

Corte de Energía • NAS • KVM ODROID • Privacidad • Verium • Punto G

ODROID

Año Seis
Num. #72
Dic 2019

Magazine

UNA COMPLETA
EXPERIENCIA INFORMATICA

Mobil

ESTACION DE TRABAJO



REPARANDO
ODROID-N2:
LO BASICO PARA
REPARAR TU
HARDWARE

ODROID BLADECENTER:
ESTACION INFORMATICA ALTO RENDIMIENTO

ANALISIS PROFUNDO IA:
USANDO OPENVINO Y OPENCV
CON EL ODROID-C2



Minería de Criptomonedas: Cómo Ganar Monedas Verium con tu ODROID

© December 1, 2019

Los Odroid XU4, desarrollados por Hardkernel Co. Ltd, son muy populares. Los sistemas SBC en general están diseñados para ser muy eficientes desde el punto de vista energético, por eso son bastante adecuados para usarlos en la minería Verium.



Protege Tu Privacidad: Por qué Deberías Crear tus Propios Dispositivos de Domótica

© December 1, 2019

Con todos los hacks, vulnerabilidades, fugas de datos y otras noticias desalentadoras que giran en torno a los servicios en la nube, los productos en la nube y las empresas de big data, es posible que estés barajando la posibilidad de crear su propia versión de estas soluciones. Timbre "inteligente" ▶



Estación de Trabajo Móvil: Utilizando un ODROID-N2 para Crear una Completa Experiencia Informática

© December 1, 2019

Eebido a que de alguna forma perdí mi contraseña de la BIOS y se me bloqueó, tuve que desprenderme de mi viejo y querido ordenador portátil, que ejecutaba Debian. En lugar de pasarme al último Thinkpad, quería ver si de alguna manera podía combinar las cuestiones que he comentado anteriormente: ▶



KVM: Diviértete con la Virtualización en ODROID-H2

© December 1, 2019

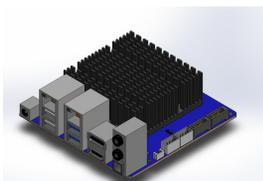
A la hora de hablar del ODROID-H2, el uso que hago de éste es de host de virtualización, ejecutar algunas máquinas virtuales de prueba para ejecutar software o probar actualizaciones de sistemas para mi empresa (p. Ej., Actualizar versiones anteriores de SO a nuevas versiones). En el foro, veo personas ▶



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

© December 1, 2019

Nuestra larga espera finalmente ha llegado a su fin: ¡Google Stadia, este servicio universal de streaming de juegos, ya ha sido lanzado! Se estrenó a fines de noviembre de 2019. Como ya se ha comentado en anteriores artículos de esta columna, para esta fecha de lanzamiento, Google Stadia SOLO estaría ▶



Desarrollando un BladeCenter ODROID-H2: Crea una Estación de Computación de Alto Rendimiento con Micro Huella

© December 1, 2019

Tras verme la inspiración del excelente modelo OpenSCAD H2 publicado en <https://forum.odroid.com/viewtopic.php?f=172&t=33824>, he creado un remix del fantástico Raspberry Blade Center para alojar 3 unidades ODROID-H2. Hice los siguientes cambios del proyecto original: Migrados los archivos a SolidWorks Fabricar un ensamblaje Alineados los mecanismos de cierre Ampliado el carro en ▶



Sobreviviendo a un Corte de Luz: Cómo Mantener un Negocio de Comercio Electrónico Durante un Corte de Energía Regional

© December 1, 2019

Imagínate si se fuera la luz en tu negocio. ¡Pero no solo en tu negocio, en tu ciudad, en tu comarca y toda tu región! A esto se enfrentó ameriDroid debido al apagado de energía de seguridad pública de PG&E que comenzó el sábado 26 de octubre de 2019 y [▶](#)



Reparado tu ODROID-N2: Cómo Recuperarse de un Cortocircuito Accidental

© December 1, 2019

En esta guía mostraremos cómo reparar el Q1 de tu ODROID-N2. El artículo original lo puedes localizar en la página wiki de Hardkernel



Creando una Aplicación de Visión en Situaciones de Baja Potencia: Usando OpenVino y OpenCV con ODROID-C2

© December 1, 2019

Este artículo te guiará a la hora de configurar un ODROID-C2 con Ubuntu * 16.04 (LTS), compilar CMake*, OpenCV y el kit de herramientas OpenVINO™ de Intel®, configurar tu Intel® NCS 2 y ejecutar algunos ejemplos para asegurarnos de que todo esté listo para que puedas crear e implementar tus [▶](#)



Cómo Configurar un NAS Básico con Samba para Compartir Archivos

© December 1, 2019

¡Usando Samba, cualquiera puede convertir una torre vieja o SBC en un servidor de archivos!

Minería de Criptomonedas: Cómo Ganar Monedas Verium con tu ODROID

© December 1, 2019 By Joe Rondx, wiki.verico.in.info ↳ ODROID-XU4, Tutoriales



Los Odroid XU4, desarrollados por Hardkernel Co. Ltd, son muy populares. Los sistemas SBC en general están diseñados para ser muy eficientes desde el punto de vista energético, por eso son bastante adecuados para usarlos en la minería Verium. La desventaja es que necesitas muchos para acumular una cantidad considerable de hashes. El coste inicial de los SBC puede ser muy alto, pero su consumo de energía es bastante bajo. Recuerda tener en cuenta todos los extras necesarios para hacer que la SBC funcione (tarjetas SD, cables de alimentación, cables de red, interruptores, refrigeración, mecanismos de montaje, etc...).

En el caso de los SBCs, el software de minería generalmente suele compilarse para 1 vía con 128 MB por subproceso. Muchos dispositivos han sido probados en términos de hashrate y se puede afirmar que la plataforma Odroid con sus CPU Octa core (Exynos5422 big.LITTLE) y 2 GB de LPDDR3 RAM

supera a cualquier otro dispositivo. En este momento, Hardkernel incluso ha lanzado una versión especial del XU4 original llamado Odroid HC1 diseñado para el clustering. Cabe señalar que a pesar de su relativamente bajo hashrate, la ración hashrate por energía sigue siendo muy buena. Además, existe un software de la comunidad que ayuda con el mantenimiento de grandes clusters. Para obtener más información, visita el sitio web oficial de ODROID <http://www.hardkernel.com>.

Imagen Optimizada de Sistema Operativo para Verium Mining

La imagen ODROID Verium Mining (por joe_rondx) incluye un sistema operativo optimizado con un minero preinstalado y otras funciones útiles. Está desarrollada para los dispositivos de la línea XU4, es decir, Odroid XU4, XU4Q, HC1, HC2, MC1.

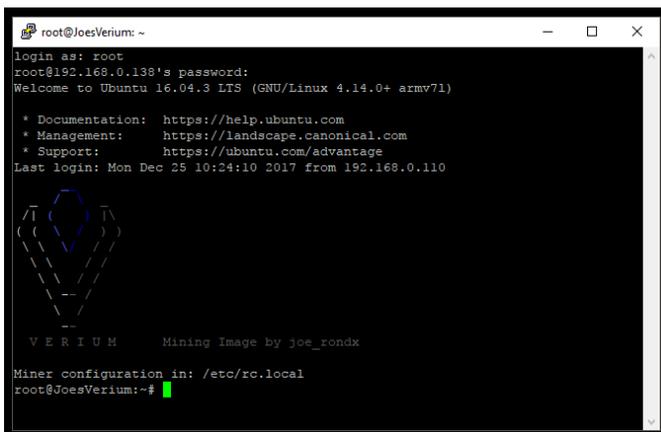


Figura 1: Iniciando sesión en la imagen de minería Verium optimizada

Características de rendimiento::

- Hugepages habilitadas (gracias a Birty & Fireworm)
- Reloj de RAM al Máximo: 933 MHz
- Downclocking CPU de los núcleos grandes: 1.9 GHz (2 GHz no valen la pena)
- Configuración óptima de dos mineros para núcleos big.LITTLE.

Scripts de ayuda:

- Registro de temperatura en verium/cpu_temp.log
- Script de estado general vrmcheck.sh
- Expansión del sistema de archivos por resize.sh

Configurar la imagen

1. Descargar la imagen desde: <https://drive.google.com/open?id=1RbXnGUh5fwmfhMQNzefCK75PajyrXjQi>
2. Graba sin extraer el archivo (la cuestión es la siguiente: la imagen consta de dos particiones, si extraes el archivo gz debes asegurarte de grabar ambas particiones y no solo la partición más voluminosa).
3. En el primer arranque, espera entre 5 a 10 minutos hasta que puedas localizar el dispositivo en la red.
4. Inicia sesión con las credenciales estándar de root/odroid.

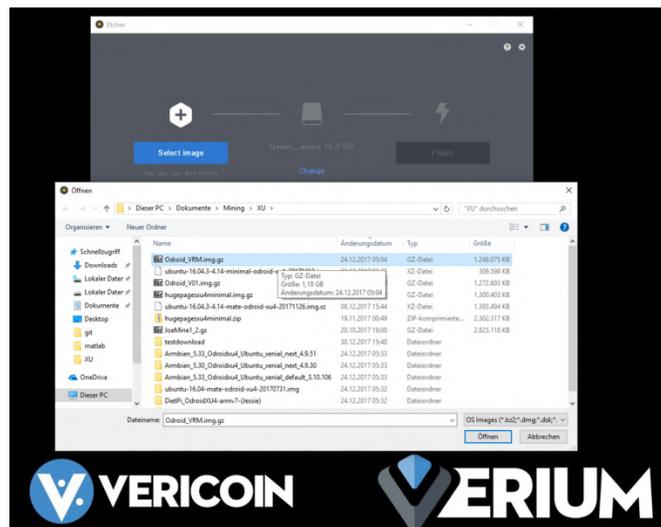


Figura 2 - Grabando la imagen de Verium usando Etcher

Configuración de la imagen

Lo primero que tienes que hacer es Actualizar la imagen. El script para hacerlo está en: https://raw.githubusercontent.com/DJoeDt/verium/master/odroid_image_update.sh. Desde la línea de comando usa:

```
$ wget
https://raw.githubusercontent.com/DJoeDt/verium/master/odroid_image_update.sh
$ chmod +x odroid_image_update.sh
```

Ahora, antes de ejecutar la actualización, asegúrate de matar al minero, yo utilicé

```
$ top
$ kill
```

Donde, nnn correspondería al pid de la instancia del minero en tu sistema. Ahora ejecuta:

```
./odroid_image_update.sh
```

Luego edita,

```
$ nano /etc/rc.local
```

Lo que hace es:

- Obtener y compilar el último minero fireworm
- Agregar el nuevo comando de un minero a /etc/rc.local
- Actualizar vrmcheck.sh (no es muy agradable, pero está bien ...)

Por qué hace esto: Resultaba que la configuración con dos mineros no es muy buena por varias razones.

Debo admitir que fui demasiado avaricioso y solo miraba el resultado H/m. Los motivos son los siguientes:

- La baja tasa de hash del minero de una vía reduce la posibilidad de enviar un share cuando se realiza la minería en grupo.
- La conexión adicional del minero provoca mucha carga de trabajo en el servidor del grupo.
- Es también así para la minería en solitario.

Ah, y un minero es más fácil de mantener, monitorear, etc. 😊 Y, por supuesto, la actualización de Fireworm Miner ofrece algunas características muy interesantes. He llegado a obtener unos 530 H/m.

Configuración personal

Comencemos con la configuración del minero. El inicio automático del minero se encuentra actualmente en el archivo rc.local.

```
$ nano /etc/rc.local
```

Al final, encontrarás dos comandos de minería que debes ajustar de acuerdo con tu configuración de grupo/solo. En realidad, no, tras la actualización, simplemente debes eliminar estos comandos y usar el comando One miner recién añadido.

Además de la configuración del minero, es posible que quieras cambiar el nombre de host y la contraseña:

```
$ nano /etc/hostname  
$ nano /etc/hosts  
$ passwd
```

También puedes expandir el sistema de archivos utilizando el script

```
$ ./resize.sh
```

Reinicia antes de continuar

```
$ reboot
```

y vuelve a ejecutar el script de cambio de tamaño

```
$ ./resize.sh
```

Red

Normalmente, el puerto Ethernet solo debe conectarse a través de DHCP. Si tienes un adaptador Wifi deberías usar

```
$ nmtui
```

Consejos XU4 y XU4Q

Como la imagen ha sido creada en un HC1, es posible que quieras revisar la configuración de la GPU: https://wiki.vericoin.info/index.php?title=Odroid#Downclocking_the_GPU.

Uso y Monitorización

Hay un simple script de estado en el directorio de inicio, invoca

```
$ ./vrmcheck.sh
```

que estampa la configuración en rc.local, las últimas 10 líneas de cada registro minero, la frecuencia de la CPU y la temperatura actual. Monitorizar ambos mineros supone un gran reto, pero los scripts de Monitor de mineros de DerrickES (<https://github.com/derricke/MinerMonitor>) o el monitor CLI de casanova (<https://github.com/bezeredi/verium-cli-monitor>) son compatibles con la configuración de los puertos (4048 y 4049 en este caso). También recomiendo minar en dos grupos para descentralizar el hasrate, mantener algunos hash activos si un grupo no está disponible y también para monitorizar.

Sugerencia de donación

Con la nueva actualización es más difícil distribuir pequeñas porciones de H/m. Básicamente, no voy a dedicar tiempo para volver a escribir esto: si ejecutas muchos Odroids con esta imagen, contempla la posibilidad de tener uno para mí, ¡muchas gracias!

1. La imagen comienza a minar de inmediato, para joe_rondx (la próxima versión incluirá fireworm). Esto no pretende ser una estafa. Contempla la posibilidad de tener una placa minando para nosotros, quizás una hora como donación inicial.
2. Si está ejecutando menos de 10 unidades (XU4, XU4Q, HC1, MC1/4), una donación inicial es todo lo que pedimos.
3. Si estás ejecutando 10-19 unidades, puede considerar tener un pequeño minero en la dirección joe_rondx.

4. Si está ejecutando 20-29 unidades, puede considerar tener un pequeño minero más en la dirección fireworms.
5. Si estás ejecutando 30-39 unidades, puede considerar cambiar el pequeño minero joes a uno grande.
6. Si estás ejecutando 40-49 unidades, puede considerar cambiar el pequeño minero fireworms a uno grande.
7. Si estás ejecutando más de 50 unidades, también podrías contemplar la posibilidad de hacer una donación al proyecto a través de su página.

Justificación: Los hasrates individuales pueden variar, pero supongamos que obtiene 450 H/m por unidad sin esta imagen. La imagen debería proporcionarle 537 H/m, lo que representa un aumento de casi el 20%. 10 unidades deberían producir 5370 H/m, mientras que un pequeño minero produce alrededor de 137 H/m, que es aproximadamente el 2.5% del hashrate total.

Otras imágenes de sistema operativo

Comenzamos con imágenes especiales de Odroid God birty: Odroid Miner Images (<https://drive.google.com/drive/folders/0B26cQdlGFXo2S3ViQ3lxaVhfUkk>). En particular, la imagen más reciente con hugepages habilitadas combinadas con mineros fireworms proporciona un impulso significativo de hashrate. Fue utilizado para optimizar la imagen. Las imágenes oficiales de Odroids (https://wiki.odroid.com/odroid-xu4/os_images/linux/start) están, por supuesto, muy bien hechas. El último Ubuntu Mate 16.04.3 (20171212) que se lanzó después de la imagen de birtys incluso tiene habilitadas las hugepages. Pero desafortunadamente usa demasiada RAM. Aún más desafortunada es la imagen Ubuntu 16.04.3 (20171213) (MINIMAL, BARE OS) que no tiene habilitadas las hugepages. He llegado a probar hasta 6 imágenes diferentes que están disponibles para la plataforma XU4. Para empezar, recomiendo la imagen DietPi (<https://dietpi.com/#download>) porque su herramienta diet-config incluida ya soporta muchas de las configuraciones que tienes que ajustar.

Adaptar el Sistema Operativo para la Minería

Todo ello en un script (ARM Miner + XU4 Setup):
 descarga el script shell desde:

https://raw.githubusercontent.com/DJoeDt/verium/master/1wayARM_XU4_VeriumMiner_install.sh.

```
$ wget
https://github.com/DJoeDt/verium/raw/master/1wayARM_XU4_VeriumMiner_install.sh
$ chmod +x 1wayARM_XU4_VeriumMiner_install.sh
$ ./1wayARM_XU4_VeriumMiner_install.sh
```

Downclocking (sí, ¡baja!) la CPU

Para evitar el estrangulamiento debido al calor, puedes aumentar tu hashrate si la CPU no funciona a 2 GHz (máx.). Incluso si puedes evitar la aceleración a 2 GHz, de todos modos no vale la pena: puede obtener 10-15 H/m más, pero te cuesta aproximadamente 2 vatios (de 12) llegar a este nivel, por lo que no compensa. Echa un vistazo a las pruebas de rendimiento: <https://www.planet3dnw.de/vbulletin/threads/428622-Odroid-HC1>. Instala la utilidad (o usa la configuración DietPi)

```
$ sudo apt-get install cpufrequtils
```

usarlo directamente

```
$ sudo cpufreq-set -c 7 -u 1.9GHz -r
```

y haz que los cambios se vuelvan permanentes creando un archivo de configuración

```
$ sudo nano /etc/default/cpufrequtils
```

con las siguientes configuraciones

```
ENABLE="true"
GOVERNOR="performance"
MAX_SPEED=1900000
MIN_SPEED=1900000
```

Ahora la CPU siempre debería funcionar a una velocidad constante. Esto también ahorra cierta cantidad de energía.

Overclocking de la RAM

En la partición FAT de arranque edita el boot.ini

```
$ sudo nano boot.ini
```

Busca

```
ddr_freq
```

y cambia el valor a 933

```
# DRAM Frequency
# Sets the LPDDR3 memory frequency
# Supported values: 933 825 728 633 (MHZ)
setenv ddr_freq 933
```

Asegúrate antes que bootz

```
# set DDR frequency
6dmc ${ddr_freq}
```

Downclocking la GPU

Instala esta utilidad

```
$ sudo apt-get install sysfsutils
```

Luego edita

```
$ sudo nano /etc/sysfs.conf
```

y agrega la siguiente línea

```
# Put GPU into powersave mode (= Downclocking it)
devices/platform/11800000.mali\:/devfreq/11800000.
mali\:/governor = powersave
```

luego inicia el servicio

```
$ sudo service sysfsutils start
```

Efecto: se reduce el consumo de energía en 0.7 - 0.8W, la temperatura del SOC será 1-3 °C más baja. Tienes información sobre cómo reducir el consumo de energía en un escenario sin monitor en <https://obihoernchen.net/1340/lower-gpu-clock-of-odroid-xu4-for-headless-servers/> Otra forma podría ser

```
$ sudo nano /etc/rc.local
```

y agregue esta línea antes de salir

```
$ echo powersave >
/sys/devices/platform/11800000.mali\:/devfreq/1180
0000.mali\:/governor
```

Configurar un archivo de intercambio Swap

Verium consume mucha memoria, de modo que aumentaremos el archivo de intercambio swap (o utiliza la configuración de DietPi).

```
$ sudo fallocate -l 1G /var/swapfile
$ sudo chmod 600 /var/swapfile
```

```
$ sudo mkswap /var/swapfile
$ sudo swapon /var/swapfile
```

compruébalo con

```
$ free -h
```

y configúralo para hacerlo permanente.

```
$ sudo echo "/var/swapfile none swap sw 0 0" >>
/etc/fstab
```

Procesos

Se puede lograr una mayor optimización comprobando el árbol de procesos

```
$ pstree -p
```

y deshabilitar/desinstalar cosas que no son necesarias. ¡MÁTALOS TODOS! Yo todavía no lo he localizado, pero si te encuentras con ads7846, elimínalo.

```
$ modprobe -r ads7846
$ tee /etc/modprobe.d/blacklist-ads7846.conf <<<
"ads7846"
```

Hardware XU4

La clave está en cambiar la cinta térmica del disipador térmico por una buena pasta térmica, lo cual hace que descienda la temperatura en 10 grados (C) con las mismas condiciones de prueba. También desciende la temperatura de la parte inferior. Para ahorrar electricidad, puedes bajar el voltaje de la fuente de alimentación. Comprueba la temperatura de la ranura de la tarjeta SD.

Usar RAM al máximo y optimizar big.LITTLE

Conocido también como "Obtener el último Hash de tu Odroid". El primer paso es usar una imagen del sistema operativo que use una cantidad mínima de RAM para el sistema. Para mostrar cómo usar los núcleos big.LITTLE y la mayor parte de la memoria, primero echaremos un vistazo a la configuración del minero.

