udev/rules.d/90-disk.rules
ACTION=="add", ENV{DEVNAME}=="/dev/
sd?", SUBSYSTEM=="block", ENV{ID_SERIAL_SHO
```

```
RT}==”57584431414130563632393 7”, RUN+=”/sbin/hdparm
-S 120 $env{DEVNAME}”
$ sudo udevadm control -R
```

Si hdparm no puede denter el disco, pruebas con otras alternativas como sdparm, que puede enviar un comando SCSI a tu disco, ordenando que se cierre en ese instante:

```
$ sudo sdparm -C stop /dev/sda
```

Existen herramientas como hd-idle (http://bit.ly/2j3zWSk) o scripts periódicos que pueden ejecutar para poner tu disco en reposo. En mi caso no llegaron a funcionar, pero asegúrate de probarlos manualmente antes de decidirte por una solución. Aquí tiene un script manual que chequea el disco (identificado por un UUID de partición) para ver su actividad en una ventana de 10s, y si no detecta actividad en el disco (transferencia de datos), utiliza sdparm para detener el disco. Puedes ejecutarlo a través de cron:

```
$ sudo wget -O /usr/local/bin/hdd-idle.sh http://bit.
ly/2k6LK7Y
$ sudo chmod a+x /usr/local/bin/hdd-idle.sh
$ sudo /usr/local/bin/hdd-idle.sh “4283-E975”
```

Debes tener en cuenta que existen herramientas y servicios que activaran tu disco periódicamente, incluso si no se tranfieren datos. Este tipo de herramientas incluyen smartctl (de smartmontools) y smartd. El servicio smartd comprueba periódicamente la integridad del disco, y si no está correctamente configurado, puede mantener activado el disco sin necesidad. Puedes consultar el hilo en http://bit.ly/2kh6b17 si no lográs averigura que el lo que mantiene tu disco activo. Deberías poder deducir el estado del disco ejecutando este comando:

```
$ sudo smartctl -d sat -n standby -a /dev/sda
```

Si aparece un error, tu disco aún está en modo de espera y debería haber dejado de girar.

Rendimiento del disco flash

Una cosa más a tener en cuenta cuando utilizamos el almacenamiento flash (eMMC o SSD) es que necesitan reajustes periodicos para mantener su velocidad. Básicamente, para escribir en un bloque de almacenamiento, necesitas borrarlo primero y esto lleva más tiempo que escribirlo. Los sistemas de archivos normales no hacen este borrado cuando eliminan datos, así que tras un rato el rendimiento de disco desciende significativamente. Para “revivir” el disco, la operación de reajuste informa al controlador de disco para que borre todos los bloques vacíos, restaurando así las velocidades de escritura. La operación de reajuste debe ser compatible con el sistema de archivos y el controlador de disco. Una vez más, usar cron una vez por semana para ejecutar fstrim puede salvarte de las ralentizaciones:

```
$ sudo crontab -e
#trim the eMMC once a week
15 0 0 * *      /sbin/fstrim / >/dev/null 2>&1
```

Regulador

El rendimiento y la teperatura también dependen directamente del regulador que utilices para la CPU. Manteniendo éste en “performance” consigues un rendimiento superior, pero también generas una gran cantidad de calor. En mis pruebas, la mejor opción fue “ondemand” modificado en base a las recomendaciones de http://bit.ly/2jfaDjw. Para activarlo, asegúrate de seleccionar governor=ondemand en /media/boot/boot.ini, y configura el resto de los parámetros dentro de /etc/rc.local (prueba los comandos antes). Los siguientes comandos funcionan para un kernel 4.9 y pueden variar para el kernel 3.10:

```
$ sudo vi /etc/rc.local
echo 1 > /sys/devices/system/cpu/cpufreq/ondemand/
io_is_busy
echo 10 > /sys/devices/system/cpu/cpufreq/ondemand/
sampling_down_factor
echo 80 > /sys/devices/system/cpu/cpufreq/ondemand/
up_threshold
```

Con la configuración anterior, la CPU aumentará la frecuencia más pronto y tendrá en cuenta el uso de E/S, haciendo que las tareas intensivas de E/S influyan en la frecuencia de la CPU. Esto te permitirá alcanzar un buen rendimiento cuando lo necesites y disminuirá el calor cuando el sistema está inactivo. En mis pruebas, los núcleos pequeños se mantenían al relentin a unos 300MHz, mientras que los grandes se mantenían a unos 200MHz.

Rendimiento de la red - MTU

Si cuentas con una red Gigabit con el cableado adecuado, puede aumentar la MTU (unidad de transmisión máxima) en el sistema de red del ODROID-XU4. Esto te permitirá enviar paquetes más grandes con menos sobrecarga y generar menos interrupciones en el extremo receptor. Sin embargo, para beneficiarte de ello, necesitarás que los dispositivos de red (switches/routers) y los dispositivos finales soporten Jumbo Frames. Lo ideal sería que los Jumbo Frames estuvieran activados en todos los dispositivos de red de tu LAN, de lo contrario, podría experimentar un descenso del tráfico o incluso que los dispositivos no puedan enviarse tráfico pesado entre sí. Por ejemplo, SSH funciona porque utiliza paquetes pequeños, pero cargar una página web o transferir un archivo bloquearía la conexión. Si decides habilitar Jumbo Frame, el valor MTU mágico de ODROID-XU4 es 6975 (http://bit.ly/2jP9zDl). Puede activarlo en el ODROID-XU4 dentro de /etc/rc.local:

```
$ sudo vi /etc/rc.local
#MTU
```

```
/sbin/ifconfig eth0 mtu 6975 up
```

Rendimiento de la red - interrupciones

Otra causa importante del bajo rendimiento de las altas tasas de trafico es el hecho de que el gestor de interrupción en el ODROID se ejecuta sobre la CPU0 por defecto, que es un núcleo pequeño. Puedea aumentar la velocidad de ren en otros 150 Mbps si mueves el gestor de interrupción de la tarjeta de red a un núcleo grande. Tengo configurado un servicio que mueve todos los gestores de interrupción USB a los núcleos grandes. Puede descargarlo e instalarlo con estos comandos:

```
$ sudo wget -O /etc/systemd/system/affinity.service \
https://raw.githubusercontent.com/mad-ady/\
odroid-ODROID-XU4-optimizations/master/affinity.ser-
vice
$ sudo systemctl enable affinity
$ sudo systemctl start affinity
```

Puede comprobar que funciona mirando dentro de /proc/interrupts y viendo los valores en la 6ª columna (correspondiente a la 4ª CPU) para el bus USB. Además, iperf reportará velocidades de descarga más rápidas. Hay otras posibles mejoras que puedes realizar, pero en mi caso no observé ninguna diferencia en lo que respecta al rendimiento. Puedes analizarlas y comentarlas en http://bit.ly/2k6JOMF.

Transferencia con sshfs/scp/sftp

Dado que SSH es un protocolo muy flexible y soporta conexiones por túnel y la transferencia de archivos, sería muy acertado usarlo a máxima velocidad. Si intentas realizar una transferencia segura (scp) en un XU4 con el proceso sshd asociado a los núcleos pequeños, obtendrás una velocidad de 15 MB/s. Si vinculas el proceso sshd a los grandes núcleos conseguirás 40MB/s. Si eres atrevido y no te importa sacrificar algo de seguridad, puedes llegar hasta los 50 MB/s reduciendo el algoritmo de cifrado utilizado. Yo lo hice con una conexión sshd diferente (en el puerto 2222) con diferentes configuraciones:

```
$ sudo wget -O /etc/systemd/system/ssh-big.service \
https://raw.githubusercontent.com/mad-ady/\
odroid-xu4-optimizations/master/ssh-big.service
$ sudo wget -O /etc/ssh/sshd_config_big \
https://raw.githubusercontent.com/mad-ady/\
odroid-xu4-optimizations/master/sshd_config_big
$ sudo systemctl enable ssh-big
$ sudo systemctl start ssh-big
```

Para montar o transferir un archivo usando este nuevo servicio ssh necesitará especificar concretamente el cifrado (ya que está desactivado por defecto porque se considera débil). Puedes hacerlo con una entrada en ~/.ssh/config sobre el cliente:

```
Host odroid-big
Hostname odroid-ip
Port 2222
Ciphers arcfour
Compression no
```

Para transferir archivos puede utilizar el siguiente comando:

```
$ scp bigfile odroid-big:/remote/path
```

Ajustar los intervalos systemd

Puede ser muy irritante tener que esperar a que systemd finalice algo que nunca terminará. Puedes modificar los tiempos de espera de systemd modificando la configuración global en /etc/systemd/system.conf:

```
DefaultTimeoutStartSec=20s
DefaultTimeoutStopSec=10s
```

Algunos servicios (como networking) configuran intervalos muy concretos y necesitarás cambiarlos igualmente:

