

CarPC • MongoDB • PiHole • Navegación GPS • ODR0ID-Go Dual Core

ODROID

Año Seis
Num. #64
Abr 2019

Magazine

LA MEJOR EMULACION RETRO X86

Con Box86

INFO Y MONITORIZACION:

LCD CLOUDSHELL2 CON OPENMEDI VAULT
TOTALMENTE TESTEADO EN DEBIAN

BITTORRENT:

COMPARTE TUS PROYECTOS CON UN
SERVIDOR DE SEMILLAS ODROID-HCI

EL PUNTO G:

TU DESTINO PARA TODOS
TUS JUEGOS ANDROID

GOOGLE HOME:

APRENDE A USAR TU PROPIO ODROID-C2
COMO UN ASISTENTE PERSONAL



Herramienta de Monitorización y Pantalla LCD CloudShell2

© April 1, 2019

He creado una herramienta de información y monitorización para CloudShell2 y deseo compartirla con la comunidad.



Monta tu Propio Servidor de Semillas BitTorrent ODROID-HC1

© April 1, 2019

BitTorrent ofrece una forma de compartir multitudes de archivos; peer-to-peer a través de archivos "torrent". En el fondo, BitTorrent permite el alojamiento distribuido y el intercambio de archivos de manera que el cargador original no tenga que proporcionar todo el ancho de banda o el espacio de almacenamiento para que [▶](#)



Navegación GPS con ODROID-GO

© April 1, 2019

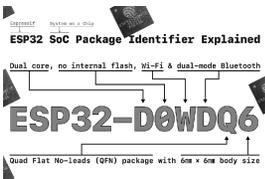
ODROID-GO es uno de esos pequeños sistemas extensibles versátiles con los que, dado algo de imaginación, puedes crear proyectos muy útiles e interesantes. Aquí tienes uno de esos proyectos, que integra un sensor GPS para mostrar mapas GPS en un ODROID-GO.



Cómo Crear un Servidor de Backup: Utilizando Clonezilla y PXE - Parte 2

© April 1, 2019

El paquete tftp-hpa es un servidor FTP muy pequeño y liviano que se utiliza casi exclusivamente para arrancar desde una red



Dual Cores para GO: Procesamiento Dual Core en ODROID-GO

© April 1, 2019

¿Cómo se puede usar este dual core en ODROID-GO? Por ejemplo, ¿cómo puede un programa escrito en el Entorno de Desarrollo Integrado de Arduino (IDE) aprovechar el dual core de ESP32? Es muy fácil usar el dual core dentro de ODROID-GO, pero primero recopilemos algo de información básica.



Google Home: Usando el ODROID-C2 como un Asistente Personal

© April 1, 2019

Este artículo describe como instalar Google Home en el ODROID-C2.



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

© April 1, 2019

¡Estos ODROID están muy animados con el sonido de los juegos Android! Desde la serie C (por ejemplo, ODROID-C2) hasta la X (por ejemplo, ODROID-XU4), jugar en tu consola favorita nunca ha sido tan divertido. Así que abre tu aplicación Google Play Store y pongámonos a descargar. Run Garfield, Run! [▶](#)



PiHole

© April 1, 2019

Este artículo trata sobre el software PiHole Linux con DNS-over-TLS en Docker. PiHole es una aplicación que permite bloquear la publicidad y los rastreadores de Internet. Actúa a modo de sumidero DNS, ideal para usarse en una red privada. Docker ha añadido soporte para la Raspberry Pi y, por lo [▶](#)



Juegos Linux: Box86 – Un Nuevo Emulador x86 para ODROID

© April 1, 2019

Este mes, he incluido algo diferente para variar. Algunos de vosotros ya lo habréis leído en los foros en <https://forum.odroid.com/viewtopic.php?f=91&t=34070>, quisiera hablar de ello: Box86, un nuevo tipo de emulador x86 para ARM que se puede ejecutar en los ODROIDS. ¿Qué es Box86? Box86 es un emulador x86 que te [▶](#)



CarPC

© April 1, 2019

<https://forum.odroid.com/viewtopic.php?f=55&t=33941> En este artículo, me gustaría presentarte mi proyecto CarPC. Este proyecto está desarrollado partiendo de cuatro componentes principales: Una pantalla a medida sobre una carcasa de unidad principal Android 2 DIN Un ODROID-XU4Q Un ZTE Blade como módem 3G y sistema de seguimiento por GPS para cuando el coche [▶](#)



Conociendo un ODROIDian: Robert Millette (@Lix-Alpha)

© April 1, 2019

Mi nombre es Rob, más conocido como Lix-Alpha en los foros ODROID y tengo 43 años. Tengo una esposa, dos hijos, un gato siamés y todos mis dientes. Me encanta jugar a los clásicos de los años 90 en mi ODROID-GO, crear música con mis dispositivos de bolsillo y reajustar [▶](#)

Herramienta de Monitorización y Pantalla LCD CloudShell2

© April 1, 2019 By @AreaScout CloudShell, Linux, Tutoriales



He creado una herramienta de información y monitorización para CloudShell2 y deseo compartirla con la comunidad. La he utilizado con OpenMediaVault y, por lo tanto, sólo ha sido testada en Debian, pero también debería funcionar en Ubuntu.

La imagen se puede descargar desde el siguiente enlace. Debes instalar las últimas actualizaciones, que incluyen una actualización del kernel desde el 4.9 a 4.14 a través de la interfaz de administración.

<https://goo.gl/YzaeMy>



Figura 1: Cloudshell2 ejecutando OpenMediaVault en toda su gloria

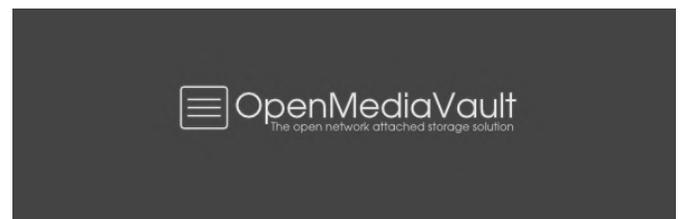


Figura 2 - OpenMediaVault

<https://goo.gl/ti7v3U>

Ilustración

El objetivo principal era el de proporcionar una vista en la que la información se pudiera leer fácilmente utilizando un tipo de letra más grande como las que utilizan otras herramientas.

El estado del RAID, la hora actual, el tiempo de actividad, el promedio de uso, la dirección IP actual y el espacio libre son solo algunos de los datos que puedes obtener. En la parte inferior tienes cuatro iconos; Los de la izquierda son:

- Estado de conexión del cable Ethernet
- Velocidad de conexión Ethernet admitida
- Estado del encendido/apagado del ventilador
- Estado de conexión del cable HDMI

La herramienta CloudShell2_lcd supervisa el estado y la temperatura del Raid. Tiene dos activadores de temperatura, uno para apagar el CloudShell2 y enviar una notificación por correo electrónico al usuario si se alcanza una cierta temperatura durante un tiempo determinado (por defecto, 85 ° C durante 15 segundos). El segundo regula el ventilador de la carcasa del CloudShell2, por defecto, lo encenderá si el XU4 alcanza una temperatura > 65 ° C y se apaga si es temperatura <60 ° C (65-5) se mantiene durante un período de 5 minutos. Todos los correspondientes scripts para el envío de correo y el inicio están instalados en /usr/local/bin

```
$ raid_mailsend
$ start-lcd
$ temperature_shutdown
```

Dentro de la interfaz de administración de OpenMediaVault (OMV), debes habilitar las notificaciones de correo. En caso de errores de HW/RAID, se informa junto con todo lo que esté dando problemas en ese momento o, alternativamente, simplemente se reportará información del sistema. [caption id="attachment_11863" align="alignnone" width="790"]

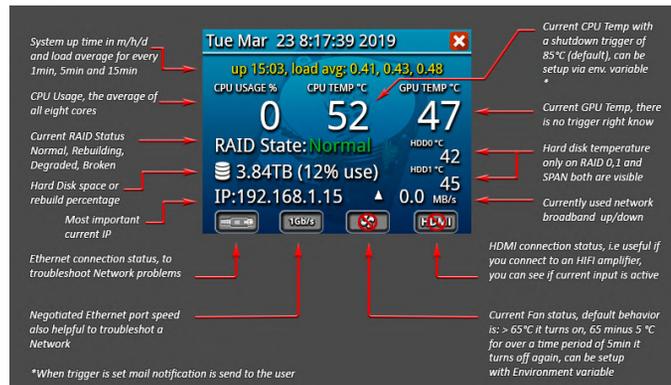


Figura 3 – Ilustración del CloudShell2/[caption]

XU4 y XU4Q son diferentes en cuanto a los valores que se pueden usar, en un XU4 (sin ventilador en SoC) podría ser mejor configurar el ventilador para que esté siempre encendido, aunque los valores pro defecto deberían ser lo suficientemente buenos para una carga de trabajo media diaria.

Instalación

Activar la pantalla LCD:

```
$ wget https://github.com/seungchar/odroid-cloudshell/raw/master/odroid-cloudshell_20170420-4_armhf.deb
$ sudo dpkg -i odroid-cloudshell_20170420-4_armhf.deb
```

Instalar dependencias:

```
$ sudo apt-get install libfreetype6 systat
```

Finalmente, instalar la herramienta de información y monitorización LCD. El driver LCD ya debería estar ejecutándose, cárgalo manualmente o reinicia en este punto.

```
$ wget http://www.areascout.at/cloudshell2-monitoring_1.0.7-2_armhf.deb
$ sudo dpkg -i cloudshell2-monitoring_1.0.7-2_armhf.deb
```

¡Diviértete con el nuevo estilo de pantalla!

Ahora puedes continuar configurando el controlador RAID con los interruptores DIP en la parte frontal de tu CloudShell2. Después, debes montar el conjunto RAID de disco duro creado recientemente dentro de la interfaz de administración de OMV.

La guía de configuración de HW RAID la puedes descargar aquí: <https://goo.gl/pwWm49>

Tendrías que saltarte los pasos 2 y 3 de esa guía.

Variables de entorno

Tanto la temperatura como la histéresis (período de tiempo) se pueden configurar individualmente a través de variables de entorno.

NETIF : el nombre de la interfaz de red, es decir, eth0, se detecta automáticamente mediante el script de inicio de la herramienta LCD, pero si falla, se puede configurar manualmente.

MAX_CPU_TEMP : Temperatura de apagado del CloudShell2 (por defecto MAX_CPU_TEMP=85).

CPU_TEMP_HYSTERESIS : Cuánto tiempo en segundos ha estado activo el MAX_CPU_TEMP para apagar el CloudShell2 (por defecto CPU_TEMP_HYSTERESIS=15).

FAN_ON_TEMP : Temperatura con la que el ventilador de la carcasa debería encenderse (por defecto FAN_ON_TEMP=65).

FAN_OFF_HYSTERESIS : Tiempo, en segundos, que FAN_ON_TEMP menos 5 °C debe mantenerse para apagar el ventilador nuevamente (por defecto, FAN_OFF_HYSTERESIS = 300).

CS_HDD_MOUNTP : Punto de montaje único o ruta del dispositivo para mostrar el espacio del disco duro, es decir, si estás utilizando una configuración de LVM compleja.

Ejemplo:

Exportar las variables de entorno dentro del script /usr/local/bin/start-lcd:

```
#!/bin/bash
export NETIF=$(ls /sys/class/net -I lo)
export FAN_ON_TEMP=0
echo 0 > /sys/class/graphics/fbcon/cursor_blink
echo none >
/sys/class/leds/blue\:heartbeat/trigger
/usr/local/bin/CloudShell2_lcd
```

Esto configurará el ventilador para que esté siempre encendido, existen varios modos de declarar las variables de entorno, ésta es solo una forma:

```
$ sudo systemctl restart cloudshell2-monitoring
```

o podría ser necesario reiniciar

La herramienta también tiene una nueva opción de línea de comando -i que puedes usar para obtener más información sobre tu controlador Raid y los discos conectados; Puede ejecutarla en cualquier momento, incluso si la herramienta ya está ejecutándose:

```
$ CloudShell2_lcd -i
```

¡Que te diviertas!

Descargo de responsabilidad: de ninguna manera me hago responsable por cualquier daño que sufra tu cloudshell2. Por favor, ten siempre muy presente lo que estás haciendo a la hora de realizar pequeños ajustes y experimentar.

Referencias

<https://forum.odroid.com/viewtopic.php?f=147&t=33860>

Monta tu Propio Servidor de Semillas BitTorrent ODRROID-HC1

© April 1, 2019 By Robbie Ferguson Linux, ODRROID-HC1, Tutoriales



BitTorrent ofrece una forma de compartir multitudes de archivos; peer-to-peer a través de archivos “torrent”. En el fondo, BitTorrent permite el alojamiento distribuido y el intercambio de archivos de manera que el cargador original no tenga que proporcionar todo el ancho de banda o el espacio de almacenamiento para que la gente pueda descargar el archivo. Esto es particularmente útil para los desarrolladores de código abierto porque les permite distribuir grandes archivos de proyectos con muy pocos gastos en general, tanto en recursos de servidor como en costes monetarios. BitTorrent también tiene el efecto de proporcionar a los usuarios finales descargas más rápidas (una vez que exista un cierto número de semillas), ya que recibirán múltiples piezas (fragmentos) de varias fuentes simultáneamente. Estos fragmentos se vuelven a ensamblar en el archivo original en el cliente BitTorrent del usuario. Pero a BitTorrent se le ha dado una mala reputación: debido a que es una tecnología estupenda para distribuir archivos de gran tamaño,

los piratas de contenido multimedia y de software no tardaron mucho en empezar a usarlo para distribuir ilegalmente material con derechos de autor.

Los medios de comunicación más conocidos tienden a informar mal de la tecnología: Internet acaba de cumplir 30 años*. La idea errónea parte de una tergiversación de la tecnología cuando las personas que realmente no la entienden son las que la explican. Piensa en la forma en que los informativos de red hablan del Bitcoin o Blockchain, o tal vez cómo se ha incluido Kodi en la lista negra de algunos dispositivos como resultado de una redacción deficiente dentro del periodismo. A los ojos de los lectores y espectadores con buenas intenciones, Bitcoin se convierte en una herramienta para atacantes de ransomware, el Blockchain; un misterioso reino digital donde los hackers del futuro cumplirán sus órdenes. Kodi está visto como una plataforma de streaming ilegal y BitTorrent es donde encontrarás descargas ilegales. Estas maravillosas

herramientas están envueltas en una especie de luz oscura, lo cual hace que una persona normal tenga miedo a usarlas. Por lo tanto, BitTorrent en sí mismo se ha convertido en algo ilegal para los usuarios que no entienden cómo funciona realmente.