El Objetivo

La minería Verium tiene que ver mucho con la RAM, de modo que querrás maximizar el uso de la memoria. ¿Cómo funciona? Verium Miner

(<https://github.com/effectsToCause/veriumMiner>)

se puede configurar para usar una cantidad diferente de RAM por subproceso. Entonces, la idea es usar 2 compilaciones mineras diferentes y hacer uso de los 2GB de RAM LPDDR3 a 933Mhz que tiene el Odroid. Además: hacer esto inteligentemente beneficiará también a los núcleos big.LITTLE de las CPU Exynos5422 Cortex™ ARM Cortex-A15 (2.0Ghz)/Cortex-A7 (1.4 Ghz) Octa core de Samsung

Detalles técnicos

La configuración del minero activa 1 vía o 3 vías donde

```
1 way => 128 MB per mining thread
3 way => 384 MB per mining thread
```

La vía del minero está configurado en

```
$ nano veriumMiner/algo/scrypt.c
```

Ahora puedes calcularlo tú mismo o simplemente confiar en mí para tener estas dos configuraciones mineras y luego ejecutar

```
5 threads @ 1 way = 640 MB
3 threads @ 3 way = 1152 MB
1792 MB total RAM
```

El sistema también necesita algo de memoria y con esta configuración solo quedan unos 60 MB libres, pero solo si usas la imagen vinculada anteriormente (he probado 6 diferentes, solo ésta funciona con -t 5 y -t 3). Una compilación de 2 vías también puede ser interesante, pero en mi caso no llego a funcionar.

Fácil instalación

He preparado dos scripts en mi repositorio git (<https://github.com/DJoeDt/verium> - sin ningún tipo de garantía):

```
1wayARM_XU4_VeriumM6iner_install.sh
3wayARM_XU4_VeriumMiner_install.sh
```

que instala los mineros en

```
~/verium/1wayminer/
~/verium/3wayminer/
```

Cómo ejecutarlos

Teniendo ambas compilaciones a mano, necesitamos administrarlas de forma adecuada. Además de los hilos de ejecución hay dos cosas que tenemos que configurar

```
# the big.little core
# the API Port
```

big cores -t 3

Los núcleos grandes deben ejecutar el minero de 3 vías en el puerto estándar 4048 con alta prioridad

```
--cpu-priority 4
--cpu-affinity 0x00F0
-b 4048
--api-bind 0.0.0.0:4048
```

Estas son las opciones a configurar. Sobre el uso de cpu-affinity (<https://wiki.verico.in/index.php?title=Cpu-affinity>). Esta es la línea completa de mi /etc/rc.local

```
# 3way -t 3 big
/root/verium/3wayminer/cpuminer -o
stratum+tcp://pool-eu.bloxstor.com:3003 -u
VEXMki29ycW5vSt3MmdM5iwHqsHux91EMr.Guide -p
GuidePwd --cpu-priority
4 -t 3 --cpu-affinity 0x00FF --api-bind
0.0.0.0:4048 &
```

Simplemente cópialo y Pruébalo a modo de donación ;):

little cores -t 5

El pequeño núcleo ejecutará el minero de 1 vía en el puerto API 4049 con menor prioridad

```
--cpu-priority 1
-b 4049
--api-bind 0.0.0.0:4049
```

donde simplemente no tocamos --cpu-affinity y, por lo tanto, se usan los 4 núcleos pequeños + 1 grande. Esta es la línea completa de mi /etc/rc.local

```
# 1way -t 5 little
/root/verium/1wayminer/cpuminer -o
stratum+tcp://pool-eu.bloxstor.com:3003 -u
VEXMki29ycW5vSt3MmdM5iwHqsHux91EMr.Guide -p
GuidePwd --cpu-priority 1 -t 5 -b 4049 --api-bind
0.0.0.0:4049 &
```

Monitorización

Si usas la API, necesitas configurar ambos puertos. Mi solución con birtys <3 webscripts se ve así: Hasta ahora he copiado index_monitor.php a index_monitor4049.php, reconfigurado

```
defined('API_PORT') || define('API_PORT', 4049);
```

en él, y simplemente lo incluí añadiendo al index.php. Puede encontrar más información sobre la monitorización aquí.

El resultado

Algunas observaciones antes de mirar H/m:

1. Nuevamente: solo la imagen vinculada anteriormente me funcionó, aunque no por defecto
2. Tienes que crear un archivo de intercambio swap (incluido en mis scripts)
3. Mentí sobre las líneas en mi rc.local, en realidad mino en dos grupos diferentes. ¡Descentralízalo!

¡Números de hashrate! 11

Originalmente ejecuté la configuración de birtys miner, que en realidad es la configuración de 1 vía. Sin ninguna opción -t, simplemente inicié 8 subprocesos y con la minería de grupo obtuve un promedio de

```
395 H/m = 1 way -t 8
```

Los dos mineros produjeron algo así como

```
195 H/m = 1 way -t 5  
250 H/m = 3 way -t 3
```

¡Feliz adición! 😊

Me pregunto cómo funciona esto con la minería en solitario.

Actualización a fireworm miner & hugepages

```
1way : 128MB -> nr_hugepages = 65. 3way : 384MB ->  
nr_hugepages = 193. 6way : 768MB -> nr_hugepages =  
386.
```

Configuración de la imagen de birtys

Configuración: imagen mínima de birtys hugepages: [unidad] Configurar hugepages

```
$ sudo nano /etc/sysctl.conf
```

a

```
$ vm.nr_hugepages=839
```

Cambiar host

```
$ nano /etc/hostname
```

```
$ nano /etc/hosts
```

y dram_freq=933 en

```
$ nano /media/boot/boot.ini
```

así como la contraseña por

```
$ passwd
```

Reiniciar antes de continuar

```
$ reboot
```

Instalación del minero Fireworm

Eliminar el viejo minero de la imagen de birtys:

```
$ rm -rf veriumMiner
```

Instala el minero más reciente con el script:

```
$ wget \
```

https://github.com/DJoeDt/verium/raw/master/nwayARM_HC1_fireworm_install.sh

```
$ chmod +x nwayARM_HC1_fireworm_install.sh  
$ ./nwayARM_HC1_fireworm_install.sh
```

Configuración de inicio automático de minero

Configuración de inicio automático

```
$ nano /etc/rc.local
```

```
# Verium Miner Configuration  
# big cores  
$ nice --10 /root/verium/nwayminer/cpuminer  
-o stratum+tcp://vrm.n3rd3d.com:3332 -u  
joe_rondx.1 -p joe  
-t 3 -1 1 --cpu-affinity 0x00F0 --cpu-priority 2  
--api-bind 0.0.0.0:4048 --no-color >>  
/root/verium/nwayminer/3waymine.log &  
  
$ sleep 5 # delay for hugepages allocation  
  
# little cores
```

```
$ /root/verium/nwayminer/cpuminer
-o stratum+tcp://eu.vrm.mining-pool.ovh:3032 -u
joe_rondx.HC1_1 -p joe
-1 4 --cpu-affinity-stride 1 --cpu-affinity-
oneway-index 0 --cpu-priority 0
--api-bind 0.0.0.0:4049 -b 4049 --no-color >>
/root/verium/nwayminer/1waymine.log
```

¡Ejecuta el minero 3way primero!

```
-t 3 -1
```

Primero porque deberías hacer un uso máximo de las hugepages. En segundo lugar, ejecuta solo 1way

```
-1 4
```

donde 1 hilo se ejecuta sin hugepages.

Hashrate

```
=> 400 + 137 = 537 H/m @ 1.9 GHz
```

Potenciar el dispositivo

Las especificaciones dicen 4 Amps @ 5V para un Odroid. Pero hay mucho más a tener en cuenta si deseas potenciar tu plataforma Referencias

<https://wiki.verico.in/index.php?title=Odroid>

[https://ameridroid.com/products?](https://ameridroid.com/products?keywords=aluminum)

[keywords=aluminum](https://ameridroid.com/products?keywords=aluminum)

[https://forum.odroid.com/viewtopic.php?](https://forum.odroid.com/viewtopic.php?f=93&t=27239)

[f=93&t=27239](https://forum.odroid.com/viewtopic.php?f=93&t=27239)

[http://www.thinkwiki.org/wiki/How_to_use_cpufreq](http://www.thinkwiki.org/wiki/How_to_use_cpufreq_utils)

[utils](http://www.thinkwiki.org/wiki/How_to_use_cpufreq_utils) [https://wiki.odroid.com/odroid-](https://wiki.odroid.com/odroid-xu4/os_images/linux/ubuntu_4.14/ubuntu_4.14)

[xu4/os_images/linux/ubuntu_4.14/ubuntu_4.14](https://forum.odroid.com/viewtopic.php?f=146&t=28895&sid=873dc51d2cf97257c807b99826f91525)

[https://forum.odroid.com/viewtopic.php?](https://forum.odroid.com/viewtopic.php?f=146&t=28895&sid=873dc51d2cf97257c807b99826f91525)

[f=146&t=28895&sid=873dc51d2cf97257c807b99826f91525](https://forum.odroid.com/viewtopic.php?f=146&t=28895&sid=873dc51d2cf97257c807b99826f91525)

Protege Tu Privacidad: Por qué Deberías Crear tus Propios Dispositivos de Domótica

December 1, 2019 By Bo Lechnowsky, www.ameridroid.com Desarrollo, ODROID-C2, ODROID-N2, ODROID-XU4



Con todos los hacks, vulnerabilidades, fugas de datos y otras noticias desalentadoras que giran en torno a los servicios en la nube, los productos en la nube y las empresas de big data, es posible que estés barajando la posibilidad de crear su propia versión de estas soluciones.

Timbre "inteligente"

¿Ha leído las recientes noticias sobre un "timbre inteligente" vendido por una gran empresa que envía la contraseña WiFi de tu casa en texto plano a través de Internet, lo cual permite a los hackers obtener potencialmente acceso a tu red doméstica? ¿Y cómo la misma compañía se jactaba de rastrear los "truco o trato" cuando iban de casa en casa en Halloween?

¿Has leído las noticias de este verano sobre una popular "cerradura inteligente" que permitía a los hackers desbloquear tus puertas sin conocer tu contraseña o conseguir la clave?



Figura 1 - Imagen: Chase Dardaman, Jason Wheeler

desde <https://techcrunch.com/2019/07/02/smart-home-hub-flaws-unlock-doors/>:

"Dardaman dijo que cualquier centro conectado directamente a Internet sería explotable de forma remota. Los investigadores encontraron cinco dispositivos vulnerables usando Shodan, un motor de búsqueda de dispositivos y bases de datos disponibles públicamente".

¿Has oído que Alexa, Siri y Google Assistant pueden ser pirateados por un láser a una distancia de hasta 350 pies? Las noticias de esta explotación la pueden encontrar [aquí:](https://www.vice.com/en_in/article/3kxwvy/alexa-siri-and-google-assistant-can-be-hacked-remotely-with-lasers)
https://www.vice.com/en_in/article/3kxwvy/alexa-siri-and-google-assistant-can-be-hacked-remotely-with-lasers

Estos son solo tres ejemplos de compañías de big data que gestionan muy mal la privacidad y seguridad de sus clientes. Una de las razones por las que estos "productos de big data" se piratean con tanta frecuencia es porque las empresas de big data a menudo se preocupan menos por la seguridad que por los beneficios y por poner un producto rápidamente en el mercado. Además, generalmente suelen vender muchos de los productos en cuestión, lo que los convierte en un objetivo muy jugoso para los piratas informáticos.

Algunas ventajas para crear tus propias versiones de estos productos son:

- Tienes libertad para ejercercitar tu creatividad y pasión para desarrollar tus productos.
- Aprenderás nuevas habilidades y agudizarás las habilidades ya existentes.
- Puedes añadir funciones que otros no ofrecen y que se adaptan a tus necesidades particulares.
- Puedes sentirte orgulloso de crear tus propias soluciones y reparar tus propias cosas.
- Puedes "tirarte a la piscina" con tus elementos de seguridad, si quieres, o puedes confiar en la "seguridad opaca de terceros", (https://en.wikipedia.org/wiki/Security_through_obscurity) o una combinación de ambos, algo que puede funcionar mucho mejor para un producto único en lugar de un producto vendido a miles o millones de clientes por una empresa de Big Data.
- Existen muchos proyectos de código abierto que pueden usarse como punto de partida para tu solución. A diferencia de una "solución enlatada" proporcionada por una compañía de big data, el software de código abierto puede ser auditado por cualquier persona para comprobar si existen problemas de seguridad o privacidad.

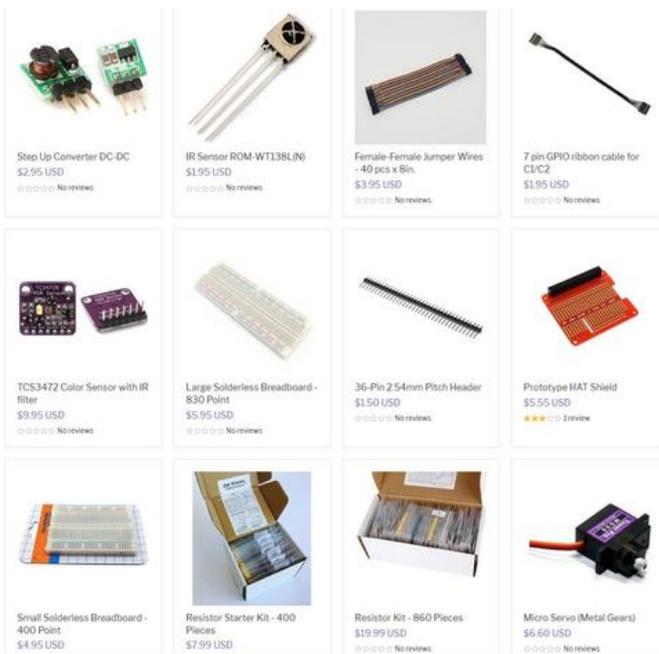


Figura 2 - Lista de recursos

El artículo original se puede ver en el siguiente enlace:
<https://ameridroid.com/blogs/ameriblogs/privacy-why-you-should-probably-make-your-own-home-automation-devices>

Estación de Trabajo Móvil: Utilizando un ODROID-N2 para Crear una Completa Experiencia Informática

© December 1, 2019 By Pieterjan Montens ODRROID-N2, Tutoriales



Durante algunos años he anhelado dos cosas:

- Un "ordenador portátil" con un teclado de alta calidad y un trackball
- Una "estación de trabajo" basada en arquitectura ARM para ser más productivo

Inesperadamente y debido a que de alguna forma perdí mi contraseña de la BIOS y se me bloqueó, tuve que desprenderme de mi viejo y querido ordenador portátil, que ejecutaba Debian (suelo viajar con un ordenador portátil Linux y Windows. No, no puedo pensar en una sola razón para hacerme con un Mac) Pero en lugar de pasarme al último Thinkpad, quería ver si de alguna manera podía combinar las cuestiones que he comentado anteriormente: una estación de trabajo basada en ARM "móvil" donde podía elegir todos los componentes, sin soldadura o cinta adhesiva, simplemente componentes de serie. Sería de bajo consumo. Se adaptaría a los lugares en

los que trabajaría, usando pantallas si están disponibles, energía, etc., solo necesitaría llevar los componentes que necesitaría para mi viaje. Sería de bajo presupuesto. Así que, me fui con una pequeña lista de compras.

Ya sabía que usaría mi teclado y mi trackball ergonómicos, porque son mis herramientas de entrada más hábiles. Simplemente necesitaba una unidad informática, una pantalla y una batería. Estos dos últimos componentes fueron fáciles de encontrar, Amazon tiene muchos:

- Un panel estándar de 13.3" 1080p que de alguna manera funciona, aunque no poder cambiar el brillo y el contraste es un error: debería haber elegido algo mejor.
- Una fuente de alimentación de 25000 mAh con salidas USB y 12V, esto es importante. La mía es de Krisdonia y a diferencia de la pantalla, estoy muy contento con ella.

Elegir una unidad informática basada en ARM, lo suficientemente potente como para "trabajar" con (cierta carga informática y gráficos decentes) y con un medio de almacenamiento robusto (es decir, que no sea microSD) fue un poco más difícil: ya tenía mucha experiencia en Raspberry Pi (Esto fue antes de que saliera la 4) con proyectos de hogar y soluciones de monitorización, pero se echaba en falta como plataforma de trabajo cotidiana. Tras una pequeña comparación de alternativas disponibles, elegí el ODROID-N2 4GB, principalmente por el robusto almacenamiento eMMC. Aquí tienes una lista más completa de lo que ofrece:

- Arquitectura BIG.little: quad Cortex-A73 "GRANDE" y Cortex-A53 dual "pequeño", lo que lo convierte en un hexacore heterogéneo, junto con 4 GB de RAM DDR4
- Una decente GPU Mali-G52
- Memoria eMMC (hasta 128 GB)
- Gigabit ethernet, HDMI (4k @ 60hz), 4 puertos USB3 y
- Viene con un sistema de enfriamiento pasivo, que actúa a modo de soporte para toda la unidad

La instalación fue muy fácil (Debian Buster con el kernel 4.9.190-odroidn2-arm64), y en poco tiempo tenía instalado mi entorno de trabajo habitual y era completamente funcional. ¡Funcionaba! Lo usé realmente para trabajar. Estoy escribiendo este artículo con él. Es un hardware que encontraría en un teléfono inteligente, es pequeño, he conectado mis periféricos y todo va genial. También minimiza mi impacto medioambiental, ya que el consumo de energía es muy limitado en comparación con un ordenador portátil estándar, y bastante menor que un ordenador de escritorio (la pantalla es el elemento que más consume energía, al igual que en los teléfonos inteligentes).

Entonces, ¿Cómo es trabajar en ARM?

Bueno, para empezar: gracias a Debian que tiene compilada toda la distribución para múltiples arquitecturas, tengo acceso a casi todas mis herramientas. Algunos software (propietario) como la estación en la nube Synology, no están disponibles para arquitecturas ARM. Slack tampoco está

disponible, pero ofrecen una interfaz web bastante viable.

No tuve ningún problema con las herramientas del día a día, como vim, node, npm, aunque Docker es un poco lento de compilar. No obstante, ¡incluso GIMP y FreeCAD funcionan! Trabajar con la batería también es factible, hasta ahora nunca he logrado vaciarla por completo.

Puntualmente, alguna pestaña se bloquea en Firefox. De cualquier forma, la navegación es algo muy trivial, aunque nada comparable a una máquina i7 de 16GB. Las aplicaciones web pesadas para el cliente son otro tema y a veces, un poco lentas. Sin embargo, cambiar a Chrome para Google Docs ayuda bastante. No obstante, no se ha convertido en mi único "ordenador" que llevo conmigo ... Todavía llevo un Thinkpad la mayor parte del tiempo, y resultan ser bastante complementario. Simplemente necesito un poco más de tiempo para configurarlo todo al llegar y para retirarlo todo cuando me voy.

Esto, sin embargo, no será el final de mis aventuras con ARM. Todavía tiene algunas debilidades en las que se debería trabajar, para que un dispositivo como este satisfaga todas mis necesidades. Como, por ejemplo:

- Decenas de núcleos. BIG.little sería genial, ¡Simplemente añadir más!
- ¡Más RAM!
- Mejor almacenamiento (¿NVMe?)
- Redes de calidad integradas (wifi, ethernet, Bluetooth)
- 2 salidas de video
- Cifrado completo de disco basado en hardware: estos dispositivos son fáciles de perder y pueden contener datos confidenciales.

No olvidemos que se espera que Microsoft lance pronto un híbrido de Surface impulsado por ARM: puedo terminar trabajando exclusivamente en ARM antes de lo que creía posible.

Pero, ¿cómo es comparado con la Raspberry Pi 4?

Poco después de comprar el ODROID, salió el Raspberry Pi 4, totalmente sin previo aviso, para sorpresa de casi todos. La mayoría de los lectores

probablemente querrán saber cómo ambas plataformas ARM se comparan entre sí. Esta breve y subjetiva comparación no incluirá la ventaja de la existencia de la formidable comunidad de raspberry, que es la razón principal por la que elegí raspberry cuando empecé mi aventura con la arquitectura ARM. Solo me centraré en el uso específico que yo hago de estos dispositivos, aunque no debemos olvidar que ambas plataformas realmente no van dirigidas al mismo segmento de mercado. En general, las diferencias técnicas con la raspberry 4 impulsada con A72 son pequeñas:

- Si bien la CPU ODROID A73 es una arquitectura completamente diferente, solo ofrece algunas mejoras con respecto a la A72, tal y como se detalla en anandtech:
<https://www.anandtech.com/show/10347/arm-cortex-a73-artemis-unveiled>
- El ODROID de HardKernel tiene 2 núcleos A53 "pequeños" adicionales, como los utilizados por Raspberry Pi 3 (que tiene 4 de ellos). De modo que

obtienes el equivalente informático de una Raspberry Pi 4 y una media de la Raspberry Pi 3.

- Ambos tienen puertos Gigabit ethernet y USB 3
- ODROID viene con su propio disipador de calor, mientras que la raspberry se acelera rápidamente sin tener uno.
- La GPU ODROID Mali G52 debe ser al menos dos veces más potente que la videocore 6 de Raspberry Pi (850Mhz, 6.8 Gpix/s frente a 500Mhz, 2.5 GPix/s)
- El chipset integrado ODROID no incluye wifi o bluetooth, necesita un pequeño dongle ... esto es una pena, porque los RaspBerry incluyen ambos.

Aunque me han encantado la doble salida micro-HDMI de la Raspberry Pi 4, el almacenamiento micro-SD es muy poco fiable a la hora de trabajar diariamente. El almacenamiento eMMC es, en mi opinión, el argumento decisivo a favor del ODROID, en comparación con el Raspberry Pi 4.

El artículo original lo puedes encontrar en https://medium.com/@pieterjan_m/reinterpretation-of-the-mobile-workstation-e8dc95d279f9.