```
$ sudo vi /etc/systemd/system/network-online.target.
wants/\
networking.service
TimeoutStartSec=30sec
```

Rendimiento

Aquí tienes algunas estadísticas de rendimiento usando anteriores ajustes y una red Gigabit. El cliente es un C2 que ejecuta Ubuntu 16.04, mientras que el servidor es el XU4. Las instrucciones de descarga y subida son relativas al ODROID-XU4. El disco conectado al XU4 tiene una velocidad de escritura de 110MB/s. El sistema de transferencias de archivos transferió un archivo de 8GB lleno de ceros (dd if =/dev/cero of=zero bs=1M count=8000 conv=fsync).

Test	Command	Download	Upload
iperf	**ODROID-XU4$** iperf -s -i 10 -N **c2$** iperf -c 192.168.1.9 -t 30 -i 10 -r	925Mbps (115MB/s)	820Mbps (102MB/s)
samba	**c2$** sudo mount -t cifs -o username=odroid,password=odroid //odroid-ip/Seagate-XFS /media/share **c2$** cp /media/share/zero /dev/null **c2$** cp zero /media/share/zero.1	79MB/s (98MB/s with a PC client)	46MB/s
nfs	**c2$** sudo mount -t nfs -o rw odroid-ip:/nfs/Seagate-XFS **c2$** cp /media/share/zero /dev/null **c2$** cp zero /media/share/zero.1	102MB/s (111MB/s with a PC client)	56MB/s
scp	**c2$** scp odroid-big:/media/seagate/xfs/zero /dev/null **c2$** scp zero odroid-big:/media/seagate/xfs/zero.1	38MB/s (52MB/s with a PC client)	31MB/s (52MB/s with a PC client)

Ten en cuenta que parte del rendimiento depende de tu cliente. Logre mejor rendimiento con un PC Linux que con el ODROID-C2 como cliente.

Tiene total libertad para comentar posibles optimizaciones en el hilo de soporte en http://bit.ly/2j3Ddke.

EL ASOMBROSO ODROID-VU8

TABLET ODROID PORTATIL TODO EN UNO CON PANTALLA TACTIL Y UNA RESOLUCION DE 1024 X 768

editado por Rob Roy (@robroy)

Figura 1 - El nuevo ODROID-VU8 ofrece una imagen clara y nítida

El ODROID-VU8 es una nueva pantalla multi-táctil de 8 pulgadas para el ODROID-C2 y C1+ que está disponible en http://bit.ly/2k8bML5 por 90$. Ofrece a los usuarios la posibilidad de crear proyectos todo-en-uno como Tablet, consolas de juegos, sistemas de información de ocio y sistemas embebidos. Simplemente enchufa un adaptador de corriente de 5V/4A a la LCD y enciende el C2 o C1 +. Esta pantalla táctil LCD de amplio ángulo de visión y alta calidad está específicamente diseñada para trabajar tanto con Android como con Linux. Puede encontrar información más detallada en http://bit.ly/2iY960M.

Especificaciones

- TFT-LCD de 8 pulgadas
- Resolución de pantalla: 1024x768 píxeles (relación 4: 3)
- Entrada táctil capacitiva de 10 dedos (ID USB 18D1: 4E12)
- Control del brillo de la luz de fondo con PWN GPIO ODROID
- Consumo de energía máximo: 1.1A / 5Volt (sólo LCD y controlador de la pantalla)
- Ángulo de visión: Izq. 75, Der. 75, Arriba 75, Abajo 75 grados
- Dimensiones de la pantalla: 189 x 149 x 29 mm
- Tamaño de pantalla: 162 x121,5 mm (área activa)

Componentes de hardware

A. TFT LCD 8 inch ensamblada + Pantalla multi-táctil de 10 puntos
B. Carcasa trasera de plástico
C. Placa convertidor
D. Placa de doble conexión HDMI
E. 8 tornillos de 3,5 mm
F. Cable puente para 2 puertos
G. Cable USB Micro-a-Micro (aproximadamente 8cm)
H. Cable USB Micro-a-Tipo A (aproximadamente 20cm)
I. Fuente de Alimentación 5V/4A

Figura 2 - Kit ODROID-VU8

Figura 3 – 2 Puertos USB host y Puerto Ethernet

Figura 4 - Rejillas de ventilación y línea de corte para el puerto GPIO de 40 pines

Figura 5 - Entrada PSU 5V/4A

Ansamblaje

Ten mucho cuidado a la hora de montar el ODROID-VU8.

Figura 6 - Conecta la placa de doble conexión HDMI a la placa convertidor. Es más fácil montar la placa de doble conexión HDMI con este ángulo

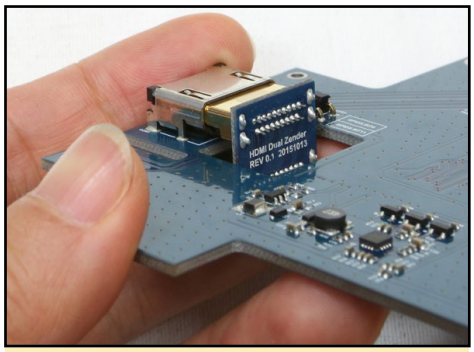


Figura 7 – Empuja hasta que encaje y revisa el conector

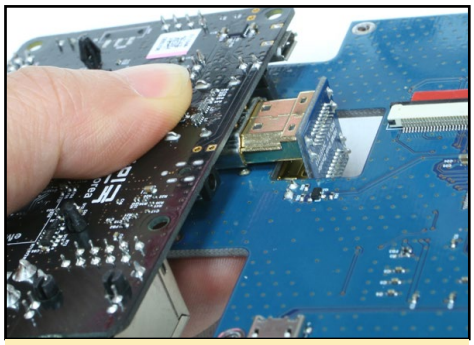


Figura 8 - Conecta el ODROID-C2/CI+ al otro conector de la placa de doble conexión HDMI

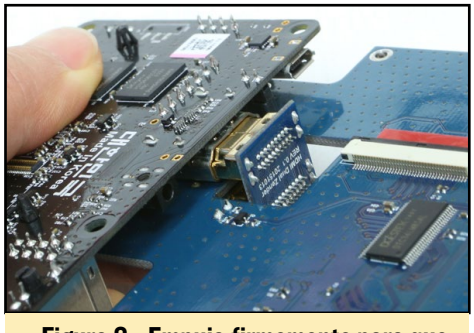


Figura 9 - Empuja firmemente para que la conexión HDMI sea estable

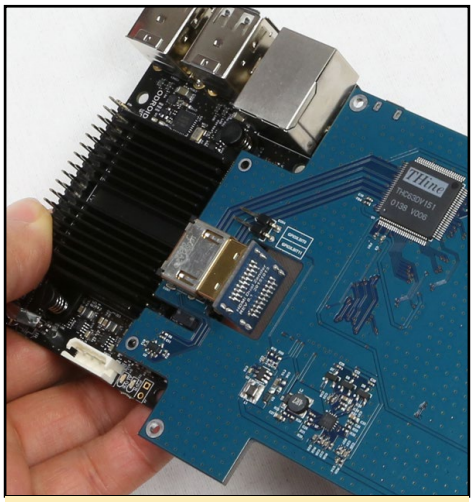


Figura 10 - La señal HDMI está conectada

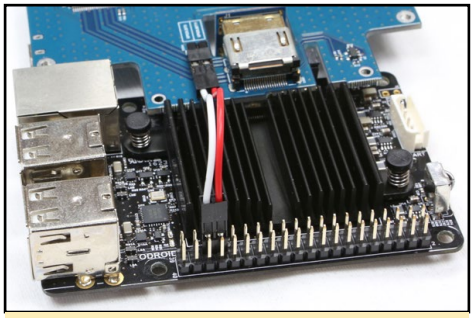


Figura 11 - Conecta un extremo del cable puente al pin GPIO #33 y #35, y el otro extremo a la placa convertidor

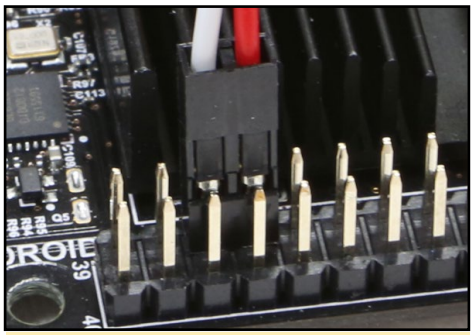


Figura 12 - Comprueba la disposición de los pines en relación al cable puente

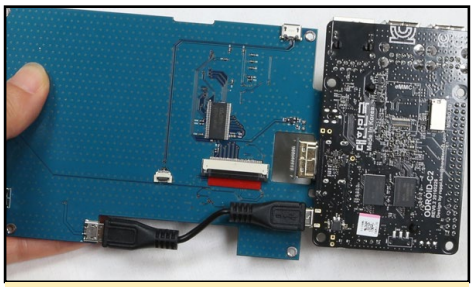


Figura 13 – Conecta el cable USB micro a micro, que proporciona al C2 o CI+ la energía a través de la placa convertidor

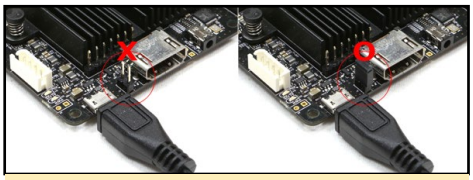


Figura 14 - Los dos pines puente deben estar conectados para dar entrada a la energía a través del conector microUSB

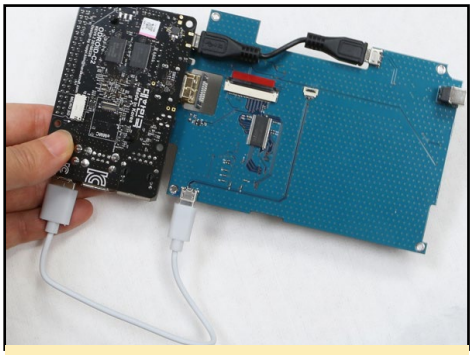


Figura 15 - Conecta al cable microUSB para la señal táctil entre el C2/CI + y el convertidor

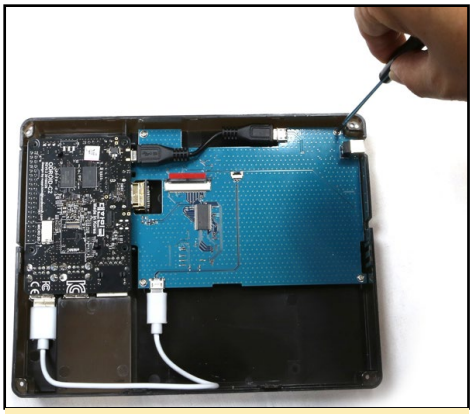


Figura 16 - Ubica la placa ensamblada en la carcasa y coloca los tornillos

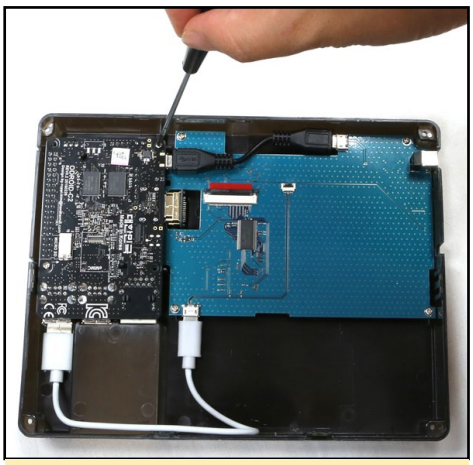


Figura 17 - Coloca los tornillos en los orificios de la placa C2/CI+

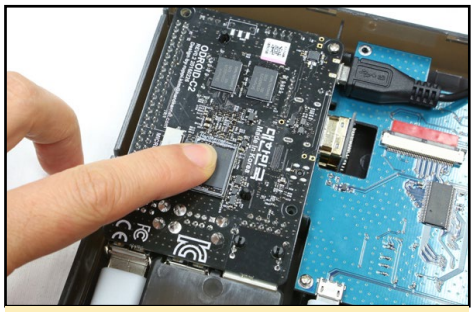


Figura 18 - Instala el Módulo eMMC / tarjeta MicroSD. No es posible cambiar el módulo eMMC o la tarjeta microSD tras completar el montaje.

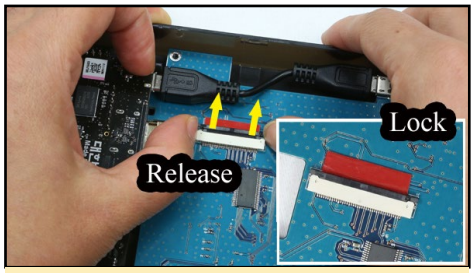


Figura 19 – Desliza hacia afuera la ranura del conector de 40 pines

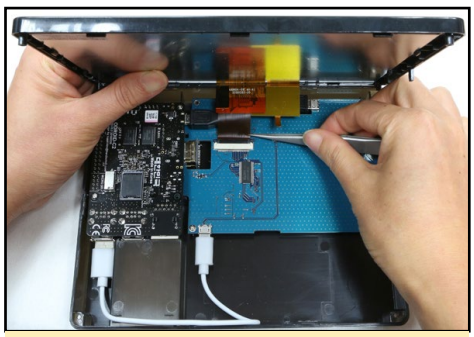


Figura 20 - Inserta el FPCB de 40 pines en la ranura del conector. La señal LCD va a la placa convertidor a través de este FPCB

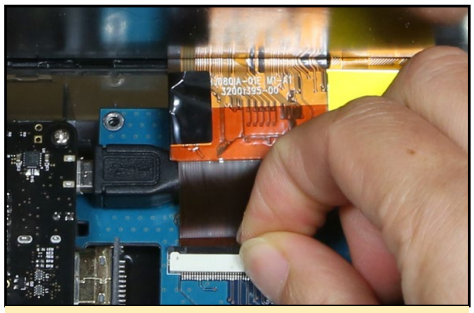


Figura 21 - Bloquea la ranura

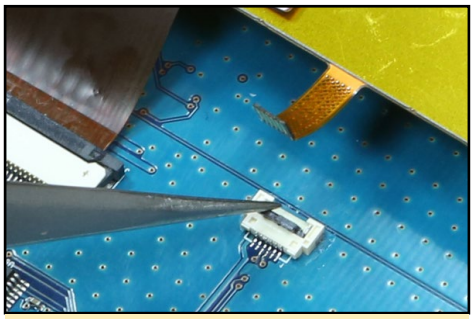


Figura 22 - Abre el conector de 6 pines. La puerta debe abrirse 90 grados

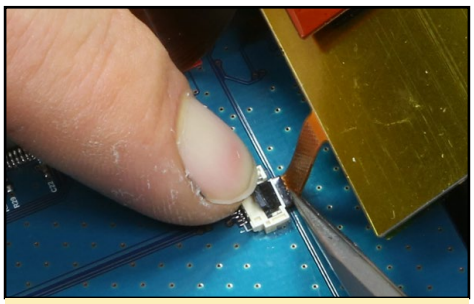


Figura 23 - Inserta el FPCB de 6 pines en el conector con cuidado con ayuda de unas pinzas. La parte oscura debe insertarse correctamente. La señal de la pantalla táctil va a la placa convertidor a través de este FPCB de 6pin. Este es uno de los pasos más difíciles. Si el FPCB de 6 pines no se inserta correctamente, se soltará y la señal táctil se desconectará

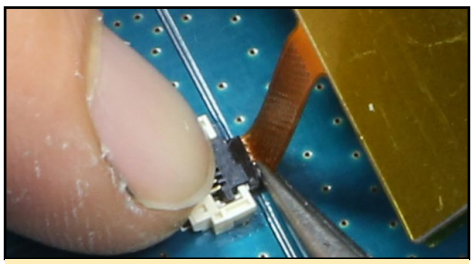


Figura 24 - Cierre la puerta

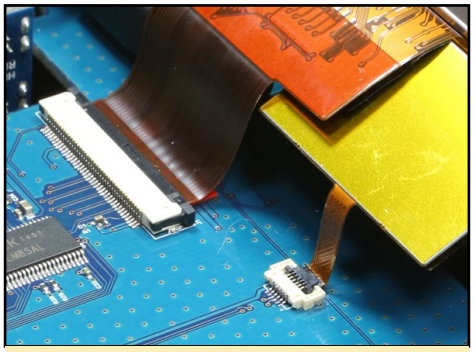


Figura 25 - Comprueba las conexiones FPCB de 40 pines y 6 pines

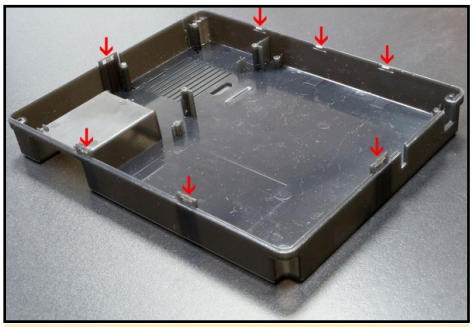


Figura 26 - Comprueba los ganchos y los orificios de bloqueo en la carcasa trasera antes de colocar la parte frontal

Figura 27 - Cierra la carcasa. Asegúrate de no dañar el sistema y los frágiles FPCBs que hay dentro de la carcasa

Figura 28 - Presiona los lados de la caja posterior a la vez y razonablemente fuerte para enganchar las piezas de bloqueo

Figura 29 - Inserta y aprieta los tornillos en los cuatros agujeros de las esquinas en los laterales

Software

En Android, selecciona “ODROID-VU8” en ODROID Utility mientras está conectado a un monitor HDMI.

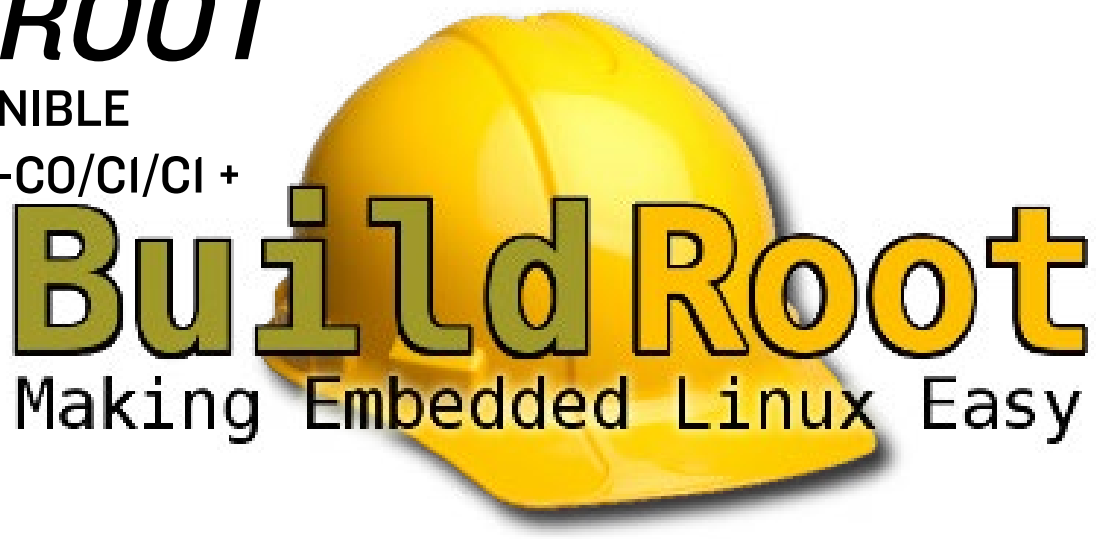
En Linux, abre boot.ini con un editor de texto y descomenta estas líneas:

```
setenv m “1024x768p60hz”
setenv vout “dvi” # C2
setenv vout_mode “dvi” # C1/C0
```

BUILDROOT

AHORA DISPONIBLE PARA ODROID-C0/C1/C1 +

por @Brian.K



Buildroot es una herramienta muy útil que simplifica y hacer más eficiente el proceso de compilación de Linux. La herramienta no sólo es fácil de usar, sino que también cuenta con una gran variedad de opciones avanzadas como la compilación cruzada y el desarrollo de software adicional con paquetes. Buildroot se maneja en gran medida a través de menús, lo cual permite rápidamente configurar y compilar cosas en tan sólo 15 minutos. Esta guía te enseñara a través de unos sencillos pasos a utilizar Buildroot en el ODROID-C1 + y el ODROID-C0.

Antes de poder utilizar las imágenes generadas por Buildroot en tu ODROID-C1 + y ODROID-C0, tiene que preparar la tarjeta SD o eMMC. Primero, descarga el código fuente de buildroot para ODROID-C1 + / C0 desde GitHub:

```
$ git clone --depth 1\
 https://github.com/hardkernel/
buildroot.git
```

Además, vamos a necesitar un compilador cruzado para que el gestor de arranque y el kernel compilados puedan ejecutarse en el ODROID:

```
$ wget\ http://releases.linaro.
org/archive/14.09/components/\
toolchain/binaries/gcc-linaro-
arm-none-eabi-4.9-2014.09_linux.
tar.xz
$ sudo tar\
 Jxvf gcc-linaro-arm-none-ea-
bi-4.9-2014.09_linux.tar.xz \
 -C /opt/toolchains/
$ export ARCH=arm
$ export CROSS_COMPILE=arm-none-
eabi-
$ export PATH=/opt/tool-
chains/gcc-linaro-arm-none-ea-
bi-4.9-2014.09_linux/bin/:$PATH
```

A continuación, crea el ejecutable BuildRoot:

```
$ cd buildroot
$ make odroidc1_defconfig
```

Tras una compilación satisfactoria, puedes editar las opciones de compilación mediante el siguiente comando:

```
$ make menuconfig
```

Finalmente, vamos a compilar todos los archivos y crear la imagen rootfs. Ten en cuenta que necesitarás tener acceso a Internet, ya que Buildroot descargará cualquier código fuente de los paquetes adicionales incluidos:

```
$ make
```

Tras la compilación, deberías tener el siguiente árbol de archivos:

```
output/images/
+-- Image
+-- boot.ini [1]
+-- boot.vfat
+-- meson8b_odroidc.dtb
+-- rootfs.ext2
+-- rootfs.ext4
+-- rootfs.tar
+-- sdcard.img
+-- bl1.bin.hardkernel
+-- sd_fusing.sh
`-- u-boot.bin
```

El archivo boot.ini es el archivo de configuración utilizado para el uboot de ODROID-C1+/C0. Una vez finalizado el proceso de compilación, tendrás una imagen llamada “sdcard.img” en el directorio output/images/. Graba el “sdcard.img” de arranque en una tarjeta SD o eMMC usando “dd” en Linux/OSX o Win32 Disk Imager en Windows:

```
$ sudo dd if=output/images/sd-
card.img of=/dev/sdX conv=fsync
```

Una vez grabada la imagen, inserta la tarjeta SD o eMMC en tu ODROID y enciéndelo. Tu nuevo sistema debería estar listo para usarse. Para comentarios, preguntas y sugerencias, visita el post original en http://bit.ly/2jqGRsO.

SKY FORCE RELOADED

¿A QUIEN DE NOSOTROS NO LE GUSTA UN BUEN SHOOT'EM UP? ¡MALDICION BALAS!

por Bruno Doiche

Podemos empezar de inmediato. Y para los tiempos que corren, el equipo Infinite Dreams ha creado el que creo que es el más refinado de los shoot 'em creados hasta la fecha para la plataforma Android. ¡Sky Force Reloaded se acerca bastante a lo que podría ser una verdadera obra de arte! Si te gusta los juegos llenos de desafíos, éste tiene todas las papeletas para que lo pongas en página de inicio de tu Android. Has sido advertido

https://play.google.com/store/apps/details?id=pl.idreams.SkyForceReloaded2016

¡Este juego tiene unos gráficos fantásticos, y es un buen motivo para que compres un joystick USB si aún no tiene uno!

DOMOTICA CON HOMEBRIDGE

INTEGRA FACILMENTE TODAS LAS TECNOLOGIAS DE TU HOGAR

por @korinel

La domótica es una forma estupenda de añadir todo tipo de funcionalidades a tu hogar tanto en pequeñas como en grandes dimensiones. Ya sea para encender las luces, ajustar el termo de agua caliente o bloquear las puertas, Existe una gran variedad de gadgets que puedes comprar para controlar esta tecnología de forma automática o desde tu Smartphone

Sin embargo, muchos de estos gadgets necesitan sus propias apps o herramientas para poder ejecutarse, lo cual complica las cosas si tienes varios gadgets con diferentes API. Aquí es donde entra HomeBridge. Este ingenioso servidor NodeJS te puede ayudar a unificar varios gadgets de domótica a través de la API HomeKit de Apple, permitiendo que tu ODROID gestione la conexión entre tu iPhone y tus diversos sistemas de luces inteligentes, sistemas de bloqueo y otros dispositivos IoT, tiendo en cuenta que Apple normalmente no te permitiría conectar a un sistema HomeKit.

No he visto instrucciones específicas para hacer funcionar HomeBridge sobre un ODROID-C2, así que decidí investigar un poco, ya que es una gran plataforma de código abierto. Me llevó un tiempo, pero finalmente conseguí una instalación limpia de homebridge. La mayor parte de la información procede de http://bit.ly/2jZr2u3, pero es necesario realizar algunos pasos adicionales para que las cosas funcionen correctamente.

Al principio la dificultad a estaba en la instalación de los nodos, ya que la versión disponible en los repositorios arch64 (ya que nuestro ODROID-C2 es de 64 bits) estaba bastante obsoleta. Para esto, crea un directorio temporal en tu ODROID y luego, ejecuta los siguientes comandos para preparar tu sistema:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt-get install libavahi-compat-libdnssd-dev g++ git make
```

A continuación, compilaremos la versión más reciente de Node.js. Ten en cuenta que la versión de Node.js puede cambiar, así que comprueba la versión más reciente y modifica estas instrucciones según sea necesario:

```
$ sudo curl -sL https://deb.
nodesource.com/setup_6.x | sudo
-E bash -
apt-get source nodejs
$ dpkg-source -x nodejs_6.9.4-
1nodesource1~xenial1.dsc
$ cd nodejs-6.9.4
$ dpkg-buildpackage -us -uc
```

El paso anterior llevará bastante tiempo, así que vete a tomar unas cuantas tazas de té mientras esperas. Una vez completado, instala Node.js:

```
$ cd ..  # Here are your local
.deb packages
$ sudo mv *.deb /var/cache/apt/
archives
$ sudo apt-get install nodejs
```

pués, instalaremos HomeBridge y sus dependencias:

```
$ sudo npm install -g --unsafe-
perm homebridge hap-nodejs node-
gyp
$ cd /usr/lib/node_modules/home-
bridge/
$ sudo npm install --unsafe-perm
bignum
$ cd /usr/lib/node_modules/hap-
nodejs/node_modules/mdns
$ sudo node-gyp BUILDTYPE=Release
rebuild
```

Ahora configuraremos el servicio HomeBridge, con los respectivos archivos de apoyo y usuarios:

```
$ sudo adduser --system home-
bridge
$ sudo mkdir /var/lib/homebridge
$ sudo chown homebridge /var/lib/
homebridge
```

Para configurar el servicio se necesita algo más de trabajo. Descarga los dos archivos de este enlace: http://bit.ly/2dRZSSY. Edita homebridge.service de forma que la línea 11 se lea del siguiente modo:

```
ExecStart=/usr/bin/homebridge
$HOMEBRIDGE_OPTS
```

Coloca el archivo homebridge en /etc/default y homebridge.service en /etc/systemd/system Las instrucciones para crear el config.json y la instalación de módulos adicionales de homebridge la puedes encontrar en http://bit.ly/1QK07by.

A continuación, coloca tu config.json en /var/lib/homebridge y asegúrate de que los anteriores archivos sean legibles por el usuario "homebridge". Carga e inicia el servicio del sistema:

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable home-
bridge
$ sudo systemctl start homebridge
$ sudo systemctl status home-
bridge  # Just a status check
```

Ahora ya estás listo para empezar a crear plugins para tus diferentes dispositivos de domótica. Si alguien pudiera actualizar uno de los repositorios oficiales para incluir la versión aarch64 más reciente de NodeJS, simplificaría considerablemente el proceso. Si estás utilizando un ODROID de 32 bits, como el C1, este proceso debería ser más fácil. Para comentarios, preguntas y sugerencias, visita el post original en los foros de ODROID en http://bit.ly/2iBbaIZ.

TAP 'N' SLASH

SIMPLES RECORTES CON RAPIDOS REFLEJOS

por Bruno Doiche

Entre las mejores cosas que hay a la hora de jugar con tu ODROID es

el hecho de no tener que lidiar con juegos demasiado complejos. Hay momentos en los que eres lo suficientemente perezoso como para no querer ni siquiera abrir ese cajón para coger tu mando USB, aunque tote apetece jugar a un divertido juego. Entonces, ¿Qué podemos hacer? Disfrutar de Tap 'N' Slash, un rápido y divertido juego que requiere buscar un equilibro entre reaccionar rápido y querer reducir el sinfín de monstruos de las mazmorras. ¡No te vas a creer lo rápido que te vas a enganchar en este increíble juego!

https://play.google.com/store/apps/details?id=com.invictus.tapnslash

¡No te confíes demasiado e intenta encontrarte entre las espadas!

JUEGOS LINUX

COMPARANDO LA VELOCIDAD Y RENDIMIENTO DE PPSSPP

por Tobias Schaaf (@meveric)

En la actualidad, es difícil comparar el rendimiento de los juegos entre los diferentes dispositivos ODROID. Los juegos que hacen uso de ciertas prestaciones de las placas, como la recompilación dinámica, son muy pocos, especialmente cuando se trata del ODROID-C2. Ningún juego de PS1, N64 o NDS requiere recompilación dinámica para la optimización y se ejecutan recurriendo casi en su totalidad al hardware, y así funcionan bien en el ODROID-C2. Sin embargo, hay una excepción: PPSSPP, el emulador Playstation Portable. De hecho, implementa un recompilador dinámico ARM64 que también nos permite ejecutar juegos de PSP en el C2. Así que pensé que sería hora de hacer una nueva comparativa y ver cómo encaja el C2 en el mundo de la emulación. Ten en cuenta que esta comparativa está hecha bajo Linux y el rendimiento en Android es totalmente diferente a lo que puedes encontrar en este artículo.

Configuración de PPSSPP

PPSSPP tiene muchas y muy diversas de opciones para optimizar el rendimiento y la apariencia de los juegos. Yo usare las opciones que vienen por defecto para comparar los diferentes dispositivos y desde ahí partiré para ver que opciones puedo cambiar para mejorar el rendimiento. La configuración por defecto es la siguiente:

Frameskipping 3

Auto Frameskip: ON

Mipmapping: OFF

Lazy texture caching: ON

Spline/Bezier curves quality: Medium

Texture coord speedhacks (speedup): ON

Lower resolution for effects (reduces artifacts): Balanced

Basandome en mi experiencia, puedo decir que esta configuración combianda con una resolución 2xPSP, normalmente funciona con la mayoría de los juegos. Al principio, quería comparar las principales placas de ODROID (C1, C2, U2/U3, XU3/XU4), pero cuando empecé con el C1, pronto descubrí que, debido al problema alfa de esta placa, la mayoría de los juegos no llegaban ni siquiera a iniciarse y aquellos que lo hacían tenían problemas gráficos. Pasé de 32bpp a 24bpp, los 16bpp tampoco ayudaron. Los 16bpp provocaban un error al iniciar y el emulador no llegaba a arrancar, Los 24bpp lo mostraba todo distorsionado. Todavía no he probado los núcleos Libretro de PPSSPP en el C1, porque quise comparar el mismo emulador y los mismos ajustes en todas las placas. Al final, me di por vencido con el C1 y sólo comparé el resto de placas.

Primero comparé el rendimiento con diferentes resoluciones, para comprobar lo alto que podía poner los gráficos antes de que el juego dejara de responder adecuadamente. También verás a menudo el término FXAA en mis resultados, se trata de un shader post-procesamiento que me gusta usar en el emulador PPSSPP. He capturado algunas imágenes en alta definición de diferentes resoluciones con y sin FXAA para que puedas compararlas. Estas imágenes fueron tomadas con un ODROID-XU3 y muestran que juegos son lo que se ven con mayor definición y con shader FXAA. Puede encontrar las imágenes en http: // bit. Ly / 2jZjlDQ. La resolución estándar de una PSP es 480 × 272, de modo que una resolución 2xPSP sería 960x544 y así sucesivamente.

Tekken 6

Por alguna razón, a la gente le gustar bastante este juego, aunque realmente no entiendo por qué. Me parece muy aburrido y está mal programado. Mientras que otros juegos de lucha funcionan a un 60 FPS estables, Tekken 6 se queda en los 10-15 FPS con retardos y saltos de imagen. Aún así, es muy popular.

El juego está programado de una forma que lo hace muy difícil emular. Su tasa de fotogramas siempre es muy baja y en mi opinión, los gráficos y los efectos no justifican esa alta demanda de rendimiento necesaria para ejecutarse correctamente. Vamos a ver cómo se comporta este juego en las diferentes plataformas.

C2

Por lo general, el C2 no fue capaz de ejecutar correctamente este juego. Los videos eran un poco lentos y la jugabilidad era incluso peor.

Video

2xPSP: 20-30 FPS, en su mayoría entre 20-25, algo lento

2xPSP + FXAA: 17-23 FPS, en su mayor parte en torno a los 20 FPS

Figura 1 - Tekken 6

3xPSP: 18-25 FPS, principalmente entre los 20-23 FPS, algo lento

3xPSP + FXAA: 13-15 FPS, no tan lento como cabría de esperar

En el juego

2xPSP: 1-14 FPS, lento pero jugable

2xPSP+FXAA: 09-11FPS, muy lento

3xPSP: 09-11 FPS, muy lento

3xPSP+FXAA: 08-09FPS, muy lento

De nuevo, Tekken 6 demuestra ser un juego mal programado, que es demasiado severo con el sistema.

Sin embargo, me sorprendió que el rendimiento de C2 fuese tan malo en comparación con el rendimiento del U3, que es 3-4 años más antiguo.

U2/U3

En el U3, olvidé de cambiar la re-solución de los videos y sólo probé la resolución 2xPSP en la intro. El ODROID-U3 manejaba este juego sorprendentemente bien, lo cual me sorprendió.

Video

2xPSP: 50-60 FPS, en su mayoría 60

En el juego

2xPSP: 25-30 FPS, en su mayoría 30

2xPSP + FXAA: 20-30 FPS, dependiendo del campo de batalla

3xPSP: 25-30 FPS, no tan estable a 30 FPS

3xPSP + FXAA: 15-20 FPS, en su mayoría 18 FPS, que es sorprendentemente alto.

Estos resultados fueron bastante mejores de esperado. Teniendo en cuenta que el hardware de ODROID-U3 es del 2012 con sólo una GPU Mali 400MP2, el juego era bastante estable y rápido, y la jugabilidad era realmente fluida.

XU3/XU4

Me sorprendieron un poco los resultados del XU3. Si sólo me fijo en los números, el XU3 parece incluso peor que la U3, aunque la experiencia de juego fue bastante agradable. En este juego probe a desactivar el llamado frameskipping debido al elevado rendimiento del XU3/XU4. Esto ayudó bastante con las películas que se reproducian sorprendentemente bien, pero en el propio juego, el frameskipping se hacía necesario de todos modos.

Video

2xPSP: velocidad máxima sin frameskipping, de otro modo 25-60 FPS

2xPSP+FXAA: velocidad máxima sin frameskipping, de otro modo 25-60 FPS

3xPSP: velocidad máxima sin frameskipping, de otro modo 25-60 FPS

3xPSP + FXAA: 25-60 FPS con frameskipping

En el juego

2xPSP: 15 FPS, permaneciendo constante a 15 FPS sin retardos

2xPSP + FXAA: 16-22 FPS, ligeramente borroso pero sin retardos

3xPSP: 15-20 FPS, todavía sin retardos, excepto en los primeros planos de los personajes al inicio y al final de los combates

3xPSP + FXAA: 12-15FPS, sorprendentemente fluido con ligeros retardos

Me sorprendió que incluso con 3xPSP + FXAA funcionase tan bien. El juego va fluido, aunque de cara a la velocidad, probablemente sea mejor quedarse en 2xPSP. En general, Tekken 6 ha demostrado ser un juego difícil de ejecutar en cualquier sistema. Sin embargo, tanto el U2/U3 como el XU3/XU4 son capaces de manejarlo bastante bien, y no debería ser difícil encontrar la configuración perfecta para el juego.

Hexyz Force

Me gusta bastante este juego, no sólo porque me gustan los juegos RPG en general, sino porque también es un juego impresionante en lo que respecta a los requisitos. El juego se ejecuta sólo a 30 FPS en lugar de 60 FPS como ocurren en otros juegos, lo que significa que normalmente se puedes asignar más recursos a los gráficos. Los videos y andar por las zonas están tan bien implementados que se puede aumentar la calidad sin llegar a percibir problemas de rendimiento. Sólo en las peleas puede experimentar algo de receso, pero como éstas son por poco tiempo, es probable que quieras asumir algo de lentitud durante las peleas a cambio de una tener una mejor calidad gráfica en general. Las imágenes de http://bit.ly/2jZjlDQ que muestran lo que puede llegar hacer el FXAA y las resoluciones más altas, fueron tomadas de este juego. Se puede ver muy bien cómo los gráficos de juego mejoran con respecto a los originales de la PSP.

C2

Video