Cuando descargas un archivo con BitTorrent, como una distribución de Linux IMG o ISO, puedes optar por dejarlo en tu servidor de BitTorrent. Esto hace que tu copia del archivo se convierta en una "semilla", lo que significa que otras personas pueden descargar fragmentos de tu archivo con su propio cliente BitTorrent durante el tiempo que decidas dejar como "semilla" el archivo. Un archivo con muchas semillas se descargará muy rápido y repartirá el uso del ancho de banda a través de la red. Es una forma sutil, aunque significativa de devolver anónimamente una distribución de Linux porque en esencia estás donando ancho de banda de descarga a futuros descargadores. Ubuntu está oficialmente disponible a través de BitTorrent, al igual que muchas otras distribuciones de Linux, incluido mi propio proyecto, NEMS Linux. BitTorrent es beneficioso en general: para mí, como una persona que desea obtener sus archivos IMG rápidamente, y para el siguiente usuario que quiera hacer lo mismo. BitTorrent reduce el coste de distribución para el desarrollador de la distribución y me ofrece una pequeña forma de "devolver el favor" al proporcionar una semilla. Además, garantiza un tiempo de actividad óptimo para las descargas, ya que, si una semilla se desconecta, el resto se recuperarán sin que se note la diferencia.

Aunque puedes instalar un cliente de BitTorrent como Transmission o Deluge en tu ordenador, para seguir generando torrent, tendrás que mantener ese software activo en todo momento. Si eres como yo, preferirás una solución dedicada; algo que no consuma mucha energía, pero que cuente con el espacio de almacenamiento y la potencia necesaria para hacer un excelente trabajo de descarga y siembra de torrents 24/7, sin tener que dejar nada ejecutándose en el ordenador. Aquí es donde entran en juego el ODROID-HC1 o el ODROID-HC2. Estos dos productos de Hardkernel son esencialmente un SOC ODROID-XU4 integrado en una bonita carcasa NAS

doméstica. Puedes pensar en ellos como un ordenador dedicada de 8 núcleos impulsado por Linux con 2 GB de RAM y un único disco duro. No hay salida de video, de modo que es ideal como servidor Debian sin monitor siempre encendido. Personalmente, yo usaré un ODROID-HC1 porque quiero usar un SSD y hacer que este pequeño servidor sea completamente sólido. Si prefieres utilizar un disco giratorio de 3.5", puedes optar por el ODROID-HC2 y seguir las mismas indicaciones que detallo a continuación. Elige el modelo que te gustaría usar según el tipo de almacenamiento del que dispongas.

Lo que necesitarás:

- ODROID-HC1
- Fuente de alimentación adecuada (por ejemplo, conector en forma de cilindro de 5V 4A para HC1)
- Carcasa para el ODROID-HC1
- Tarjeta MicroSD para el Sistema Operativo y el Software
- Unidad SSD de 2.5" para el almacenar los torrent
- Opcional: Batería RTC



Figura 1 - Mi servidor de semillas BitTorrent ODROID-HC1 con un SSD Kingston de 960 GB

Paso 1: Descarga mi imagen base de Debian Buster y grábala en tu tarjeta MicroSD. <https://baldnerd.com/sbc-build-base/>

Paso 2: Instala tu SSD en el chasis HC1 y coloca la carcasa. Inserta la tarjeta MicroSD y el cable Ethernet y reinicia.

Paso 3: Busca la dirección IP de tu dispositivo verificando la fuente DHCP de tu red. Conéctate via SSH con el nombre de usuario/contraseña por

defecto: baldnerd/baldnerd y conviértete en root escribiendo: sudo su

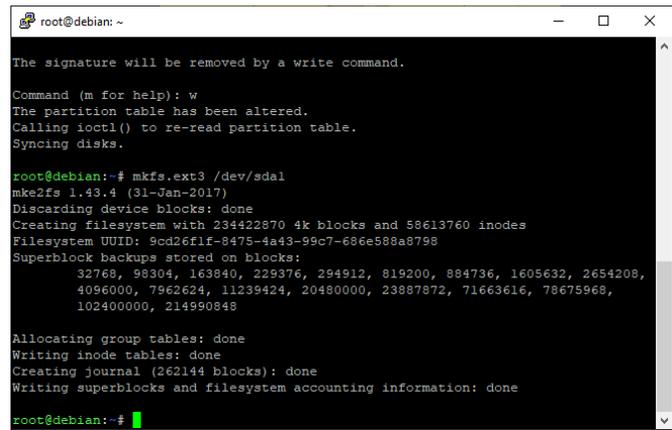
Step 4: Verifica tu fecha/hora con el comando date. Si tu fecha/hora está mal, los certificados serán rechazados, lo cual significará que apt no funcionará y tendrá un sinfín de problemas. Así que, lo primero es lo primero: debes verificar que la fecha y la hora sean correctas. Si son incorrectos, configúralo usando el comando generado por esta útil herramienta: <https://baldnerd.com/nerdgasms/linux/date/>

Y como el ODRROID-HC1 tiene un reloj en tiempo real, sugeriría añadir una pila RTC. Esto evitará contratiempos si tu HC1 se queda sin energía durante un tiempo prolongado, lo cual restablecería el reloj del sistema a noviembre de 2016.

Paso 5: Tu SSD se encuentra en /dev/sda. Te recomiendo que primero particiones la unidad con una única partición que use toda la unidad. Para hacer esto, escribe los siguientes comandos para eliminar la(s) partición(es) existente(s) y crear una nueva, teniendo en cuenta que esto borrará el contenido de la unidad, así que, si ésta no es una unidad nueva, primero haz una copia de seguridad de su contenido:

```
fdisk /dev/sda
d [enter] If there is more than one partition,
delete each
n [enter x 5]
If asked "Do you want to remove the signature?"
answer: y [enter]
w [enter]
```

Paso 6: Paso 6: Formatea tu nueva partición: mkfs.ext3/dev/sda1



```
root@debian: ~
The signature will be removed by a write command.
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
root@debian:~# mkfs.ext3 /dev/sda1
mke2fs 1.43.4 (31-Jan-2017)
Discarding device blocks: done
Creating filesystem with 234422870 4k blocks and 58613760 inodes
Filesystem UUID: 9cd26f1f-8475-4a43-99c7-686e588a8798
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
102400000, 214990848
Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done
root@debian:~#
```

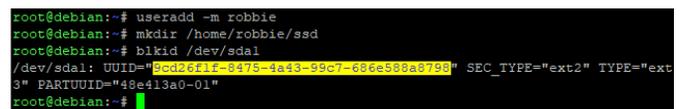
Figura 2 - Formateando /dev/sda1

Paso 7: Crea un nuevo usuario de Linux. Para este ejemplo, yo crearé "robbie". Reemplaza éste por tu propio nombre de usuario:

```
useradd -m robbie
usermod -aG sudo robbie
passwd robbie
```

Paso 8: crea una carpeta donde montaremos el SSD dentro de la carpeta de inicio de ese usuario y la haremos permanente. Yo la llamaré "ssd": mkdir /home/robbie/ssd && chattr +i /home/robbie/ssd

Paso 9: Podrías haberlo anotado en el Paso 6 tras formatear la partición, pero por si acaso localizaremos el UUID con este comando: blkid /dev/sda1



```
root@debian:~# useradd -m robbie
root@debian:~# mkdir /home/robbie/ssd
root@debian:~# blkid /dev/sda1
/dev/sda1: UUID="9cd26f1f-8475-4a43-99c7-686e588a8798" SEC_TYPE="ext2" TYPE="ext3" PARTUUID="48e413a0-01"
root@debian:~#
```

Figura 3 - Prepara tu punto de montaje y localiza tu UUID

D)

Paso 10: Haz que el SSD se monte automáticamente en el arranque:

```
nano /etc/fstab [enter]
Añade esta línea al final del archivo,
reemplazando 3333 ... por tu UUID real del Paso 9
y robbie por tu nombre de usuario real del Paso 7:
UUID=33333-33333-33333-33333 /home/robbie/ssd ext3
errors=remount-ro,noatime,discard 0 1
Press CTRL-O, [Enter] to save.
Press CTRL-X to exit.
```

Paso 11: Verifica que la unidad SSD este montada correctamente:

```
mount -a [enter]
mount | grep sda1 [enter]
```

Si la unidad se ha montado correctamente, deberías ver algo como esto:

```
root@debian-stretch:~# mount | grep sda1
/dev/sda1 on /home/robbie/ssd type ext3
(rw,noatime,discard,errors=remount-
no,stripe=8191,data=ordered)
```

```
root@debian:~# nano /etc/fstab
root@debian:~# mount -a
root@debian:~# mount | grep sda1
/dev/sda1 on /home/robbie/ssd type ext3 (rw,noatime,discard,errors=remount-ro,stripe=8191,data=ordered)
root@debian:~#
```

Figura 4: Asegúrate de que tu unidad esté montada en tu nuevo punto de montaje

Si no has tenido éxito, asegúrate de reparar o eliminar la línea /etc/fstab antes de reiniciar, de lo contrario, es posible que no pueda iniciar el sistema.

Paso 12: Ahora estamos listos para configurar el servidor BitTorrent. Usaré Transmission Daemon, que está disponible en los repositorios:

```
apt update && apt install transmission-daemon
```

Paso 13: Crea una carpeta en el SSD para almacenar las descargas y asígnale los permisos correctos:

```
mkdir /home/robbie/ssd/torrents
chown -R debian-transmission:debian-transmission
/home/robbie/ssd/torrents/
```

Paso 14: Es hora de detener Transmission para que podamos editar la configuración. Si olvidas detener el demonio, tus cambios no se guardarán: `systemctl stop transmission-daemon`

Paso 15: Modificar el archivo de configuración

```
Abre el archivo de configuración para editarlo:
nano /etc/transmission-daemon/settings.json
Cambia el valor de download-dir por
/home/robbie/ssd/torrents
Dado que nuestro servidor BitTorrent será
accesible solo en nuestra LAN (no lo abriremos
para el acceso remoto), podemos desactivar la
lista blanca de IP. Cambia rpc-whitelist-enabled a
false
Guarda tu configuración presionando CTRL-O
[intro], seguido de CTRL-X para salir.
```

Paso 16: Inicia Transmission una vez más: `systemctl start transmission-daemon`

Paso 17: Reinicia tu nuevo servidor simplemente escribiendo: `reboot`

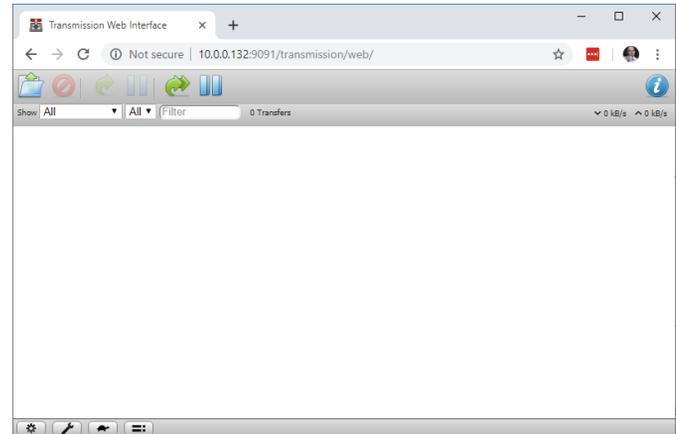


Figura 5 - Transmission está instalado y funcionando, accesible desde un navegador web en nuestra LAN

¡Ya está! Estamos en marcha. Si todo ha salido bien, podrá abrir la interfaz web visitando `http://ip:9091` desde tu ordenador, donde ip es la IP real de su nuevo servidor BitTorrent. El nombre de usuario por defecto de la interfaz web es `transmission` y la contraseña también es `transmission`. Puedes cambiar estos valores deteniendo el demonio como lo hiciste en el Paso 14, y editando los valores dentro del archivo de configuración antes de reiniciar el demonio.



Figura 6 - Abre un archivo Torrent

Puedes empezar a usar tu nuevo Servidor de semillas BitTorrent ODR0ID-HC1. Dentro de la interfaz web, presiona el icono "Open Torrent" en la parte superior izquierda.

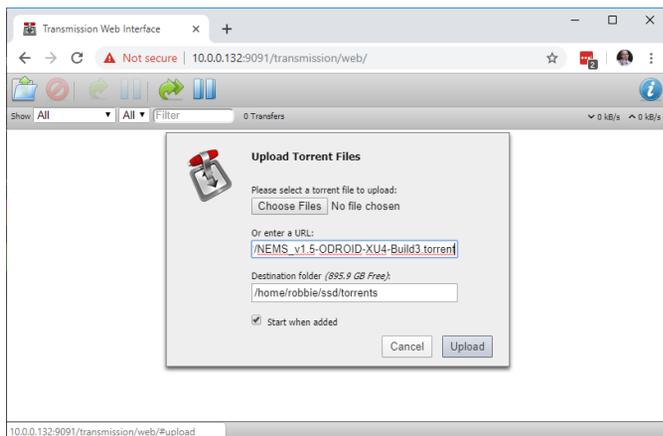


Figura 7 - Agrega el archivo Torrent o la URL para descargar

Añade la URL al archivo torrent que deseas cargar, o carga el archivo .torrent si lo tiene almacenado localmente. Si quieres probar con el torrent NEMS Linux, visita <https://nemslinux.com/download/nagios-for-odroid.php> y copia el enlace en cualquiera de los archivos .torrent. Cuando el torrent empiece a descargarse, se guardará en /home/robbie/ssd/torrents, que es tu partición /dev/sda1 en tu SSD/HDD.

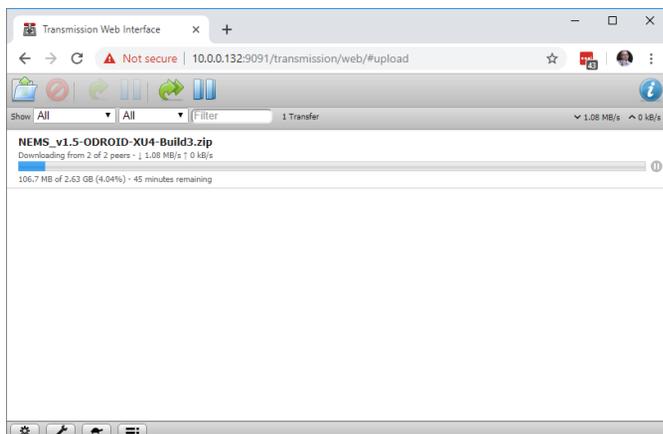


Figura 8 - Tu servidor de semillas BitTorrent está perfecto y se encuentra descargando tu primer Torrent

Una última reflexión: a pesar de que tu nuevo servidor de semillas BitTorrent ODROID-HC1 solo será visible para sus dispositivos LAN internos, nunca es bueno tener un sistema online utilizando las contraseñas por defecto. Reinicia tu servidor y luego inicia sesión en SSH con el nombre de usuario y la contraseña que creaste en el Paso 7. Si puedes iniciar sesión correctamente, elimina el usuario predeterminado: userdel -f -r baldnerd

En la edición del próximo mes de ODROID Magazine, daremos el siguiente paso juntos y aprenderemos a añadir y configurar el intercambio de archivos en red para que podamos abrir nuestros archivos descargados directamente desde un ordenador de nuestra red.

Nota

* Internet tiene 50 años, se inventó en 1969. Sin embargo, es la World Wide Web recientemente ha cumplido 30 años. Algunos de medios de comunicación más conocidos han confundido Internet con la World Wide Web, afirmando que Internet tiene solo 30 años. Dicho sea a su favor, la palabra "Internet" se ha usado erróneamente como sinónimo de "World Wide Web" en la lengua vernácula moderna. Sin embargo, son dos servicios distintos.