KVM: Diviértete con la Virtualización en ODRROID-H2

© December 1, 2019 By Tobias Schaaf ODRROID-H2, Tutoriales



A la hora de hablar del ODRROID-H2, el uso que hago de éste es de host de virtualización, ejecutar algunas máquinas virtuales de prueba para ejecutar software o probar actualizaciones de sistemas para mi empresa (p. Ej., Actualizar versiones anteriores de SO a nuevas versiones). En el foro, veo personas que tienen dificultades a la hora hacer funcionar VMware o Citrix Hypervisor, principalmente debido a los NICs no compatibles y otros componentes de la placa. Siempre me ha desconcertado un poco, ya que conozco la infraestructura KVM que es parte del kernel de Linux, la cual permite una fácil virtualización, y dado que las NICs y otras cosas de Linux (Ubuntu, Debian, etc.) funcionan bien, la virtualización debería funcionar sin ningún problema. Cuando mencioné esto por primera vez en el foro, me pidieron que hiciera una guía, ya que parecía que nadie lo conocía. Así que veamos qué tenemos que decir al respecto: primero, explicaremos un poco lo que realmente vamos a hablar, y qué herramientas y componentes voy utilizar.

KVM

KVM está integrado en el núcleo, en realidad significa "máquina virtual basada en el núcleo" y utiliza el núcleo de Linux para ejecutar máquinas virtuales (VM) sobre él. Esto te permite ejecutar máquinas virtuales directamente en cualquier Linux que ejecutes en tu ODRROID-H2 (o cualquier otro PC que ejecute Linux).

QEMU

Qemu - Quick Emulator es conocido como un emulador bastante rápido y rico en funciones que te permite emular diferentes sistemas y componentes de hardware. Se puede usar en combinación con KVM para emular hardware como discos duros, unidades de CD-ROM, NIC, etc., aunque envía los comandos de los dispositivos a la capa de virtualización de KVM. Permite funciones avanzadas como capturas de pantalla para máquinas virtuales.

libvirt

Libvirt es una API de código abierto para configurar plataformas VM como KVM, Xen, VMWare o QEMU. La lista de hipervisores compatibles es bastante larga e incluso incluye contenedores como LXC. Lo usaremos como interfaz de usuario, tanto de forma gráfica como de línea de comandos para hacer cosas con nuestras máquinas virtuales en KVM.

Primer Escenario

Ahora que hemos decidido usar KVM, veamos qué necesitamos para ejecutar máquinas virtuales y cómo instalar máquinas virtuales de Linux y Windows, así como controlarlas. Puesto que somos amantes de los ODRUIDs, lo hacemos, por supuesto, en nuestro ODRUID-H2, pero como he dicho anteriormente, también funcionará en cualquier otro PC/Servidor basado en x86 que ejecute Linux. Ten esto en cuenta cuando me refiera al ODRUID-H2 de aquí en adelante, ya que también se aplicará a cualquier otro sistema.

Requisitos

- ODRUID-H2
- Sistema operativo instalado (Debian o Ubuntu para un inicio fácil)
- Conexión a Internet

En este primer escenario, no necesitamos mucho: solo un ODRUID-H2 con un Linux instalado de tu elección. Como estoy a favor de Debian para las tareas del servidor, usaré Debian Buster como sistema operativo de referencia, pero los comandos se aplican también a Ubuntu y deberían funcionar de igual forma.

Instalación y configuración

Supongamos que ya has instalado una distribución de Linux completamente funcional con un entorno de escritorio (DE) y Network Manager y tu usuario por defecto es "odroid".

```
$ sudo apt install virt-manager
$ sudo adduser odroid libvirt
$ sudo reboot
```

¡Felicitaciones, ya has terminado y puedes empezar a crear máquinas virtuales! Esto no ha sido tan difícil, ¿verdad?

Virtual Machine Manager (virt-manager)

El Virtual Machine Manager (virt-manager) es una interfaz gráfica que puedes usar para crear y configurar tus máquinas virtuales, crear instantáneas y todo tipo de cosas.

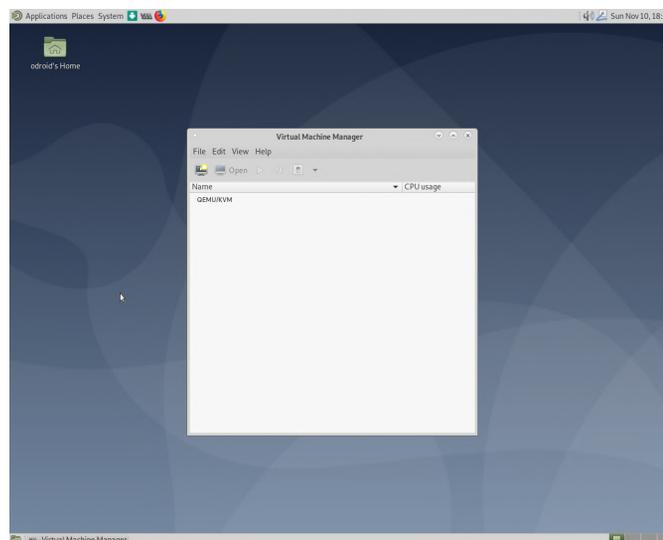


Figura 1 - Virtual Machine Manager en Debian Buster MATE Desktop

Virtual Machine Manager es lo que necesitas para crear, configurar e interactuar con tus máquinas virtuales. Se puede usar para monitorizar recursos como el uso de la CPU y otras cosas. No quiero entrar demasiado en detalles, digamos que hay mucho que puedes hacer con él, y es posible que quieras leer algunos tutoriales avanzados. Por ahora, vamos a lo simple y veamos qué necesitamos para crear una VM.

Creando nuestra primera VM Aún conservo una imagen de instalación de Debian Stretch, que voy a usar en este escenario. Puedes montar archivos .iso directamente en máquinas virtuales similares a VMWare y otros hipervisores. Solo hay un detalle que no estás de más que conozcas: por defecto, virt-manager busca todas las imágenes en /var/lib/libvirt/images. Utiliza esto por defecto para todas las imágenes, tanto tus discos duros, como los archivos ISO que desees montar. Puede agregar más almacenamiento si quieres, pero por ahora, conservaremos este y copiaremos la imagen de Debian Stretch que tengo en esta carpeta para que sea localizada directamente. Después de esto, haremos clic en el botón de la esquina superior izquierda de nuestro Virtual Machine Manager para

crear una nueva VM. Se abrirá un asistente que te guiará a través de todo el proceso.

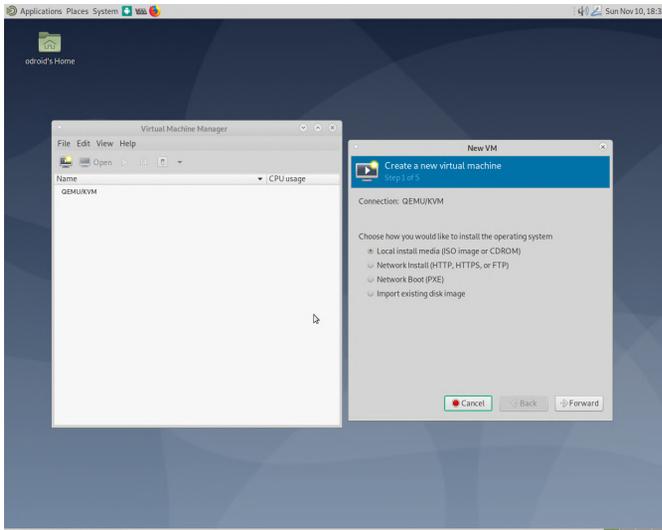


Figura 2: Configurar una nueva máquina virtual con virt-manager es muy fácil

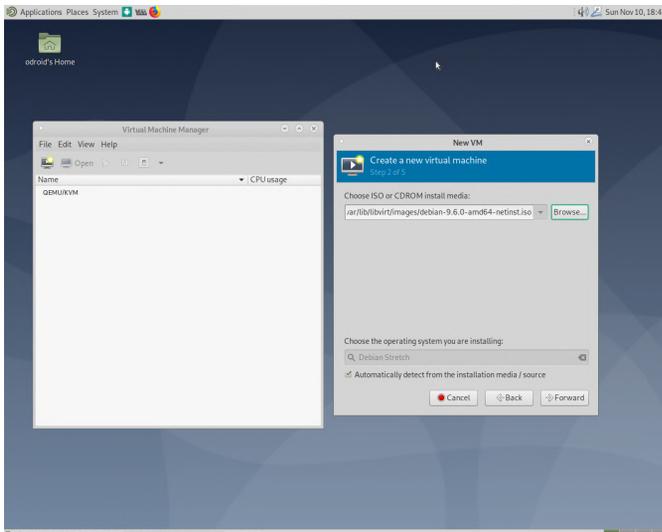


Figura 3 - Seleccionando la imagen netinstall de Debian 9 como CDROM

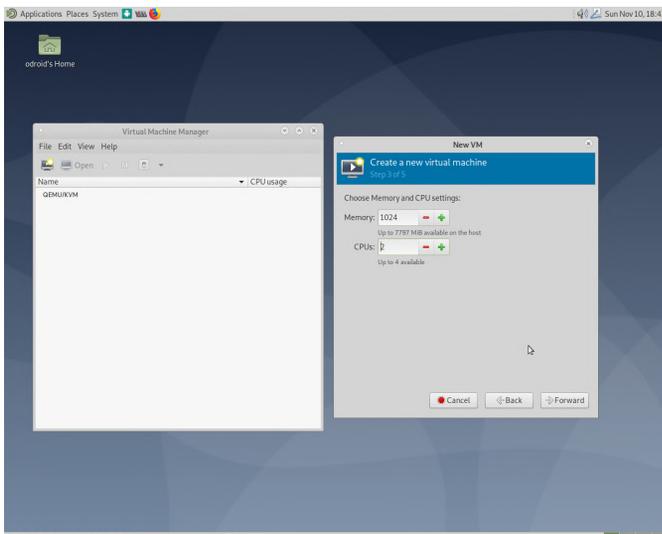


Figura 4: Elegí 2 CPU para demostrar que las máquinas virtuales pueden manejar múltiples núcleos

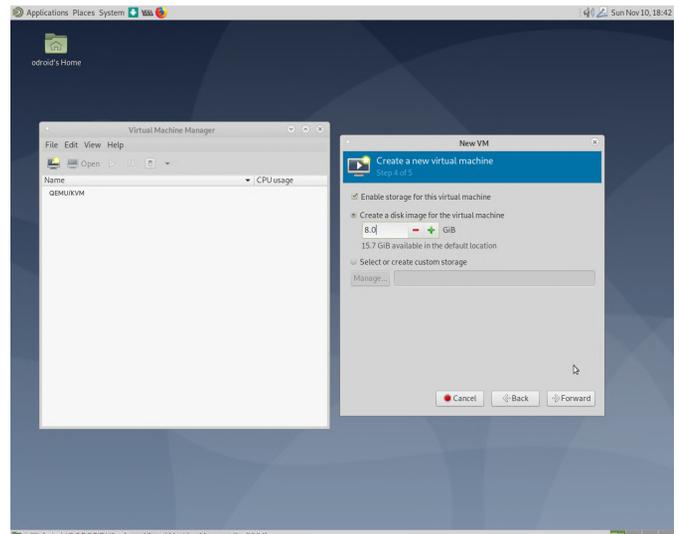


Figura 5: Solo quiero una pequeña máquina virtual de prueba, así que 8 GB es suficiente como espacio en disco

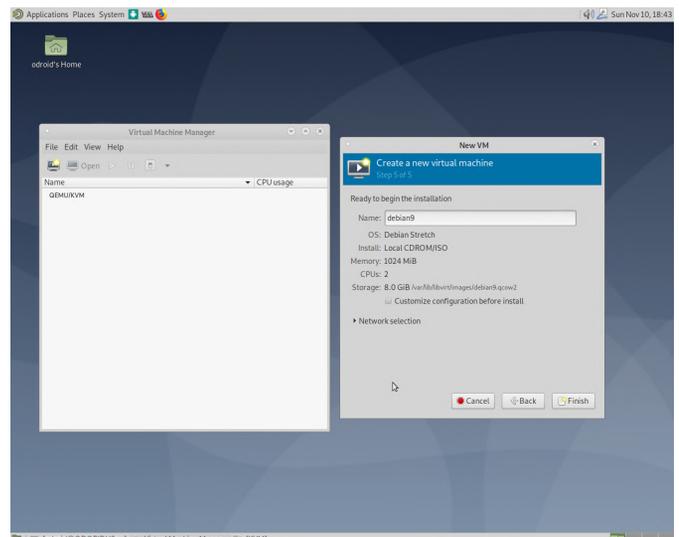


Figura 6: Asigna un nombre a la máquina virtual y listo.

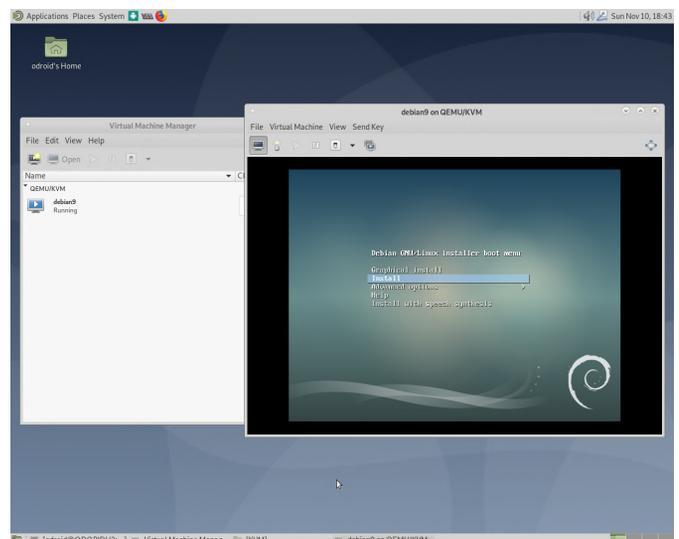


Figura 7 - La VM inicia y arranca la imagen del instalador para Debian 9

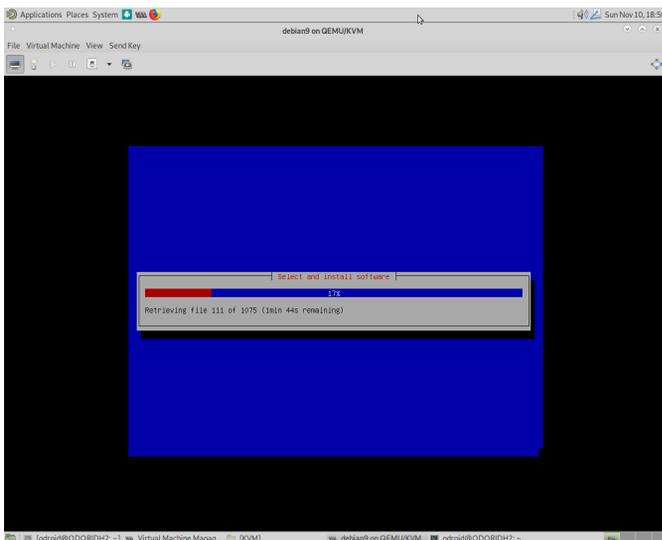


Figura 8: Pasar por la configuración estándar e instalar el escritorio MATE

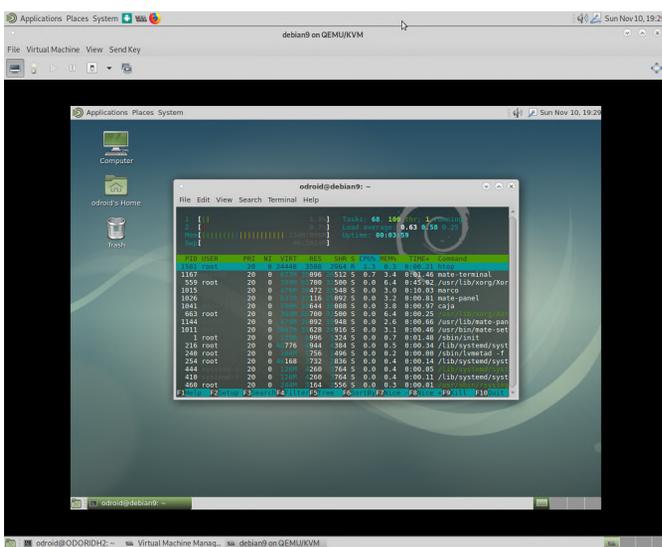


Figura 9: El escritorio Mate en Debian Stretch ejecutándose en una VM

Todo el proceso es bastante fácil y directo. Y como podemos ver, incluso una instalación de un entorno de escritorio gráfico es bastante sencilla y puede usarse sin problemas. Por otro lado, también podrías usar una simple instalación de terminal y conectarte a tu VM a través de SSH.

Instalar una VM con Windows

Instalar Windows es parecido, pero tiene algunos matices. Mientras que Linux es bastante factible a la hora manejar sistemas virtuales y ser un sistema virtual en sí mismo, Windows no es tan bueno con estas cosas, y aunque podríamos instalar una máquina Windows de la misma manera que lo hicimos con el sistema Linux, el rendimiento no sería tan bueno, ya que Windows no maneja la virtualización lo suficientemente bien. Para esto, es

mejor instalar Windows directamente con soporte de controladores para la virtualización, especialmente para que KVM tenga el mejor rendimiento posible. En primer lugar, necesitamos descargar algunos controladores. Afortunadamente para nosotros, el proyecto Fedora se encarga de la mayoría de las cosas, y simplemente tenemos que descargar una imagen precompilada para todos los controladores que necesitamos. En <https://bit.ly/2s3rAUK>, hay varios enlaces de imágenes estables y las imágenes más recientes con los controladores necesarios, ya sea como un CD o incluso como una imagen de disquete para descargar. Yo descargué la imagen CD estable que en el momento de escribir este artículo era: virtio-win-0.1.171.iso y la coloqué en la carpeta de imágenes como he explicado anteriormente. A continuación, podemos empezar a crear una nueva máquina virtual igual que la que hicimos con Linux, excepto con la última pantalla donde verificaremos la opción "Customize configuration before install". Aquí, necesitamos cambiar algunas configuraciones de hardware para la VM. Comenzamos agregando nuevo hardware, donde "Storage" ya debería estar seleccionado. Cambiamos el tipo de dispositivo a un dispositivo CDROM y seleccionamos nuestra imagen virtual que tiene todos los controladores.

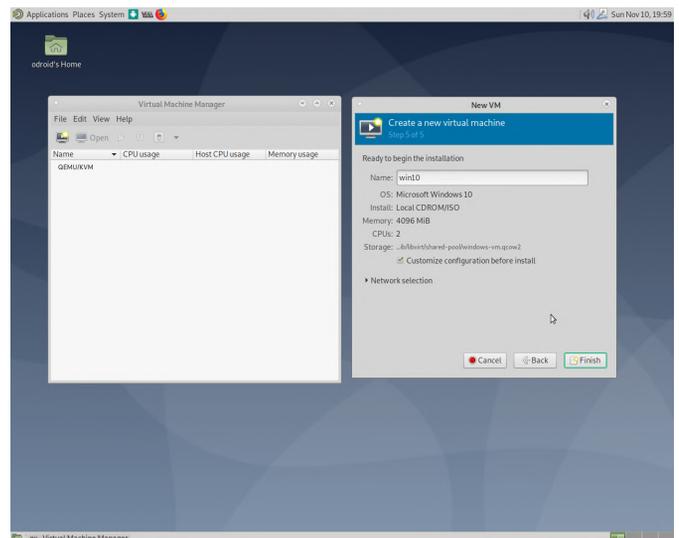


Figura 10: Asegúrate de seleccionar la configuración personalizada.

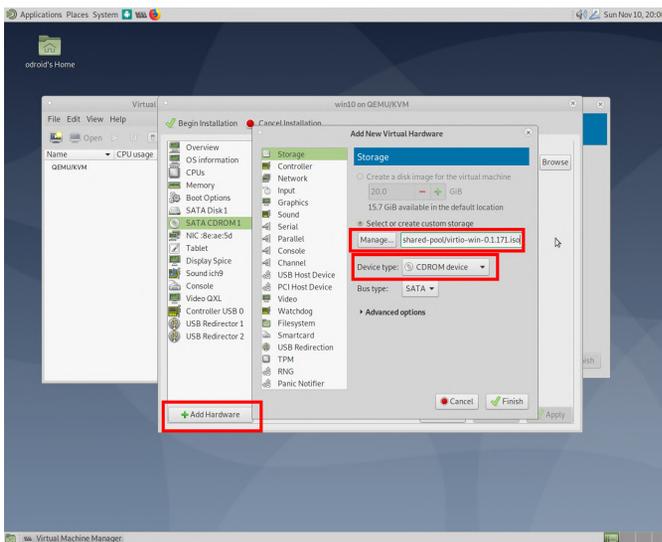


Figura 11 - Añadir un nuevo dispositivo CDROM con virtio ISO

A continuación, necesitamos ajustar parte del hardware existente; en concreto, nuestro adaptador de red (NIC) y el disco que queremos usar. Ambos dispositivos tienden a ser lentos en este tipo de virtualización con Windows, especialmente el rendimiento del disco, que en mi caso se ha convertido en un problema continuo bajo KVM. Animo a cualquiera que vaya a usar máquinas virtuales de Windows (servidor o cliente) a indagar sobre este tema, estoy abierto a cualquier opción o sugerencia para mejorar el rendimiento.