xPSP: 25-30 FPS, en su mayoría a velocidad máxima

2xPSP + FXAA: 18-30 FPS, en la mayoría de los casos alrededor de los 23-25 FPS, un poco lento

3xPSP: 23-30 FPS, principalmente en lo más alto 20s o incluso a velocidad máxima

3xPSP + FXAA: 14-20 FPS, en su mayoría 16-17 FPS, un poco lento

Figura 2 - Hexyz Force

Caminando por las zonas

2xPSP: 28-30 FPS, en su mayoría 30

2xPSP + FXAA: 24-30 FPS, en su mayoría 30 FPS

3xPSP: 27-30 FPS, en su mayoría 30

3xPSP + FXAA: 16-18 FPS, la imagen da algunos saltos mientras caminas

Combates

2xPSP: 15-30 FPS, principalmente en lo más alto 20s

2xPSP + FXAA: 13-23 FPS, un poco lento, cercano a los 20

3xPSP: 10-30 FPS, a menudo en lo

más bajo 20s o inferior

3xPSP + FXAA: 8-17 FPS, en su mayoria por debajo de 15

Me sorprendió ver unos FPS tan bajo en el C2. Este juego se ejecuta muy bien en otros dispositivos, el C2 tiene potencial GPU como para manejar fácilmente los efectos. No obstante, con la configuración 2xPSP puedes ejecutarlo a máxima velocidad, así que tengo esperanza.

U2/U3

Olvidé cambiar de gráficos para los videos, aunque en 2xPSP se ejecutó a maxima velocidad y espero resultados similares en 2xPSP + FXAA y 3xPSP.

Caminando por las zonas

2xPSP: 30 FPS

2xPSP+FXAA: 30 FPS, algo borroso

3xPSP: 30 FPS

3xPSP + FXAA: 23-24 FPS, se ve bien, pero imagen tienes saltos

4xPSP: 30 FPS

4xPSP + FXAA: 15 FPS, imagen con muchos saltos

Combates

2xPSP: 25-30 FPS, a veces incluso puede bajar hasta los 20 FPS

2xPSP+FXAA: 25-30FPS, borroso

3xPSP: 25-30 FPS, puede bajar a los 15 FPS cuando hay mucho movimiento

3xPSP + FXAA: 15-18 FPS, muy lento con esta configuración

El juego se ejecuta bastante bien en el U2/U3. Incluso fui capaz de probar las resoluciones 4xPSP mientras caminaba, lo cual se veía genial. La configuración 2xPSP + FXAA funcionaba muy bien, pero la "baja resolución" hacia que la experiencia en su conjunto se volviera un poco borrosa. Descubrí que ajustando el frameskipping a 1x en lugar de 3x reducía los saltos de imagen. En general, cualquier juego a 30 FPS sólo se debería ejecutar con 1x Frameskipping, o sino se experimenta saltos de imagen. El juego posiblemente funcione mejor con 3xPSP en el U2/U3, al menos en los combates. Puede que quieras probar 2xPSP + FXAA para ver el resultado. No obstante, este juego en general ofrece una muy buena experiencia sobre el U2/U3.

XU3/XU4

Fije el Frameskipping en 1 para hacer las pruebas en el XU3/XU4.

Video

2xPSP: 26-30 FPS, en su mayoría 30 FPS, parece un poco lento

2xPSP + FXAA: igual que 2xPSP

3xPSP: igual que 2xPSP

3xPSP + FXAA: igual que 2xPSP

4xPSP: igual que 2xPSP

4xPSP + FXAA: 28-30 FPS, mejor aún que 2xPSP, lo cual es un poco raro

Me di cuenta que en el XU3/XU4, los videos que se reproducen en ffmpeg tienen un poco de retardo. Esto no significa que el hardware no pueda reproducirlo, sino que algo no funciona bien. Por ejemplo, ejecutar una película 1080p H264 en ffplay sobre el U3 funciona perfectamente. El hardware se acerca a su límite, pero en la mayoría de los casos, puede reproducir la película perfectamente. La misma película en el XU3/XU4, que tiene el doble de rendimiento que el U3, muestra salto continuamente, a pesar de que el uso de los núcleos CPU están sólo al 30-50%. Puedo ver efectos similares en los juegos. Desactivar el frameskipping puede mejorar la calidad del vídeo, aunque no es necesario.

Simplemente para que quede constancia, a partir de 5xPSP + FXAA, empiezan a caer los frames y se vuelve lento. Sin FXAA, incluso la resolución 9xPSP se ejecuta a 30 FPS. 10xPSP se ejecuta en su mayoría a 30 FPS pero podría bajar a los 22-24 FPS, volviéndose lento. Es increíble, si te paras a pensarlo un momento. 9xPSP es un renderizado interno de 4320x244, y funciona muy bien en el XU3/XU4. ¡Es un juego muy bien optimizado!

Caminando por las zonas

2xPSP: velocidad máxima

2xPSP + FXAA: velocidad máxima

3xPSP: velocidad máxima

3xPSP + FXAA: velocidad máxima

4xPSP: velocidad máxima

4xPSP + FXAA: velocidad máxima

En los videos ocurre algo similar, el rendimiento cae a 5xPSP + FXAA. Sin FXAA incluso a 9xPSP es reproducible. La tasa de fotogramas suele caer cuando se cargan nuevas texturas en memoria. Una vez que los datos están en memoria, no aparecen retardos.

Combates

2xPSP: 30 FPS, pero puede caer a 15 FPS (frameskip) aunque se mantiene a máxima velocidad (sin retardos)

2xPSP + FXAA: 30 FPS, puede caer a 18-19 como mínimo, sin retardos

3xPSP: igual que 2xPSP

3xPSP+FXAA: igual a 2xPSP+ FXAA

4xPSP: igual que 2xPSP

4xPSP+FXAA: igual que 2xPSP, puede caer a 15 FPS, no se aprecian retardos

Aquí hay mucho margen para experimentar. He mejorado la calidad de las curvas Spline/Bezier, desactivando el caché de textura vagas y deshabilitando la baja resolución de los efectos. Este tipo de configuración hizo que el juego pareciera más espectacular, aun cuando continuaba ejecutandose a máxima velocidad con el frameskipping fijado en 1. Me gusta bastante este juego por su rendimiento y jugabilidad, nos muestra claramente lo que podemos llegar a conseguir con la optimización de los juegos. El juego se ve y se ejecuta como si estuviera hecho para el ODROID XU3/XU4. Sugiero usar 3xPSP + FXAA y a partir de aquí realizar cambios. Este juego también muestra muy bien lo que puede llegar a hacer FXAA, tal y omo se ve en las capturas de pantalla a las que hemos hecho referencia anteriormente.

Asphalt Urban GT 2

Para este juego, cambié la configuración por defecto

Desactivé Frameskipping

Active "Disable slower effects"

Desactivé "Texture coord speedhack (speedup)", que sorprendentemente hace más lento este juego.

Este es un juego muy interesante desde el punto de vista de la emulación. Lo he utilizado para realizar pruebas de rendimiento desde que empecé a utilizar PPSSPP. Es un poco complicado de emular y los resultados varían bastante entre las diferentes versiones, desde no poder ejecutarlo a funcionar a pleno rendimiento. Todavía presenta algunos problemas con los efectos, pero eso lo hace perfecto para tenerlo como punto de referencia y ver la evolución del proyecto PPSSPP.

El juego tiene una tasa de fotogramas muy variable. No se mantiene en 30 o 60 FPS, sino que va saltado continuamente. Incluso en el menú, se puede apreciar que los FPS van desde los 30 a los más de 50 FPS, lo que significa que la velocidad del juego va variando. En el propio juego ocurre los mismo, pero generalmente a una escala más pequeña entre 30-34 FPS.

El juego reacciona muy mal al frameskipping. De hecho, fue el primer juego en el que tuve que desactivar el frameskipping para aumentar la velocidad. He descubierto que configurar el frameskipping demasiado alto provoca saltos en el juego, lo cual hace que parezca que va muy lento, aunque se supone que es posible ejecutarlo a máxima velocidad con frameskipping. Aún así, fijar Frameskipping en 1 puede mejorar la velocidad sin tener saltos de imagen.

C2

Menú

2xPSP: 30/45 FPS, varía, retraso en la música

3xPSP: 30/45 FPS, igual que 2xPSP

Seleccionar coche

2xPSP: 19-23 FPS, muylento

2xPSP+FXAA: 19-20 FPS,muy lento

3xPSP: 21-23 FPS, muy lento

3xPSP+FXAA: 14-15FPS, muy lento

Carreras

2xPSP: 18-21 FPS, muy lento

2xPSP+FXAA: 16-18FPS, muy lento

3xPSP: 15-18 FPS, muy lento

3xPSP+FXAA: 11-14FPS, muy lento

No he logrado una configuración en la que este juego se ejecutara sin retardos. Incluso a 2xPSP con Frameskipping fijado en 1 aparecen saltos de imagen, aún así es jugable a una velocidad del 85% la mayor parte del tiempo.

Figura 3 - Asphalt Urban GT 2

U2/U3

Menú

2xPSP: máxima velocidad

3xPSP: máxima velocidad

Seleccionar coche

2xPSP: 56 FPS, en su mayoría bien

2xPSP + FXAA: 49 FPS, desfase en la música

3xPSP: 56 FPS, en su mayoría bien

3xPSP + FXAA: 28 FPS, música muy retardada

Carreras

2xPSP: Máxima velocidad en las carreras normales la mayor parte del tiempo, 1x Nitro causa una bajada de 1-2 FPS, 2x y 3x Nitro en torno a los 19/33 FPS, imagen speeder 10 FPS

2xPSP + FXAA: Carrera normal 23-25 FPS, lento, 2x y 3x Nitro en torno a los 17 FPS, imagen speeder 10 FPS

3xPSP: Carrera normal 17-19 FPS, muy lengo, 2x y 3x Nitro en torno a los 11 FPS

3xPSP + FXAA: Carreras normales 13-15 FPS extremadamente lento, 2x y 3x Nitro en torno a los 8-10 FPS

Fijando en 1 el Frameskipping mejora la velocidad, en 2xPSP + FXAA y 3xPSP se puede jugar con esta configuración. Cualquier valor mayor que 1 hará que el juego se vuelva lento, ya que demasiado skipping hará que la imagen tenga saltos. Lo mejor es mantener el juego en 2xPSP. Con Frameskipping ajustado en 1, el juego es totalmente jugable y muy divertido en el U2 / U3.

XU3/XU4

Menú

2xPSP: máxima velocidad

2xPSP + FXAA: máxima velocidad

3xPSP: máxima velocidad

3xPSP+FXAA: 45-49 FPS, algo lento

4xPSP: máxima velocidad

Seleccionar coche

2xPSP: velocidad al 50%, aunque con frameskipping en 1 alcanza máxima velocidad

2xPSP + FXAA: velocidad al 50%, aunque con frameskipping en 1 alcanza máxima velocidad

3xPSP: velocidad al 50%, aunque con frameskipping en 1 alcanza máxima velocidad