Sobre el Autor

Robbie Ferguson es el dueño de Category5 Technology TV y creador de NEMS Linux. Su programa de televisión se encuentra en <https://category5.tv/> y su blog es <https://baldnerd.com/>

Navegación GPS con ODROID-GO

© April 1, 2019 By Stefan Schreiter ↗ ODROID-GO, Mecanico



ODROID-GO es uno de esos pequeños sistemas extensibles versátiles con los que, dado algo de imaginación, puedes crear proyectos muy útiles e interesantes. Aquí tienes uno de esos proyectos, que integra un sensor GPS para mostrar mapas GPS en un ODROID-GO.

<https://github.com/ripper121/OdroidGoOSMGPSOffline>

El firmware necesario se puede obtener desde esta ubicación: <https://goo.gl/PgYF6C>

El menú de teclas es el siguiente:

- **JoyPad** = Te mueves
- **A/B** = Te acercas/alejas (el nivel de zoom va de 5 a 16, dependiendo de los niveles de zoom de los mosaicos que tengas)
- **Select/Start** = Brillo
- **Menu** = Reiniciar Movimiento



Figura 1 - Mapa offline OSM en Youtube

Generar Mosaicos

Como cabría esperar, las limitaciones de diseño del ODROID-GO no permiten mapas en vivo. Por lo tanto, un mapa debe estar predefinido y almacenado para su uso posterior. Los mosaicos son esencialmente estos mapas offline.

El código fuente se puede obtener desde el repositorio [git](https://github.com/ripper121/OdroidGoOSMGPSOffline):

Los mosaicos se pueden descargar en el sistema MS Windows usando la aplicación TileDownloader.exe de: <https://bit.ly/2HDSLRh> (EDITOR: el acortador de URL de Google no funciona para este enlace)

El comando de una sola línea (reemplaza las coordenadas específicas de tu ubicación) crea un mosaico y lo recupera:

```
> TileDownloader.exe -URL
https://a.tile.openstreetmap.org/{z}/{x}/{y}.png
-z 5 -left -0.489 -bottom 51.28 -right 0.236 -top
51.686
```

Donde se pueden especificar las siguientes opciones:

- -URL : TileServer
- -z : Zoom level

y los límites del recuadro de selección son:

- -left : min-longitude
- -bottom : min-latitude
- -right : max-longitude
- -top : max-latitude

En algunas regiones, se puede usar una coma ‘,’ en lugar del punto ‘.’, para los valores del recuadro de selección

Se pueden usar otros servidores de mosaicos del siguiente enlace:

https://wiki.openstreetmap.org/wiki/Tile_servers

También puedes obtener mosaicos adicionales de: <https://goo.gl/DRusXL>

Los mapas offline también se pueden almacenar en una tarjeta SD como puedes ver:



Figura 2 – ODROID-GO con un video youtube de un mapa GPS OSM offline

Utiliza un nivel de zoom de 5 a 14, con el marcador de posición como círculo rojo.

Este “desplazamiento” es una manipulación de las coordenadas GPS a través del JoyStick. Los siguientes pasos te permiten generar mosaicos (Mapa offline) para tu área:

- 1. Abre Maperitive.exe (<http://maperitive.net/>)
- 2. Mueve el mapa a tu posición favorita
- 3. MAP->Set Geometry Bounds
- 4. MAP->Set Printing Bounds

Ahora está fijada el área del mapa que quieres exportar.

- 5. TOOLS-> Generate Tiles (esto puede llevar algo de tiempo, dependiendo de la resolución del nivel de zoom)
- 6. Ahora encontrarás algunos archivos PNG en la carpeta Maperitive/tiles
- 8. Abre Flexxi.exe (<https://sourceforge.net/projects/flexxi-image-resizer/>)
- 9. Importa la carpeta Tiles
- 10. Cambia el tamaño de la imagen a 240x240px (el mejor tamaño para la pantalla GO)
- 11. Convierte a archivos JPG
- 12. Guarda
- 13. Ahora tienes tus mosaicos con el tamaño y formato correctos
- 14. Copia la carpeta “TILES” (todo en mayúsculas) en el directorio raíz de su tarjeta SD

Abre Arduino IDE y graba el código en tu GO. Este es el Código Arduino:

```
#include

#define DISPLAY_WIDTH 320
#define DISPLAY_HEIGHT 240
#define TILE_SIZE 240

bool firstRun = true;
double zoom = 10;
double lat_rad = 50.8225313, lon_deg = 12.7508936;
double tileX = 0, tileY = 0;
double old_lat_rad = 0, old_lon_deg = 0;
double old_zoom = 0;
double old_tileY = 0, old_tileX = 0;
uint8_t brightness = 127;

void setup()
```

```

{
  // put your setup code here, to run once:
  Serial.begin(115200);
  GO.begin();
  GO.battery.setProtection(true);
  GO.lcd.clear();
  GO.lcd.setCursor(0, 0);

  if (!SD.begin()) {
    GO.lcd.println("Card Mount Failed");
    Serial.println("Card Mount Failed");
    return;
  }
  uint8_t cardType = SD.cardType();

  if (cardType == CARD_NONE) {
    GO.lcd.println("No SD card attached");
    Serial.println("No SD card attached");
    return;
  }

  Serial.print("SD Card Type: ");
  if (cardType == CARD_MMC) {
    GO.lcd.println("MMC");
    Serial.println("MMC");
  } else if (cardType == CARD_SD) {
    GO.lcd.println("SDSC");
    Serial.println("SDSC");
  } else if (cardType == CARD_SDHC) {
    GO.lcd.println("SDHC");
    Serial.println("SDHC");
  } else {
    GO.lcd.println("UNKNOWN");
    Serial.println("UNKNOWN");
  }

  uint64_t cardSize = SD.cardSize() / (1024 *
1024);
  Serial.printf("SD Card Size: %lluMB
", cardSize);
  GO.lcd.printf("SD Card Size: %lluMB
", cardSize);

  delay(3000);

  GO.lcd.clear();
  GO.lcd.setCursor(0, 0);
}

//setBrightness(uint8_t brightness),
void loop()
{
  GO.update();

```

```

if (GO.JOY_X.isAxisPressed() == 1) {
  lon_deg += 0.0001;
  delay(10);
}
if (GO.JOY_X.isAxisPressed() == 2) {
  lon_deg -= 0.0001;
  delay(10);
}
if (GO.JOY_Y.isAxisPressed() == 2) {
  lat_rad += 0.0001;
  delay(10);
}
if (GO.JOY_Y.isAxisPressed() == 1) {
  lat_rad -= 0.0001;
  delay(10);
}
if (GO.BtnA.wasPressed() == 1) {
  zoom++;
}
if (GO.BtnB.wasPressed() == 1) {
  zoom--;
}
if (GO.BtnSelect.isPressed() == 1) {
  GO.lcd.setBrightness(brightness);
  brightness--;
}
if (GO.BtnStart.isPressed() == 1) {
  GO.lcd.setBrightness(brightness);
  brightness++;
}

if (lat_rad > 85.05112878)
  lat_rad = 85.05112878;
if (lat_rad < -85.05112878) lat_rad =
-85.05112878; if (lon_deg > 180)
  lon_deg = 180;
if (lon_deg < -180) lon_deg = -180; if (zoom >
16)
  zoom = 16;
if (zoom < 5) zoom = 5; if (brightness > 254)
  brightness = 254;
if (brightness < 1)
  brightness = 1;

//redraw only when something has changed
if (old_lat_rad != lat_rad || old_lon_deg !=
lon_deg || old_zoom != zoom || firstRun) {
  double posX, posY, fractpart, intpart;
  //calculate from coordinates to tile numbers
  tileX = long2tilex(lon_deg, zoom);
  tileY = lat2tiley(lat_rad, zoom);
  //fractional part is the position of the your
  coordinats in the tile

```

```

    posX = modf(tileX , &intpart);
    posY = modf(tileY , &intpart);
    posX = (posX * TILE_SIZE) + (abs(DISPLAY_WIDTH
- TILE_SIZE));
    posY = (posY * TILE_SIZE);

    //redraw only when something has changed
    if (uint16_t(old_tileX) != uint16_t(tileX) ||
uint16_t(old_tileY) != uint16_t(tileY) ||
old_zoom != zoom || firstRun) {
        String path = "/TILES/" +
String(uint16_t(zoom)) + "/" +
String(uint32_t(tileX)) + "/" +
String(uint32_t(tileY)) + ".jpg";
        Serial.println(path);
        if (SD.exists(path)) {
            Serial.println("File found.");
            GO.lcd.clear();
            GO.lcd.setCursor(0, 0);
            //drawJpgFile(fs::FS &fs, const char
*path, uint16_t x = 0, uint16_t y = 0, uint16_t
maxWidth = 0, uint16_t maxHeight = 0, uint16_t
offX = 0, uint16_t offY = 0, jpeg_div_t scale =
JPEG_DIV_NONE),
            GO.lcd.drawJpgFile(SD, path.c_str(),
(abs(DISPLAY_WIDTH - TILE_SIZE)));
        } else {
            GO.lcd.println("");
            GO.lcd.println("Debug:
File not found.");
            Serial.println("File not found.");
        }
        firstRun = false;
    }

    GO.lcd.drawRect(int32_t(posX), int32_t(posY),
4, 4, RED);
    GO.lcd.fillRect(0, 0, abs(DISPLAY_WIDTH -
TILE_SIZE), DISPLAY_HEIGHT, BLACK);
    GO.lcd.setCursor(0, 0);
    GO.lcd.println("Battery:");

GO.lcd.println(String(GO.battery.getPercentage())
+ "%");
    GO.lcd.println("Lon_deg:");
    GO.lcd.println(String(lon_deg, 6));
    GO.lcd.println("Lat_rad:");
    GO.lcd.println(String(lat_rad, 6));
    GO.lcd.println("Zoom:");
    GO.lcd.println(String(zoom));

```

```

    Serial.println(String(tileX, 6));
    Serial.println(String(tileY, 6));
    Serial.println(String(posX));
    Serial.println(String(posY));
    Serial.println(zoom);
    Serial.println(String(lon_deg, 6));
    Serial.println(String(lat_rad, 6));

    old_tileX = tileX;
    old_tileY = tileY;
}

old_lat_rad = lat_rad;
old_lon_deg = lon_deg;
old_zoom = zoom;
}

double long2tilex(double lon, double z)
{
    return (double)((lon + 180.0) / 360.0 * pow(2.0,
z));
}

double lat2tiley(double lat, double z)
{
    return (double)((1.0 - log( tan(lat * M_PI /
180.0) + 1.0 / cos(lat * M_PI / 180.0)) / M_PI) /
2.0 * pow(2.0, z));
}

double tilex2long(int x, int z)
{
    return x / pow(2.0, z) * 360.0 - 180;
}

double tiley2lat(int y, int z)
{
    double n = M_PI - 2.0 * M_PI * y / pow(2.0, z);
    return 180.0 / M_PI * atan(0.5 * (exp(n) - exp(-
n)));
}

```

Referencias

<https://forum.odroid.com/viewtopic.php?f=162&t=33629>
https://youtu.be/-4kA_Khlvus
<https://github.com/ripper121/OdroidGoOSMGPSOffline>
<https://youtu.be/BQWwTZANGIE>

Cómo Crear un Servidor de Backup: Utilizando Clonezilla y PXE – Parte 2

© April 1, 2019 By @lazgo Linux



Este es un artículo de dos partes que fue adaptado de los foros ODROID. La primera parte la puedes encontrar en <https://magazine.odroid.com/article/how-to-create-a-backup-server-using-clonezilla-and-pxe-part-1/>.

PXElinux

En los dos ejemplos anteriores, hemos configurado DHCP para indicar a un cliente PXE que intente iniciar la descarga del archivo denominado "pxelinux.0" desde un servidor específico. Este archivo es parte del paquete pxelinux de los cargadores de arranque. Así que sigue adelante, instala pxelinux en el mismo sistema en el que instales tftp-hpa. Puedes encontrar la página de inicio de Syslinux en [https://www.syslinux.org/wiki/index.php ...](https://www.syslinux.org/wiki/index.php...) ux_Project y los documentos de pxelinux en <https://www.syslinux.org/wiki/index.php?>

[title=PXELINUX](#) si alguna vez tiene planes para ampliar tu servidor PXE más allá de este pequeño ejercicio de Clonezilla, considera estudiarlo.

Configura tftp-hpa y haz que Clonezilla se inicie en la red

El paquete tftp-hpa es un servidor FTP muy pequeño y liviano que se utiliza casi exclusivamente para arrancar desde una red. Es muy ligero en cuanto a seguridad, así que no lo implementes en un entorno hostil a menos que tenga una o dos capas de seguridad que lo cubran. Al ser tan liviano su configuración es bastante simple, aunque no muy estandarizada. Lo más importante es configurar la carpeta raíz para tftpd. Mucha gente usa ubicaciones como /var/lib/tftpboot o /srv/tftpboot o incluso /opt/tftpboot. He visto ejemplos de configuraciones en las que el usuario crea el directorio /tftp, pero creo que es mejor no saturar el directorio / con más

carpetas. Al final, opté por /srv/tftp.root simplemente porque quería que me llamase la atención. Después de instalarlo en la ubicación elegida, crea el directorio y luego edita el archivo de configuración. La ubicación del archivo de configuración varía bastante dependiendo del dispositivo que esté utilizando. Puede ser /etc/default/tftpd-hpa o /etc/xinetd.d/tftp, o incluso /etc/conf.d/in.tftpd, por lo que deberás consultar la documentación de las distribuciones para asegurarte de que lo configures correctamente. A efecto de este "HOWTO" usaré /var/lib/tftpboot como directorio raíz.

En primer lugar, vamos a poner pxelinux.0 en su lugar. La ubicación de la instalación volverá a depender de tu distribución, pero generalmente estará en algún lugar de /usr/share o /usr/lib. Después de copiarlo, crearemos un directorio para guardar los archivos que extraeremos del archivo .zip de Clonezilla y otro directorio para guardar nuestros parámetros de arranque.

```
$ sudo cp /usr/lib/PXELINUX/pxelinux.0
/var/lib/tftpboot
$ sudo mkdir /var/lib/tftpboot/cz
$ sudo mkdir /var/lib/tftpboot/pxelinux.cfg
```

Luego copia el archivo zip de Clonezilla, extráelo y copie algunos archivos en su ubicación:

```
$ sudo cp /path/to/clonezilla-live-2.6.0-37-
amd64.zip /var/lib/tftpboot/cz/
$ cd /var/lib/tftpboot/cz/
$ sudo unzip -X *.zip
$ sudo cp /var/lib/tftpboot/cz/syslinux/*.c32
/var/lib/tftpboot
```

Luego crearemos el archivo de configuración del cargador de arranque PXE. Este "HOWTO" no será nada elegante, ni siquiera bonito. Simplemente cargará lo que le digamos después de una cuenta atrás de 10 segundos:

```
$ cd /var/lib/tftpboot/pxelinux.cfg
$ sudo nano default
```

Coloca lo siguiente en el archivo de configuración "default" y no olvides cambiar la dirección IP en la última línea. Como nota al margen, encontré esta

plantilla en algún lugar online hace algún tiempo atrás, aunque no recuerdo dónde:

```
# The default menu style - using vesa menu in this
example
DEFAULT vesamenu.c32
# If you have a png image in the tftpr directory
you can specify it here like so:
# Menu Background image.png
# Prompt user for selection
prompt 0

#Global label identifier
label Clonezilla
    # Set this entry as the default selection
    menu default
    # Actual viewable label text
    MENU LABEL Clonezilla
    # The timeout for the entry is a bit
unclear, but 100 should be equivalent to 10
Seconds.
    TIMEOUT 100
    TOTALTIMEOUT 100
    # The kernel image to load. This entry
would actually reside at
/var/lib/tftpboot/cz/live/vmlinuz
    #The path is relative to /var/lib/tftpboot or
your tftp root directory
    kernel cz/live/vmlinuz
    # The initrd relative to the
/var/lib/tftpboot directory and specifying the
netboot server, protocol, and file
    # In this example the tftp protocol is
used on server 192.168.123.1. The file is
filesystem.squashfs
    append initrd=cz/live/initrd.img boot=live
username=user union=overlay config components
quiet noswap edd=on nomodeset nodmraid locales=
keyboard-layouts= ocs_live_run="ocs-live-general"
ocs_live_extra_param="" ocs_live_batch=no
net.ifnames=0 nosplash noprompt
fetch=tftp://192.168.123.1/cz/live/filesystem.squa
shfs
```

Ahora reinicia el servicio tftpd e inicia un ordenador cliente con PXE. Debería funcionar.