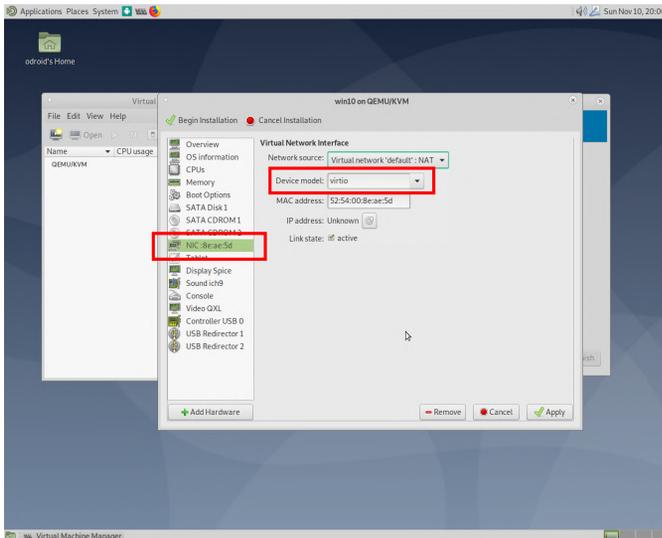


Figura 12: NIC debe ser un modelo de dispositivo: virtio para un mejor rendimiento

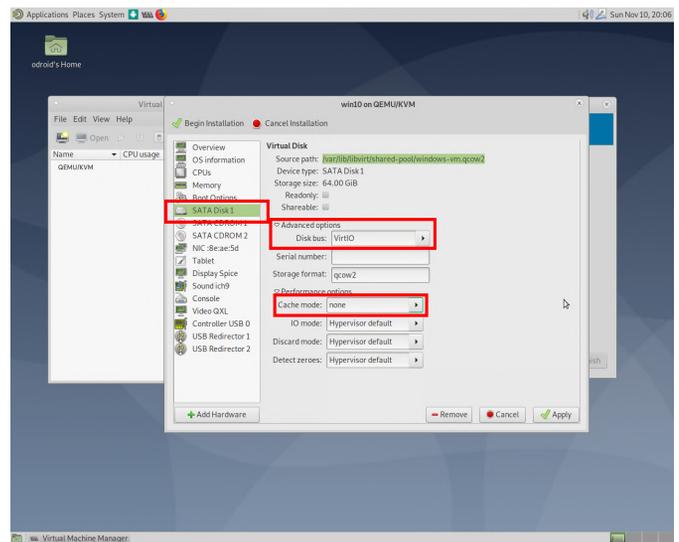


Figura 13: Sin el bus de disco VirtIO, la velocidad del disco podría llegar a ser tan lenta como 1 MB/seg.

El adaptador de red debe cambiarse al modelo de dispositivo virtio para obtener el mejor rendimiento. Lo mismo ocurre con el bus de disco de nuestro disco duro virtual. En las opciones de rendimiento, debe configurar el modo de caché en "none". Esto debería aumentar el rendimiento de la escritura, que es el mayor problema de los hosts de Windows. El formato de almacenamiento qcow2 es el predeterminado para las máquinas virtuales, pero hay quien recomienda usar "raw" como formato de almacenamiento. Aunque esto puede aumentar el rendimiento, también elimina algunas características, como las instantáneas de las máquinas virtuales, así que asegúrate de no necesitarlas. Finalmente, haz clic en el botón "Begin Installation" en la esquina superior izquierda de la ventana de configuración, y KVM iniciará nuestra nueva VM y arrancará el CD de Windows.

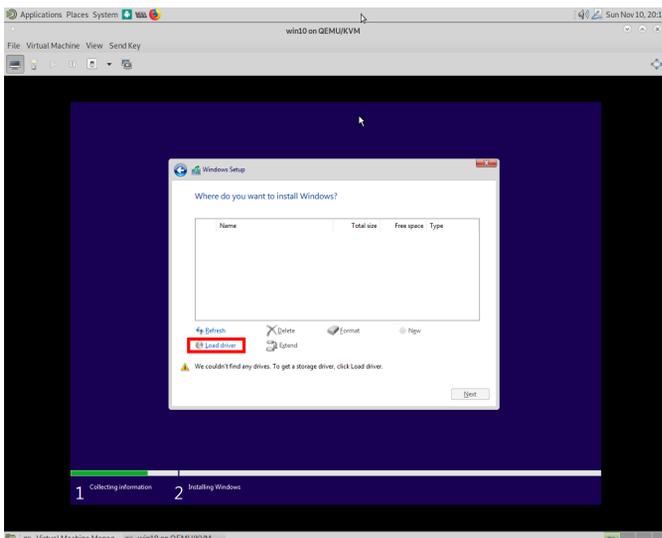


Figura 14: Windows no sabe cómo manejar discos VirtIO

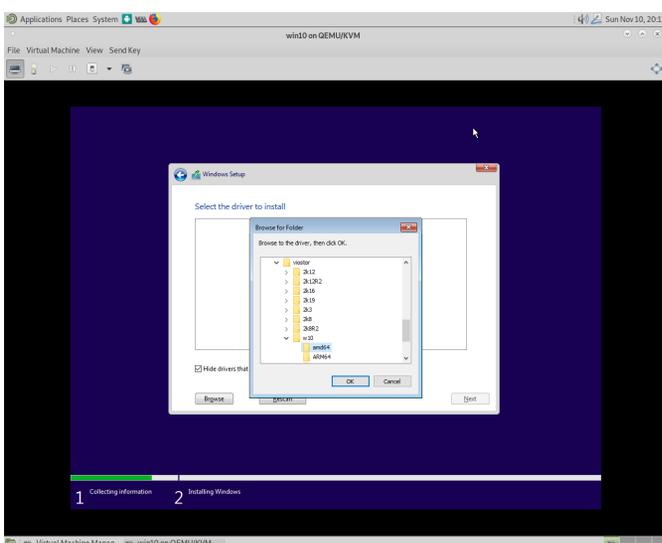


Figura 15 - Instalación de controladores adicionales durante la instalación de Windows

Cuando inicies por primera vez la configuración de Windows y vayas a seleccionar el disco duro en el que quieres instalar Windows, verás que la lista está completamente vacía. Esto se debe al hecho de que Windows no tiene ningún controlador para las unidades de disco basadas en VirtIO. Una característica de Windows muy utilizada en Windows 95 o 98, pero prácticamente olvidada actualmente, es necesaria para que todo funcione: la instalación de controladores adicionales durante la instalación de Windows. Para esto, selecciona el botón "Load drivers" en la esquina inferior izquierda de la ventana de instalación y navega hasta el CD que montamos en la segunda unidad de CD. Dirígete a la carpeta "viostor" y luego al sistema operativo y la arquitectura que desees instalar. En mi caso es w10/amd64, ya que instalé Windows 10 en una placa de 64 bits (ODROID-

H2). Si tu sistema operativo no está en la lista, elige uno que sea similar, debería funcionar de todos modos. Tras instalar el controlador, la configuración encontró el disco duro y pude continuar instalando Windows como de costumbre. Tras completar la configuración y reiniciar el sistema, Windows se cargó con normalidad, pero se detuvo nuevamente cuando intentó conectarse a Internet. No instalé los controladores de la red cuando instalé los controladores del disco duro. Para ser honesto, no sé si podría haberlo hecho, y no lo intenté. Probablemente podría haber instalado todos los controladores necesarios en aquel momento, pero no es importante ya que puedes continuar sin ellos e instalar los controladores más tarde.

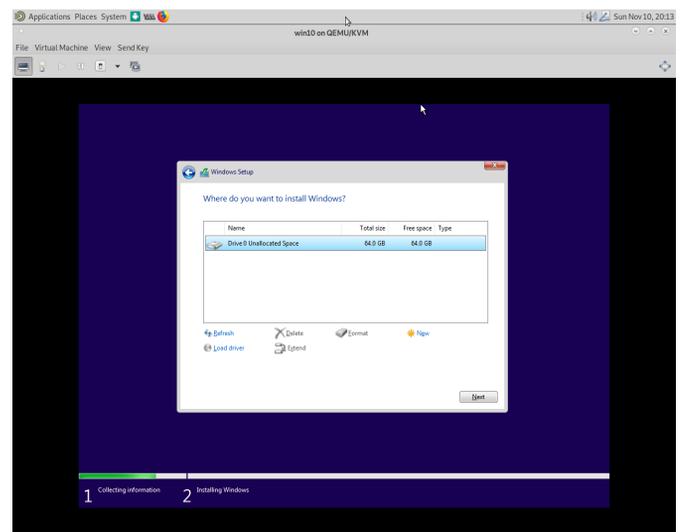


Figura 16 - Después de instalar los controladores para el bus de disco VirtIO, Windows encontró el HDD

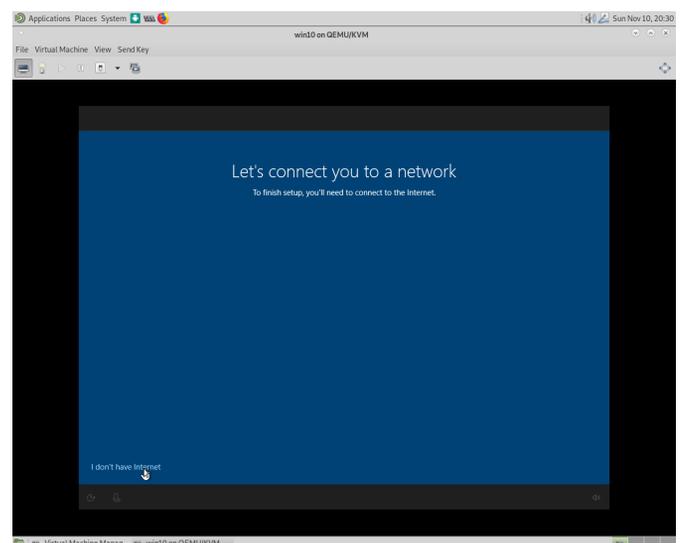


Figura 17 - La red fue algo a posteriori

Una vez que Windows esté en funcionamiento, simplemente navega hasta Sistema → Administrador

de dispositivos en Windows, y encontrará los dispositivos donde faltan las unidades. Haz clic derecho sobre ellos y selecciona actualizar controladores, luego navega hasta el CD con los controladores virtio. No necesitas seleccionar la carpeta correcta de los controladores, ya que Windows los encontrará por sí solo.

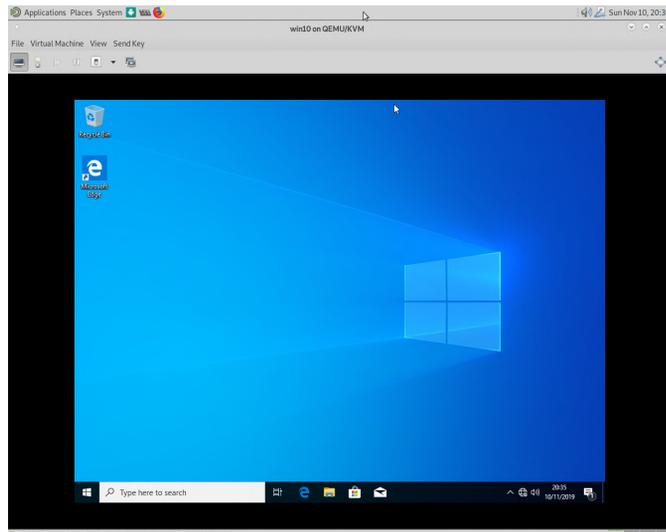


Figura 18: Windows arrancó bien incluso sin red

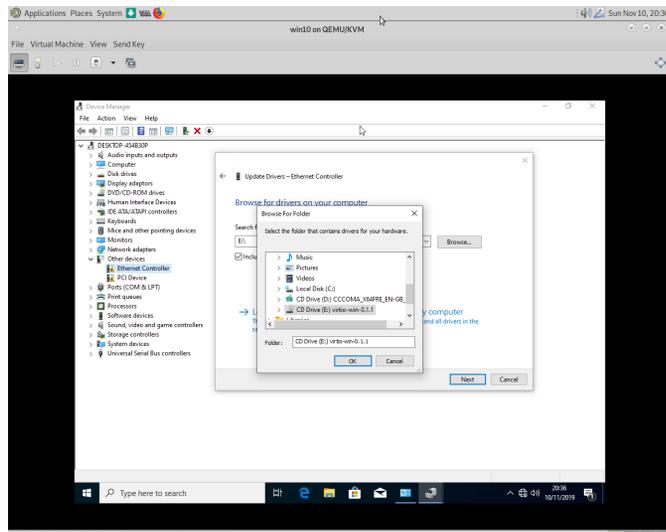


Figura 19 - Instalando de controladores que faltan; simplemente selecciona el CD y Windows hace el resto

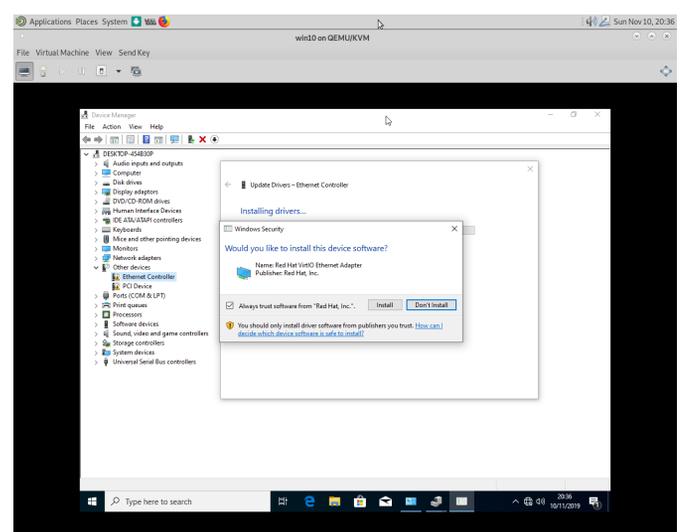


Figura 20: Windows te pregunta si debes instalar el controlador

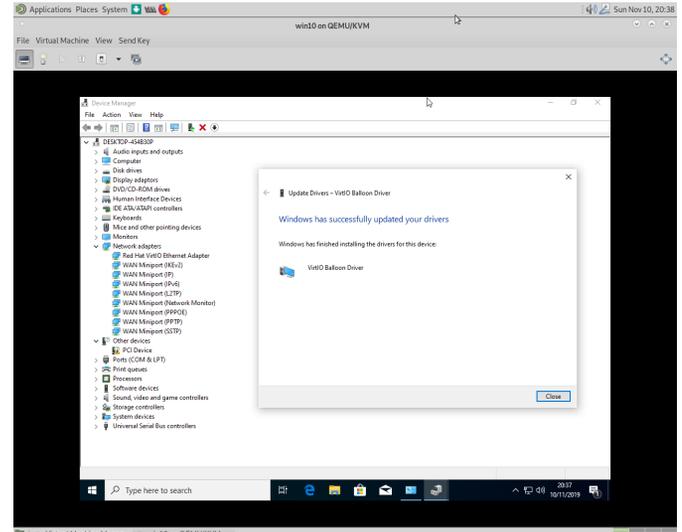


Figura 21 - Instalación del controlador Balloon del dispositivo PCI

Instalar el **Balloon Driver** "PCI Device" suscita cierta duda. Algunas personas afirman que el sistema funciona mejor sin él, pero te dejaré que decidas tú mismo. Lo que hace es llenar la RAM con un pseudo proceso, y dependiendo de la RAM que necesite el sistema operativo host o el sistema operativo invitado para ejecutar las aplicaciones, se expande o se reduce su tamaño, y con ello permite que el host o el invitado utilicen más RAM, dependiendo de sus necesidades. Llegado a este punto, Windows debería estar funcionando y debería comportarse como cualquier otra máquina virtual de Windows.

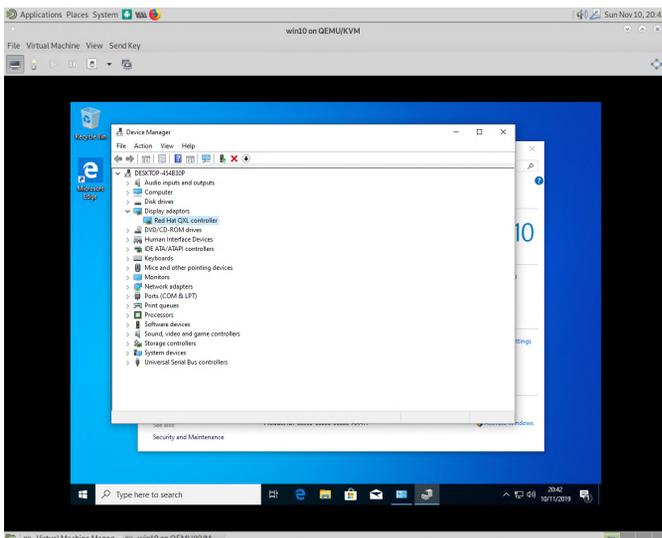


Figura 22: Ahora podemos ver como Windows hace lo que mejor sabe hacer

No obstante, puede instalar controladores adicionales. Spice es una interfaz gráfica para conectarte a la pantalla de la VM. En lugar de usar VNC, como otros hipervisores, Spice es usado por defecto en QEMU. Es más rápido y responde mejor que VNC. Para Windows hay herramientas especiales para invitados que pueden mejorar los gráficos y otras características de Windows. Verlo como si fuera VMWare o VirtualBox como herramientas/ampliaciones para invitados solo para QEMU/Spice. Puedes descargarlas e instalarlas desde <https://www.spice-space.org/download/binaries/spice-guest-tools/>.

Puedes comprobar si está instalado correctamente verificando tu adaptador de gráficos tras la instalación de spice-guest-tools.

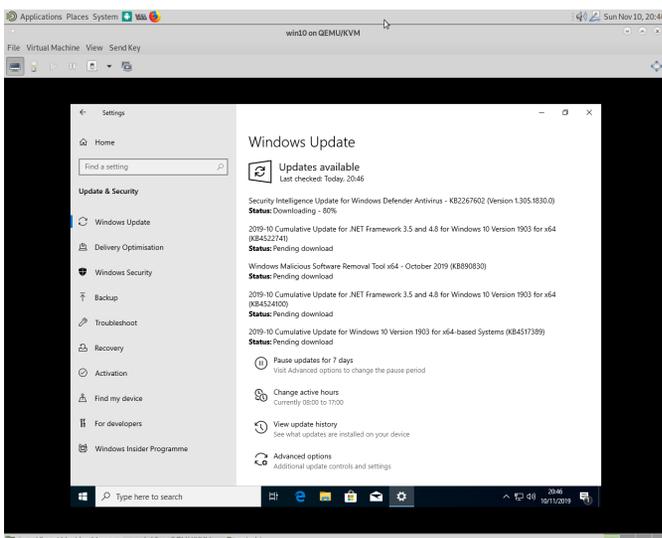


Figura 23: el controlador Red Hat QXL es el nuevo controlador de GPU para tu VM con herramientas de invitado spice instalada

Hay una característica de la que quiero hablar, ya que también se ha utilizado en otros hipervisores: instantáneas. El Virtual Machine Manager puede manejar instantáneas de máquinas virtuales muy bien, igual que VMWare, o VirtualBox o cualquier otro hipervisor. Simplemente abre la pestaña Instantánea en tu VM, haga clic en el símbolo más y crea una instantánea.