3xPSP + FXAA: velocidad al 50%, con frameskipping en 1 ligeramente por debajo de la máxima velocidad, se percibe un poco de retardo.

4xPSP: velocidad al 50%, aunque con frameskipping en 1 alcanza máxima velocidad

4xPSP + FXAA: velocidad del 50%, frameskipping en 1 no ayuda, la velocidad alcanza solo el 67%, lo cual quiere decir que es más bien lento, frameskipping 2 alcanza máxima velocidad pero la imagen salta un poco.

Carreras

2xPSP: Máxima velocidad con frameskipping 1

2xPSP + FXAA: Máxima velocidad con frameskipping 1

3xPSP: Máxima velocidad con frameskipping 1

3xPSP + FXAA: Máxima velocidad con frameskipping 1

4xPSP: ligeramente por debajo de la velocidad máxima, 95%

4xPSP + FXAA: un poco por debajo de la velocidad máxima, 80-95%

Me decepciono un poco el hecho de tener que usar frameskipping 1 incluso en 2xPSP, pero activándolo permite incluso que el juego vaya fino en 3xPSP + FXAA, de hecho, se ve bastante bien. Parece ser que la GPU está bastante bien utilizada. Los efectos especiales, como cuando se conduce con 2xNitro, no repercute para nada en el rendimiento del sistema, mientras que en el U3 y C2, esto provocó una considerable bajada de la tasa de fotogramas. Otros efectos como la "imagen speeder" sólo causa una ligera caída de la tasa de fotogramas, mientras que en otros dispositivos aparece una enorme ralentización durante unos segundos.

Final Fantasy VII - Crisis Core

Este juego tiene muchas escenas, como son las escenas cortas de video, escenas en las que se camina o habla con la gente, escenas de combate y peleas contra enemigos. En consecuencia, el juego puede comportarse de diferente modo en cada situación, y encontrar una con-

Figura 4 - Crisis Core

figuración que satisfaga por igual a todas las escenas puede ser algo un tanto complicado. Así que, durante mis pruebas, me dedique a buscar una buena experiencia en conjunto, sin centrarme en un determinado aspecto del juego.

C2

Videos

2xPSP: 16-30 FPS, en su mayoría más de 20

2xPSP+FXAA: 13-19 FPS, algo lento

3xPSP: 15-30 FPS, en la mayoría de los casos más de 20

3xPSP + FXAA: 13-17 FPS, mayoritariamente en torno a 15

En el Juego/Combates

2xPSP: 13-30 FPS, principalmente por debajo de 20 FPS, lento

2xPSP + FXAA: 13-18 FPS, en su mayoria 15 FPS, lento

3xPSP: 13-22FPS, sobre todo 15 FPS

3xPSP + FXAA: 8-16 FPS, mayoritariamente 13 FPS o menos, lento

Este juego por desgracia no ofrece una experiencia muy buena sobre el C2, aunque posiblemente sea jugable en 2xPSP. Una vez más, el C2 muestra resultados bastante malos a la hora de ejecutar juegos de PSP bajo Linux.

U2/U3

Videos

2xPSP: máxima velocidad

2xPSP + FXAA: máxima velocidad

3xPSP: máxima velocidad

3xPSP + FXAA: 25-30 FPS

Los resultados son los típicos del U2/U3, la reproducción de vídeo a través de ffmpeg en PPSSPP es especialmente buena. Posiblemente, las escenas de video se vean mejor en el U2/U3.

En el juego/Combates

2xPSP: máxima velocidad

2xPSP + FXAA: En la mayoría de los casos a máxima velocidad, algunas escenas pueden bajar hasta 25 FPS

3xPSP: en su mayor parte a máxima velocidad, algunas escenas caen a 25 FPS

3xPSP + FXAA: 17-27 FPS con saltos de imagen frecuentes.

Puesto que este es también un juego de 30 FPS, ejecutarlo con Frameskipping fijado en 1 probablemente sea lo mejor. Ejecutar el juego en 2xPSP + FXAA presenta buen aspecto, y tiene muy buen rendimiento. A diferencia de Hexyz Force, Final Fantasy VII - Crisis Core no se vuelve borroso con la configuración 2xPSP + FXAA, ésta es una buena forma de mejorar los gráficos del juego. Además, parece funcionar de un modo muy estable en el U2/U3.

XU3/XU4

Vídeo

2xPSP: 15-30 FPS, en torno a los 30 FPS, aun así no se observa lentitud.

2xPSP + FXAA: igual que 2xPSP

3xPSP: igual que 2xPSP

3xPSP + FXAA: igual que 2xPSP

4xPSP: igual que 2xPSP

4xPSP + FXAA: similar a 2xPSP, más frecuente en los 20 que en los 30 FPS

Una vez más, se hace patente la mala gestión de los videos con ffmpeg. Al mismo tiempo que hay desfases, continúan apareciendo saltos de imagen de vez en cuando, incluso a resoluciones bajas, no es nada grave y las películas no se ven afectadas por estos saltos de imagen.

En el Juego/Combates

2xPSP: 15-30 FPS, en su mayor parte máxima velocidad a 30 FPS

2xPSP + FXAA: igual que 2xPSP

3xPSP: igual que 2xPSP

3xPSP + FXAA: igual que 2xPSP

4xPSP: similar a 2xPSP, aunque mayoritarimante a 20 FPS, se puede volver lento por unon o dos segundos.

4xPSP + FXAA: igual que 4xPSP

A estas alturas, no debería ser una sorpresa que el XU3/XU4 supere a cualquier otra plataforma. Probablemente lo mejor sea ejecutar este juego a 3xPSP + FXAA. Sin embargo, he observado al-

Figura 5 - Soul Calibur

gunos problemas gráficos en este juego. Algunos efectos tenían como resultado una especie de luz de arco iris. No estoy seguro de cual es la causa, ni si sólo ocurre con mi imagen o con cualquier otro XU3/XU4 también. No obstante, no recuerdo haberlo visto antes.

Soul Calibur

Me encanta este juego de lucha y lo preferiría al Tekken a cualquier hora. Adoraba jugar a Soul Calibur en mi Dreamcast, y la versión para PSP también es muy buena.

C2

Video

2xPSP: 18-24 FPS, lento

2xPSP+FXAA: 15-20 FPS

3xPSP: 16-22 FPS

3xPSP+FXAA: 12-15 FPS

En el Juego/Combates

2xPSP: 13-20 FPS, no hay retardos

2xPSP+FXAA: 13-16 FPS, algo lento

3xPSP: 13-16 FPS, mejor que 2xPSP + FXAA

3xPSP+FXAA: 10-11FPS, muy lento

Una vez más el C2 me decepcionó, Por lo general el juego funciona bastante bien en el resto plataformas, esperba algo mas del C2. En 2xPSP debería haber llegar al menos a los 20 o 30 FPS, aunque gracias a Frameskipping, 2xPSP apenas tiene retardos.

U2/U3

Vídeo

2xPSP: 40-60 FPS, sin retardos, mayoritariamente supera los 50 FPS

2xPSP + FXAA: 35-50 FPS, puede aparece algún pequeño desfase, en la mayoria de los casos supera los 40 FPS, rara vez se acerca a los 50

3xPSP: 38-60 FPS, sin retardos visibles, en su mayoria supera los 50 FPS

3xPSP + FXAA: 28-30 FPS, observé 26 FPS una vez, pero la mayoría de las veces llega a los 29 FPS

En el Juego/Combates

2xPSP: 44-60 FPS, sin retardos, por encima de 50 o incluso los 60 FPS

2xPSP + FXAA: 36-45 FPS, sin retardos, en torno a los 40 FPS

3xPSP: 36-60 FPS, sin retardos con 45-50 FPS, pero a veces llega a 60 FPS

3xPSP + FXAA: 23-26 FPS, ligeros saltos de imagen, especialmente cuando la cámara pasa a toda velocidad

El juego se ejecuta muy bien en el U2/U3, y los combates son muy divertidos. Si tuviera algunos frames más este juego funcionaría muy bien a 3xPSP + FXAA, pero como tiene ligereos saltos de imagen, ajustarlo a 3xPSP o 2xPSP + FXAA es la mejor opción.

XU3/XU4

Con el XU3/XU4, he experimentado sin Frameskipping, pero al final resultó que algunas escenas necesitaron Frameskipping a 1x o 2x, pero a parte de eso el juego funcionaba perfectamente.

Videos

2xPSP: 24-60 FPS, aparecen pequeños saltos, desactivando el Frameskipping mantiene los FPS en 60

2xPSP + FXAA: baja a 55 FPS sin Frameskipping, sino igual que 2xPSP

3xPSP: igual que 2xPSP

3xPSP + FXAA: necesita Frameskipping, 23-57 FPS, se aprecias saltos

4xPSP: igual que 2xPSP

4xPSP + FXAA: 20-32 FPS con Frameskipping, los videos empiezan a tener algunos saltos

No usar el Frameskipping sería lo mejor, pero el juego lo necesita, dependiendo de la resolución que elijas. 3xPSP+FXAA y Frameskipping fijado en 1 o 2 es la mejor opción.

En el juego/Combates

2xPSP: dependiendo del fondo, incluso sin Frameskipping se ejecuta a 60 FPS, pero es mejor jugar con 1x Frameskipping a máxima velocidad, aunque bajemos a 30 FPS

2xPSP + FXAA: requiere Frameskipping, pero con 1x Frameskipping también se ejecuta a máxima velocidad

3xPSP: con 1x Frameskipping funciona a máxima velocidad a 30 FPS

3xPSP + FXAA: se ejecuta mayoritariamente bien en 1x Frameskipping, pero puede ralentizarse en determinados movimiento o golpes finales, usando 2x frameskipping el juego se mantiene entre 20-30 FPS sin retardos o saltos visibles.

4xPSP: funciona bien con 1x Frameskipping

4xPSP + FXAA: funciona perfectamente con 2x Frameskipping

Realmente me sorprendió ver 4xPSP + FXAA trabajando de este modo, pero incluso a 3xPSP + FXAA ya se ve estupendamente y se ejecuta sin problemas.

GTA - China Town Wars

Este juego también es bastante difícil de emular, pero es muy divertido, y aunque todas las acciones las llevas a cabo en el mismo "campo", el fun-

Figura 6 - GTA China Town Wars

cionamiento de éstas es muy diferente. Caminar por la ciudad, conducir un coche lento, conducir un coche rápido (incluso conducir despación con un coche rápido) y luchar contra los enemigos con diferentes armas, todo reacciona de manera diferente, de modo que basaré mis resultados en una media global. Aunque conducir coches generalmente es lo que más esprime el sistema en la mayoria de los casos.

C2

En el juego

2xPSP: 9-15 FPS, dependiendo del coche, puede ser incluso menor.

Fue una experiencia pesima que no esperaba para nada. Este juego funciona muy mal en el C2. Posiblemente se pueda jugar en 2xPSP con Frameskipping fijado en 1, pero está muy lejos de ofrecer una experiencia sin retardos y saltos. Ni siquiera me molesté en probar cualquier otra resolución superior con o sin FXAA activado, ya que ambas configuraciones degradaban aún más el rendimiento.