Lecturas PXE adicionales

<https://help.ubuntu.com/community/UbuntuLTSP/roxyDHCP> <== Algo antiguo pero muy bueno
<https://clonezilla.org/livepxe.php> <== El documento que inspiró este HOWTO

<https://clonezilla.org/clonezilla-SE/> <==

Configuración de un servidor Clonezilla dedicado para una implementación a gran escala

https://wiki.gentoo.org/wiki/Home_router <==

Usando dnsmasq e iptables para convertir un servidor con dos NIC en un router

Parte tres: Creando un servidor de archivos con el sistema de archivos de red

El protocolo de intercambio de archivos más común en todo ecosistema nix* tiene que ser el Sistema de archivos de red. NFS es lo que podríamos llamar tecnología antigua. Las primeras versiones fueron desarrolladas internamente por Sun Microsystems en 1984. Pronto cumplirá 35 años y su última revisión importante fue en 2016. Entonces, ¿por qué todavía se está usando un protocolo de intercambio de archivos tan antiguo? Bueno, por la misma razón por la que sigue usándose. Simplemente funciona. Puedes usar NFS para compartir los directorios /home de los usuarios desde un único servidor, por lo que, independientemente del sistema nix* en el que inicien sesión en tu red, tendrán exactamente el mismo contenido en /home/nombre usuario. Puedes usar NFS para almacenar los sistemas de archivos raíz de estaciones de trabajo sin disco o imágenes de máquinas virtuales. Básicamente, cualquier parte del sistema de archivos de un ordenador nix* puede residir remotamente en un servidor NFS. El recurso compartido de red se puede montar en el momento del arranque con una entrada en /etc/fstab o manualmente con el comando "mount".

NFS no es perfecto. En la década de 1980, la mayoría de los esfuerzos de desarrollo del software se centraron en la "funcionalidad" y no en la "seguridad". Aunque ha habido intentos de añadir más seguridad a NFS en los últimos veinte años, ninguna de las soluciones propuestas ha sido fácil de configurar y de mantener. Por lo tanto, en este momento realmente no existe un estándar que aporte seguridad a NFS por defecto. Existe una propuesta interesante (<https://tools.ietf.org/html/draft-cel-nfsv4-rpc-tls-01>) que podría tener un gran impacto si llega a adoptarse. Por ahora, creo que es más seguro instalar tu servidor de archivos NFS detrás de un buen firewall y restringir el acceso al mismo con una buena

configuración. Si tiene tiempo para aprender, puede echar un vistazo a LDAP y Kerberos o incluso RPCSEC GSS.

Casi todas las distribuciones modernas de Linux tienen el servidor NFS integrado en su núcleo como módulo. Lo que les falta es la interfaz de usuario para configurarlo y controlarlo. En Debian, el paquete se llama nfs-kernel-server y en la mayoría del resto de distribuciones se llama nfs-utils. Adelante, instálalo ahora.

El siguiente paso es decidir dónde almacenarás las copias de seguridad y crear el directorio en tu servidor de archivos. Para este "HOWTO" usaremos /srv/backups como directorio compartido. Primero, crearemos el directorio y cambiaremos su propiedad al usuario y grupo con menos privilegiados:

```
$ sudo mkdir /srv/backups
$ sudo chown nobody:nogroup /srv/backups
```

El archivo de configuración /etc/exports controla qué directorios se comparten y de qué forma los usuarios y/o los ordenadores pueden acceder a ellos. Una copia de la página de guía de /etc/exports la tienes disponible en <https://linux.die.net/man/5/exports> o ejecutando "man export" en un intérprete de comandos bash.

El archivo exports tiene tres secciones por línea. Empieza con el directorio que compartirá el servidor, seguido de por quién puede acceder a él y el tipo de opciones para compartir y el acceso que tendrá esa persona. A modo de ejemplo:

```
/foo/bar    lazlo(rw, sync, root_squash)
```

Significa que tengo acceso de lectura/escritura a /foo/bar, el servidor sincronizará las escrituras pendientes en el disco antes de aceptar una nueva, y si alguien actúa en este recurso compartido como root, su uid y gid cambiarán de root:root (0:0) a nobody:nogroup (65535: 65535) para la acción dada.

Otro ejemplo:

```
/home/lazlo *(r,no_root_squash)
```

Llegado este momento, puedes reiniciar el servicio NFS y montar tu recurso compartido en un sistema

remoto. Pruébalo creando algunos archivos y directorios y luego ejecuta "ls -al" para asegurarte de que todo sea propiedad de nobody:nogroup. Para montar manualmente el recurso compartido en el Cliente Linux, deberás instalar el paquete del cliente NFS "nfs-common" en los sistemas basados en Debian. El paquete "nfs-utils" para la mayoría de las distribuciones incluye los programas cliente y servidor. Una vez que lo hayas instalado, puedes crear un nuevo subdirectorio en tu carpeta de inicio y montar el recurso compartido (nos olvide cambiar la dirección IP en el ejemplo):

```
$ mkdir nfstest
$ sudo mount -t nfs 192.168.123.1:/srv/backups
./nfstest
$ mkdir nfstest/testdir
$ dd if=/dev/zero of=nfstest/testdir/testfile
bs=1M count=10000 status=progress
```

Una vez hecho esto comprueba la propiedad:

```
$ ls -al nfstest/testdir
```

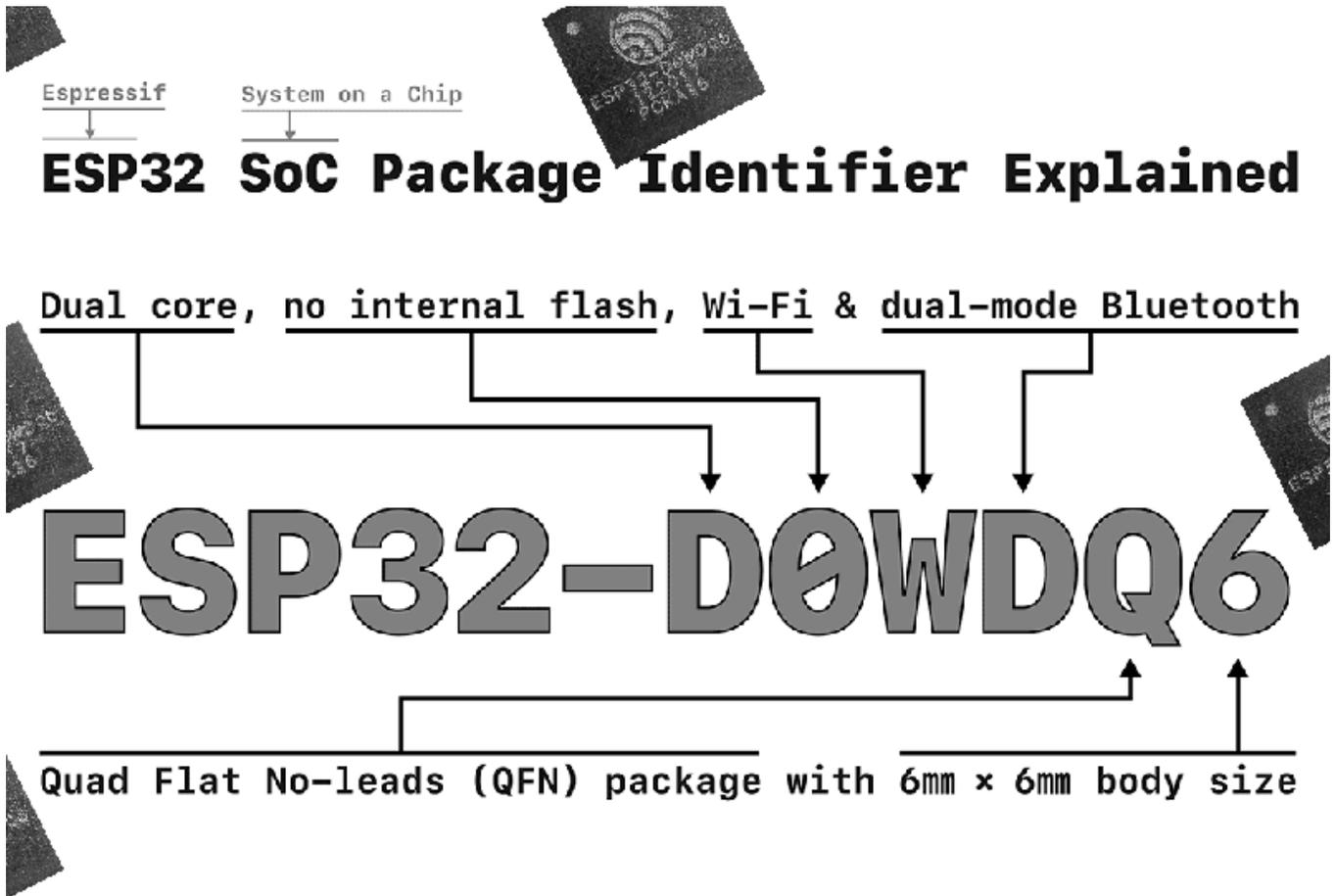
El propietario del archivo debe ser nobody y el grupo debe ser nogrup. Si ese es el caso, enhorabuena: estás listo para usar tu servidor de archivos con Clonezilla.

Lecturas Adicionales

Si quieres o lo necesitas, puedes reajustar NFS con una amplia variedad de formas para adaptarlo mejor a tu entorno. Una simple búsqueda en Google de "guía ajustes nfs" te proporcionará una excelente base. Solo recuerda que existen cosas tales como "ajustes excesivos" y "puntos de rendimientos decrecientes".

Dual Cores para GO: Procesamiento Dual Core en ODROID-GO

© April 1, 2019 By Dave Prochnow ODRROID-GO, Tutoriales



¿Quién lo hubiera adivinado? Pero está ahí en la página 1 de la hoja de especificaciones técnicas del ESP32 WROVER: “Existen dos núcleos de CPU que pueden ser controlados individualmente, ...”

Y podemos encontrar otra referencia al dual core en la página 7: “ESP32-D0WDQ6 contiene dos microprocesadores LX6 de 32 bits Xtensa® de baja potencia”.

En fin, es interesante saberlo, pero ¿cómo se puede usar este dual core en ODROID-GO? Por ejemplo, ¿cómo puede un programa escrito en el Entorno de Desarrollo Integrado de Arduino (IDE) aprovechar el dual core de ESP32?

Es muy fácil usar el dual core dentro de ODROID-GO, pero primero recopilemos algo de información básica.

Cuando se ejecuta un esquema “tradicional” de Arduino, es decir, un programa que contiene las funciones setup() y loop(), solo se usa uno de los núcleos. Ese núcleo es el Core 1, el segundo núcleo

(empiezan desde el 0). Puedes probarlo por ti mismo agregando la llamada xPortGetCoreID() a tus funciones de configuración y bucle. Cuando se redirige al Serial Monitor del IDE de Arduino (ver Figura 1), deberías ver el valor del núcleo actual, Core 1.



Figura 1 - El Serial Monitor de Arduino se activa con este botón.

Ahora, el código original utilizado para el descubrimiento e implementación de los dual cores en este artículo procede de un artículo de Rui Santos de randomnerdtutorials.com y de un blog del 14 de febrero de 2019, y de un post sobre SparkFun

Electronics en sparkfun.com. Si deseas obtener información adicional sobre la programación de dual cores en Arduino, empieza con estos dos recursos. Luego puede hacer que tu ODROID-GO haga dos cosas al mismo tiempo.

Instrucciones paso a paso

1. Introduce este ejemplo de esquema Arduino:

```
/*
 * Dual Core Demonstration for
 * ODROID-GO
 *
 * Derived from code by Rui Santos
 * at randomnerdtutorials.com
 *
 * Mentioned in SparkFun Electronics
 * blog on February 14, 2019
 * at sparkfun.com
 *
 * ODROID Magazine
 * by Dave Prochnow
 */

// Instantiate one task for separate execution
// inside core 0

TaskHandle_t Task0;

void setup() {
  // put your setup code here, to run once:

  // All of our dual core results will be
  // displayed on the Arduino Serial Monitor

  Serial.begin(115200);

  // The native core used by the Arduino is Core
  1
  // Use xPortGetCoreID() to confirm

  Serial.print("Setup() and Loop() run in Core
");
  Serial.println(xPortGetCoreID());

  // Define the function
xTaskCreatePinnedToCore(
  // for the core 0 task

  xTaskCreatePinnedToCore(
    task0Code, // task function
    "Task0", // task name
```

```
10000, // stack size allotment in words
NULL, // task input parameter
1, // task priority
&Task0, // task handle; as instantiated above
0); // designate core

// A very short delay for displaying the Task 0
// core information
vTaskDelay(.25);
}

// Create the core 0 task function; which will run
// forever

void task0Code( void * pvParameters) {
  Serial.print("This is Task0 running in Core ");
  // Label the core with xPortGetCoreID
  Serial.println(xPortGetCoreID());
  int x = 0;
  String xPreamble = "Core 0 - ";
  // You can create an infinite loop with for(;;)
  for(;;){
    Serial.println(xPreamble + x);
    x++;
  }
}

void loop() {
  // put your main code here, to run repeatedly:
  // This function will run in Core 1
  int y = 0;
  String yPreamble = "Core 1 - ";
  for(;;) {
    Serial.println(yPreamble + y);
    y = y + 2;
  }
}
```

2. Compila este programa y cárgalo en el ODROID-GO (consulta la Figura 2).

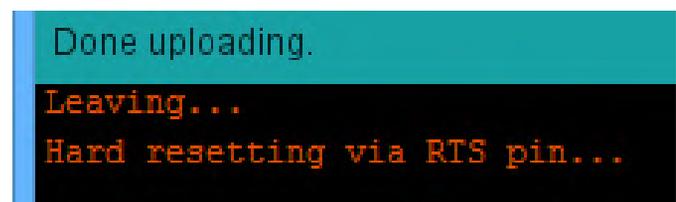


Figura 2: Tu esquema se ha compilado correctamente y se ha cargado en ODROID-GO cuando veas esta respuesta en el monitor.