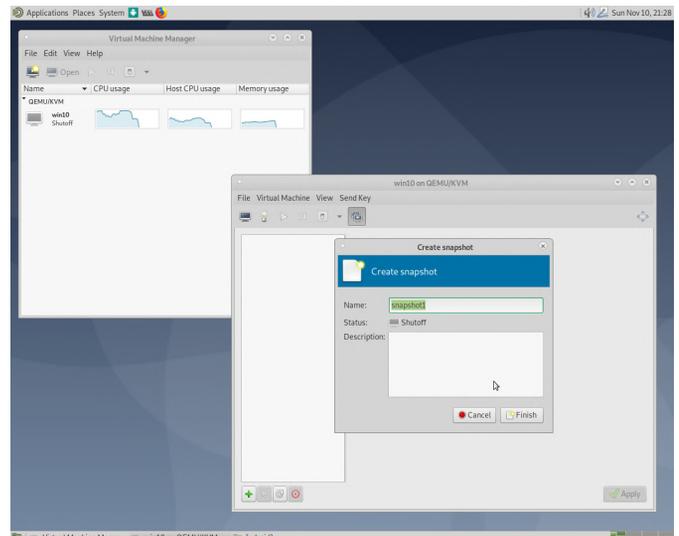


Figura 24: Creación de una nueva instantánea para nuestra VM de Windows

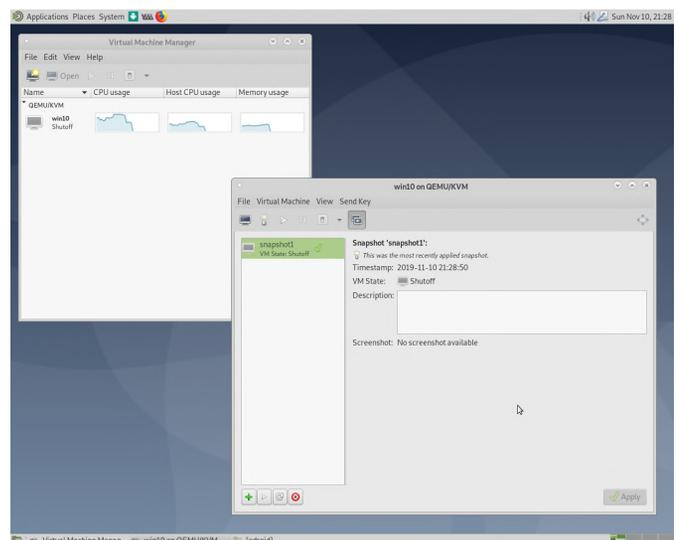


Figura 25: Detalles de la instantánea: puedes tener varias instantáneas para una máquina

Las instantáneas son una buena manera de volver a un estado anterior, lo que las hace esencial a la hora de probar cosas nuevas que podrían afectar el sistema (por ejemplo, la actualización del sistema operativo), ten en cuenta que las instantáneas pueden reducir el rendimiento general y aumentar mucho el uso del disco. Supongamos que tiene una VM con dos discos: uno para el sistema operativo y

otro con un tamaño de 20 GB como almacén de datos. Rellena el disco de datos al máximo (20 GB) y realiza una instantánea. Luego intercambia todos los datos en el disco de datos. Si verificas en el sistema host, donde se encuentra tu archivo qcow2 del disco de datos, notarás que el tamaño del archivo ha aumentado aproximadamente 40 GB. Aunque el límite de tamaño del disco es de 20 GB, ahora necesita 20 GB adicionales para almacenar la diferencia entre la instantánea y los nuevos datos. Ten esto en cuenta a la hora de trabajar con instantáneas, y también que las instantáneas solo deberían ser una solución temporal, no una forma de organizar o respaldar tus datos.

Hasta la próxima

En el siguiente artículo sobre este tema, hablaré de características avanzadas, como las agrupaciones de almacenamiento compartido y la migración en tiempo real de máquinas virtuales de un host a otro, lo que significa mover una VM mientras se ejecuta de un PC a otro sin interrupción. También quiero presentar "virsh", que es la herramienta de línea de comandos para libvirt, y también mostraré cómo puedes conectarte a tus hosts de virtualización desde un sistema remoto, para que no necesites una instalación de escritorio para ejecutar tus máquinas virtuales.

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

© December 1, 2019 By Dave Prochnow Android, Juegos, ODROID-C2, ODROID-N2



Nuestra larga espera finalmente ha llegado a su fin: ¡Google Stadia, este servicio universal de streaming de juegos, ya ha sido lanzado! Se estrenó a fines de noviembre de 2019.

Como ya se ha comentado en anteriores artículos de esta columna, para esta fecha de lanzamiento, Google Stadia SOLO estaría disponible para los suscriptores de la Edición Stadia Founder. Como recordarás, este paquete de 129\$ incluía un mando, Chromecast Ultra para TV, 3 meses de Stadia Pro y una copia "GRATIS" de Destiny 2. Si no te habías suscrito a la Edición Founder, tendrás que esperar tres meses para poder experimentar Stadia.

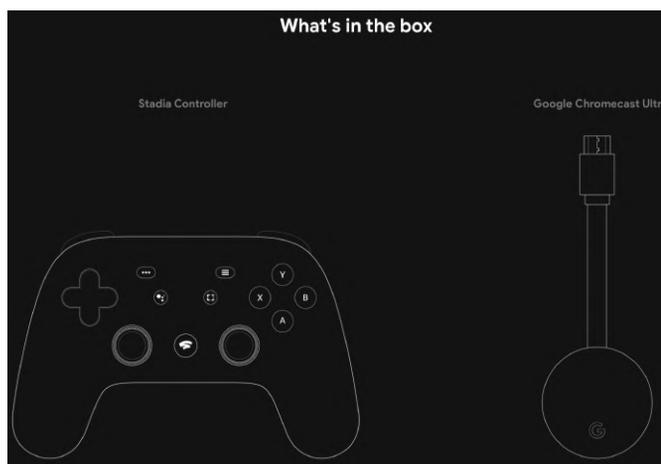


Figura 1: Los suscriptores de Founder's Edition obtienen un componente de hardware con su compra

Tras concluir el periodo de tres meses de los suscriptores Founder, los suscriptores normales podrán unirse a la fiesta del streaming de juegos. Esta opción de Stadia Pro tendrá un coste de 9.99\$ al mes (tres meses de esta opción ya estaban incluidos con la suscripción de Founder's Edition) más el coste de

comprar cualquier juego. De momento no hay nada gratis.

En el evento Made by Google de octubre, se mencionó que se lanzaría un "nivel gratuito" en 2020. Esta opción incluirá sonido estéreo y transmisión de juegos de 1080p. La opción gratuita contrasta con Stadia Pro mencionado anteriormente, streaming de juegos 4K, color HDR, sonido envolvente 5.1 y un juego gratis al mes. Independientemente de tu nivel de suscripción, recibirá al menos 60 frames por segundo (fps) al reproducir los juegos.



Figura 2 - El mando Stadia

Desafortunadamente, hubo algunos "detalles" que fueron mencionados en el evento de octubre que decepcionaron a muchos de los fieles de Stadia.

Dentro de estas cuestiones que sorprendieron a algunos seguidores de Stadia estaba la limitación de los dispositivos móviles que podrán acceder a Stadia en el momento de su lanzamiento. Solo los teléfonos Pixel 3 y 3a (y presumiblemente la nueva familia de teléfonos inteligentes Pixel 4) y las tablets Chrome OS (por ejemplo, Pixel Slate, HP Chromebook X2, etc.) tendrán acceso. Además, el mando Stadia solo funcionará en modo inalámbrico con Chromecast Ultra. Sin embargo, cualquier ordenador con un navegador Google Chrome funcionará con el controlador Stadia utilizando una conexión USB con cable y tendrá acceso a Stadia. Además, se podrá utilizar mandos de terceros con Chrome en el ordenador desde donde ejecutes Stadia.

Bring Stadia to your TV with the included Chromecast Ultra.³

Play on your TV in up to 4K Ultra HD with HDR.²

Stunning graphics at up to 60 FPS.¹

Works seamlessly with the Stadia Controller.⁴



Figura 3: Haz que Tu televisor se convierta en un "juego" Stadia con Chromecast Ultra7

En cuanto a los juegos, Google había indicado previamente que habría disponibles 31 juegos el día del lanzamiento de Stadia. Puesto que la mayor parte de estos títulos ya se habían lanzado, este objetivo se alcanzaría muy posiblemente (sin olvidar que dos de estos 31 títulos son estándares de DOOM). Afortunadamente, muchos desarrolladores de juegos quieren subirse al tren de Stadia, así que esta lista de juegos está aumentando. De hecho, tanto Red Dead Redemption 2 como Orcs Must Die 3 deberían unirse a la fiesta de Stadia en marzo de 2020.

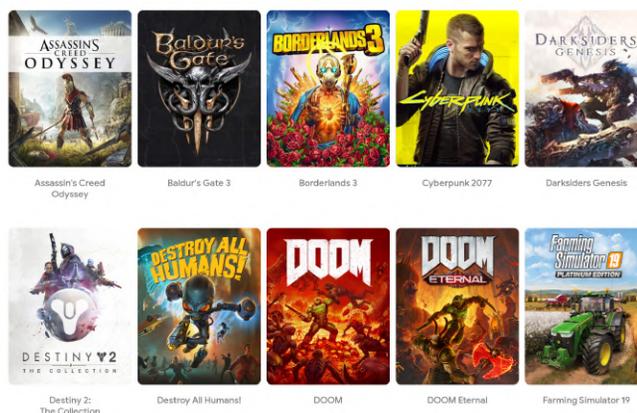


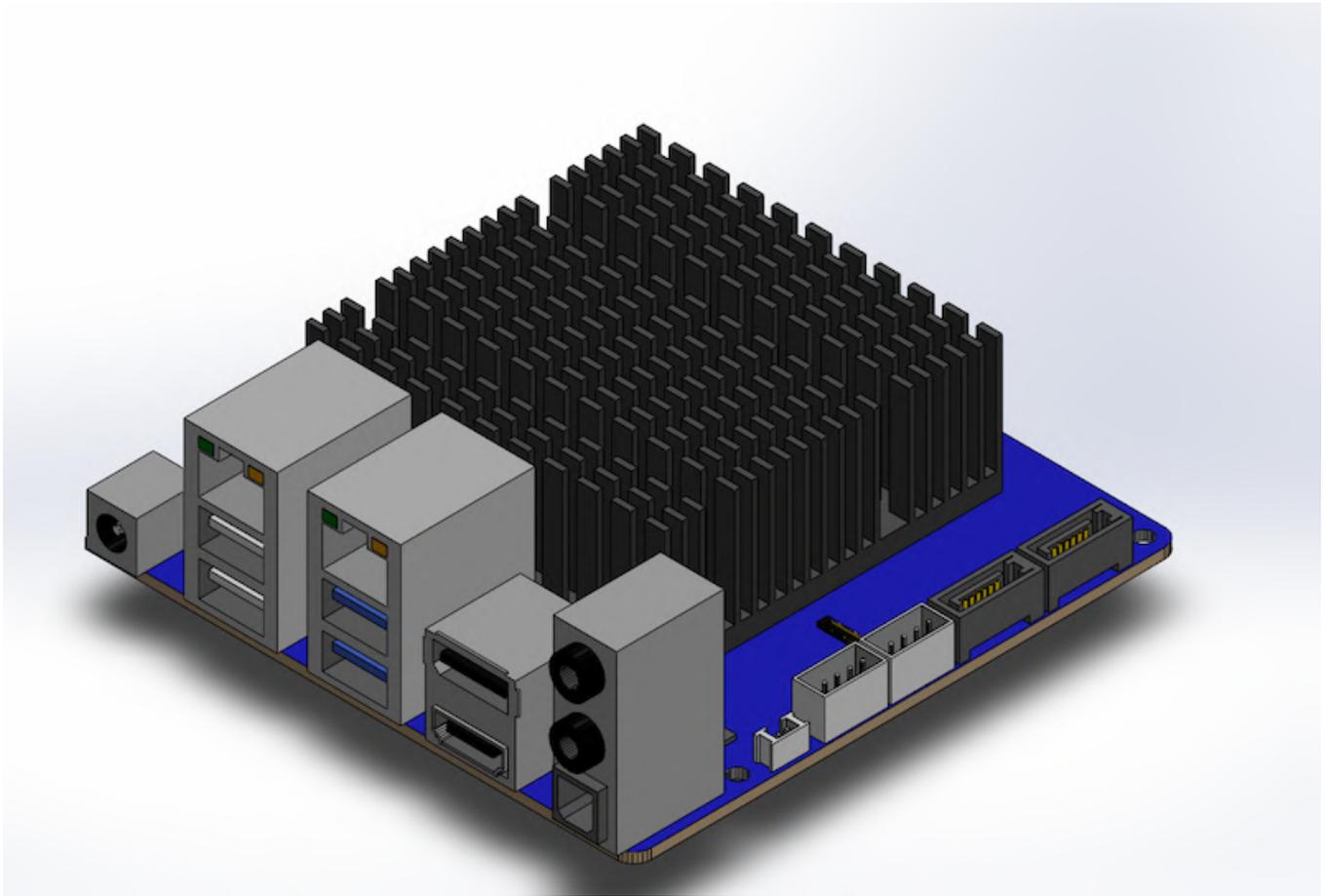
Figura 4 - Una muestra de juego Stadia; Destiny 2: the collection es gratis para los suscriptores de Founder

¿Es este un inicio lleno de baches para Stadia o simplemente un problema menor que quedará en el olvido fácilmente? Según el jefe de Stadia, Rick Osterloh, "Stadia tiene como objetivo ofrecer los mejores juegos jamás creados en casi cualquier pantalla".

Si aún estás un poco perdido con todo este mundo de Stadia, hay un video hecho por Google en <https://www.youtube.com/watch?v=Pwb6d2wK3Qw> que explica "cómo funciona Stadia" o, como también es conocido, "Stadia 101".

Desarrollando un BladeCenter ODR0ID-H2: Crea una Estación de Computación de Alto Rendimiento con Micro Huella

© December 1, 2019 By @rvalle ODR0ID-H2, Mecanico



Tras verme la inspiración del excelente modelo OpenSCAD H2 publicado en <https://forum.odroid.com/viewtopic.php?f=172&t=33824>, he creado un remix del fantástico Raspberry Blade Center para alojar 3 unidades ODR0ID-H2. Hice los siguientes cambios del proyecto original:

- Migrados los archivos a SolidWorks
- Fabricar un ensamblaje
- Alineados los mecanismos de cierre
- Ampliado el carro en 2 mm para que se mantuviera en su lugar
- Pilares creados para los insertos de latón PSM por Heatlok y Minitch

Los archivos de SolidWorks están disponibles en <https://www.thingiverse.com/thing:3929164>.

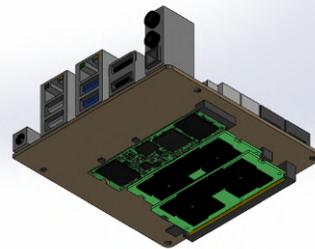


Figura 1: Para un modelo más realista, aquí tienes un ensamblaje del ODR0ID-H2 con periféricos y disipador de calor

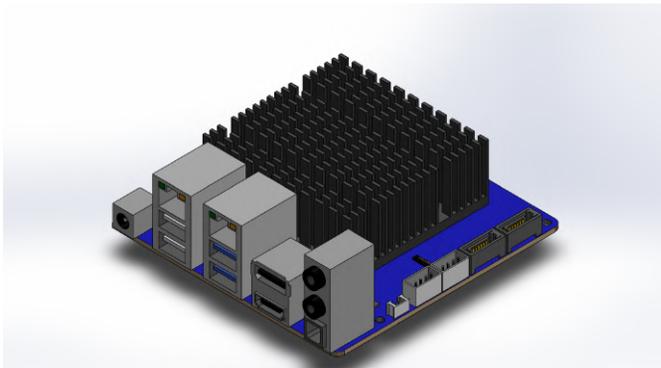


Figura 2: Aquí tienes otra vista del ensamblaje del ODROID-H2 con periféricos y disipador de calor

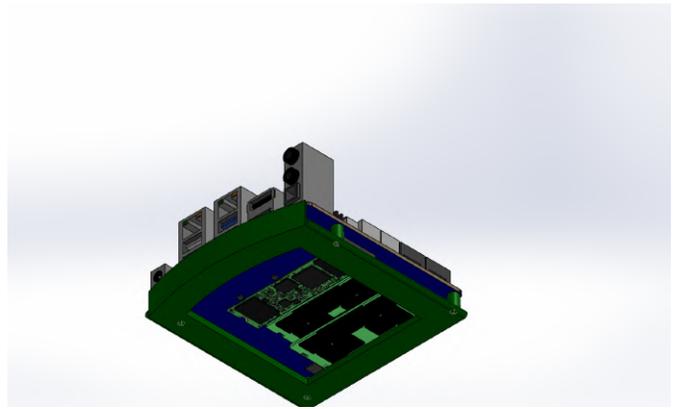


Figura 6 - El diseño de la cajita

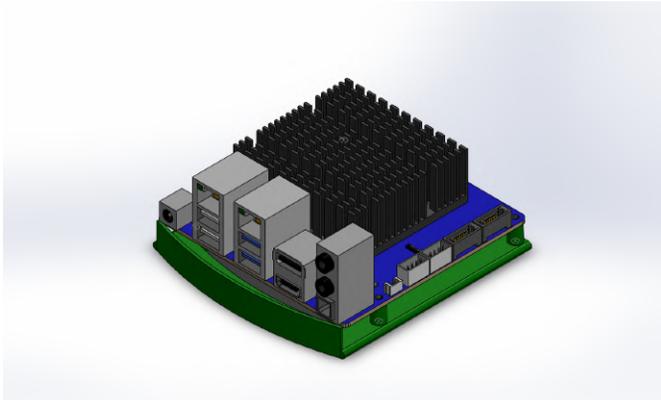


Figura 3 - El diseño de la cajita

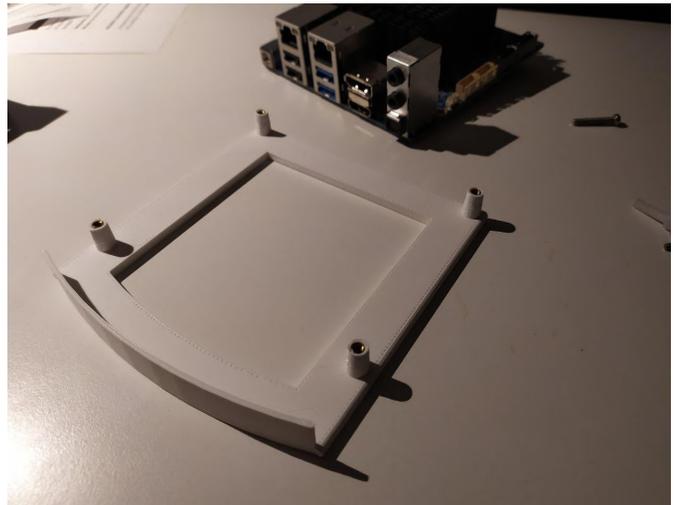


Figura 7: La primera impresión en 3D que funciona muy bien, pero necesita pilares más gruesos

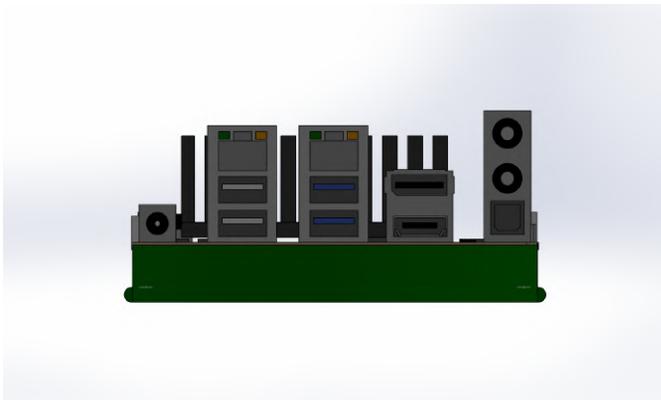


Figura 4 - El diseño de la cajita

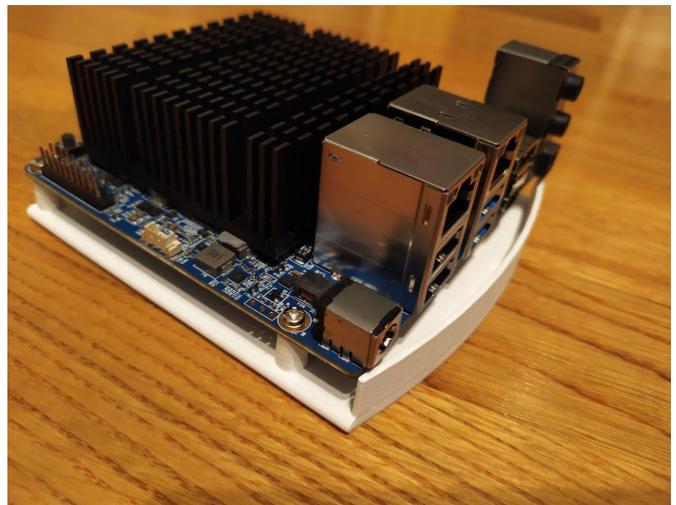


Figura 8: La primera impresión en 3D que funciona muy bien, pero necesita pilares más gruesos

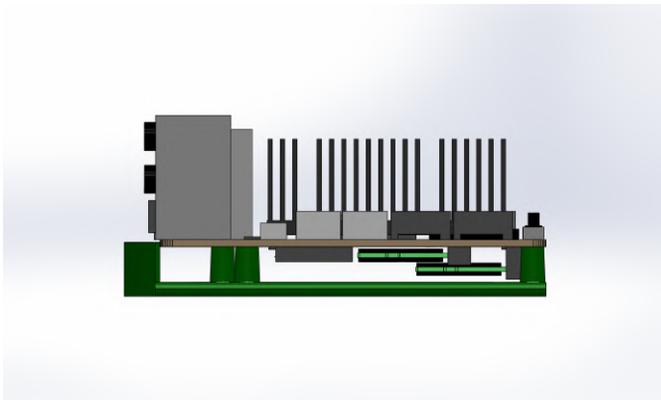


Figura 5 - El diseño de la cajita

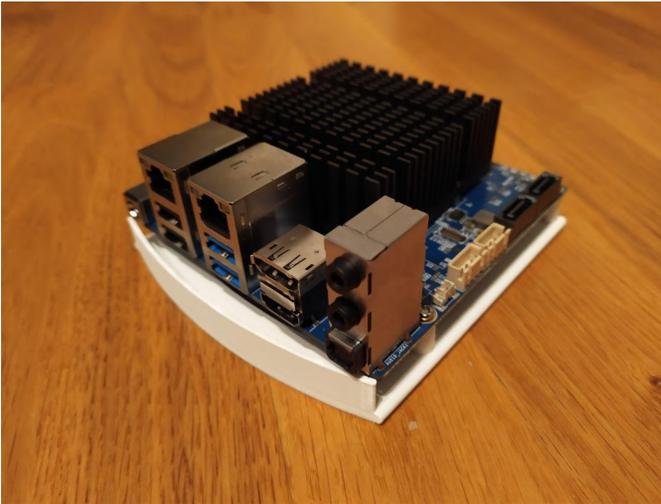


Figura 9: La primera impresión en 3D que funciona muy bien, pero necesita pilares más gruesos