U2/U3

En el juego

2xPSP: 25-30 FPS máxima velocidad, cae a 20 FPS en las persecuciones

2xPSP + FXAA: igual que 2xPSP

3xPSP: 20-30 / 30 FPS, mayoritariamente 22-25 FPS, puede bajar a los 20 FPS en persecuciones a alta velocidad.

3xPSP + FXAA: 15-20 FPS, mayoritariamente 17-18 FPS, bajando a los 13 FPS en las persecuciones a alta velocidad

Aunque 2xPSP y 2xPSP + FXAA responde casi igual, el juego a penas se beneficia del FXAA, así que probablemente lo mejor es usar la configuración 2xPSP. Las resoluciones superiores como la 3xPSP pueden funcionar también, pero en algunos casos, puede que necesites algo de velocidad extra en el juego, de modo que te sugiero que bajes a 2xPSP con Frameskipping configurado en 1.

XU3/XU4

En el juego

2xPSP: a máxima velocidad, excepto si conduces los coches más rápidos

2xPSP + FXAA: igual que 2xPSP

3xPSP: igual que 2xPSP

3xPSP + FXAA: igual que 2xPSP

4xPSP: igual que 2xPSP

4xPSP + FXAA: algo lento, especialmente con los coches más rápidos

Este juego funciona mejor de lo esperado en esta plataforma Probablemente sea mejor ejecutarlo a 3xPSP + FXAA o 3xPSP, ya que FXAA no parece mejorar demasiado los gráficos en este juego.

Conclusiones

Me he sentido muy decepcionado con el rendimiento del C2 a la hora de ejecutar PPSSPP, lo que me lleva a la conclusión de que el recompilador dinámico ARM64 no está funcionando correctamente o no está todavía totalmente implementado, ya que esperaba resultados similares a los del U2/U3, sobre todo porque el C2 tiene mejor GPU. El U2/U3, el X2, y posiblemente también el ODROID-X son dispositivos muy consolidados, y no presentan grandes problemas al ejecutar juegos PSP. Teniendo en cuenta que estos dispositivos ODROID son de hace cinco años, pone de manifiesto lo que este hardware todavióa es capaz de hacer.

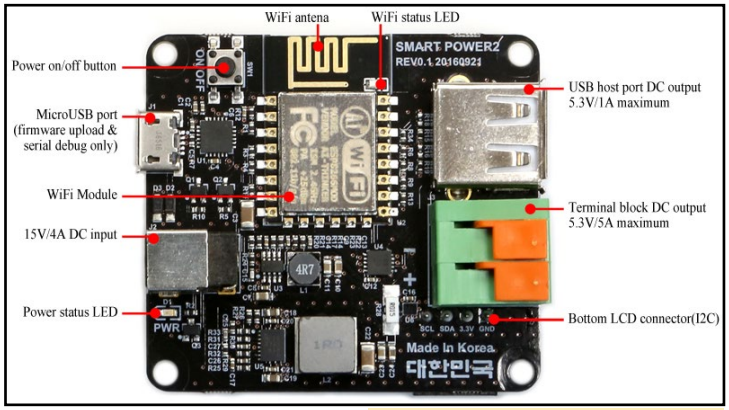
No es de extrañar que el XU3/XU4 supere a cualquier otra placa ODROID cuando hablamos de emulación PSP. La CPU es lo suficientemente rápida como para emular la mayoría de los juegos a máxima velocidad, y junto con su potente GPU, podemos aumentar la resolución de renderizado y activar shaders como FXAA. He jugado y finalizado un par de juegos en el U3 y XU3, y bajo mi experiencia personal, puedo decir que PPSSPP funciona muy bien en los ODROIDs y los gráficos son impresionantes. Espero que los problemas con el C2 puedan solucionarse pronto, y que al final veamos un rendimiento PPSSPP al menos similar a los lejendaríos U2/U3.

SMART POWER2

TU MEJOR AMIGO PARA EL AHORRO ENERGETICO

editado por Rob Roy (@robroy)

El SmartPower2 es una fuente de alimentación Wi-Fi que controla el voltaje y el encendido/apagado de un dispositivo conectado, disponible para su compra en la tienda Hardkernel por 40$ en http://bit.ly/2j3hhcv. También puedes monitorizar la corriente de carga y el consumo de energía de forma remota desde tu smartphone, tablet o PC haciendo uso de la interfaz de usuario de la página web integrada. La guía completa, los esquemas, el archivo de diseño PCB y la información detallada están disponibles en la Wiki de Hardkernel en http://bit.ly/2jVXvOC.



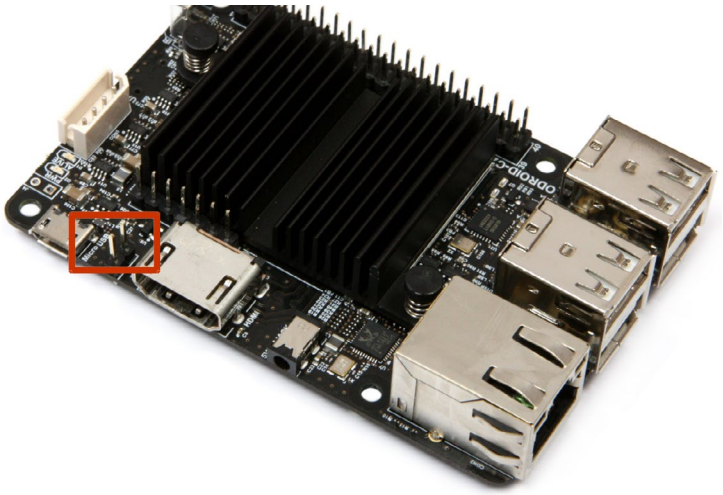
Esquema de la placa SmartPower2 con anotaciones

Monitorizando de forma remota el SmartPower2 a través del Smartphone

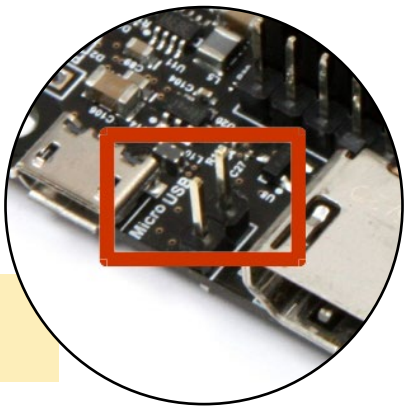
CONSUMO ELECTRICO DEL ODROID-C2

UN SIMPLE TRUCO QUE HACE MILAGROS

editado por Rob Roy (@robroy)



Si posee un ODROID-C2, puede reducir el consumo de energía significativamente ajustando uno de los conectores. Por defecto, la placa se suministra con el conector J1 en la posición “on”, lo cual permite que la placa sea alimentada a través del puerto OTG. Sin embargo, si tu ODROID-C2 utiliza la clavija DC 5V, puede estar consumiendo energía innecesariamente. De acuerdo con las mediciones, cuando el dispositivo se mantiene a ralentí con una consola serie, el consumo de energía esta en torno a los 1,7W. Una vez retirado el conector, tal y como se muestra en la Figura 1, el consumo cae a 1,2 W, un ahorro de 0,5 W. Esta modificación también reducirá el calor que produce la propia placa. Para comentarios, preguntas y sugerencias, visita el post original en http://bit.ly/2jSIlsl.



Conector JI retirado

CONOCIENDO UN ODROIDIAN

RICHARD BOWN (@RICHARD-G8JVM)

editado por Rob Roy (@robroy)

Por favor, háblanos un poco sobre ti

Ahora estoy jubilado, trabajaba como ingeniero RF. Empecé con la reparación y calibración de equipos de prueba RF, luego pasé a diseñar y desarrollar teléfonos móviles y estaciones base.

También soy un radioaficionado desde 1974. Vivo en el condado rural de Shropshire, Inglaterra, no muy lejos de la frontera con Gales. Estudié electrónica industrial en una universidad al oeste de Inglaterra, centrada más bien en la electrónica digital. Llevo casado con mi esposa, Sue, 44 años y tenemos dos hijos. El mayor tiene su propia familia y trabaja para un importante fabricante de automóviles en el sureste de Inglaterra, se encarga de detectar y solucionar problemas de fabricación. Mi hijo menor tiene su propio negocio de calefacción y fontanería en Shropshire. Puedes conocer más sobre mí y mis proyectos en http://g8jvm.com.

¿Cómo empezaste con los ordenadores?

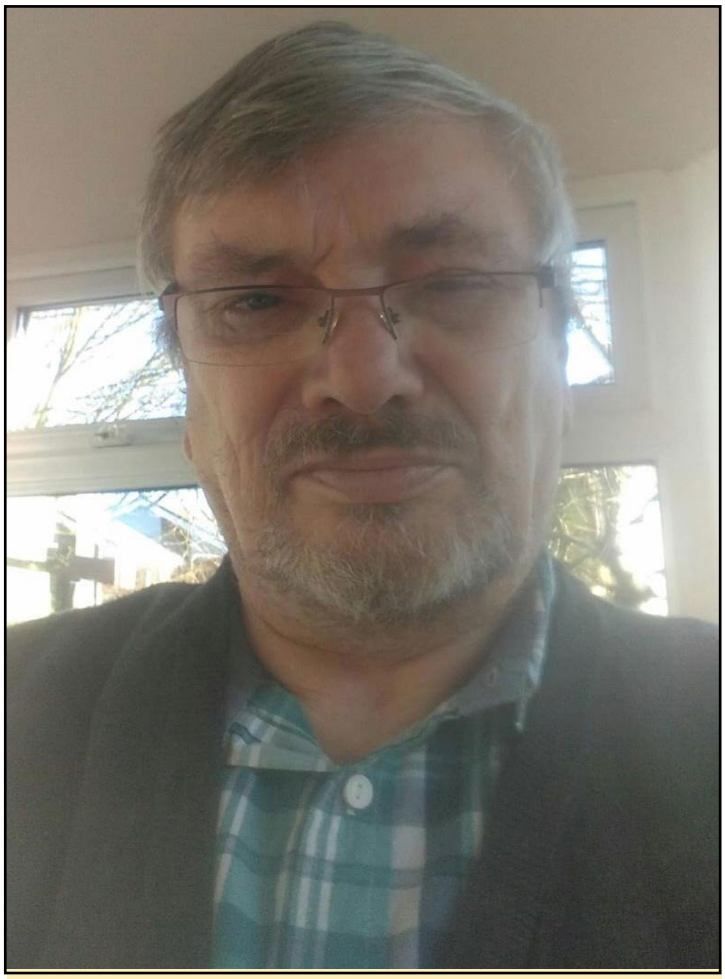
He tenido cierto interés por los ordenadores desde los años 80, además desarrolle mi propia máquina basada Z80 mientras estaba en la universidad. Aprendí un montón de programación de código de instrucciones, pero a medida que se hicieron más populares los idiomas de alto nivel, pronto olvidé esas habilidades. Empecé a usar Linux hace 22 años mientras me dedicaba a la radio por paquetes, no tanto para hablar o usar el buzón de mensaje a 1200 baudios, sino más bien para implementar una red TCP/IP de alta velocidad en el Reino Unido, similar a la de Alemania. Sin embargo, me di cuenta que el deseo por la tecnología en el Reino Unido era algo escaso, así que volví a mi principal inquietud por la radio comunicación, una señal débil que trabaja en VHF a frecuencias de microonda. Esta área actualmente cuenta con un software muy útil que es de gran ayuda, pero utiliza anchos de banda muy estrechos para extraer las señales del área de ruido.

¿A quién admiras en el mundo de la tecnología?

Admiro a Joe Taylor K1JT, por su trabajo en el software de señales débiles.

¿Qué te atrajo a la plataforma ODROID?

Inicialmente, empecé con un C1+ para darle movimiento a una cámara IP. Luego descubrí que también podría utilizarlo para Kodi. Miré la Raspberry Pi, pero las especificaciones del C1+ eran bastante mejores, aunque su uso estaba menos extendido dentro de la comunidad de la radio comunicación.



Richard es un radio aficionado amateur desde hace mucho tiempo

¿Cómo usas tus ODROIDs?

Más tarde me hice con un XU4 para substituir mi ordenador principal. Me las arreglé para conseguir que muchas de las aplicaciones que utilizo en Linux pudieran ejecutarse en el XU4, como la suite WSJTX. También uso Live MUF con ayuda de Dave G7RAU, ya que esta aplicación necesita Mono para ejecutarse. El XU4 aloja otro software relacionado con la radio comunicación, así como Piklab que se ejecuta sobre QT4, el cual tiene una interfaz gráfica simple, agradable y fácil de usar, a diferencia de MPLAB. Desafortunadamente, he dejado de usar recientemente el XU4 y he regresado a una pequeña plataforma x86_64, ya que necesitaba ejecutar varias aplicaciones al mismo tiempo, lo cual superaba la capacidad del XU4. He cambiado mi C1+ por un C2 que ejecuta Ubuntu 16.04, una plataforma mucho más estable capaz de soportar dos cámaras IP y una cá-

El radio-zulo de Richard, donde pasa gran parte de su tiempo libre

mara USB en movimiento, además de ser una plataforma más rápida para Kodi.

¿Cuál es tu ODROID favorito y por qué?

Me gusta la XU4, porque es una plataforma con un rendimiento mayor que el C1 o el C2.

¿Qué innovaciones te gustaría ver en futuros productos de Hardkernel?

Me gustaría ver un receptor de infrarrojos (IR) integrado en todos los modelos de ODROID para poder usar un mando a distancia, y más memoria integrada para manejar los gráficos más rápido. Tiene que ser capaz de poder iniciar varios programas que utilicen representaciones gráficas, especialmente cuando si utilizan mapas de alta resolución.

¿Qué aficiones e intereses tienes aparte de los ordenadores?

Estoy muy involucrado en el tema del radio comunicación para aficionados. He ocupado el callsign G8JVM desde 1974 (CEPT clase 1). También practico la fotografía y colecciono ferrocarriles (4mm GWR circa 1935).

¿Qué consejo le darías a alguien que quiere aprender más sobre programación?

Muy simple: no rendirse. Todos tenemos que empezar en algún momento, y no dejarse llevar por las criticas esporádicas. Leer mucho y no tener miedo de estropear algo, ya que ello puede convierte en un desafío.

La esposa de Richard Susan en Monument Valley, Utah