3. Deja el ODROID-GO conectado a tu PC e inmediatamente, apaga el interruptor de encendido.

4. Ahora abre el Serial Monitor y ajusta los baudios a 115200 como se muestra en la Figura 3.

```
Core 0 - 43915
Core 1 - 87732
Core 0 - 43916
Core 1 - 87734
Core 0 - 43917
Core 1 - 87736
Core 0 - 43918
Core 1 - 87738
Core 0 - 43919
Core 1 - 87740
Core 0 - 43920
Core 1 - 87742
Core 0 - 43921
Core 1 -
 Autoscroll  Show timestamp Newline 115200 baud
```

Figura 3: Asegúrate de que los baudios del Serial Monitor están configurados en 115200.

5. Enciende y apaga el interruptor de alimentación del ODROID-GO muy rápidamente. Este rápido cambio permitirá que solo se envíe una breve ráfaga de información del dual core al Serial Monitor.

6. Desplázate hasta la parte superior de la impresión del Serial Monitor y revisa las líneas que aparecen después de la información de inicio del ESP32. Debería ver un par de IDs de núcleo y una larga lista de valores que son calculados por cada núcleo, algo similar a lo que puedes ver en la Figura 4.

```
Setup() and Loop() run in Core 1
This is Task0 running in Core 0
Core 1 - 0
Core 1 - 2
Core 0 - 0
Core 1 - 4
Core 0 - 1
Core 1 - 6
Core 0 - 2
Core 1 - 8
Core 0 - 3
Core 1 - 10
Core 0 - 4
Core 1 - 12
```

7. Ahora adapta esta técnica a tus propios desafíos de Arduino (ve la Figura 5) y enséñale a tu ODROID-GO cómo recorrer y mascar código al mismo tiempo.

```
16 // .Instantiate one task for separate execution
17 // .inside core 0
18
19 TaskHandle_t Task0;
20
21 void setup() {
22     // .put your setup code here, to run once:
23
24     // .All of our dual core results will be
25     // .displayed on the Arduino Serial Monitor
26
27     Serial.begin(115200);
28
29     // .The native core used by the Arduino is Core 1
30     // .Use xPortGetCoreID() to confirm
31
32     Serial.print("Setup() and Loop() run in Core ");
33     Serial.println(xPortGetCoreID());
34
35     // .Define the function xTaskCreatePinnedToCore()
36     // .for the core 0 task
37
38     xTaskCreatePinnedToCore(
39         task0Code, // .task function
40         "Task0", // .task name
41         10000, // .stack size allotment in words
42         NULL, // .task input parameter
43         1, // .task priority
44         &Task0, // .task handle; as instantiated above
45         0); // .designate core
46
47     // .A very short delay for displaying the Task 0
48     // .core information
49     vTaskDelay(.25);
50
51 }
```

Figura 5 - Adapta tu propio código a las tareas del Core 0 en ODROID-GO

Google Home: Usando el ODROID-C2 como un Asistente Personal

© April 1, 2019 By @codewalker ➔ Android, ODROID-C2, Tutoriales



El software de asistencia personal de Google llamado Google Assistant integra una gran cantidad de servicios, tanto internos como de terceros, permitiendo a los usuarios escuchar música, controlar la reproducción de videos o fotos o recibir actualizaciones de noticias... todo ello controlado totalmente por la voz. Los dispositivos de Google Home también tienen soporte integrado para la automatización del hogar, lo cual permite a los usuarios controlar los electrodomésticos inteligentes con su propia voz.

En primer lugar, abre un navegador en el ODROID-C2 y visita <http://opengapps.org>.

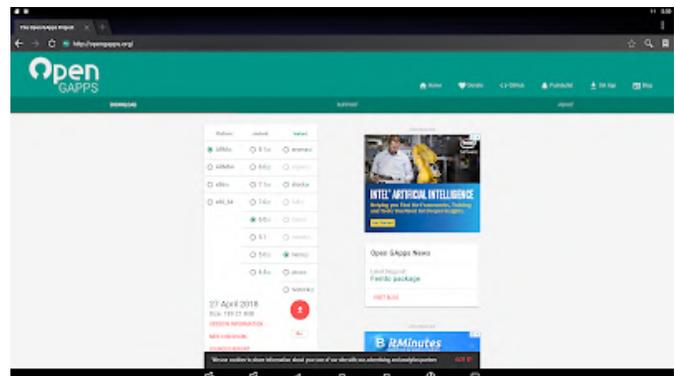


Figura 1 - Selecciona la variante "nano" y presiona Download

A continuación, abre la aplicación ODROID Utility



Figura 2 - Haz clic en el menú y selecciona "Package install from storage"

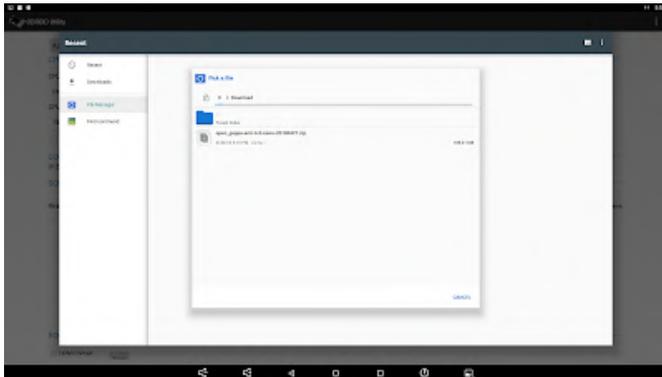


Figura 3 - Selecciona el archivo zip que acabas de descargar

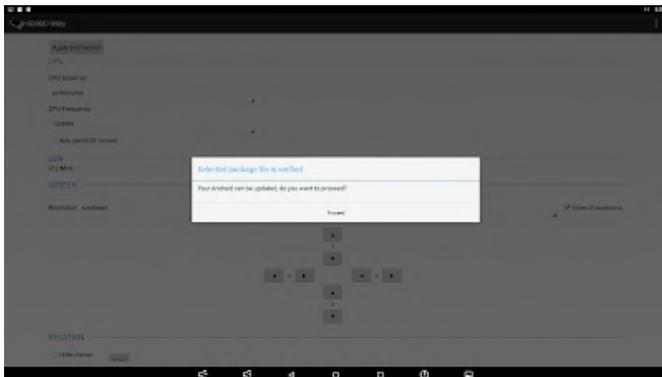


Figura 4 - El archivo zip ha sido utilizado

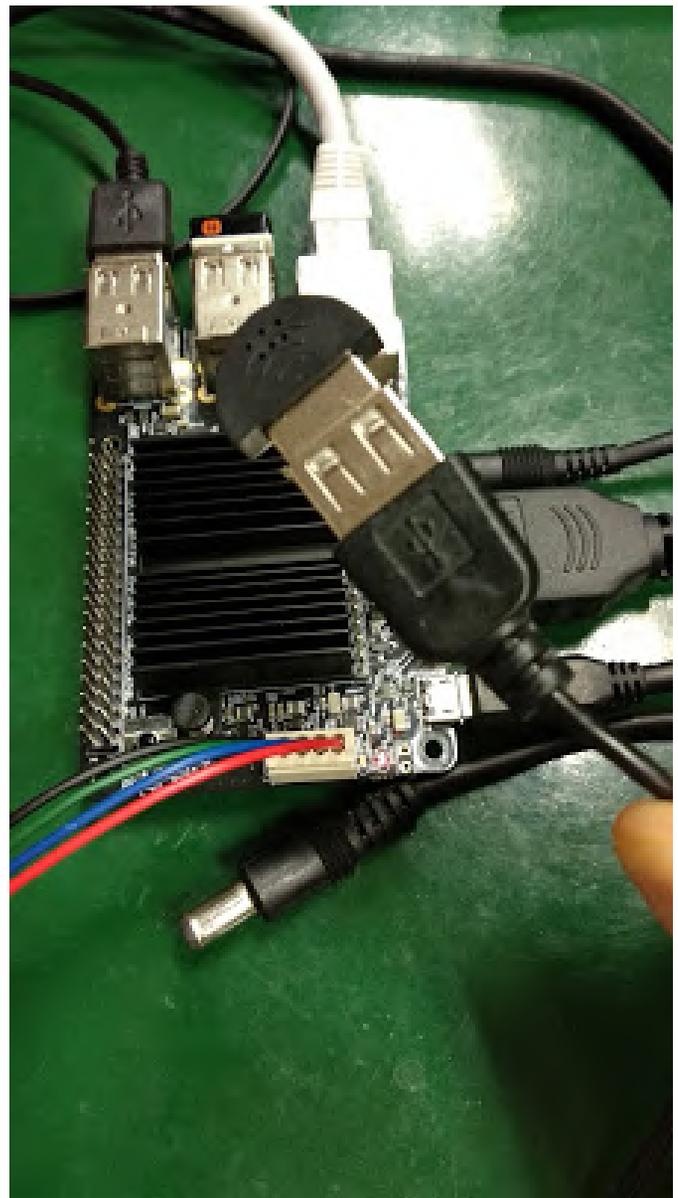


Figura 5 - Conecta el micrófono USB

Otra posibilidad es utilizar la cámara USB ODROID (http://www.hardkernel.com/main/products/prdt_info.php?g_code=G146883099080)



Figura 6 - Cámara USB ODROID

Después, abre la aplicación de Google.

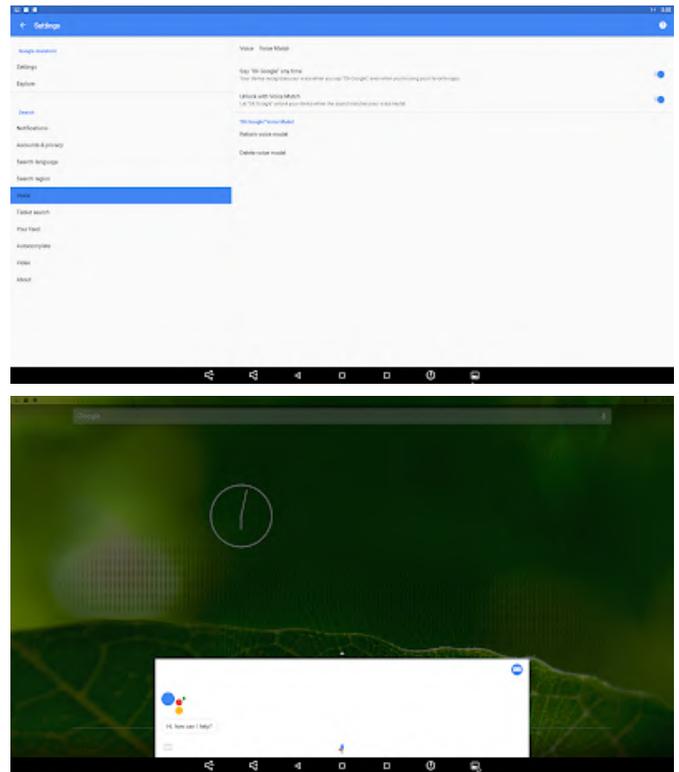


Figura 8 - Instala Google Assistant

Actualmente, el ODROID-C2 solo es compatible con el idioma inglés. Para comentarios, preguntas y sugerencias, visita el post original en <https://codewalkerster.blogspot.com/2018/04/>.

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

© April 1, 2019 By Dave Prochnow Android, Juegos, ODRROID-C2, ODRROID-XU4



¡Estos ODRROID están muy animados con el sonido de los juegos Android! Desde la serie C (por ejemplo, ODRROID-C2) hasta la X (por ejemplo, ODRROID-XU4), jugar en tu consola favorita nunca ha sido tan divertido. Así que abre tu aplicación Google Play Store y pongámonos a descargar.

Run Garfield, Run!

Recientemente actualizado y gratis, Garfield Rush continúa intentando atrapar a ese mezquino gato (Figura 1). Al igual que antes, hay un montón de cosas para coleccionar, 18 nuevos trajes para desbloquear, y -te encantan o los odias- más y más videos de recompensa "gratis" para ver. La actualización a la versión 5.0 todavía es gratis, así que átate los cordones y vete de casa.



Figura 1: Garfield Rush es un interminable juego de carreras con un toque animado.

Fortnite Free Forever

El mega exitoso Fortnite acaba de recibir un nuevo parche que supuestamente ofrece jugabilidad a 60 fps para determinados dispositivos Android. ¿Serán los dispositivos ODROID uno de los pocos, uno de los afortunados, uno para el juego? Junto con una tasa de fotogramas mejorada, Fortnite vuelve a regalarnos un pase de batalla de la temporada 8 gratis. No intentes "romperte el dedo del pie" -una pésima referencia a una legendaria erupción volcánica de 1889 en Indonesia: la temporada 8 incluye un volcán, como se muestra en la Figura 2, con un par de ríos de lava. Estos ríos dan origen a nuevas aventuras en "Lazy Lagoon" y "Sunny Steps". Simplemente ten cuidado con el Cañón pirata.



Figura 2 - Geesch, ahora lava, Fortnite continúa superándose día tras día

Ver: <https://www.youtube.com/watch?v=3aMtD0Klp2I>

Tweet, Tweet

Un pequeño pajarito me dijo que Human: Fall Flat podría llegar a Android dentro de uno o dos meses. Recordarás que esta aplicación de No Brake Games gira en torno a un entorno de sueño inducido por la física (Figura 3) con un componente de juego llamada Bob. Las tareas cotidianas de Bob incluyen rompecabezas, sorpresas y naturalmente, peligros. Para enfrentarse a estos desafíos, Bob debe tirar, empujar, transportar, escalar y abrirse camino hacia el éxito. Human: Fall Flat, con más de 5 millones de razones para introducirse en el mercado de Android, ya es un lanzamiento de éxito en consolas y PC (\$ 14.99 USD). Pero este pajarito podría equivocarse: este es, por supuesto, ¡el mismo pajarito que me dijo que ODROID-N1 se lanzaría pronto!



Figura 3: Bob, Bob, Bob, ¿En qué te has metido ahora?

Ver: <https://www.youtube.com/watch?v=-Edk59BqSEU>

Un mundo alocado, desenfrenado, indomable, salvaje y excéntrico

Crescent Moon Games, los desarrolladores de Morphite y Ravensword Shadowlands, ahora nos traen un juego de plataformas irreal. Linn: Path of Orchards (Gratis; \$ 2.99 USD en compras dentro de la aplicación) exteriormente se ve como cualquier otro rompecabezas de plataformas. El único problema es que el mundo del juego en sí mismo juega contra ti, ¡intentando lanzarte fuera del mundo! Tan extraño como suena, Linn se mueve y hace surcos en un intento de hacer que te caigas del mundo antes de que puedas completar cada nivel (Figura 4). Tu homónimo en este animado mundo similar a M.C. Escher es Aban, un autodenominado “exótico” guardián de la naturaleza. Todo lo que tienes que hacer es ayudarlo a rejuvenecer el antiguo árbol de la luz. Buena suerte con ello.



Figura 4 - Agárrate a Aban o sus cortinas en Linn: Path of Orchards

¿Defensor de la libertad = Terrorista?