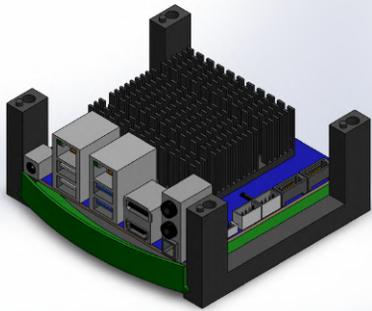


Figura 10: Decidí hacer un hack inusual, dejando que cada conector de audio se integrara en el siguiente bloque

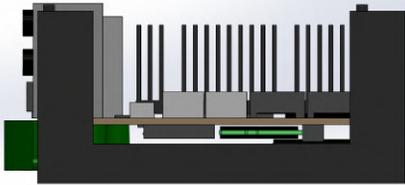


Figura 11: el bloque de audio está más alto que el soporte, como puedes ver en esta vista lateral

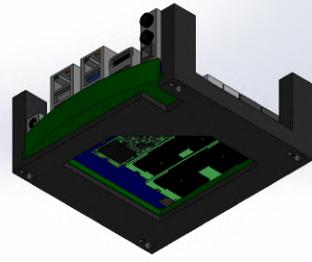


Figura 12: Se ha creado un zócalo en la base del soporte para insertar el conector de audio del bloque H2 anterior



Figura 13: Aquí puedes ver cómo cada conector de audio se integra en el siguiente bloque, ahorrando espacio para permitir 8 unidades en 3U

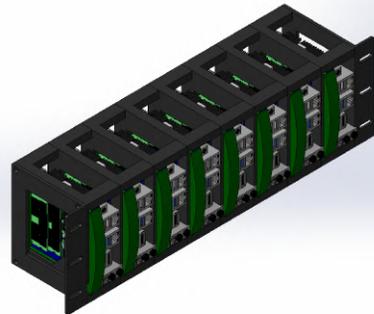


Figura 14: una vez que los juntas, el centro del blade se ve así

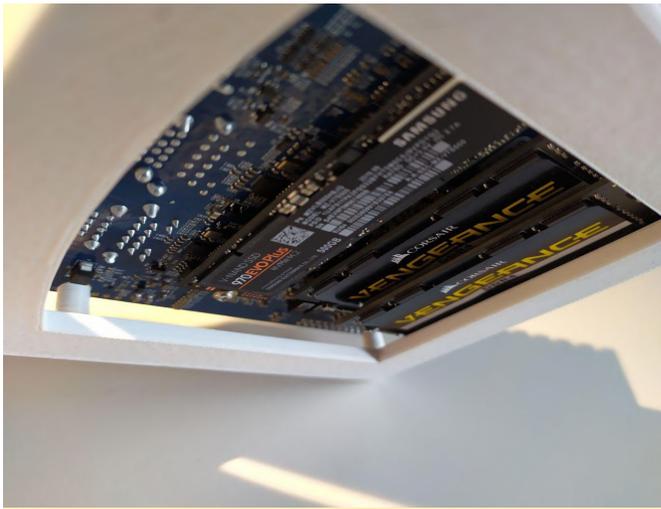


Figura 15 - Después de conectar la RAM y SSD



Figura 18: La segunda impresión de montaje, donde las agarraderas de alineación funcionan muy bien, aunque los agujeros deben limpiarse de plástico para que puedan hacer su trabajo correctamente.

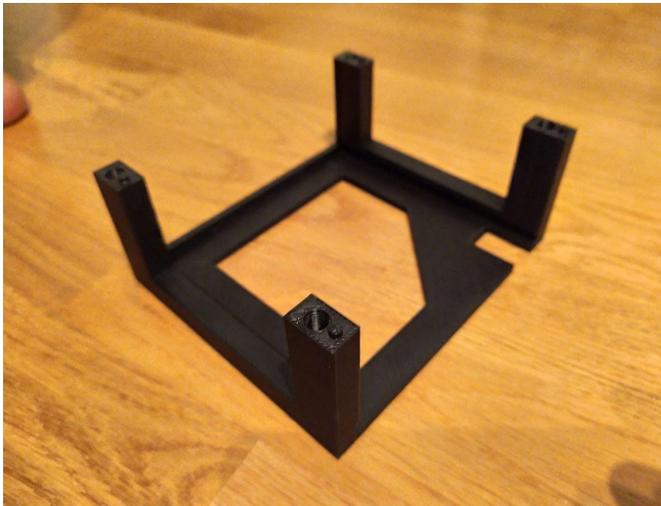


Figura 16 - La primera impresión de montaje

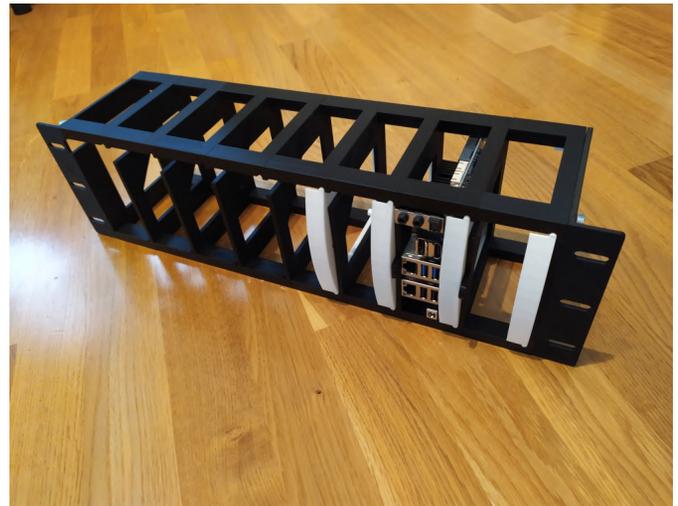


Figura 19: todo el bastidor ensamblado, donde implementé un sistema de ajuste a presión para que las cuchillas no se salgan del soporte con facilidad, con un clic y una pestaña auxiliar para sacarlas, similar a los cables Ethernet

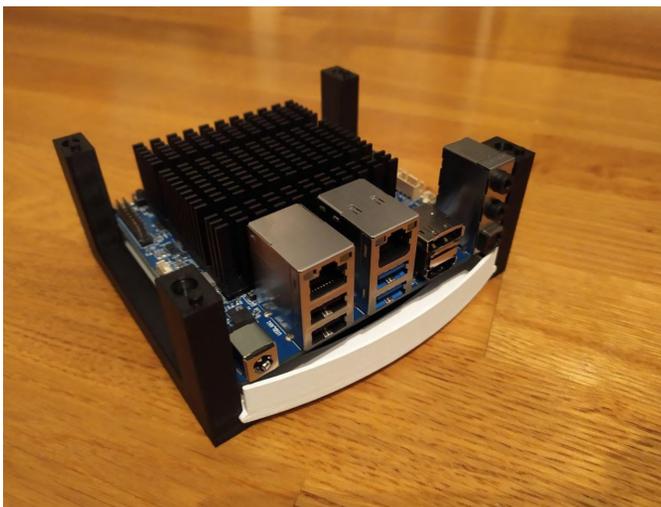


Figura 17 - La primera impresión de montaje

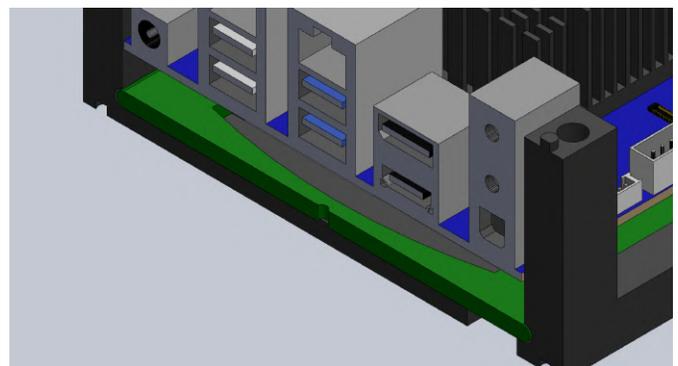


Figura 20: Aquí hay una sección para ver el sistema de bloqueo funcionando

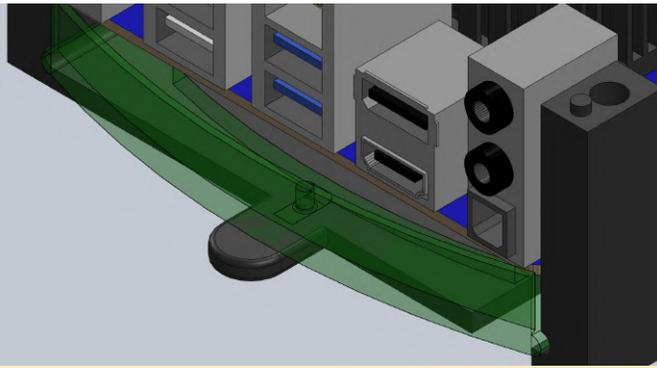


Figura 21: Aquí puedes ver cómo interactúa la cajita y la pestaña

- 8x cajitas H2 con insertos recomendados (ya sea versión de insertos o autorroscante), aunque requiere insertos de latón.
- 8x soportes h2 (versión vainilla, pressfit o pressfit & tab) recomendada la que tiene pestaña.
- 1x oreja izquierda
- 1x oreja derecha

Equipo

- 8x placas H2
- 3U disponible en un rack (de pequeña profundidad/rack de red está bien)
- PSU



Figura 22: el conjunto final montado en un bastidor, con 3 unidades ODROID-H2 instaladas

Cuestiones pendientes

No he probado la versión de tornillos autorroscantes de la cajita La fuente de alimentación distribuida no es apta para rack.

Comentarios de diseño de ODROID-H2 Para un uso particular del H2 como una hoja de rack, he encontrado los siguientes problemas:

- Los conectores de audio están demasiado altos y deben mantenerse dentro de los límites de los otros conectores.
- La ubicación de los interruptores de encendido/reinicio es un inconveniente, y sería mejor tenerlos en la parte delantera de la cajita
- La posición de los LED también es un inconveniente

Hardware Requerido

- 4x varillas, M6x44.5 cm (metal o nylon)
- 4x tuercas M6
- 32x Inserciones de latón HeatLok Mx opcionales de PSM Internacional o similar.
- 32x Tornillos opcionales de M3x5mm para la versión de inserción
- 32x Tornillos autorroscantes opcionales de 5-6 mm de largo (el orificio es de 2.5 mm de ancho)

Componentes impresos

Para obtener más información, comentarios, sugerencias y preguntas, visita la página del proyecto en <https://www.thingiverse.com/thing:3929164> o la publicación del foro ODROID en <https://forum.odroid.com/viewtopic.php?t=36780&p=272149>.

Sobreviviendo a un Corte de Luz: Cómo Mantener un Negocio de Comercio Electrónico Durante un Corte de Energía Regional

© December 1, 2019 By www.ameridroid.com Tutoriales



Imagínate si se fuera la luz en tu negocio. ¡Pero no solo en tu negocio, en tu ciudad, en tu comarca y toda tu región! A esto se enfrentó ameriDroid debido al apagado de energía de seguridad pública de PG&E que comenzó el sábado 26 de octubre de 2019 y duró hasta la tarde del miércoles 30 de octubre y que afectó a su almacén del norte de California.



Figura 1

Nuestro Desafío

El personal de ameriDroid no quería retrasar el envío de paquetes a sus clientes, y tampoco quería que los clientes recurrieran al teléfono para ponerse en contacto con ellos. Así que sabían que no podían

simplemente relajarse y no hacer nada, hasta que volviese el suministro eléctrico.



Figura 2

Nuestra infraestructura

Puesto que ameriDroid es un distribuidor de ordenadores de placa reducida (<https://ameridroid.com/collections/single-board-computer>), minorista (<https://ameridroid.com/collections/all>), y mayorista (<https://ameridroid.com/pages/corporate-orders>), siempre hemos querido estar íntimamente familiarizados con los productos que vendemos. Además de la investigación y la teoría, usar nuestros productos en el mundo real es una de las mejores maneras de hacerlo. Desde el principio decidimos que funcionaríamos tanto como fuera posible utilizando únicamente los productos que vendemos. También estamos muy comprometidos a minimizar nuestro impacto sobre el medio ambiente. Hacemos esto de varias formas:

- No recurrimos al papel en las operaciones que no sean: cajas de cartón y material de embalaje que utilizamos para el envío, etiquetas de envío y recibos de embalaje.
- Reciclamos cualquier exceso de cartón, papel y material de embalaje que obtenemos de otras fuentes.
- Nuestro personal de envío utiliza Tablet Android para la mayoría de su trabajo en el almacén.
- Nuestra infraestructura de almacén funciona con SBC ODROID basados en (<https://ameridroid.com/collections/odroid>) con un sistema Intel integrado de baja potencia (<https://ameridroid.com/products/beelink-sii-mini->

pc) (para los periféricos que se ejecutan con Windows, como balanzas digitales e impresoras de etiquetas), esto nos aporta redundancia y contar con bajos consumos de energía.



Figura 3



- Nuestro sistema de seguridad de video incorpora unidades de cámara Raspberry Pi diseñadas a medida con un ODROID-U3 que actúa como un DVR, que es un sistema extremadamente eficiente.



Figura 5

- El sistema de seguridad de las instalaciones de nuestro sitio funciona con un ODROID-XU4Q y sensores

inalámbricos de movimiento y de puerta con batería, también extremadamente eficientes.

- Nuestra instalación de almacenamiento periférico funciona con una matriz solar autónoma, pero está demasiado lejos de nuestro almacén principal para actuar como fuente de alimentación.

Tenemos conexión a Internet desde 3 proveedores diferentes por redundancia, solo uno terminó siguió activo al 100% durante el tiempo que duro el apagón regional:

- Comcast: inactivo debido a un fallo regional de equipos.
- Verizon: marginalmente operativo con alta latencia, con un 50% de pérdida de paquetes debido a la saturación de la red.
- Pacific.net Bonded DSL - Totalmente operativo

Puesto que nuestra empresa matriz (<http://respectech.com/>), nos proporciona la infraestructura principal de equipamiento de industria, tuvimos que configurar una red temporal para lograr que ameriDroid trabajase con la conexión Pacific.net. El rack de servidores de Respectech requiere 30A de energía para funcionar, por lo que no era factible alimentar estos servidores basados en Windows y CentOS con nuestra solución autónoma. Los sistemas telefónicos de ameriDroid también funcionan fuera de la infraestructura de Respectech. Afortunadamente, las llamadas de ameriDroid se transfieren automáticamente a los teléfonos móviles de nuestro personal cuando se produce un corte de luz.

Nuestra solución



Figura 6

En nuestro primer día de funcionamiento sin red eléctrica, recurrimos a ameriBus para que nos proporciona energía con su convertidor de onda sinusoidal con picos de 4000W. El ameriBus tiene asientos perimetrales tipo limusina, así que instalamos este convertidor para conseguir que el ameriBus actuase como una sala de conferencias portátil con un equipamiento de demostración para nuestras soluciones SBC en ferias, conferencias y cuando visitáramos a nuestros clientes de la costa oeste. También nos fue muy útil en nuestra situación de desconexión eléctrica. Aunque este sistema funcionaba perfectamente bien, el ameriBus consumía aproximadamente un galón de combustible cada 4 horas en ralentí debido a su gran motor v10. Sabíamos que podíamos hacerlo mejor. Al día siguiente, recorrimos a un convertidor de onda sinusoidal de 1000W que se usa en el sistema de energía solar de nuestras instalaciones de almacenamiento periférico y lo conectamos a la furgoneta ameriDroid. La furgoneta tiene un motor de 6 cilindros mucho más pequeño. Este cambio nos permitió trabajar con aproximadamente 1 galón de combustible por cada 8 horas de funcionamiento. Puesto que operamos con SBC, tablet e impresoras térmicas de baja potencia, 1000 W de corriente eran más que suficientes. Una pequeña dificultad añadida fue que incluso durante el día, nuestro personal de envío tuvo que usar faros LED para trabajar en el almacén ya que la iluminación del techo no podía encenderse.



Figura 7

El resultado

Perdimos unas cuantas horas de productividad para poder configurar las soluciones de distribución de

energía autónoma y montar una red temporal para nuestra infraestructura de envío. Aparte de esto, estuvimos casi completamente operativos durante el tiempo del apagón de energía que duró 5 días, la mayoría de los pedidos de los clientes salieron a tiempo. Los pocos que no lo hicieron salieron al día siguiente. ¡Estaremos aún más preparados para el próximo desafío, gracias a las posibilidades que brindan los ordenadores de placa reducida!

Referencia

<https://ameridroid.com/blogs/ameriblogs/news-operating-e-commerce-business-during-regional-power-outage>

Reparado tu ODROID-N2: Cómo Recuperarse de un Cortocircuito Accidental

December 1, 2019 By Justin Lee ODROID-N2, Mecanico



En esta guía mostraremos cómo reparar el Q1 de tu ODROID-N2. El artículo original lo puedes localizar en la página wiki de Hardkernel disponible en: <https://wiki.odroid.com/odroid-n2/hardware/repairs>

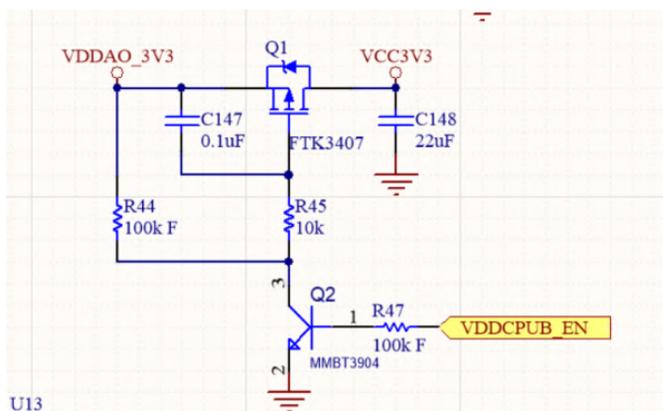


Figura 1 - Esquema del conmutador de carga discreta con FTK3407

Causas por las que falla el transistor Q1

Las sobrecargas de corriente, incluso de corta duración, pueden causar daños progresivos a un

MOSFET, a menudo con un pequeño aumento de temperatura notable antes del fallo. Los MOSFET a menudo tienen un pico de corriente nominal alto, pero estas ratios generalmente suponen corrientes pico que solo duran 300 μ seg más o menos. La sobrecarga de corriente acumulativa puede causar daños.

Las causas del fallo Q1 son generalmente de circuitos limitadores de corriente incorporados en conmutadores de potencia integrados para evitar que la carga SW se destruya durante las sobrecargas. En el caso de SW de carga discreta, el circuito limitador de corriente no está integrado, de modo que el P-CH FET FTK3407 está dañado debido a la sobrecarga de corriente que hemos mencionado anteriormente.

- Cuando la eMMC se monta invertida, la sobrecarga fluye a VCC3V3 debido al cortocircuito del nodo FLASH_1V8 hacia GND.
- Error de conexión GPIO o cortocircuito GPIO en el puerto E/S de 40 pines.

Fallo de funcionamiento - debido a fallos

Tal y como se muestra a continuación, el conmutador de carga no suministra el nodo VCC3V3 debido a los daños y su fuente de alimentación no proporciona energía, de modo que el eMMC, USB, Ethernet y el pin IO no funcionan.

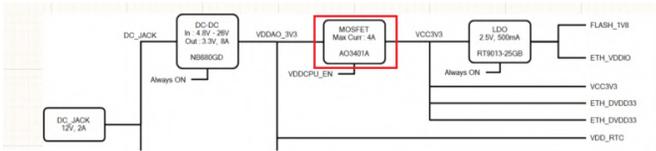


Figura 2 - Diagrama de bloques MOSFET que está dañado

Procedimiento de reparación

Nota:

- También puede reemplazar AO3407A alternativamente ya que FTK3470 no es fácilmente accesible.
- Asegúrate de dejar solos los componentes R44, R45 y C147 cerca de Q1. Las piezas pequeñas pueden salir volando al usar el soldador.

Paso 1. Básicamente, se puede ver que el Q1 está dañado por su apariencia tal y como se muestra en la figura 3.

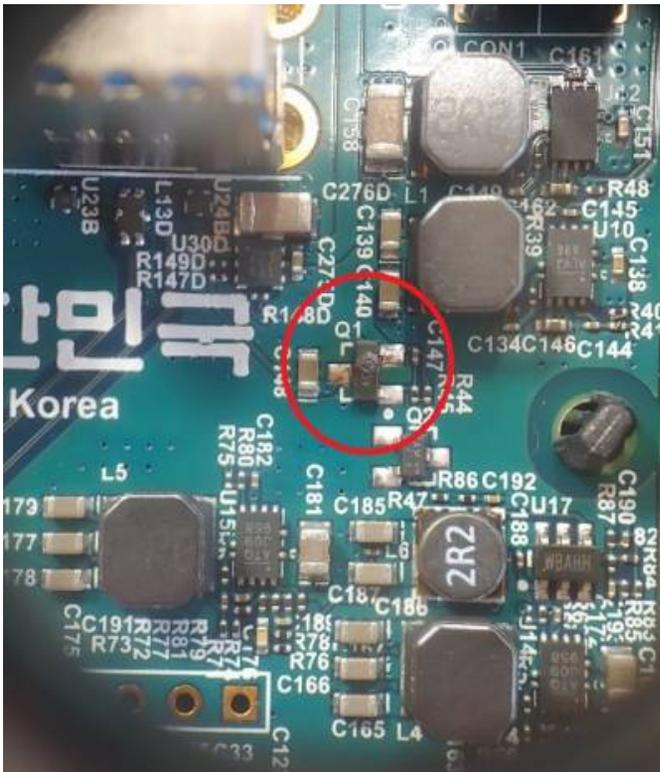


Figura 3 - Q1 resaltado en rojo

Paso 2. Para separar fácilmente Q1, debe añadir un poco de soldadura en la yema del Q1.

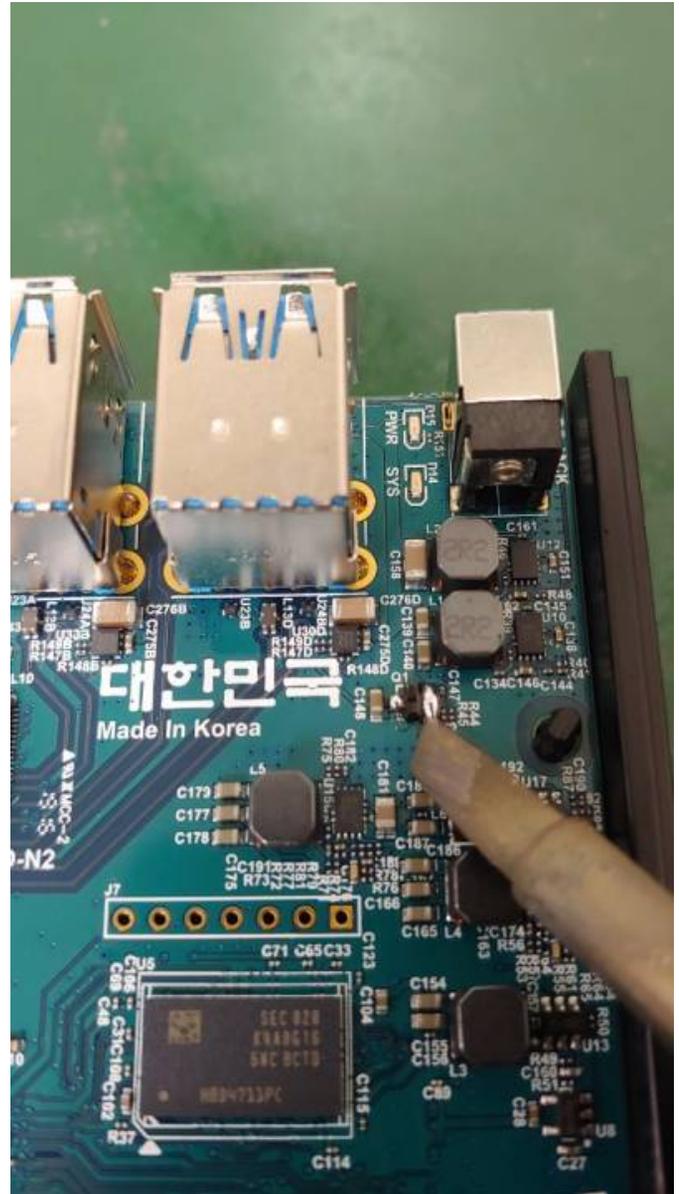


Figura 4: Soldadura añadida a Q1 para facilitar la extracción

Paso 3. Si tienes un soldador de repuesto, úsalo tal y como se muestra en la imagen. Si no tiene un soldador de repuesto, será algo más complicado que usar un único soldador, aunque se podría intentar fundir la soldadura de las tres yemas al mismo tiempo y luego retirar la pieza.

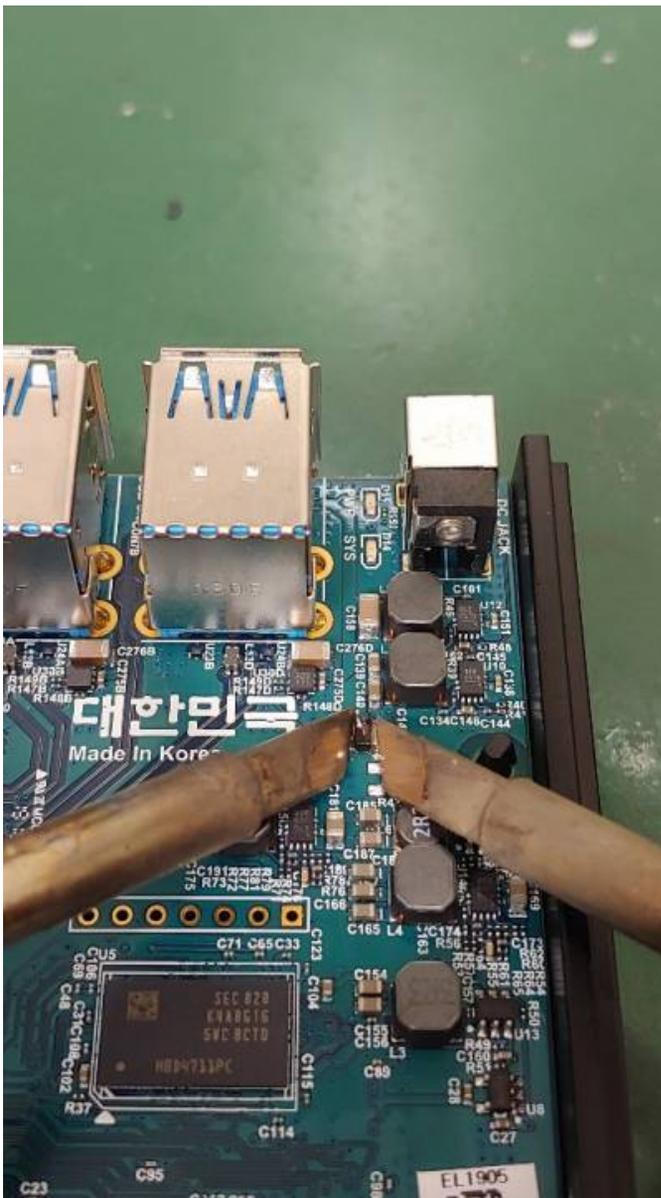


Figura 5: Usando 2 soldadores para eliminar Q1, calentando todas las yemas al mismo tiempo

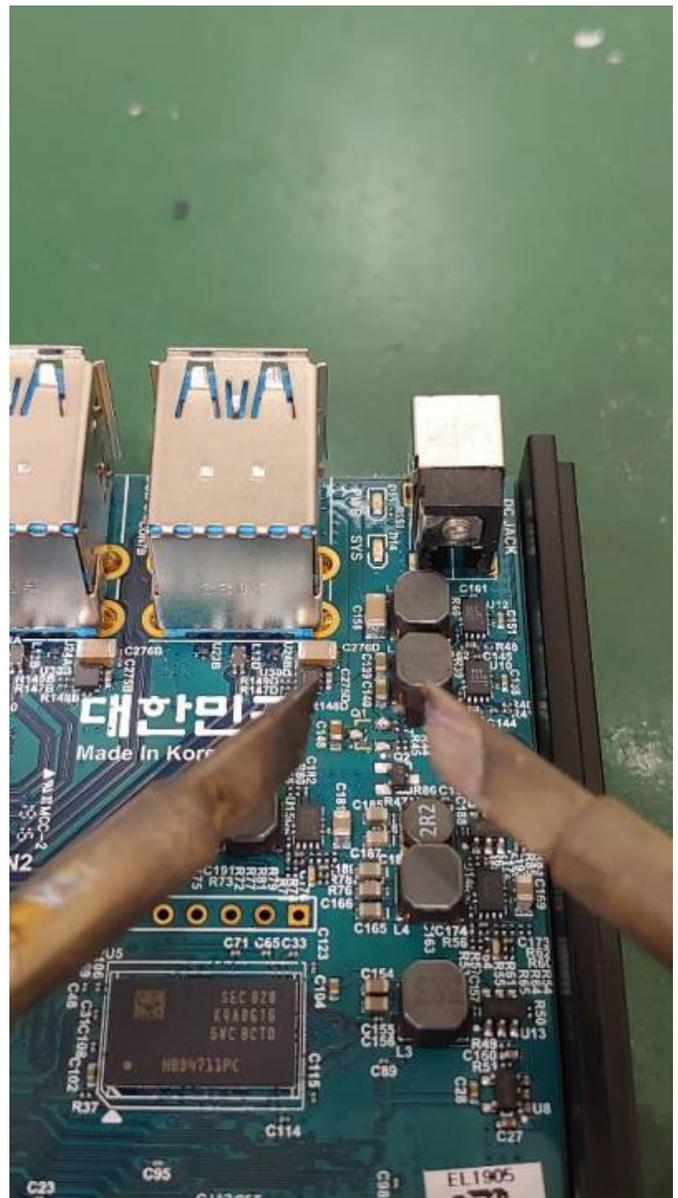


Figura 6 - Q1 retirado

Paso 4. Retira el resto de la soldadura de la yema. Usa una mecha de soldadura para retirar la soldadura.

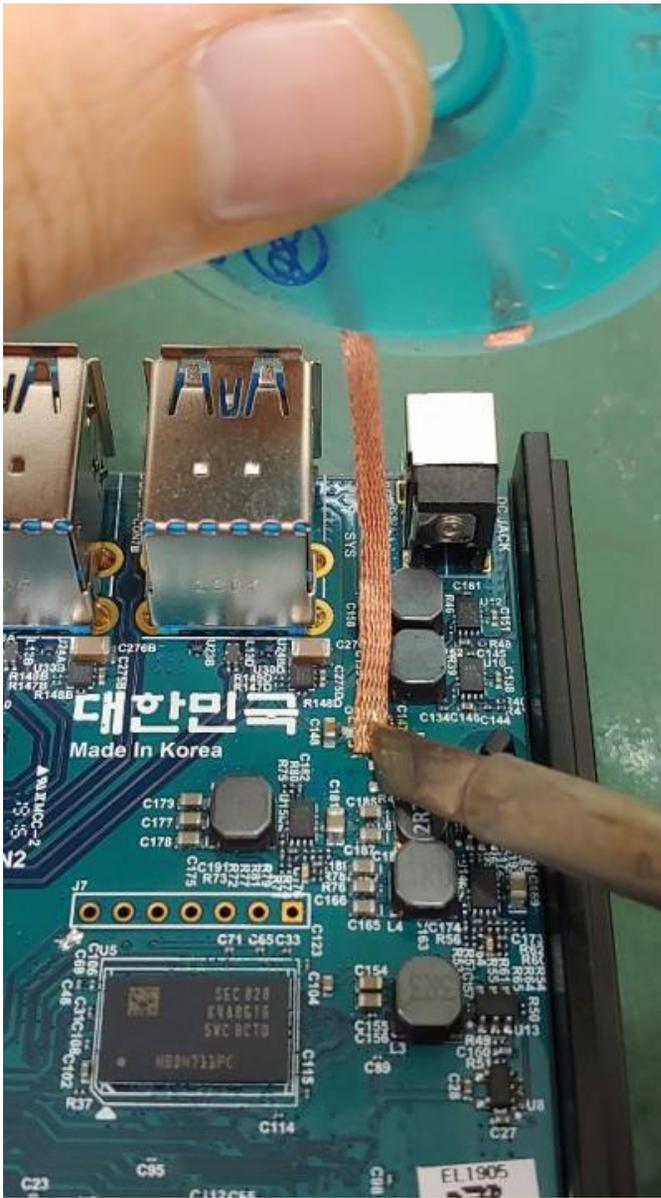


Figura 7 - Mecha de cobre utilizada para eliminar la soldadura residual

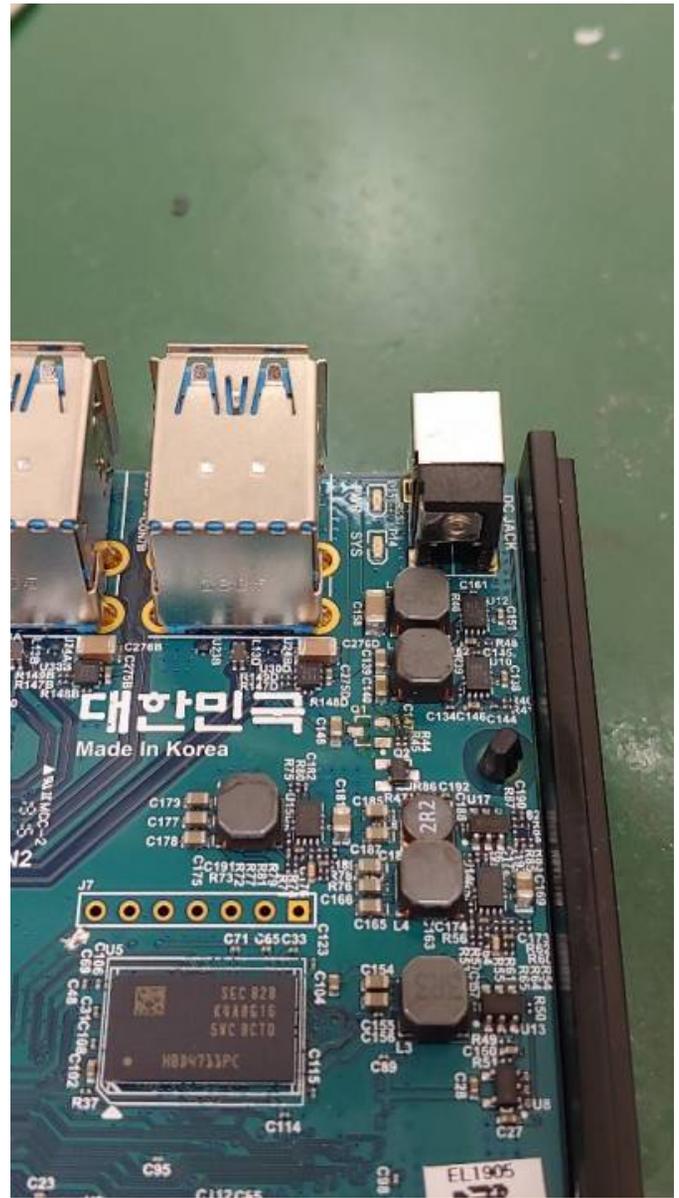


Figura 8 - Soldadura eliminada

Paso 5. Usa pinzas SMD y suelda el nuevo Q1 en su lugar.

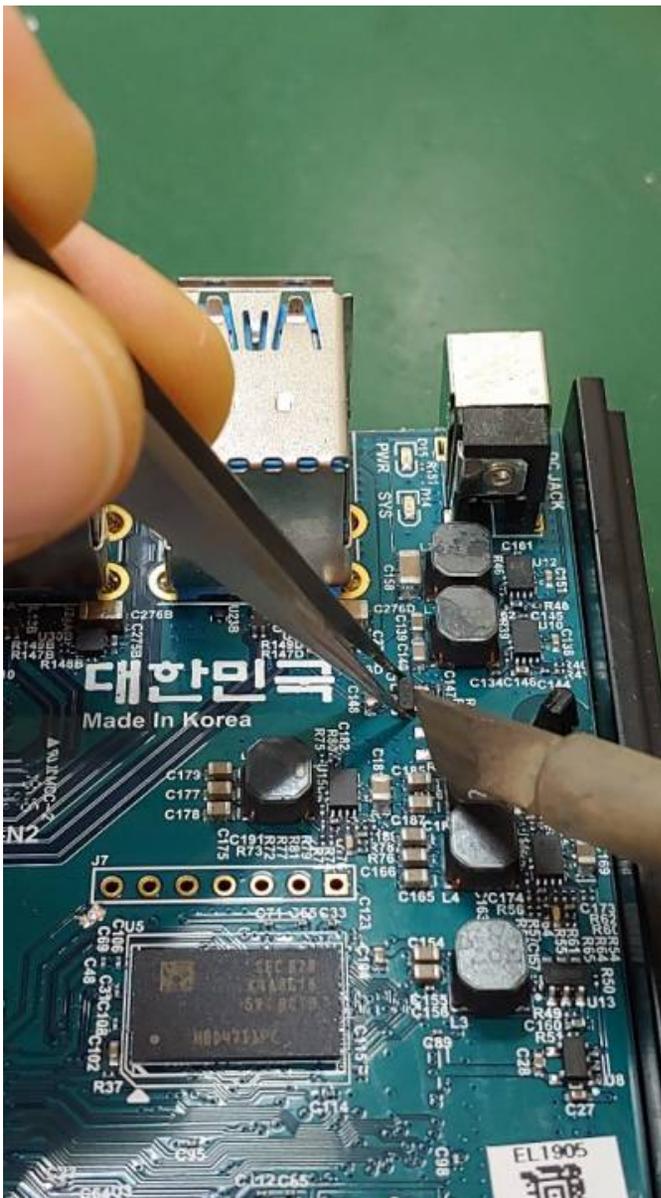


Figura 9 - Nuevo MOSFET colocado en su lugar con pinzas



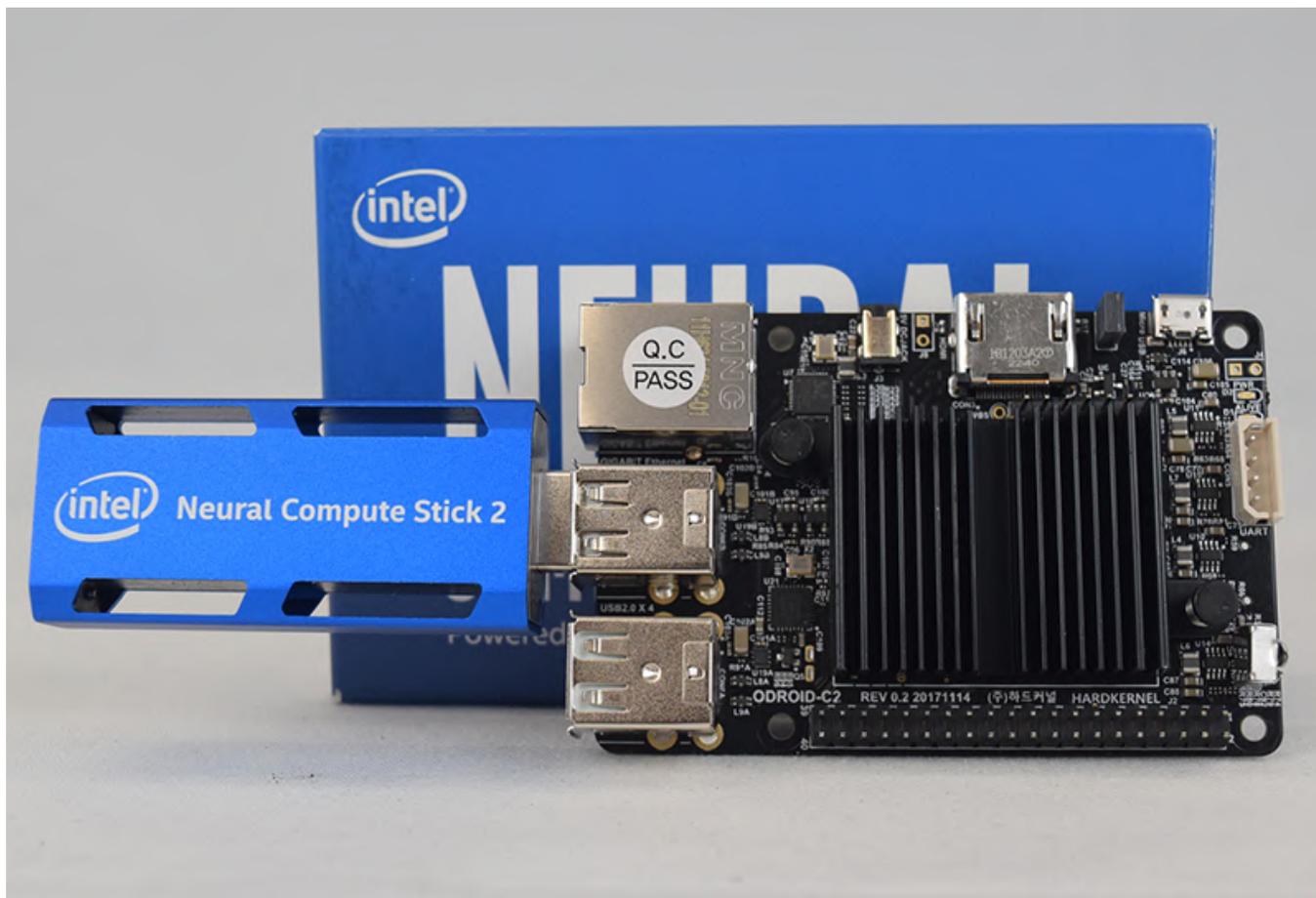
Figura 10 - Nueva soldadura MOSFET en su lugar

Conclusión

Esperamos que esta guía te haya ayudado a reparar tu ODR0ID-N2. Después de reemplazar Q1 (P-CH FET), tu USB, Ethernet, conector de expansión de 3.3V de potencia y el arranque eMMC deberían funcionar perfectamente.