Holy cow, Ndemec Creations regresan con una venganza. Los desarrolladores que contagiaron la diversión con Plague, Inc., han lanzado un juego de estrategia llamado Rebel, Inc. (Figura 5). En realidad, a mediados de febrero fue lanzado Rebel, Inc. (GRATIS; compras desde la aplicación por 2.99\$) para Android y desde entonces los jugadores lo han estado descargando en masa. Una vez que centres tu cabeza en el concepto, Rebel, Inc. podría volverse tan adictivo como Plague, Inc. Esto equivale a aproximadamente a 100 millones de descargas de esta app. Ka-ching. En Rebel, Inc. hay cinco regiones del mundo, representadas con magníficos detalles gráficos (Figura 6), que están despertando cierta inquietud. Suena bastante realista, ¿eh? Tu función es apoyar a los gobiernos, reprimir la insurgencia y convertirte en el patriota de la gente. Simplemente un día normal en la oficina, ¿verdad?



Figura 5: Planea tu próximo movimiento con mucho cuidado o podrías ser derrocado



Figura 6 – Esa pista de aterrizaje de helicóptero te va a costar

Go Android, Go – Top Ten de Marzo de ODRUID Magazine

10. Plunder Kings – GRATIS – ¡Reventando el top ten de por sí!

9. Asphalt 9: Legends – GRATIS

8. Minecraft – 6.99\$

7. The War of Mine – 14.99\$/Pack de expansión para desarrolladores 1.99\$

6. Crashlands – 4.99\$

5. Fire Emblem Heroes – GRATIS

4. The Escapists 1 and 2 – 6.49\$ & 6.99\$; ¿Quién lo habría pensado, verdad?

3. Shadowgun Legends – GRATIS

2. Riptide GTP Series – 2.99\$ – ¡Este podría ser el mejor juego de carreras de Android!

1. PUBG Mobile – GRATIS; Con +10M descargas, ¿No debe estar nada mal, verdad?

Finalmente, Auto Android

Apple CarPlay dominó el Consumer Electronics Show (CES) 2019, pero ahora Google está llevando a cabo una muy buena jugada. En un evento mediático celebrado en Mountain View, California, el 1 de abril, Alphabet, una empresa matriz de Google, dejó caer una noticia sorprendente que cogió por sorpresa a una selecta audiencia de periodistas y ejecutivos de la industria del automóvil

A partir de la primera semana de abril, todos los próximos vehículos Dodge / Chrysler / JEEP rechazarán Apple CarPlay en favor de un producto Android desarrollado en secreto. Según la vicepresidenta ejecutiva de marketing, April Foole, “la presencia de Android en un automóvil totalmente estadounidense será presentado con un increíble vehículo de acero inoxidable en edición limitada”.

“Actualmente Alphabet, Dodge/Chrysler/JEEP y Google están extremadamente orgullosos de presentar el Dodge 2020”

“Esto es apasionante”, gritó Elon Musk.

¡Feliz día de los inocentes, Amigos!

PiHole

© April 1, 2019 By BlueMedia.com ODRROID-XU4, Tutoriales



Este artículo trata sobre el software PiHole Linux con DNS-over-TLS en Docker. PiHole es una aplicación que permite bloquear la publicidad y los rastreadores de Internet. Actúa a modo de sumidero DNS, ideal para usarse en una red privada. Docker ha añadido soporte para la Raspberry Pi y, por lo tanto, para todos los procesadores ARM.



Hace poco me encontré por casualidad con la imagen Docker DNS-over-TLS (DoT) del repositorio qdm12 (<https://github.com/qdm12/cloudflare-dns-server>) en

GitHub. Un “proxy” DNS que acepta solicitudes DNS y las envía cifradas a Google (8.8.8.8), CloudFlare (1.1.1.1) u otros servidores DoT.

Esto podría implementarse fácilmente en mi ODROID XU4 y Raspberry Pi 3 (dos dispositivos para que, si un dispositivo falla, continúa funcionando internet en casa). El cambio rápido del servidor DNS en el “Fritzbox” (famosa serie de router alemana) por las dos direcciones IP de los dispositivos permite que las consultas DNS de mi red doméstica sean cifradas y no simple texto sin formato. Mientras me ocupaba del tema del DNS, me topé con PiHole (Docker Image). Un servidor DNS que responde a las consultas DNS, a menos que vayan a servidores de publicidad. Así pues, podemos implementar un bloqueador de rastreo y adware de bajo nivel, incluso para los dispositivos en los que no se puedan instalar un bloqueador de publicidad.

Ambos, combinando PiHole con DNS-over-TLS, permiten, a través de la implementación de dos

dispositivos, una configuración DNS a prueba de fallos, cifrada, sin publicidad y sin rastreo para toda la familia. En este post me gustaría explicar brevemente cómo implementar todo esto. Este tutorial va dirigido tanto a mi audiencia principal, los fanáticos de ODROID XU4 como a los usuarios de Raspberry Pi porque la configuración es idéntica.

Instalar Docker

Instalar Docker y Docker-Compose es fácil. Abre una conexión SSH y escribe los siguientes comandos:

```
$ sudo su
# cd ~

curl -fsSL https://get.docker.com -o get-docker.sh
# Docker repositories import of Docker, Inc.
sh get-docker.sh # Import repositories

apt install docker-ce python3-pip # install Docker
and pip3

systemctl enable docker-ce # start Docker at
startup
systemctl start docker-ce # start Docker now

pip3 install docker-compose # install Docker
Compose
```

Docker Compose es responsable de la organización de varios contenedores Docker y simplifica la tarea enormemente.

Instalar PiHole y proxy DoT

Si tenemos Docker y Docker Compose instalados, ya podemos embarcarnos en nuestra tarea. Iniciaremos un servidor PiHole y dos DNS-over-TLS. He dirigido el servidor DoT a diferentes servidores DNS emisores, de modo que uno envía solicitudes a Cloudflare y el otro a Google Public DNS. Ambos tienen una disponibilidad de entre el 99% y el 99,99%. Como ningún sistema de resolución de DNS puede garantizar un tiempo de actividad del 100%, recomendaría utilizar siempre dos proveedores. La imagen Docker de qdm12 ofrece, además de los dos servidores ya mencionados, soporte para quad9, quadrant y cleanbrowsing. De cualquier forma, puedes profundizar en [entrypoint.sh](#) tú mismo y añadir tus propios proveedores.

Así que, en primer lugar, clonamos el repositorio Git del servidor DoT y lo configuramos. Si esto funciona, integraremos PiHole a través de `acoplable-compose.yml`.

```
apt install git

cd ~ && git clone
https://github.com/qdm12/cloudflare-dns-server.git
./dns-server

cd dns-server
```

En el `docker-compose.yml` puedes ajustar la configuración. Entre otras cosas, aconsejaría desactivar `BLOCK_MALICIOUS`, ya que esto bloquea demasiado para mi gusto. `VERBOSE` lo he fijado en 0. Y para los proveedores, como he dicho antes, tienes la oportunidad de elegir entre google, cloudflare, quad9, cleanbrowsing y quadrant. Tienes más información sobre las opciones de configuración en el sitio `README` del repositorio Git. Luego, puede compilar e iniciar `compose-stack` con `docker-compose up -build #optional -d`. Si se han realizado cambios en los archivos de configuración, yo siempre iniciaría la pila con `-build`. De lo contrario, `docker-compose up -d` sería suficiente para arrancarlo y `docker-compose` para apagarlo. ¿Cómo podemos ver ahora si el servidor DNS está funcionando o no? Para esto contamos con la pequeña herramienta `dig`:

```
apt install dig

dig blaumedia.com @ 127.0.0.1
```

El comando debería devolver algo como esto:

```
; << >> DiG 9.11.3-1ubuntu1.3 Ubuntu << >>
blaumedia.com @ 127.0.0.1
;; global options: + cmd
;; Got answer:
;; - >> HEADER << - opcode: QUERY, status:
NOERROR, id: 42751 ;; flags: qr rd ra; QUERY: 1,
ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT
PSEUDO SECTION: ; EDNS: version: 0, flags :; udp:
4096 ;; QUESTION SECTION: ; Blaumedia.com. IN A ;;
ANSWER SECTION: blaumedia.com. 86400 IN A
159.69.38.130 ;; Query time: 100 msec ;; SERVER:
127.0.0.1 # 53 (127.0.0.1) ;; WHEN: Wed Feb. 13
18:14:59 CET 2019 ;; MSG SIZE rcvd: 58
```

¿Te las has arreglado hasta ahora? ¡Muy bien! Luego convertimos un servidor DoT en dos y añadiremos PiHole. Para simplificar las cosas, primero voy a mostrarte mi archivo docker-compose.yml y luego te explicaré exactamente lo que he estado haciendo. Mi archivo docker-compose.yml:

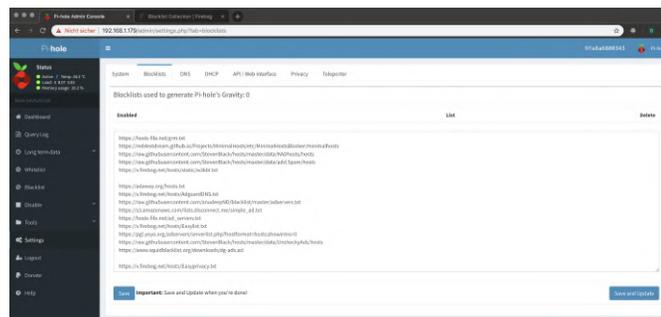
```
version: '2' services: cloudflare-dns-tls: build.restart: always image: qmcgaw / cloudflare-dns-server container_name: cloudflare-dns-tls environment: - VERBOSITY = 0 - VERBOSITY_DETAILS = 0 - BLOCK_MALICIOUS = off - LISTENING PORT = 53 - PROVIDER = cloudflare Networks: dnsbridge: ipv4_address: 10.5.0.2 google-dns-tls: build.restart: always image: qmcgaw / cloudflare-dns-server container_name: google-dns-tls environment: - VERBOSITY = 0 - VERBOSITY_DETAILS = 0 - BLOCK_MALICIOUS = off - LISTENING PORT = 53 - PROVIDER = google Networks: dnsbridge: ipv4_address: 10.5.0.3 pihole: image: pihole / pihole: 4.2.1 dns: - 127.0.0.1 - 192.168.1.1 ports: - "53: 53 / tcp" - "53: 53 / udp" - "80: 80 / tcp" - "443: 443 / tcp" cap_add: - NET_ADMIN Networks: dnsbridge: ipv4_address: 10.5.0.4 environment: ServerIP: DEVICE_IP (eg 192.168.1.100) TZ: Europe / Berlin DNS1: 10.5.0.2 DNS2: 10.5.0.3 WebPassword: PIHOLE_PASSWORD volumes: - '/ etc / pihole /: / etc / pihole /' - '/etc/dnsmasq.d/:/etc/dnsmasq.d/' restart: always Networks: dnsbridge: driver: bridge ipam: config: - subnet: 10.5.0.0/16 gateway: 10.5.0.1
```

Hay muchos cambios, así que te recomiendo simplemente sobrescribir tu archivo docker-compose.yml por el mío. En primer lugar, debe configurar en la línea 47 la IP de tu dispositivo que ejecutará PiHole. Con `ip route get 1.1.1.1 | awk '{print $ NF; exit}'` puedes descubrir rápidamente cuál es (Fuente: AskUbuntu.com). En la línea 51, he especificado la contraseña para la interfaz web de PiHole. ¿Qué estamos haciendo aquí exactamente? Para que los dos servidores DoT sean accesibles para PiHole, creamos un puente de red en la parte inferior del archivo. Esto nos permite asignar direcciones IP fijas a los contenedores individuales. Docker normalmente gestiona esto para que los contenedores siempre puedan ser identificados por el nombre del contenedor entre ellos. Pero los servidores DNS como dnsmasq de PiHole no conocen los servidores de nivel superior con sus nombres de

dominio, aunque esperan las direcciones IP. ¿Cómo podría resolver estos dominios en todo caso? Así que proporciona a los contenedores direcciones IP fijas; fácil. Esto lo determinamos en la línea 49 y 50 como servidor emisor para PiHole. En las líneas 34 y 35 configuramos los servidores DNS del contenedor, que se utilizan, por ejemplo, durante el arranque de los procesos. El primero siempre debe ser 127.0.0.1, el segundo puede ser cualquier otra cosa. Pongo aquí la IP de mi Fritzbox (router). En las líneas 53 y 54 montamos la carpeta de configuración de PiHole en nuestro sistema host para que no se elimine en cada proceso de compilación y permanezcan inamovible. Eso es todo, el resto ya deberías conocerlo o es evidente. Con `docker-compose up -build -d` ahora podemos compilar e iniciarlo todo.

Insertar canales de publicidad PiHole

Accedemos a la interfaz web de PiHole con nuestro navegador simplemente introduciendo la IP de la Raspberry Pi/oDroids. [Figura 02] PiHole WebUI. Siguiendo el enlace llegas a la interfaz. En el lado izquierdo, pincha en "Login" e introduce la contraseña que acaba de definir en la configuración. ¡Voilà! La mejor prueba es que las consultas DNS están funcionando con dig. Si todo funciona perfectamente, ahora podemos añadir listas para PiHole que son las responsables de bloquear la publicidad. Suelo alimentar mis listas exclusivamente desde firebog.net. Puedes añadir las listas a PiHole desde Settings -> Blocklists.



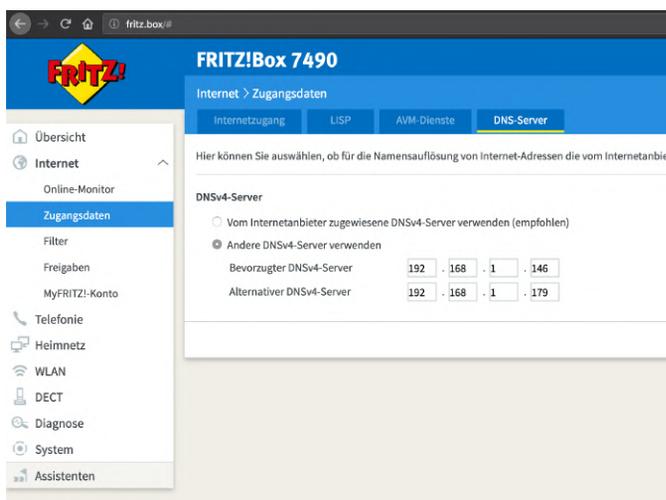
PiHole añade listas de bloqueo

Dado que hay más personas que usa Internet en mi hogar, prefiero bloquear menos direcciones DNS. Es decir, si hay que algo que no funciona debido a los agujeros negros de DNS, es molesto para todos. Por supuesto, al final esto es así para todos. Finalmente,

haga clic en save and update y se bloquearán las consultas de DNS incluidas en las listas. Las listas se actualizan automáticamente por la noche a las 00:00.

Para finalizar

Ahora tiene un servidor PiHole que recibe tus solicitudes de DNS cifradas. Si estás seguro de que todo funciona, ahora puede colocar tu servidor PiHole como un servidor DNS en tu router. Nosotros tenemos un "FritzBox" y la configuración se ve así:



Configurar el servidor DNS FritzBox

Introduce la dirección IP interna de tu dispositivo PiHole y guarda la configuración. Como puedes ver, he introducido dos IPs. En primer lugar, el ODROID XU4, luego viene mi PI3 de reserva. De ahora en adelante, las consultas de DNS de toda la red son filtradas a través de PiHole -> DNS-over-TLS -> Cloudflare / Google / Quad9 / ...

Referencias

<https://blaumedia.com/blog/odroid-raspberry-pi-pihole-dns-over-tls-docker/>

<https://github.com/qdm12/cloudflare-dns-server>

Juegos Linux: Box86 – Un Nuevo Emulador x86 para ODROID

© April 1, 2019 By Tobias Schaaf Linux, Tutoriales



Este mes, he incluido algo diferente para variar. Algunos de vosotros ya lo habréis leído en los foros en <https://forum.odroid.com/viewtopic.php?f=91&t=34070>, quisiera hablar de ello: Box86, un nuevo tipo de emulador x86 para ARM que se puede ejecutar en los ODROIDS.

¿Qué es Box86?

Box86 es un emulador x86 que te permite ejecutar aplicaciones Linux x86 en tu ODROID. Te permite ejecutar programas (o probablemente más que nada juegos) que no fueron escritos para placas ARM y no son de código abierto, como es el caso de la mayoría de los programas comerciales. Como la mayoría de las veces estos juegos no lanzaban para ARM, normalmente nos quedábamos sin jugar en SBCs como el ODROID, aunque el dispositivo es capaz de ejecutarlos sin problema si existiesen para ARM.

¿Existen otras soluciones para ejecutar software x86 en ARM?

Sí: qemu, por ejemplo, es capaz de emular un PC x86 al completo, dándote la posibilidad de ejecutar básicamente todo lo quieras. Emula GPU, CPU y otros dispositivos como LAN y USB. El problema de esta solución es que es muy lenta ya que emula un PC entero y, aunque funciona, es tan lento que prácticamente no se puede usar. Ya he tenido éxito con Windows 2000 y XP ejecutándolos en qemu (demostrando que realmente se puede ejecutar casi todo), pero tarda muchísimo y es muy lento.

Otro ejemplo sería DOSBox, que también emula un PC x86, aunque con un estilo muy antiguo 286 – 486 que cuenta con un conjunto muy reducido de instrucciones. Esta solución funciona bastante bien y te permite ejecutar juegos y programas antiguos basados en DOS, o incluso puede usarse para ejecutar Windows 3.11. Aunque es mucho más rápido, está muy limitado por la arquitectura antigua que se supone que ejecuta y, por lo tanto, estará

limitado a los juegos antiguos de DOS, y quizás a las aplicaciones de Windows 3.11.

Una solución que ya analicé en el pasado era ExaGear, que permite ejecutar aplicaciones x86 en ARM emulando las instrucciones de la CPU, pero no el PC al completo. Es mucho más rápido que qemu y puede ejecutar aplicaciones Linux x86, incluido "wine", que te permite ejecutar aplicaciones de Windows en Linux.

Exploré en profundidad las posibilidades de ExaGear y escribí un par de artículos al respecto, además de crear una lista de compatibilidad en <https://oph.mdrjr.net/meveric/other/ExaGear/>, que muestra muy bien el rendimiento de ExaGear. De hecho, probablemente sea el software más maduro que existe actualmente que te permita ejecutar aplicaciones x86 en ARM. Lamentablemente, Eltechs decidió cancelar ExaGear para dispositivos ARM de escritorio, lo cual significa que ya no puedes conseguirlo.

¿Cuál es la diferencia con Box86?

Qemu es demasiado lento para ser útil y DOSBox está demasiado limitado para ejecutar aplicaciones modernas de Windows/Linux. Pero si ExaGear es tan bueno, ¿cuál es la razón por la que conseguir un nuevo tipo de emulador, aparte del hecho de que ExaGear fue cancelado?

Pues bien, ExaGear, aunque bastante bueno, tenía un defecto muy grande: lo ejecutaba todo en un entorno i386 (x86) que, además, en el caso de los drivers GPU, limitaba el uso de software solo a aplicaciones que no requerían aceleración 3D. Básicamente, cualquier juego que usara gráficos "modernos" no funcionaba, ya que requería OpenGL y nuestros ODROID carecen de los drivers necesarios, en especial para i386.

Aquí es donde Box86 puede destacar. Muchas de las librerías requeridas para ejecutar una aplicación son redirigidas a las librerías de tu placa ARM, lo que significa que, en lugar de usar SDL2 desde i386, Box86 puede usar SDL2 desde tu sistema host ARM. Esto incluye también los drivers de la GPU. Si una aplicación requiere OpenGL para ejecutarse, ya no pedirá los drivers OpenGL i386, sino los drivers OpenGL ARM, y aunque los ODROID no tienen

OpenGL nativo, gracias a @ptitSeb, disponemos de un empaquetador OpenGL a OpenGL ES llamado gl4es que nos permite ejecutar muchas aplicaciones OpenGL. Pero no solo se utiliza OpenGL desde el sistema ARM en lugar del sistema i386. Hay muchas otras librerías, como SDL, OpenAL, X11, etc., que también se pueden utilizar desde el entorno ARM nativo en lugar de usarlo todo desde i386.

Esto, además, significa que necesitas muchas menos librerías de i386 que de ExaGear, el cual requiere un entorno i386 completo. Esto, combinado con la emulación x86, abre un campo completamente nuevo de aplicaciones que no funcionaban antes.

¿Es Box86 el santo grial para x86 en ARM?

Posiblemente sea el santo grial para x86 en ARM y el hecho de que sea de código abierto permitirá que muchas personas desarrollen y contribuyan al proyecto, lo cual a la larga puede tener como resultado la creación de un excelente paquete de software, de hecho. Actualmente Box86 está muy verde. Es un proyecto muy reciente y todavía tiene un largo camino que recorrer. Aún no admite un montón de cosas y tiene errores, pero lo bueno es que podemos ver los cambios casi a diario y el software sigue mejorando día tras día. Además, al ser de código abierto, este software podría volverse famoso bastante rápido y, con ello, más desarrolladores podrían corregir errores y mejorar el software.

Ya contamos con algunos juegos muy buenos que puedes ejecutar; algunos de los cuales hacen buen uso del potencial OpenGL de la configuración (el más famoso, probablemente sea Neverwinter Nights) y estoy bastante seguro de que pronto aparecerán más juegos.



Figura 1 – Pantalla de inicio de Neverwinter Nights



Figura 2 - Jugabilidad de Neverwinter Nights.

Entonces, ¿Qué es lo que funciona en este momento?

El número de aplicaciones soportadas sigue siendo muy bajo, y básicamente se recurre al método de prueba y error para averiguar si una aplicación es compatible o no. Sin embargo, ya hay algunos juegos que parecen funcionar bastante bien en Box86. Airline Tycoon Deluxe, por ejemplo, debería funcionar a máxima velocidad (https://www.gog.com/game/airline_tycoon_deluxe), se trata un juego de administración muy interesante dentro de la enorme serie de juegos Tycoon, que tiene mucho encanto y diversión.



Figura 3 - Pantalla de inicio de Airline Tycoon Deluxe



Figura 4 - Jugabilidad de Airline Tycoon Deluxe

Neverwinter Nights Diamond Edition es un juego de rol muy intenso dentro del universo de Dungeon & Dragons. Actualmente, solo el Enhanced Edition Pack se puede comprar en GoG (https://www.gog.com/game/neverwinter_nights_enhanced_edition_pack). Este es sin duda el juego más impresionante a nivel gráfico para el sistema, ya que utiliza el gl4es de @ptitSeb para convertir los gráficos OpenGL a OpenGL ES. Sin embargo, es bastante exigente, pero debería funcionar bien a resoluciones bajas, entre 800x600 y 720p.

Cook Serve Delicious (https://www.gog.com/game/cook_serve_delicious) es un buen juego de gestión/reacción del tiempo en el que preparas las comidas para tus clientes. Al igual que Airline Tycoon Deluxe, debería funcionar a máxima velocidad incluso en 1080p, aunque utiliza algo de OpenGL para los efectos, gracias a gl4es de @ptitSeb, funciona perfectamente.

(https://www.gog.com/game/undertale) también funciona a 1080p, aunque no hay mucho que hacer con respecto a los gráficos. Gods Will Be Watching (https://www.gog.com/game/gods_will_be_watching) también parece funcionar bien. No siempre a máxima velocidad, pero nada de lo que tengas que preocuparte.



Figura 5 - Pantalla de inicio de Freedom Planet



Figura 6 - Jugabilidad de Freedom Planet

Otro juego impresionante es Freedom Planet (https://www.gog.com/game/freedom_planet) que se siente y se juega un poco como el Sonic, pero con mucha más acción. El juego tarda bastante tiempo en cargarse, lo que actualmente pasa con muchas aplicaciones, pero una vez completada la carga inicial, el juego se carga mucho más rápido. Sufre de algunos problemas leves de velocidad: la introducción y las secuencias de video están a máxima velocidad, pero en el juego es un poco lento, aun así es jugable e impresiona bastante

Otro juego muy conocido es Faster than Light (https://www.gog.com/game/faster_than_light) que también funciona, pero tiene un tiempo de carga excesivo al principio junto con algunos problemas de sonido. La velocidad no es del 100%, pero con este tipo de juego, no es necesario tener máxima velocidad. El juego es muy bueno, y paso bastante tiempo jugando en el ODROID. También hay otros juegos que funcionan en mayor o menor medida, como la mayoría de los juegos AGS (Adventure Game Studio) games (https://www.gog.com/games?devpub=wadjet_eye_games&page=1&sort=popularity) pero varían mucho en cuanto a velocidad y rendimiento, como la serie "Eschalon Books (https://www.gog.com/game/eschalon_book_i) que usa OpenGL en muchos lugares. Y ni siquiera lo intentaría con ExaGear por este motivo, aunque también es bastante lento en Box86, por ahora. World of Goo (https://www.gog.com/game/world_of_goo) también parece iniciarse, pero se vuelve demasiado lento a la hora de jugar. DEFCON (<https://www.gog.com/game/defcon>) parece funcionar tras un largo período de carga, aunque es bastante lento.



Figura 7 - Pantalla de inicio de World of Goo



Figura 8 - Jugabilidad de World of Goo

Inherit the Earth (https://www.gog.com/game/inherit_the_earth) parece funcionar bien, pero también puede ejecutarse en ScummVM, así que no es un gran problema (https://www.gog.com/game/pixeljunk_shooter) se inicia, pero es demasiado lento para jugar. De modo que sí, ya hay unos cuantos juegos que puedes ejecutar con Box86 y estoy seguro de que pronto aparecerán más.

¿Qué pasa con los juegos de Windows y Wine?

ExaGear tiene la gran ventaja de que es capaz de ejecutar Wine bajo un entorno i386, lo que te permite ejecutar juegos de Windows en tu ODROID. Funciona bastante bien, siempre y cuando los juegos no requieran aceleración 3D para ejecutarse. Entonces, ¿Box86 ofrece lo mismo? Lamentablemente no. Actualmente, está limitado sólo a aplicaciones de Linux, aunque esto puede cambiar en el futuro, después de todo, wine también es "simplemente" una aplicación Linux.

¿Cómo se puede utilizar Box86?

Así es como normalmente utilizarías Box86:

1. Visita <https://github.com/ptitSeb/box86>
2. Descarga el código
3. Compíllalo tú mismo
4. Añade un repositorio con las librerías i386 necesarias para jugar a tus juegos Configura tus variables de entorno tal y como se describe en <https://github.com/ptitSeb/box86/blob/master/USAGE.md>
5. Si aún no tienes gl4es, repite los pasos 1-5 para gl4es también
6. Intenta hacer que las cosas funcionen.

Los pasos 1 a 3 pueden llegar a ser un engorro, y el paso 4 puede ser complicado si no sabes cómo obtener las librerías, incluso el paso 5 puede ser un tanto horrendo, ya que a menudo incluirá los mismos comandos. Para facilitar las cosas, me he tomado la libertad de preparar un archivo .deb, que espero que gestione los Pasos 1 a 5, aunque no te garantizo que funcione. Tiene el binario en su interior, así como las librerías i386 más utilizadas que pude encontrar, que es una simple copia de las librerías de Debian. También he escrito un pequeño script delante de Box86 que incluye las librerías que contiene el paquete, así como las rutas de acceso para las aplicaciones más utilizadas que tienen sus propias

librerías. Muchos juegos vienen con una carpeta lib, lib32, x86 o algo similar que almacenará las librerías que utilizan en sus programas, y el script incluirá automáticamente estas rutas en la ruta de búsqueda de librerías.

Esto significa que deberías poder simplemente ir a una carpeta del juego al que te gustaría jugar (por ejemplo: /home/odroid/GOG Games/Airline Tycoon Deluxe/game/) y ejecutar tu juego desde una terminal escribiendo "Box86" (por ejemplo: "box86 ATDeluxe").

Sin embargo, esto no funcionará con todas las aplicaciones y es posible que tengas que usar los comandos del Paso 5 para añadir tus propias librerías o carpetas. Si descubres que faltan algunas librerías, tienes total libertad para informarme de ello y las añadiré a mi paquete por comodidad.

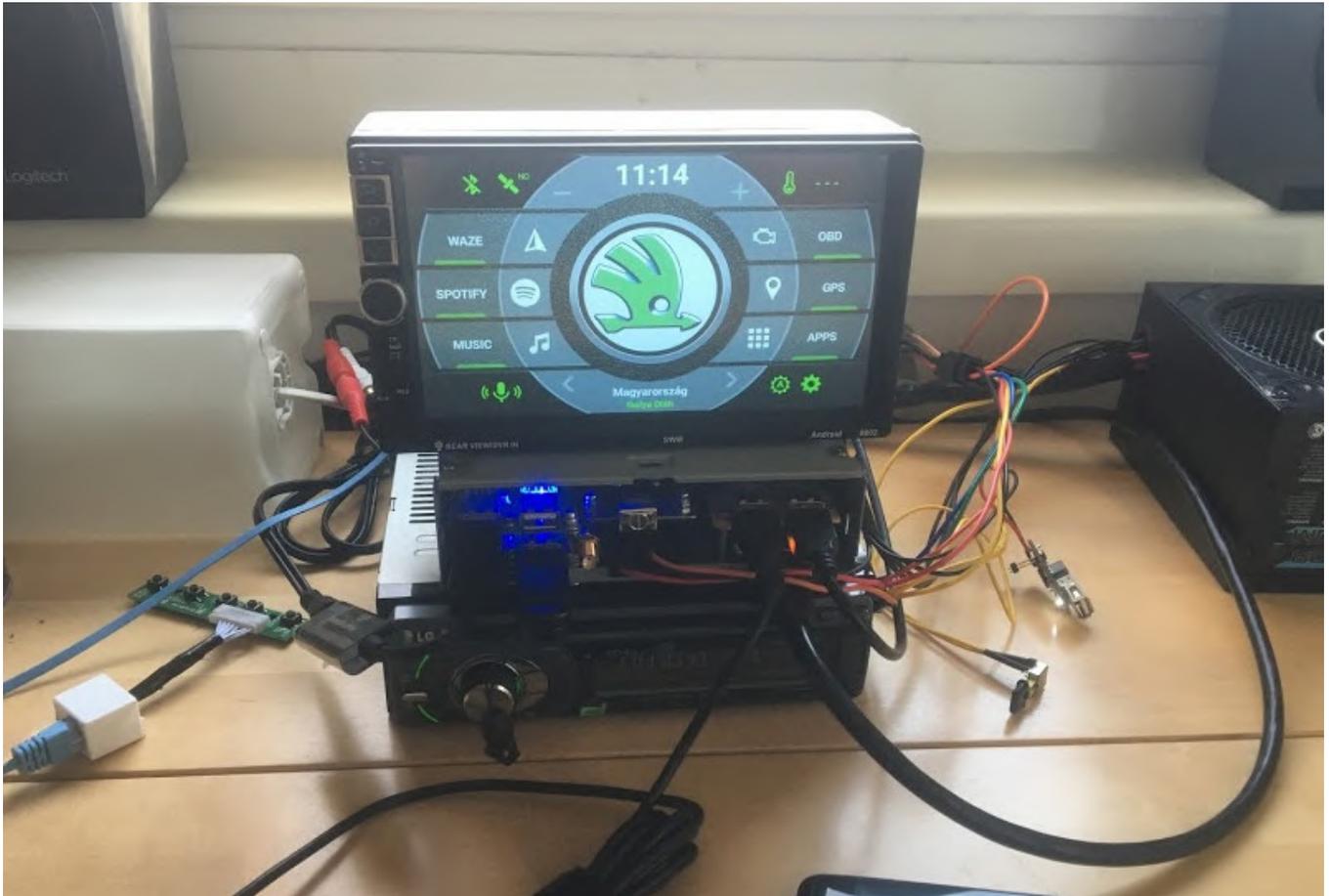
El paquete se puede instalar en mis sistemas operativos basados en Debian desde mi repositorio con el siguiente comando:

```
$ apt-get install box86-odroid
```

Ten en cuenta que esto solo está disponible para los sistemas operativos armhf (32 bits) por ahora, así que probablemente no podrás usarlo en el ODROID-C2 o en el ODROID-N2.