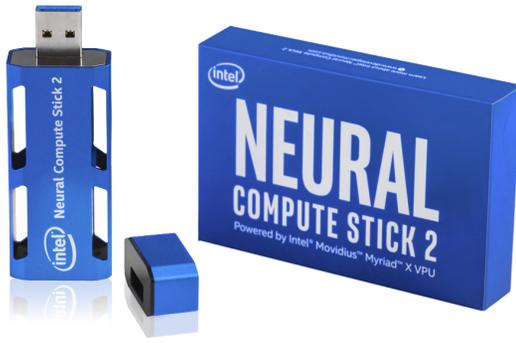
Creando una Aplicación de Visión en Situaciones de Baja Potencia: Usando OpenVino y OpenCV con ODRROID-C2

© December 1, 2019 By software.intel.com Linux, ODRROID-C2, Tutoriales



La distribución Intel® del kit de herramientas OpenVINO™ y el Intel® Neural Compute Stick 2 (Intel® NCS 2) son el complemento perfecto para aplicaciones de visión en entornos de desarrollo de baja potencia. La configuración en tantas y diferentes arquitecturas ofrece muchas oportunidades. Las plataformas ARM como ARM64 se están volviendo cada vez más comunes para los desarrolladores que crean y exportan soluciones para ordenadores de placa reducida (SBC) de bajo consumo. Estos pueden tener requisitos muy diferentes en comparación con los entornos informáticos tradicionales x86. Aunque la distribución Intel® del kit de herramientas OpenVINO™ proporciona una instalación binaria para múltiples entornos, incluido el popular SBC Raspberry Pi*, la versión de código abierto del kit de herramientas Intel® OpenVINO™ ofrece a los desarrolladores la oportunidad de compilar el kit de herramientas y la aplicación de puerto(s) para muy

diversos entornos. ODRROID-C2 de HARDKERNEL CO., LTD es un ordenador similar a la Raspberry Pi. El ODRROID-C2 es una plataforma ARM64 con un potente procesador de cuatro núcleos y mucha RAM (2 GB) para múltiples aplicaciones. Este artículo te guiará a la hora de configurar un ODRROID-C2 con Ubuntu * 16.04 (LTS), compilar CMake*, OpenCV y el kit de herramientas OpenVINO™ de Intel®, configurar tu Intel® NCS 2 y ejecutar algunos ejemplos para asegurarnos de que todo esté listo para que puedas crear e implementar tus aplicaciones del kit de herramientas Intel® OpenVINO™.



Paquete Neural Compute Stick

Aunque estas instrucciones han sido escritas para el ODROID-C2*, los pasos deberían ser prácticamente los mismos para otros SBC ARM* 64 como el ODROID-N2 siempre que tu entorno esté utilizando un sistema operativo de 64 bits. Si tu dispositivo utiliza un sistema operativo de 32 bits que admite el conjunto de instrucciones ARMv7, visita este artículo de ARMv7: <https://intel.ly/2DyJjpt>. Las instrucciones generales sobre cómo crear y utilizar la distribución de código abierto del kit de herramientas OpenVINO™ con Intel® Neural Compute Stick 2 y el Intel® Movidius™ Neural Compute Stick original, las puedes encontrar en el artículo: <https://intel.ly/2P9kQga>.

Hardware

Asegúrate de cumplir con los siguientes requisitos antes de empezar. Esto asegurará que todo el proceso de instalación se realice sin problemas:

- SBC ARMv7 como el Orange pi PC Plus
- AL MENOS una tarjeta microSD de 8GB. Puede utilizar el módulo eMMC integrado si tienes uno, aunque necesitarás una tarjeta microSD para escribir el sistema operativo en la placa.
- Intel® Neural Compute Stick 2
- Conexión a Internet Ethernet o red inalámbrica compatible
- Adaptador de corriente continua dedicado
- Teclado
- Monitor HDMI
- Cable HDMI
- Dispositivo de almacenamiento USB
- Ordenador Windows*, Ubuntu* o macOS * independiente (como el que estás usando en este momento) para escribir la imagen del instalador en el

dispositivo con un lector de tarjetas microSD compatible

Configuración de tu entorno de compilación

Esta guía asume que estás utilizando el usuario root y no incluye sudo en los comandos. Si ha creado otro usuario y ha iniciado sesión con ese usuario, ejecuta los comandos como root para instalarlos correctamente.

Asegúrate de que el software de tu dispositivo está actualizado:

```
$ apt update && apt upgrade -y
```

Algunas de las dependencias del kit de herramientas no tienen binarios ARMv7 precompilados y deben compilarse desde la fuente; esto puede aumentar el tiempo de compilación significativamente en comparación con otras plataformas. Preparar el kit de herramientas requiere de los siguientes pasos:

- Instalar herramientas de compilación
- Instalar CMake* desde la fuente
- Instalar OpenCV desde la fuente
- Clonar el kit de herramientas

Los pasos se detallan a continuación:

Instalar las herramientas de compilación

Instalar build-essential:

```
$ apt install build-essential
```

Esto instalará y configurará los compiladores GNU C y GNU CPlusPlus. Si todo finaliza con éxito, continúa e instala CMake* desde la fuente.

Instalar CMake* desde la fuente

La versión de código abierto del kit de herramientas Intel® OpenVINO™ (y OpenCV, a continuación) usa CMake* como su sistema de compilación. La versión de CMake en los repositorios de paquetes tanto para Ubuntu 16.04 (LTS) como para Ubuntu 18.04 (LTS) está demasiado desactualizada para nuestro uso y no existe un binario oficial para la plataforma, por lo que debemos compilar la herramienta desde la fuente. En este momento, la versión estable más reciente de

CMake es 3.14.4. Para empezar, coge CMake de la página de GitHub*, extráelo y dirígete a la carpeta extraída:

```
$ wget
https://github.com/Kitware/CMake/releases/download
/v3.14.4/cmake-3.14.4.tar.gz
$ tar xvzf cmake-3.14.4.tar.gz
$ cd ~/cmake-3.14.466
```

Ejecuta el script de arranque para instalar las dependencias adicionales e iniciar la compilación:

```
$ ./bootstrap
$ make -j4
$ make install
```

El paso de instalación es opcional, pero recomendado. Sin él, CMake se ejecutará desde el directorio de compilación. El número de tareas que utiliza el comando make se puede ajustar con el parámetro -j; se recomienda fijar este número para que coincida con el número de núcleos de tu plataforma. Puede verificar el número de núcleos de tu sistema utilizando el comando `grep -c ^processor /proc/cpuinfo`. Ten en cuenta que fijar un número demasiado alto puede provocar el desbordamiento de la memoria y por tanto fallar la compilación. Si el tiempo te lo permite, te recomiendo ejecutar de 1 a 2 tareas. CMake ahora está completamente instalado.

Instalar OpenCV desde la fuente

El kit de herramientas Intel® OpenVINO™ utiliza el potencial de OpenCV para acelerar la inferencia basada en la visión. Aunque el proceso CMake para el kit de herramientas Intel® OpenVINO™ descarga OpenCV, si no hay una versión instalada para las plataformas compatibles, no existe una versión específica para las plataformas ARMv7. Como tal, debemos compilar OpenCV desde la fuente. OpenCV requiere algunas dependencias adicionales. Instala lo siguiente desde tu administrador de paquetes (en este caso, apt):

- git
- libgtk2.0-dev
- pkg-config
- libavcodec-dev
- libavformat-dev

- libswscale-dev

Clona el repositorio desde la página OpenCV GitHub*, prepara el entorno de compilación y compila:

```
$ git clone https://github.com/opencv/opencv.git
$ cd opencv && mkdir build && cd build
$ cmake -DCMAKE_BUILD_TYPE=Release -
DCMAKE_INSTALL_PREFIX=/usr/local ..
$ make -j4
$ make install
```

OpenCV ahora está completamente instalado.

Descargar el código fuente e instalar dependencias

La versión de código abierto del kit de herramientas Intel® OpenVINO™ está disponible desde GitHub. La carpeta del repositorio es `dldt`, para Deep Learning Development Toolkit.

```
$ git clone https://github.com/opencv/dldt.git
```

El repositorio también tiene submódulos que deben cogerse:

```
$ cd ~/dldt/inference-engine
$ git submodule init
$ git submodule update --recursive
```

El kit de herramientas Intel® OpenVINO™ tiene varias dependencias de compilación. El script `install_dependencies.sh` las busca por ti. Se han realizado algunos cambios en el script para que se ejecute correctamente a las plataformas ARM*. Si tienes algún problema a la hora de ejecutar el script, debes instalar cada dependencia individualmente. Para las imágenes que incorporan un shell no compatible con POSIX de Bash, este script (a partir de 2019 R1.1) usa palabra clave de función y un conjunto de corchetes dobles que no funcionan para shells que no son Bash. Usando tu editor de texto favorito, realiza los siguientes cambios.

Línea 8 Original:

```
function yes_or_no {
```

Editar Línea 8:

```
yes_or_no() {
```

Line 23 Original:

```
if [[ -f /etc/lsb-release ]]; then
```

Editar Linea 23:

```
if [ -f /etc/lsb-release ]; then
```

El script también intenta instalar dos paquetes que no son necesarios para ARM: gcc-multilib y gPlusPlus-multilib. Deben eliminarse del script o el resto de paquetes deberán instalarse de forma independiente.

Ejecuta el script para instalar:

```
$ sh ./install_dependencies.sh
```

Si el script finaliza con éxito, estás listo para compilar el kit de herramientas. Si algo ha fallado, asegúrate de instalar las dependencias que hemos mencionado e inténtalo nuevamente.

Compilación

El primer paso, para empezar la compilación, es decirle al sistema la ubicación de la instalación de OpenCV. Usa el siguiente comando:

```
$ export OpenCV_DIR=/usr/local/opencv4
```

El kit de herramientas utiliza un sistema de compilación CMake para guiar y simplificar este proceso de compilación. Para compilar tanto el motor de inferencia como el complemento MYRIAD para Intel® NCS 2, usa los siguientes comandos:

```
$ cd ~/dldt/inference-engine
$ mkdir build && cd build
$ cmake -DCMAKE_BUILD_TYPE=Release
  -DENABLE_MKL_DNN=OFF
  -DENABLE_CLDNN=OFF
  -DENABLE_GNA=OFF
  -DENABLE_SSE42=OFF
  -DTHREADING=SEQ
  ..
$ make
```

Si el comando make falla debido a un problema con una librería OpenCV, asegúrate de informar al sistema de la ubicación de tu instalación de OpenCV. Si la compilación se completa con éxito, el kit de herramientas Intel® OpenVINO™ está listo para ejecutarse. Las compilaciones esta ubicadas en:

```
/inference-engine/bin/armv7/Release/
```

Verificando la instalación

Después de completar con éxito la compilación del motor de inferencia, debes verificar que todo esté configurado correctamente. Para verificar que el kit de herramientas y el Intel® NCS 2 funcionan en tu dispositivo, completa los siguientes pasos:

- Ejecuta el programa de muestra benchmark_app para confirmar que todas las librerías se cargan correctamente.
- Descargar un modelo formado
- Selecciona una entrada para la red neuronal
- Configura el controlador USB Intel® NCS 2 Linux*
- Ejecuta benchmark_app con el modelo y la entrada seleccionados.

```
Sample Programs: benchmark_app
```

El kit de herramientas Intel® OpenVINO™ incluye algunos programas de muestra que utilizan el motor de inferencia e Intel® NCS 2. Uno de los programas es benchmark_app, una herramienta para estimar el rendimiento de la inferencia de aprendizaje profundo. La puedes encontrar en:

```
~/dldt/inference-engine/bin/intel64/Release6
```

Ejecuta el siguiente comando en la carpeta para probar benchmark_app:

```
$ ./benchmark_app -h
```

Debe aparecer un cuadro de diálogo de ayuda que describa las opciones disponibles del programa.

Descargar un modelo

El programa necesita un modelo para pasar a la entrada. Los modelos para el kit de herramientas Intel® OpenVINO™ en formato IR se pueden obtener de la siguiente forma:

- Usando del Optimizador de modelos para convertir un modelo existente de uno de los marcos compatibles al formato IR para el motor de inferencia
- Usando la herramienta Model Downloader para descargar un archivo de Open Model Zoo

- Descargando los archivos IR directamente desde download.01.org

En nuestro caso, descargar los archivos directamente es más fácil. Usa los siguientes comandos para obtener un modelo de reconocimiento de edad y género:

```
$ cd ~
$ mkdir models
$ cd models
$ wget
https://download.01.org/opencv/2019/open_model_zoo
/R1/models_bin/age-gender-recognition-retail-
0013/FP16/age-gender-recognition-retail-0013.xml
$ wget
https://download.01.org/opencv/2019/open_model_zoo
/R1/models_bin/age-gender-recognition-retail-
0013/FP16/age-gender-recognition-retail-0013.bin
```

El Intel® NCS 2 requiere modelos optimizados para el formato de punto flotante de 16 bits conocido como FP16. Tu modelo, si difiere del ejemplo, puede requerir la conversión utilizando el Optimizador de modelos para FP16.

Entrada para la red neuronal

El último elemento requerido es la entrada para la red neuronal. Para el modelo que hemos descargado, necesitas una imagen de 62x62 con 3 canales de color. Este artículo incluye un archivo que contiene una imagen que puede usar, y se usa en el siguiente ejemplo. Copia el archivo en un dispositivo de almacenamiento USB, conecta el dispositivo a tu placa y usa los siguientes comandos para montar la unidad y copiar su contenido a una carpeta llamada OpenVINO en tu directorio de inicio:

```
$ lsblk
```

Utiliza el comando `lsblk` para listar los dispositivos de bloque disponibles y toma nota de tu unidad USB conectada. Usa tu nombre en lugar de `sdX` en el siguiente comando:

```
$ mkdir /media/usb
$ mount /dev/sdX /media/usb
$ mkdir ~/OpenVINO
$ cp /media/archive_openvino.tar.gz ~/OpenVINO
$ tar xvzf ~/OpenVINO/archive_openvino.tar.gz
```

La carpeta OpenVINO ahora debe contener dos imágenes, un archivo de texto y una carpeta llamada `squeezenet`. Ten en cuenta que el nombre del archivo puede ser diferente; debe coincidir con lo que has descargado de este artículo.

Configurar el controlador USB Intel® NCS 2 Linux*

Es necesario agregar algunas reglas `udev` para permitir que el sistema reconozca los dispositivos USB Intel® NCS 2. Dentro del archivo `tar.gz` adjunto hay un archivo llamado `97-myriad-usbboot.rules.txt`. Se debe descargar en el directorio de inicio del usuario. Sigue los siguientes comandos para agregar las reglas a tu dispositivo:

Si el usuario actual no es miembro del grupo de usuarios, ejecuta el siguiente comando y reinicia tu dispositivo:

```
$ sudo usermod -a -G users "$(whoami)"
```

Mientras estás conectado como usuario en el grupo de usuarios:

```
$ cd ~
$ cp 97-myriad-usbboot.rules.txt
/etc/udev/rules.d/97-myriad-usbboot.rules
$ udevadm control --reload-rules
$ udevadm trigger
$ ldconfig
```

El controlador USB debería instalarse correctamente ahora. Si Intel® NCS 2 no se detecta al ejecutar las demos, reinicia tu dispositivo e inténtalo de nuevo.

Ejecutando `benchmark_app`

Cuando se descarga el modelo, hay una imagen de entrada disponible y el Intel® NCS 2 está conectado a un puerto USB, usa los siguientes comandos para ejecutar `benchmark_app`:

```
$ cd ~/dldt/inference-engine/bin/intel64/Release
$ ./benchmark_app -I ~/president_reagan-62x62.png
-m
~/models/age-gender-recognition-retail-0013.xml
$ -pp ./lib -api async -d MYRIAD
```

Esto ejecutará la aplicación con las opciones seleccionadas. El parámetro `-d` le dice al programa qué dispositivo usar para la inferencia: MYRIAD activa

el complemento MYRAID, utilizando el Intel® NCS 2. Una vez que el comando se ejecute con éxito, el terminal mostrará estadísticas para la inferencia. Si la aplicación se ejecutó correctamente en tu Intel® NCS 2, entonces el kit de herramientas Intel® OpenVINO™ y el Intel® NCS 2 están configurados correctamente para usarlos en tu dispositivo.

Inferencia en el borde

Ahora que has confirmado que tu ARMv7 está configurado y funciona con Intel® NCS 2, puede comenzar a compilar e implementar tus aplicaciones de inteligencia artificial o utilizar una de las aplicaciones de muestra precompiladas para probar su uso. A continuación, intentaremos hacer una clasificación de imagen simple usando SqueezeNetv1.1 y una imagen descargada en la placa. Para simplificar las cosas, el archivo adjunto contiene tanto la imagen como la red. La red SqueezeNetv1.1 ya se ha convertido al formato IR para ser usado por el motor de inferencia.

El siguiente comando tomará la imagen cat.jpg que fue incluida en el archivo, usará el modelo de red squeezeNet1.1, cargará el modelo con el

complemento MYRIAD en el Intel® NCS 2 conectado e inferirá la salida. Como antes, la ubicación de la aplicación de muestra es:

```
/inference-engine/bin/armv7/Release/  
  
$ ./classification_sample -i ~/OpenVINO/cat.jpg -m  
~/OpenVINO/squeezenet/squeezenet1.1.xml -d  
MYRIAD
```

El programa generará una lista de los 10 resultados principales de la inferencia y un promedio del rendimiento de la imagen.

Si has llegado hasta aquí, tu dispositivo está configurado, verificado y listo para empezar a crear prototipos y desplegar tus propias aplicaciones de inteligencia artificial utilizando la potencia del kit de herramientas Intel® OpenVINO™.

Para obtener información más completa sobre las optimizaciones del compilador, consulta nuestro Aviso de optimización en: <https://intel.ly/33FbQUU>.

Referencia

<https://software.intel.com/en-us/articles/ARM64-sbc-and-NCS2>

Cómo Configurar un NAS Básico con Samba para Compartir Archivos

© December 1, 2019 By Miguel Alatorre, ameriDroid Technician Linux, Tutoriales



Aunque tener un hardware hecho específicamente para aplicaciones NAS es lo ideal, ¡No significa que sea imposible montar uno con un viejo ordenador [o con un ordenador de plaza reducida] ¡Usando Samba, cualquiera puede convertir una torre vieja o SBC en un servidor de archivos!

NOTA: Aunque un servidor Samba se puede configurar y ejecutar en sistemas operativos Windows, esta guía está centrada en el sistema operativo Linux.

Ahora inicia tu distribución favorita de Linux en tu máquina. Nota del editor: en caso de que tengas curiosidad, la imagen del blog muestra un ODROID-XU4 y un SSD de 2.5". Instala la última versión de Samba con los siguientes comandos:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install samba
```

Antes de configurar Samba, asegúrate de que la unidad esté montada y sea accesible. Para montarlo, introduce `sudo mount {Unidad} {Punto de montaje}`, aunque asegúrate de sustituir `{Unidad}` y `{Punto de montaje}` por la unidad y el directorio correctos, respectivamente. [Algo como esto: `sudo mount /dev/sda1/ mnt/extdrive`]

NOTA: Para encontrar la unidad correcta, usa el comando `fdisk -l`.

Desde aquí, debemos sumergirnos en el archivo `smb.conf` para configurar el servidor. Antes de jugar con la configuración, no es mala idea hacer una copia de seguridad en caso de que algo salga mal. Para hacerlo, introduce:

```
$ sudo cp /etc/samba/smb.conf
/etc/samba/smb.conf.bak
```

Ahora viene la parte divertida, para editar el archivo `"* .conf"`, usa tu editor de texto favorito o escribe:

```
$ sudo nano /etc/samba/smb.conf
```

Para un simple servidor, introduce algo similar a lo siguiente al final del archivo:

```
[Share]
comment = Shared Files
path = /path/to/share/destination
writable = yes
guest ok = yes
browsable = yes
```

Eso es todo, para acceder al recurso compartido en Windows, introduce `\Samba.Server.IP.Address` en la barra de búsqueda del explorador de archivos; para macOS, selecciona la pestaña "Ir" e introduce `smb://Samba.Server.IP.Address` [en cualquier caso, reemplaza `Samba.Server.IP.Address` con la dirección IP de tu servidor Samba]. Para Linux, el proceso

puede ser diferente según la distribución, aunque deberías poder abrir el administrador de archivos y seleccionar la opción "Conectar al host remoto" e introducir `smb://Samba.Server.IP.Address`.

[Nota del editor: tenga en cuenta que Samba tiene múltiples opciones de seguridad, y la configuración anterior es una configuración básica para almacenar archivos que no contienen datos confidenciales. Si desea almacenar y acceder a datos confidenciales, estudia la seguridad de Samba con más profundidad.]

La fuente de este artículo está disponible en: <https://ameridroid.com/blogs/ameriblogs/how-to-set-up-a-basic-nas-using-samba> Para obtener más actualizaciones de productos y tutoriales, síguenos en YouTube, nuestro blog de ameriDroid, Facebook, Twitter e Instagram!