CarPC

© April 1, 2019 By @Bence9218 Android, ODRROID-XU4, Mecanico, Tutoriales



<https://forum.odroid.com/viewtopic.php?f=55&t=33941>

En este artículo, me gustaría presentarte mi proyecto CarPC. Este proyecto está desarrollado partiendo de cuatro componentes principales:

- Una pantalla a medida sobre una carcasa de unidad principal Android 2 DIN
- Un ODRROID-XU4Q
- Un ZTE Blade como módem 3G y sistema de seguimiento por GPS para cuando el coche está parado
- Una unidad LG LAC5900RN como amplificador



Figura 1 - La pantalla tal y como se ve tras la instalación

Las principales características de este proyecto terminado incluirán:

- Android 7 (LineageOS 14.1)
- Estado del tráfico en vivo
- La posibilidad de reproducir música desde un almacenamiento local, así como desde Spotify

- Diagnóstico del motor y datos en tiempo real a través del zócalo OBDII.

Hay una serie de funciones que todavía están en desarrollo en estos momentos, como son un perfil manos libres Bluetooth (HFP) para las llamadas de manos libres a través de Bluetooth (sí, tengo una solución para esto) y un software de seguimiento por GPS para el ZTE Blade para rastrear la ubicación de mi coche en cualquier momento y que éste me avise cuando se mueva sin mí (sin la señal Bluetooth de mi teléfono).

Pantalla

Tenía una unidad Android 2 DIN que encajaba perfectamente en el salpicadero de mi coche. He estado usando esta unidad recientemente hasta que la bloqueé cuando intentaba rootear su sistema operativo. Así que decidí comprar un panel de visualización con un mejor ángulo de visión, un panel táctil y una placa de unidad HDMI.

La carcasa cuenta con tres botones retroiluminados, un potenciómetro (un codificador) y una toma USB. Para que funcionaran los botones, utilicé una placa de teclado USB y soldé los tres botones al F1, F2 y 0. El codificador está conectado al volumen principal de la unidad LG (amplificador) y puedes apagar o encender la pantalla presionando el codificador. La toma USB simplemente está conectada al hub USB alimentado de la pantalla (que a su vez está conectado al ODROID).

En cuanto al software, necesitaba recompilar el sistema operativo desde la fuente para calibrar correctamente la pantalla táctil (eGalax 0eef 0001). Era la forma más fácil de hacer que los botones funcionaran debidamente. Creé mi propio mapa de distribución de teclas (archivo .kl) para las funciones "Inicio", "Atrás" y "Cambio de aplicación".

Por último, la carcasa cuenta con cinco tomas:

- Alimentación
- HDMI
- USB (Pantalla táctil, botones frontales, toma de USB frontal)
- RJ45 (botones de la placa de pantalla HDMI)
- Jack (control remoto de volumen)



Figura 2 - Algunas de las salidas del módulo de pantalla

Cable USB OBDII ELM327

Por supuesto, tenía la interfaz del convertidor de USB a RS232 equivocada, así que necesitaba solucionar este pequeño problema. La convertí desde chip CH340 al chip PL2303 que es compatible con el software.

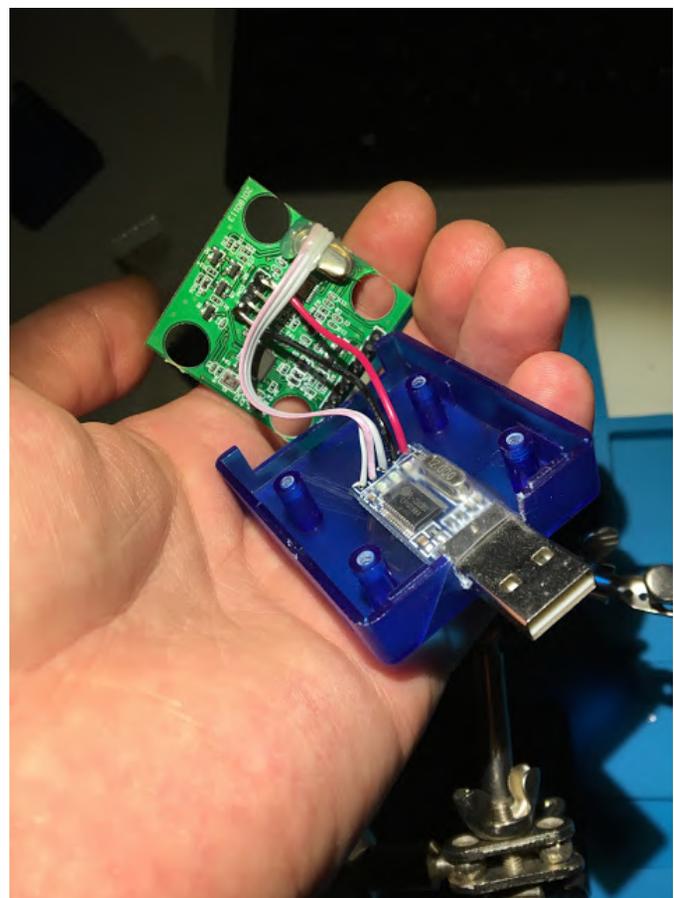


Figura 3 - Cambiando el chips

ZTE Blade

El kernel de producción de ODROID no contiene el módulo del kernel Remote Network Driver Interface Specification (RNDIS) que es necesario para compartir tu conexión a Internet desde tu teléfono Android al ODROID a través del USB. Re compilé el kernel con este módulo y creé un script de inicio de usuario en el teléfono, que activa automáticamente la función de anclaje a red USB cuando el teléfono está conectado.



Figura 4 - Después de la recompilación

Unidad principal LG

También he añadido una conexión RCA de entrada en la parte posterior.

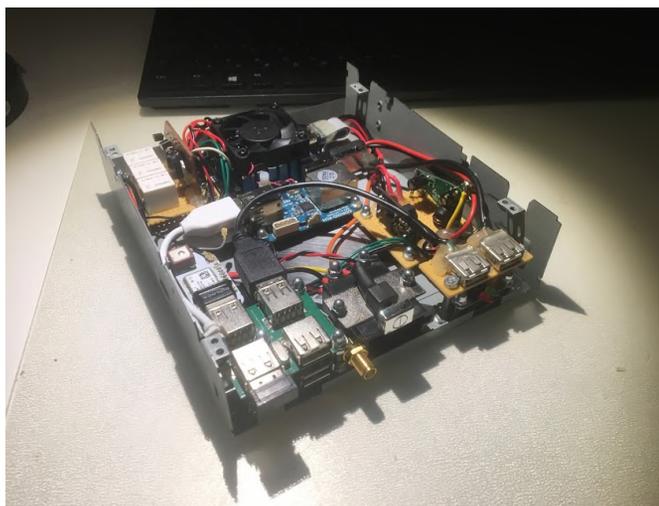


Figura 5 - Clavija RCA de entrada

ODROID

El mayor reto fue asegurarme de que el ODROID se cerrara correctamente cuando desconectara el encendido del coche, y que luego la corriente eléctrica también desapareciera. Creé mi propia fuente de alimentación y placas de control de potencia, que parecían funcionar. Si el encendido está apagado, el sistema se apaga y, un minuto después, se corta la alimentación. Tengo una toma USB especial que se mantiene con corriente cuando el encendido del automóvil está apagado, lo cual permite mantener activo el ZTE blade.

Conociendo un ODROIDian: Robert Millette (@Lix-Alpha)

© April 1, 2019 👤 By Rob Roy ➦ Conociendo un ODROIDian



Por favor h́ablanos un poco sobre ti. Mi nombre es Rob, ḿas conocido como Lix-Alpha en los foros ODROID y tengo 43 ańos. Tengo una esposa, dos hijos, un gato siamés y todos mis dientes. Me encanta jugar a los cĺasicos de los ańos 90 en mi ODROID-GO, crear ḿusica con mis dispositivos de bolsillo y reajustar el Coche radiocontrol de mi hijo. Trabajo como diseńador gŕafico freelance creando diseńos 3D para medios impresos y publicidad. Mi hija tiene 5 ańos y mi hijo tiene 8 ańos, ambos son canadienses-japoneses (japoneses por mi esposa). Mi esposa dirige la oficina de una gran empresa de alquiler de herramientas en Montreal.

Happy New Year 2019

今年もどうぞよろしくお願いたします
from: Robert, Emi, Sakura, Taiyo & Koyuki



Imagen 1 - Los chavales de Rob

Nací en una región de la Provincia de Quebec, en Canadá, llamada Mont-Tremblant. Para cualquier persona que le guste esquiar, conocerá este lugar, sin duda, ya que recientemente ha sido comprado por Aspen Group. Pero ahora vivo en Montréal. Estudié psicología, pero lo dejé antes de terminar y monté mi propia empresa de diseño gráfico. Sin embargo, déjame decirte que mis estudios no fueron un desperdicio para nada, ¡Recurro a la psicología casi todos los días con algunos de mis clientes!

Mi madre siempre dice que cuando era joven, siempre me gustaba cualquier cosa que tuviera muchos botones. Cuando tuvo la primera oportunidad, me compró un ordenador. Mi primera experiencia fue con un TRS-80. Recuerdo pasar horas y horas jugando con BASIC, en el televisor en blanco y negro que teníamos en el sótano. Aún continúa en la casa de mi madre en algún lugar bajo toneladas de polvo. Escribiendo esto, creo que lo probaré la próxima vez que vaya a visitarla. Para los fans de BASIC:

10 PRINT "Go to question 2"
20 GOTO 10

¿Qué te atrajo a la plataforma ODDROID? Siempre he querido tener un pequeño dispositivo portátil para jugar a mis juegos retro favoritos. Busqué por todas partes en la web y encontré muchas soluciones, pero con un precio muy alto. Además, quería tener un dispositivo que pudiera metérmelo fácilmente en mi bolsillo sin tener miedo a romperlo. Fue entonces cuando encontré el ODDROID-GO. El dispositivo era barato, pequeño y estaba bien diseñado. También me gustaba mucho el enfoque de Hardkernel, especialmente la forma en la que comercializaba el producto y cómo hablaban de él. Iban directamente al grano y no intentaron exagerar la capacidad del dispositivo. Todo estaba claro, y ese no es el caso con muchas otras compañías.

Tengo muchas otras soluciones (Waveshare GameHat, GPD Win, GPD Win 2, Clockwork GameShell), pero todas tienen sus problemas: demasiado frágiles, voluminosas, se sobrecalientan o demasiado caras. Hardkernel ha logrado encontrar el nivel perfecto de potencia y rendimiento para un dispositivo de bolsillo y créeme, muchos lo han intentado y han fracasado. Por supuesto, algunos de los dispositivos mencionados anteriormente son mucho más potentes, pero ¿Puedes llevártelos sin temor a romperlos a donde quieras que vayas?

¿Cómo usas tus ODDROIDS? Sólo tengo un ODDROID-GO por ahora, pero lo uso todos los días. Soy un jugador, pero con mi trabajo, no tengo demasiado tiempo para jugar muy a menudo. Trabajo prácticamente los 7 días a la semana, así que el único momento que tengo es cuando me voy a la cama. Así que me acuesto, me pongo cómodo y juego un poco al Wolfenstein 3D o Commander Keen hasta que me duermo. Este dispositivo es bastante fiable, tiene una buena duración de batería y puedo contar con él cada vez que lo necesito.

¿Cuál es tu ODDROID favorito y por qué? Para ser honesto, el único que uso es el GO, así que supongo que es mi favorito. ¡Me encanta este dispositivo! Tener a Commander Keen o Wolfenstein 3D en mi bolsillo es un sueño que siempre he tenido desde los años 90.

¿Qué innovaciones te gustaría ver en futuros productos de Hardkernel? Como he mencionado anteriormente,

solo tengo un OROID-GO, así que hablaré de éste. Me encantaría ver un poco más de potencia. Sería bueno poder ejecutar GBA o SNES por ejemplo. Como soy diseñador gráfico, me encantaría que la interfaz de usuario fuera un poco más de código abierto, así podría crear mi propia interfaz de usuario.

¿Qué aficiones e intereses tienes aparte de los ordenadores? Tengo 2 hijos, ¡así que son mis pasatiempos la mayor parte del tiempo! Pero, cuando tengo un algo de tiempo, me encanta andar en bicicleta de montaña, jugar con mis dispositivos de bolsillo o salir con mi cámara tratando de conseguir la toma perfecta.



Figura X - Trabajo fotográfico de Rob

¿Qué consejo le darías a alguien que quiera aprender más sobre los ordenadores? Realmente no soy programador, así que no puedo dar consejos a nivel técnico, pero por supuesto puedo dar consejos sobre el proceso de aprendizaje asociada a los ordenadores/programación. Cuando salí de la

universidad y decidí montar mi propio negocio de diseño, tenía un conocimiento mínimo de Photoshop o Illustrator, pero me despertaba cada mañana y trabajaba en ellos. Al principio, dudé mucho de mí mismo, e incluso pensé en tirar la toalla. Pero al final, con disciplina y perseverancia, los conseguí dominar.



Figura X - Trabajo de modelado 3D de Rob

Lo mismo sucedió cuando empecé a explorar el modelado 3D con 3DS Max, Maya, C4D o Solidworks. Cualquiera puede aprender cualquier cosa; Es solo cuestión de tiempo y trabajo. Empieza desde el principio, intenta dominar todos los aspectos del paquete de software que está aprendiendo y, sobre todo, no te compares con otros. Alguna gente aprende rápido, mientras que otra aprende más lentamente. Al final, si te diviertes, estás en el sitio correcto y debes poner toda tu energía en lo que estás haciendo.